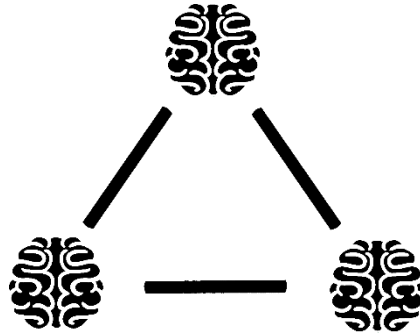




INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia de Electrónica e
Telecomunicações e de Computadores**



Interação Social entre Agentes em Ambientes de Recursos Limitados

Hélder Filipe de Oliveira Bastos
(Licenciado)

Trabalho Final de Mestrado para obtenção do grau de Mestre em Engenharia de Redes
de Comunicação e Multimédia

Orientador:

Professor Doutor Luís Filipe Graça Morgado

Júri:

Presidente: Professor Doutor Paulo Manuel Trigo Cândido da Silva

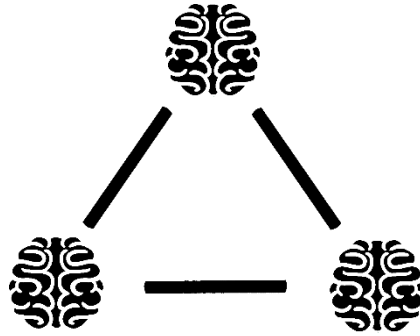
Vogal: Professor Doutor Arnaldo Joaquim Castro Abrantes

Dezembro de 2015



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia de Electrónica e
Telecomunicações e de Computadores**



Interação Social entre Agentes em Ambientes de Recursos Limitados

Hélder Filipe de Oliveira Bastos
(Licenciado)

Trabalho Final de Mestrado para obtenção do grau de Mestre em Engenharia de Redes
de Comunicação e Multimédia

Orientador:

Professor Doutor Luís Filipe Graça Morgado

Júri:

Presidente: Professor Doutor Paulo Manuel Trigo Cândido da Silva

Vogal: Professor Doutor Arnaldo Joaquim Castro Abrantes

Dezembro de 2015

Resumo

O estudo da interação social entre agentes inteligentes é cada vez mais um tema central da investigação em inteligência artificial. Atualmente inseridos numa realidade onde veículos navegam autonomamente, não é difícil imaginar cenários onde estas máquinas poderão trabalhar em conjunto, maximizando as suas capacidades e tornando-se mais eficientes na utilização de recursos e na concretização dos respetivos objetivos.

Devido à elevada complexidade computacional dos seus processos internos, ainda é difícil, sobretudo para máquinas com locomoção autónoma no mundo, desenvolver raciocínio social de forma complexa, pelo que este trabalho pretende contribuir com uma abordagem que aproxime a investigação de uma solução viável para cenários reais onde um sistema multi-agente, limitado pelos seus recursos, tem de produzir soluções em tempo real.

Este trabalho integra-se na área dos agentes inteligentes, sendo desenvolvido o tema da interação social em arquiteturas de agentes com limitação de recursos. É apresentado um estudo sobre as arquiteturas de agentes reativas e deliberativas e as respetivas potencialidades na constituição de sistemas multi-agente num contexto de limitação de recursos. É feito o estudo do problema da delegação de tarefas numa abordagem de limitação de recursos, onde se propõe um modelo de coordenação com base em mecanismos emocionais para a sua solução.

Palavras Chave: agentes inteligentes, inteligência artificial, sistemas multi-agente, modelos de emoção, raciocínio com recursos limitados, arquitetura BDI, delegação de tarefas.

Abstract

The study of social interaction between intelligent agents is increasingly becoming a central topic of research in artificial intelligence. Living in a reality where vehicles navigate autonomously, it is not hard to imagine scenarios in which these machines could work together, maximizing their skills, making better use of their resources and being more efficient on the achievement of objectives.

Due to the high computational complexity of the internal processes, it is still difficult, especially for machines with autonomous locomotion, to develop complex social reasoning, so this work aims to contribute to an approach that approximates the research to a viable solution for real scenarios where a multi-agent system, limited by its resources, has to produce solutions in real-time.

This work is part of the area of intelligent agents, developing the theme of social interaction on agent architectures working with bounded-resources. It is presented a study about the reactive and deliberative architectures and their potential in the formation of multi-agent systems in a bounded-resources context. It is also made a study about the task delegation problem from the resource-bounded point of view, proposing a coordination model, based on emotional mechanisms.

Keywords: intelligent agents, artificial intelligence, multi-agent systems, emotion models, resource-bounded reasoning, BDI architecture, tasks delegation.

Agradecimentos

Ao Prof. Dr. Luís Filipe Graça Morgado, por ter aceitado a orientação deste trabalho, pela total disponibilidade e partilha de sabedoria que imprimiram um estímulo constante para uma visão ampla e abrangente.

Ao Instituto Superior de Engenharia de Lisboa, em especial ao corpo docente e discente que, de forma direta ou indireta, me proporcionaram as melhores condições possíveis para a realização deste trabalho.

À minha companheira de viagem, paciente, dedicada e excepcionalmente compreensiva, pela sua alegria e amor.

À Prof. Paula Fonseca, pela disponibilidade e conselhos valiosos.

Ao Rafael Santos, por estes dois anos de partilha e amizade, que me ajudaram nos momentos mais difíceis deste ciclo de estudos.

Aos meus colegas de curso, em especial ao Diogo Lopes, pela disponibilidade, discussões, opiniões e sugestões que comigo partilhou.

À minha família, por me ter sempre inculcido um espírito de trabalho e responsabilidade.

A ti, companheira de viagem...

Resumo	i
Abstract	iii
Agradecimentos	v
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
1.3 Organização do Documento	4
2 Arquiteturas de Agentes Inteligentes	5
2.1 Arquiteturas Reativas	6
2.2 Arquiteturas Deliberativas	6
2.3 Arquitetura BDI	7
2.3.1 Modelo da Arquitetura BDI	9
2.4 <i>Procedural Reasoning System</i>	11
2.4.1 Arquitetura Geral	11
2.4.2 Principais Características da Arquitetura PRS	12
2.5 Plataforma <i>Jason</i>	14
2.5.1 Linguagem <i>AgentSpeak</i>	14
2.5.2 Módulo de Comunicação <i>AgentSpeak</i>	17
2.5.3 Cooperação com <i>AgentSpeak</i>	18
2.5.4 Implementação <i>Jason</i> da Linguagem <i>AgentSpeak</i>	18
2.5.5 Principais Características da Plataforma <i>Jason</i>	20
2.5.6 Outros Aspetos da Plataforma <i>Jason</i>	20

2.6	Conclusão	22
3	Sistemas Multi-Agente	23
3.1	Comunicação entre Agentes	25
3.1.1	Atos de Discurso	25
3.1.2	Linguagens de Comunicação	26
3.2	O Problema da Coordenação Multi-Agente	28
3.2.1	Métodos de Coordenação Multi-Agente	28
3.3	Interação Social	31
3.4	Raciocínio Social	31
3.4.1	Redes de Dependência	32
3.4.2	Raciocínio Social Hierárquico	34
3.5	Estruturas Organizacionais	35
3.5.1	Modelo AGR	37
3.6	Licitação	38
3.6.1	Protocolo <i>Contract Net</i>	39
3.7	Conclusão	39
4	O Problema da Limitação de Recursos	41
4.1	Coordenação de Agentes com Recursos Limitados	42
4.1.1	O Problema da Delegação de Tarefas	42
4.2	Racionalidade Limitada	44
4.3	Regulação Cognitiva	45
4.3.1	Regulação Cognitiva por Metacognição	46
4.3.2	Regulação Cognitiva de Base Emocional	47
4.4	O Modelo Emocional de Fluxo	47
4.4.1	Disposições Emocionais	48
4.4.2	Modelo de Agente	50
4.4.3	Disposição Emocional Cumulativa	52
4.4.4	Estados Motivacionais Centrais	52
4.5	Conclusão	54
5	Arquitetura de Agente Proposta	55
5.1	Organização Geral da Arquitetura de Agente Proposta	57
5.2	Subsistema Executivo	59
5.3	Subsistema Reativo	60
5.4	Subsistema Adaptativo	60
5.5	Subsistema Deliberativo	61
5.5.1	O Motor BDI	62
5.5.2	Componentes do Motor BDI	63
5.5.3	Dinâmica do Motor BDI	67

5.6	Subsistema Social	70
5.6.1	Definições da Arquitetura Social	71
5.6.2	Arquitetura do Sistema Social	72
5.7	Conclusão	78
6	Concretização e Resultados Experimentais	81
6.1	Contexto Experimental	82
6.2	Interação Social com Base em Arquiteturas Reativas	82
6.2.1	Comportamentos Reativos	83
6.2.2	Resultados Experimentais e Análise	85
6.3	Interação Social com Base em Arquiteturas Deliberativas de Planeamento Descentralizado	87
6.3.1	Resultados Experimentais e Análise	90
6.4	Interação Social com Base em Arquiteturas Deliberativas de Planeamento Centralizado	92
6.4.1	Resultados Experimentais e Análise	94
6.5	Interação Social com Base em Mecanismos Emocionais	96
6.5.1	Delegação de Tarefas de Base Emocional	97
6.5.2	Disposição Emocional dos Agentes Subordinados	98
6.5.3	Resultados Experimentais e Análise	99
6.6	Conclusão	103
7	Conclusão	105
A	Plataforma <i>Emotional Task Delegation</i>	109
A.1	Arquitetura da Plataforma <i>Emotional Task Delegation</i>	110
B	Plataforma <i>Tile World Multi-Agent Sim</i>	113
B.1	Arquitetura da Plataforma <i>Tile World MultiAgent Sim</i>	115
C	Regras de Inferência, Desejos, Planos e Ações 1	117
C.1	Regras de Inferência	117
C.2	Captor	118
C.3	Carrier	120
D	Regras de Inferência, Desejos, Planos e Ações 2	123
D.1	Regras de Inferência	123
D.2	Coordinator	124
D.3	Captor	125
D.4	Carrier	127
	Bibliografia	129

Lista de Tabelas

4.1	Situação, emoção e comportamento.	53
6.1	Política de delegação de tarefas baseada em emoção.	98

Lista de Figuras

2.1	Diagrama esquemático de uma arquitetura BDI genérica [Woldridge e Jennings, 1995].	10
2.2	Interpretador PRS.	11
2.3	Ciclo de interpretação do <i>AgentSpeak</i> [Machado e Bordini, 2002].	15
2.4	Ciclo de execução do <i>Jason</i> [Bordini et al., 2007].	19
3.1	Sistema multi-agente padrão [Jennings, 1999].	23
3.2	Integração de níveis de abstração [Morgado, 2014].	24
3.3	Exemplificação de coligações [Morgado e Gaspar, 2000].	34
3.4	Metamodelo UML do AGR [Ferber et al., 2004].	37
4.1	Delegação de tarefas.	43
4.2	Problema da delegação de tarefas.	43
4.3	Modelo geral metacognitivo.	46
4.4	Arquitetura genérica de agente de modelo de fluxo [Morgado e Gaspar, 2007].	47
4.5	Vector ED como função de δP e δA (a); relação entre os quadrantes ED e a tendência qualitativa emocional. [Morgado e Gaspar, 2008]	50
4.6	Realização de um motivador num espaço cognitivo bidimensional [Morgado e Gaspar, 2008].	51
5.1	Arquitetura de subsistemas da arquitetura de agente proposta.	57
5.2	Diagrama de contexto da arquitetura proposta.	58
5.3	Processo Executar.	59
5.4	Exemplificação de um comportamento reativo.	60

5.5	Processo de deliberação.	61
5.6	Processo de planeamento.	62
5.7	Modelo da dinâmica de um desejo.	64
5.8	Modelo de dinâmica de uma ação.	65
5.9	Modelo de dinâmica de um plano.	66
5.10	Resumo da dinâmica do motor BDI.	68
5.11	Processo Coordenação Social.	77
6.1	Arquitetura do Agente Reativo.	83
6.2	Comportamento de um agente tipo <i>captor</i>	84
6.3	Comportamento de um agente tipo <i>carrier</i>	84
6.4	Exemplo das trajetórias dos agentes reativos.	85
6.5	Exemplo do problema da falta de coordenação dos agentes reativos.	86
6.6	Organização funcional da arquitetura deliberativa.	88
6.7	Arquitetura do agente deliberativo.	89
6.8	Exemplo das trajetórias dos agentes deliberativos de planeamento descentralizado.	90
6.9	Exemplo do problema da falta de coordenação de agentes deliberativos de planeamento descentralizado.	91
6.10	Arquitetura do agente híbrido.	93
6.11	Exemplo das trajetórias dos agentes deliberativos de planeamento descentralizado.	95
6.12	Comunicação da disposição emocional no contexto da delegação de tarefas.	96
6.13	Representação do espaço cognitivo de um agente subordinado.	99
6.14	Resultados para as políticas de distribuição de tarefas uniforme, baseada na emoção e ótima.	100
6.15	Disposição emocional dos agentes ag_1 (esquerda) e ag_2 (direita) sob a política de delegação de tarefas uniforme.	101
6.16	Disposição emocional dos agentes ag_1 (esquerda) e ag_2 (direita) sob a política de delegação de tarefas baseada em emoções.	102
A.1	Arquitetura da plataforma <i>Emotional Task Delegation</i>	110
A.2	Dinâmica da plataforma de simulação de delegação de tarefas.	111
A.3	Dinâmica de um ciclo de execução do agente.	112
B.1	Demonstração da obtenção e entrega de um bloco.	114
B.2	Arquitetura da plataforma <i>Tile World MultiAgent Sim</i>	115
B.3	Dinâmica do simulador.	116

CAPÍTULO 1

Introdução

O aparecimento, nos finais do séc. XX e início do séc. XXI, do conceito de sistemas multi-agente potenciou o estudo da interação social entre sistemas inteligentes artificiais. Esse estudo teve o seu início pelo trabalho de psicólogos e sociólogos, como Daniel Merton Wegner (1948-2013) e Edgar Morin, que previamente desenvolveram estudos na área da formação de sociedades e as respetivas especificidades, que serviram de base a um desenvolvimento consolidado da área da interação social entre sistemas inteligentes artificiais.

O desenvolvimento de sistemas multi-agente permitiu a resolução de problemas de forma paralela e adaptativa, o que melhora significativamente o desempenho, distinguindo-se da computação paralela pela capacidade de adaptação ao meio envolvente, permitindo apresentar resultados que se coadunem com a situação. Com a existência de agentes em diferentes máquinas, a trabalhar em paralelo, os recursos são aproveitados ao distribuir os problemas e/ou tarefas. A implementação destes sistemas multi-agente requer uma especial atenção sobre alguns problemas sociais, e.g. a simples existência de outros agentes no meio, ainda que não colidam nos seus objetivos e recursos,

obriga a preocupações sobre as eventuais ações dos outros agentes e as suas consequências, ou as interações negativas entre agentes, ou seja, situações em que dois ou mais agentes, de forma singular ou social, competem entre si pela existência de objetivos díspares, ou pela competição na aquisição de recursos.

Para além dos problemas sociais, temas como a coordenação, o raciocínio social, a comunicação e a resolução de conflitos, são de grande importância no desenvolvimento de sistemas multi-agente, encontrando-se o estado da arte num ponto em que os recursos computacionais existentes não suportam a complexidade computacional dos processos internos destes sistemas multi-agente, ficando limitada a expansão das arquiteturas para sociedades de grande número de elementos.

Existindo o desafio da implementação dos processos sociais de um agente, caracterizados por uma elevada complexidade computacional, em sistemas onde os recursos são limitados, este trabalho tem como objetivo o desenvolvimento de soluções que permitam a implementação de sistemas multi-agente compostos por agentes capazes de interatuar utilizando o mínimo de recursos possível.

São abordadas algumas arquiteturas de agente, nomeadamente as arquiteturas reativa e deliberativa, com especial foco nas suas possibilidades de interação em ambientes de recursos limitados. Também neste trabalho é apresentada uma proposta de solução para o problema da delegação de tarefas, típico de sistemas multi-agente, em ambientes de recursos limitados, onde se demonstra o potencial da utilização de modelos de base emocional para a redução da utilização dos recursos do agente coordenador, através da geração de uma política de delegação de tarefas.

1.1 Motivação

O aparecimento de tecnologia com locomoção autónoma, com relevo para os veículos aéreos não tripulados, vulgarmente designados por *drones*, num mercado acessível à maioria da população, fornece o vislumbre de um futuro

onde máquinas, sem condicionantes humanas, poderão realizar tarefas em conjunto de forma automática e eficiente.

A utilização de *drones* é já uma realidade, verificando-se um grande potencial de aplicação destes sistemas em diversas áreas, dando-se os exemplos da fotografia, da vigilância e do transporte de bens. Verifica-se também que a utilização destes sistemas, em determinadas situações, é efetuada sem ação de controlo externo, sendo o *drone* responsável pela navegação no mundo e pela concretização dos seus objetivos. Tendo o *drone* autonomia para concretizar os seus objetivos, é legítimo afirmar que um dos passos futuros desta tecnologia será a existência de sistemas multi-agente compostos por *drones* e pessoas, que irão fomentar novas capacidades e aumentar a eficiência na realização de tarefas.

Arquiteturas variadas, implementadas em sistemas computacionais modernos, têm possibilitado o desenvolvimento de agentes inteligentes com capacidade social. No entanto, devido à elevada complexidade computacional dos processos de interação, muitas vezes ficam impossibilitadas interações sociais complexas em tempo real.

A necessidade real explanada e o desafio proporcionado pela investigação e realização de projetos inovadores são os motivos do desenvolvimento do presente trabalho.

1.2 Objetivos

Este trabalho tem como objetivos:

- o estudo de mecanismos de interação social em diferentes tipos de arquiteturas de agentes;
- o estudo das limitações dos mecanismos de interação social, estudados no ponto anterior, num contexto de limitação de recursos e possíveis soluções mitigadoras destas limitações;
- o estudo de modelos de base emocional como potencial regulador da coordenação entre agentes em ambientes de recursos limitados.

1.3 Organização do Documento

Este trabalho encontra-se dividido nos seguintes capítulos:

Capítulo 1 - Introdução: Apresenta o enquadramento deste trabalho, identificando a motivação, os objetivos e o contexto da presente dissertação.

Capítulo 2 - Arquiteturas de Agentes Inteligentes: Aborda diversas arquiteturas de agentes inteligentes, nomeadamente arquiteturas reativas, deliberativas e BDI, complementando o estudo desta última arquitetura pela análise às plataformas *Procedural Reasoning System* e *Jason*.

Capítulo 3 - Sistemas Multi-Agente: Aborda a temática dos sistemas multi-agente, onde é estudada a comunicação entre agentes e a interação social. É definido o problema da coordenação que se aprofunda pelo estudo dos métodos de coordenação: raciocínio social, estruturas organizacionais e licitação.

Capítulo 4 - O Problema da Limitação de Recursos: Apresenta a racionalidade limitada como um dos efeitos da limitação de recursos, mitigado pela regulação cognitiva. É apresentado o Modelo Emocional de Fluxo como modelo apropriado a uma regulação cognitiva de base emocional.

Capítulo 5 - Arquitetura de Agente Proposta: Propõe uma arquitetura de agente capaz de operar em ambientes de recursos limitados e de coordenar as suas ações, enquadrada numa estrutura organizacional hierárquica.

Capítulo 6 - Concretização e Resultados Experimentais: Apresenta os testes desenvolvidos e respetivos resultados, que suportam a arquitetura proposta.

Capítulo 7 - Conclusões: Conclusões sobre o trabalho efetuado: modelo de coordenação de ação proposto e arquiteturas propostas e implementadas, assim como possível trabalho futuro.

CAPÍTULO 2

Arquiteturas de Agentes Inteligentes

No contexto desta dissertação, entende-se por agente inteligente uma entidade computacional capaz de tomar a melhor decisão possível de acordo com os seus objetivos e a informação disponível [Russel e Norvig, 2009], sendo autónomo, ou seja, capaz de operar por si próprio e de modo independente de outros sistemas [Wooldridge, 2002]. Para além da autonomia, um agente inteligente apresenta características de reatividade, pro-atividade e sociabilidade. A reatividade é a capacidade de perceber o mundo e responder às suas alterações, de forma a serem atingidos os objetivos do agente. A proatividade corresponde à capacidade de tomar a iniciativa para concretizar os seus objetivos. A sociabilidade é a capacidade de um agente interagir com os restantes agentes visando a concretização dos seus objetivos [Wooldridge, 2002].

Para além das características genéricas de um agente inteligente (reatividade, proatividade e sociabilidade), dependendo da arquitetura do agente, este pode ter a capacidade de ação em tempo real, ou seja, tem de ser capaz de satisfazer restrições de tempos de resposta explícitos, ou arriscar consequências graves, incluindo a falha [Laplante e Ovaska, 2011].

2.1 Arquiteturas Reativas

As arquiteturas reativas são caracterizadas por comportamentos do tipo estímulo-resposta. Neste tipo de arquiteturas não existe raciocínio, resultando o comportamento de reações, estruturadas sob a forma de comportamentos com diferentes níveis de competência.

Murphy [Murphy, 2000] define comportamento como o mapeamento das entradas sensoriais em padrões de ações motoras. O comportamento reativo caracteriza-se como um conjunto de regras que definem, para determinado estímulo, um resposta padrão específica, sendo esta resposta enquadrada nas capacidades do agente.

Das diversas arquiteturas reativas propostas, destacam-se pela sua relevância a arquitetura de Subsunção [Brooks, 1985] e a arquitetura de Esquemas Comportamentais [Arkin, 1998].

A arquitetura de Subsunção distingue-se pela composição de comportamentos. Comportamentos complexos são definidos pela aglomeração de comportamentos mais simples que, na sua forma atômica, são módulos que definem a resposta aos estímulos provenientes dos sensores. A estruturação de comportamentos é feita por níveis de competência e os módulos de níveis mais altos podem inibir o comportamento dos níveis inferiores.

A arquitetura de Esquemas Comportamentais caracteriza-se pela especificação de comportamentos através de campos de potencial. As respostas aos estímulos (comportamentos simples) poderão ser somadas vetorialmente, produzindo comportamentos complexos.

2.2 Arquiteturas Deliberativas

As arquiteturas deliberativas, relativamente às arquiteturas reativas atrás apresentadas, são mais complexas, caracterizando-se pela capacidade de raciocínio. A complexidade das arquiteturas deliberativas resulta dos processos internos associados à manutenção e processamento de modelos do mundo que

que são utilizados no processo de raciocínio.

Destes processos destaca-se o raciocínio prático (também conhecido como raciocínio meios-fins), o qual é composto por dois processos principais: (i) decidir quais os objetivos a atingir (fins), através de um processo de deliberação; e (ii) decidir como se vai atingir os objetivos (meios), através de um processo de planeamento [Wooldridge e Jennings, 1995].

O planeamento é realizado sobre uma representação simbólica do mundo, que poderá incluir a representação de outros agentes e assumir diversas formas. Este processo é caracterizado por uma elevada complexidade computacional, o que aumenta o tempo de resposta, podendo ser um problema em situações de resposta em tempo real.

2.3 Arquitetura BDI

A arquitetura BDI (*Belief, Desire, Intention*), desenvolvida no final da década de 80, tem por objetivo criar um equilíbrio entre as arquiteturas deliberativas e as reativas. Com especial relevância em sistemas autónomos orientados por objetivos, esta arquitetura tem por base um raciocínio prático que dá à arquitetura uma maior robustez face à incerteza do mundo em que opera, ao otimizar comportamentos face à situação atual do agente.

A arquitetura BDI tem a sua base em modelos de sistemas intencionais fundamentados na psicologia humana que definem esses sistemas em termos de crenças (*beliefs*), desejos (*desires*) e intenções (*intentions*). Num determinado instante de tempo, o estado do agente BDI é definido pelo tuplo (B, D, I) onde $B \subseteq \text{beliefs}$, $D \subseteq \text{desires}$ e $I \subseteq \text{intentions}$.

A arquitetura BDI implementa, entre outros, o conceito de estado do mundo, crenças, desejos e intenções a seguir definidos.

Define-se como **estado do mundo** o conjunto de factos que caracterizam o meio em que o agente se encontra em determinado momento.

Considerando \mathbb{U} como o conjunto de todos os factos que podem caracte-

rizar o mundo, define-se o estado do mundo E da seguinte forma:

$$E \subseteq \mathbb{U} \quad (2.1)$$

É importante referir que devido às limitações naturais dos sensores de um agente, os factos que este possui sobre o mundo poderão não o descrever na sua plenitude, pelo que se define como **estado do mundo de um agente** EA :

$$EA \subseteq E \quad (2.2)$$

As **crenças** são factos sobre o estado do mundo que o agente conhece [Wooldridge e Parsons, 1999] e acredita serem verdadeiros.

Desejos são estados do mundo que o agente pretende atingir, representando o estado motivacional do sistema [Rae e Georgeff, 1995], encontrando-se o próprio agente na origem dessa mesma vontade.

Um **objetivo delegado** é definido como o conjunto de factos sobre o mundo que o agente pretende que sejam incluídos no estado do mundo, estando um agente externo na origem dessa mesma vontade.

Define-se um desejo/objetivo delegado d como um facto sobre o estado do mundo que não se observa e que o agente pretende ver concretizado:

$$d \in \overline{EA} \quad (2.3)$$

Uma **intenção** é um estado do mundo que o agente decidiu concretizar, tendo atribuído recursos para o atingir [Wooldridge e Parsons, 1999]. Os recursos atribuídos estão tipicamente relacionados com o tempo despendido a definir um plano de concretização e a concretizá-lo, mas também podem estar relacionados com a atribuição de recursos físicos.

As intenções condicionam o raciocínio prático ao: (i) guiarem o raciocínio meios-fins; (ii) constringirem o processo de deliberação; (iii) persistirem ao longo do tempo; (iv) e influenciarem as crenças sobre as quais o raciocínio prático é baseado [Wooldridge e Jennings, 1995].

Para efeitos deste trabalho, entende-se como **intenção** o compromisso de concretização de um desejo/objetivo, pela atribuição de um plano que

segundo a base de crenças é possível realizar e cuja realização irá concretizar o desejo/objetivo.

Dá-se a designação de **intenção atual** a uma das intenções que se encontram, no presente momento, em execução.

O **modelo interno** consiste no conjunto de bases de informação que, internamente, caracterizam o agente e o mundo no qual se insere. O modelo interno é individual a cada agente.

O modelo interno de uma arquitetura BDI exige a implementação das bases de crenças, desejos e planos. Dependendo da arquitetura implementada, complementos ao modelo podem ser definidos, e.g. base de intenções, alterações às crenças ou desejos.

Um **plano** é um encadeamento temporal de passos, que podem corresponder a desejos ou ações que, no caso de serem concretizados/executados sob determinadas pré-condições, permitem ao agente a concretização de um determinado desejo.

Um **plano concretizável** é um plano que é possível concretizar segundo o estado do mundo do agente.

Denomina-se **desejo possível** um desejo para o qual o agente possui um plano concretizável.

2.3.1 Modelo da Arquitetura BDI

O modelo da arquitetura BDI [Wooldridge, 2002] é composto por sete componentes principais:

1. Um conjunto de **crenças**: conjunto de factos que o agente acredita serem verdadeiros;
2. Uma função de **revisão de crenças**: transforma perceção em crenças;
3. Uma função **geradora de opções**: determina as opções disponíveis ao agente;

4. Um conjunto de **opções atuais**: possíveis cursos de ação disponíveis;
5. Uma função de **filtro**: processo de deliberação que forma intenções;
6. Um conjunto de **intenções atuais**: o foco de ação atual do agente;
7. Uma função de **seleção de ação**: determina a ação a executar.

Este modelo encontra-se representado na figura 2.1:

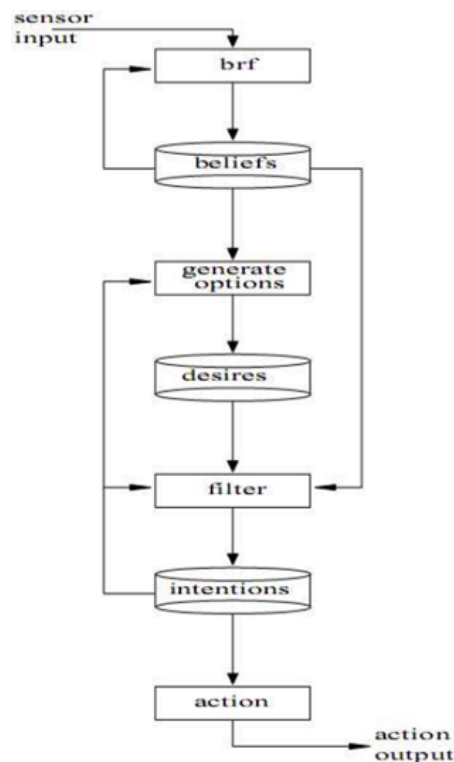


Figura 2.1: Diagrama esquemático de uma arquitetura BDI genérica [Wooldridge e Jennings, 1995].

No âmbito de uma arquitetura BDI genérica, um passo do processamento interno inicia-se com uma função de revisão de crenças que as gera. Os desejos são gerados de seguida, através de uma função de geração de opções que tem como parâmetro de entrada as crenças. Uma função de filtro analisa os desejos e as crenças e gera as intenções, das quais resulta uma ação.

De seguida, irão ser estudadas duas plataformas de programação de agentes baseadas no modelo BDI. O *Procedural Reasoning System* e o *Jason* são

referências ao estudo deste modelo, pela sua divulgação.

2.4 Procedural Reasoning System

Desenvolvido na década de 80 pelo *Artificial Intelligence Center* pertencente ao *SRI International* e posteriormente pelo *Australian Artificial Intelligence Institute*, o *Procedural Reasoning System* (PRS) é uma plataforma de programação de agentes de alto nível desenvolvida na linguagem C e baseada no modelo BDI. O PRS é uma plataforma de uso amplo em aplicações de elevada complexidade e com necessidade de robustez pelo que, a sua arquitetura se encontra bem definida e consolidada.

2.4.1 Arquitetura Geral

A arquitetura geral da plataforma PRS [Ingrand et al., 1996] é caracterizada por um ciclo de execução representado na figura 2.2.

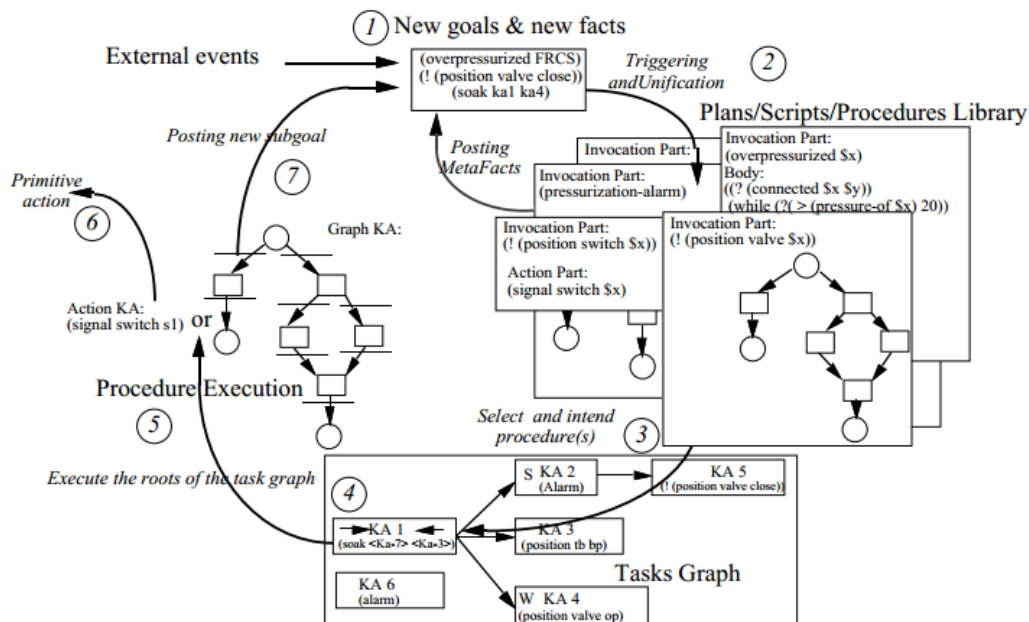


Figura 2.2: Interpretador PRS.

O ciclo de execução geral *PRS* é composto pelas seguintes atividades:

- Receção de novos eventos (do exterior e factos adicionados) e objetivos internos;
- Verificação de condições suspensas e necessárias manter e posterior seleção dos planos apropriados (baseada nos novos eventos, objetivos e crenças);
- Colocação dos planos selecionados no grafo de tarefas;
- Escolha de uma tarefa entre as raízes do grafo;
- Execução de um passo do plano ativo, da tarefa selecionada;
 - Execução do passo, se este for uma ação primitiva;
 - Adicionar objetivos, se o passo for subobjetivo.

2.4.2 Principais Características da Arquitetura PRS

Núcleo da Arquitetura PRS

O núcleo da arquitetura PRS é constituído por três componentes principais: (i) base de dados, (ii) biblioteca de planos (iii) e um grafo de tarefas.

A base de dados contém crenças que são automaticamente atualizadas à chegada de novos eventos.

A biblioteca de planos é constituída por planos. Os planos descrevem uma sequência particular de ações e testes que têm de ser verificados para se atingir determinados objetivos, ou reagir a determinadas situações.

O grafo de tarefas é um conjunto dinâmico de tarefas que se encontram em estado de execução. Este grafo mantém o registo do estado de execução da intenção e respetivos subobjetivos.

Objetivos

Os objetivos são descrições de um estado desejado pelo agente, que se encontra associado a um comportamento de alcançar/testar o respetivo estado. Os objetivos podem ser definidos com recurso aos operadores:

- **Achieve(x)**: atingir o objetivo “x”;
- **Test(x)**: testar se o objetivo “x” é verdadeiro;
- **Wait(x)**: aguarda indefinidamente até o objetivo “x” se verificar verdadeiro;
- **Preserve(x)**: enquanto um determinado comportamento é ativado, “x” é considerado sempre verdadeiro;
- **Maintain(x)**: “x” é um objetivo de natureza permanente, ou seja, permanece na base de desejos mesmo após a sua concretização;
- **Assert(x)**: adicionar “x” à base de dados;
- **Retract(x)**: remover “x” da base de dados.

Plano

O corpo de um plano define as ações a realizar para concretizar determinada intenção. Este corpo tem uma estrutura procedimental que incorpora os operadores enumerados na subsecção 2.4.2, acompanhados de estruturas de controlo padrão de linguagens de programação e.g. instruções de decisão, do tipo “if-then-else”, ou de iteração, do tipo “while”.

Grafo de Tarefas

Todas as intenções são mantidas pelo grafo de tarefas. Uma intenção é uma pilha de planos encadeados e cada plano corresponde a um conjunto de instruções ordenadas sobre quais as ações a realizar ou objetivos a atingir.

Uma nova intenção, ao ser despoletada por um objetivo proveniente de um evento externo (e.g. resultado da perceção), é adicionada à raiz do grafo

de tarefas. Se a origem do objetivo tiver proveniência num subobjetivo de um plano de uma intenção, a intenção despoletada é colocada na pilha da intenção “mãe”.

Outros Aspetos

O facto de não ter sido criada com o objetivo de implementar sistemas multi-agente repercute uma não adequação desta arquitetura a esta temática, necessitando de extensões e de ajustes para o efeito.

As intenções despoletadas por eventos exigem um esforço de programação adicional pela necessidade da implementação de uma função específica para cada problema. Esta função tem como parâmetros de entrada crenças, intenções, objetivos e planos do agente, e de saída uma nova intenção. Esta opção, embora simplifique a arquitetura, é mais propícia a erros de programação, levando a que, em situações de elevada complexidade, o programador tenha problemas em conseguir prever todos os estados possíveis para o agente e respetiva resposta do sistema.

2.5 Plataforma *Jason*

A plataforma de programação multi-agente *Jason* é *open-source* e desenvolvida na linguagem *Java*. De uma forma sumária, apresenta-se como uma implementação moderna da arquitetura *AgentSpeak*, complementada com diversas extensões.

2.5.1 Linguagem *AgentSpeak*

A linguagem *AgentSpeak* serve de padrão para linguagens de programação multi-agente sendo, de facto, um padrão de referência não apenas ao nível da arquitetura, como da semântica de programação.

A arquitetura de suporte à linguagem *AgentSpeak*, encontra-se represen-

tada na figura 2.3.

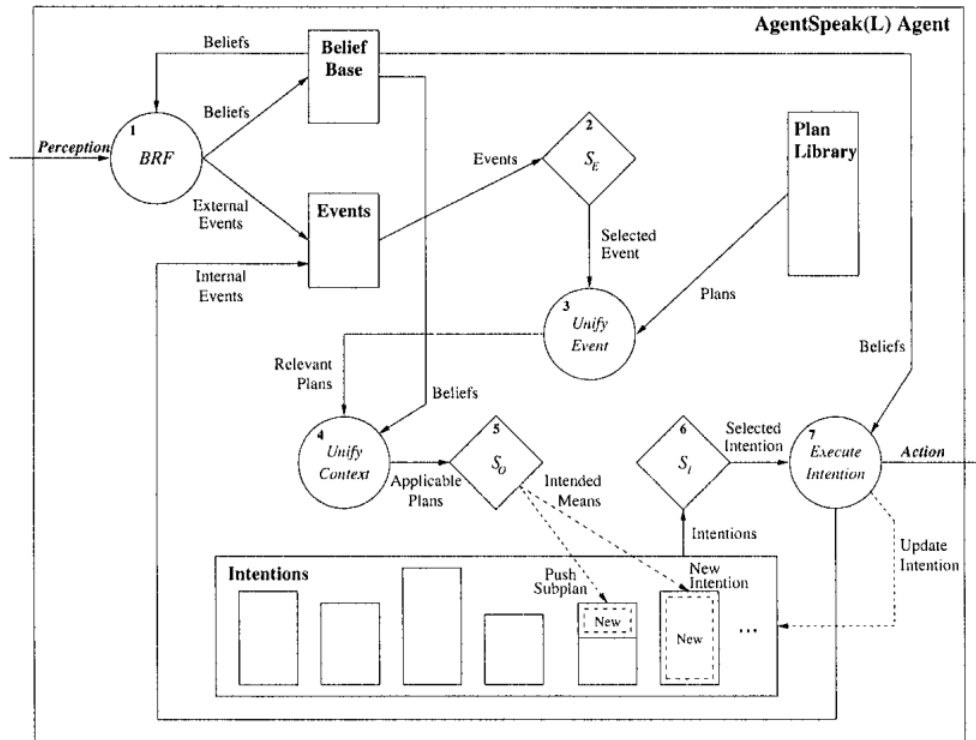


Figura 2.3: Ciclo de interpretação do *AgentSpeak* [Machado e Bordini, 2002].

Esta arquitetura é composta por sete processos [Machado e Bordini, 2002]:

- **BRF:** Função geradora de crenças. Atualiza a lista de eventos (que podem ter origem na percepção ou execução de intenções) e a lista de crenças;
- **SE:** Seleção (filtro) de eventos. Seleciona um evento para ser processado;
- **Unify Event:** Unificador de eventos. Unifica o evento selecionado com os eventos despoletadores dos planos, gerando um conjunto de planos relevantes;
- **Unify Context:** Unificador de contexto. Filtra os planos relevantes, ao unificar o contexto do plano com as crenças;
- **SO:** Seleção (filtro) de planos. Escolhe um plano do conjunto de planos

relevantes e cria uma nova intenção, ou coloca o plano no contexto da intenção que o gerou;

- **S:** Seleção (filtro) de intenções. Escolhe a intenção a ser executada;
- **Execute Intention:** Executor da intenção. Identifica o próximo passo do plano da intenção selecionada e executa (dá origem a uma ação diretamente aplicável no mundo, ou a um subobjetivo).

Definições da Linguagem *AgentSpeak*

Crença: definida por predicado, representa a informação de que o agente dispõe do mundo.

Objetivo a atingir: determinação ao agente para atingir um estado do mundo no qual o predicado é verdadeiro.

Objetivo de teste: determinação ao agente para testar se o predicado é verdadeiro.

Evento Despoletador: faz parte de um plano e indica os eventos com os quais é suposto o plano lidar. Estes eventos podem ser:

- **+b:** adição de crença;
- **-b:** remoção de crença;
- **+!g:** adição de objetivo de concretização;
- **-!g:** remoção de objetivo de concretização;
- **+?g:** adição de objetivo de teste;
- **-?g:** remoção de objetivo de teste.

Ação: representação simbólica (em predicado) da ação a realizar no mundo.

Plano: é um conjunto de ações ou subobjetivos enquadrados num contexto e com referência aos eventos despoletadores. A estrutura geral de um plano é definida por: *evento_despoletador* : *contexto* < *-corpo*.

Intenção: é uma pilha de planos parcialmente instanciados.

Agente: a circunstância do agente (estado atual), em determinado momento, é caracterizada por um conjunto de: (i) intenções; (ii) eventos; (iii) ações.

2.5.2 Módulo de Comunicação *AgentSpeak*

A arquitetura *AgentSpeak* propõe a utilização das diretivas KQML para a comunicação entre agentes. O tema da comunicação será abordado na secção 3.1.

A comunicação *AgentSpeak* é efetuada com o recuso a vários elementos elocutórios, dos quais performativos. Considerando r o agente recetor e s o agente emissor, identificam-se os seguintes performativos:

- **tell:** s tenciona que r acredite que s tem como crença o conteúdo da mensagem;
- **untell:** s tenciona que r acredite que s não tem como crença o conteúdo da mensagem;
- **achieve:** s pede a r para tentar atingir um estado do mundo onde o conteúdo da mensagem seja verdade;
- **unachieve:** s pede para r abandonar a intenção de atingir um estado do mundo onde o conteúdo da mensagem seja verdade;
- **tellHow:** s informa r de um plano;
- **untelHow:** s pede para r ignorar determinado plano;
- **askIf:** s quer saber se o conteúdo da mensagem é verdade para r ;
- **askAll:** s quer saber se o conteúdo da mensagem é verdade para todos os r ;
- **askHow:** s quer todos os planos de r para determinado evento despoletador.

2.5.3 Cooperação com *AgentSpeak*

O Coo-BDI (Cooperative BDI) [Jennings et al., 2004] foi definido para estender as linguagens de programação orientadas a agentes baseadas no modelo BDI.

A estratégia de cooperação de um agente é definida pelo programador e é caracterizada pela definição: (i) dos agentes com quem é expectável cooperar; (ii) uma política de retorno do plano; (iii) a política de aquisição do plano.

A política de retorno do plano consiste no conjunto de regras aplicadas para a troca de planos, em função de determinados eventos.

A estratégia de cooperação pode evoluir ao longo do tempo e adaptar-se, fornecendo uma maior autonomia e flexibilidade aos agentes. É modelada pelas funções:

- **trusted(Evento, Agentes)**: define o conjunto de agentes que são confiáveis para fornecer planos para um determinado evento despoletador;
- **retrievalPolicy(Evento, Policy)**: define a política sobre determinado evento, ou seja, se se deve pedir sempre planos, ou apenas pedir um quando não existe nenhum plano local para um determinado evento;
- **acquisitionPolicy(Evento, Policy)**: define o que fazer, quando um plano com origem num agente confiável e relativo a determinado evento é recebido. O plano pode ser descartado da biblioteca de planos, adicionado ou substituído.

2.5.4 Implementação *Jason* da Linguagem *AgentSpeak*

A plataforma *Jason* adapta a arquitetura *AgentSpeak*, sendo importante fazer referência aos processos: (i) atualização da base de crenças; (ii) verificar o correio (mensagens recebidas); (iii) filtro de mensagens; (iv) execução da intenção.

Esta plataforma efetua uma separação entre o processo de atualização da

base de crenças e o processo de geração de eventos. Esta separação permite aumentar a coesão das partes, garantindo que o sistema se comporta sempre da mesma forma face à alteração da base de crenças.

A base de crenças comporta as crenças dos outros agentes, sendo distinguidas das próprias pela indicação do remetente. Desta forma, o agente sabe que determinado agente tem determinada crença, podendo tomá-la como sua, ou não.

O processo de verificação do correio (mensagens recebidas) faz parte do módulo de comunicação da arquitetura e tem como objetivo verificar se a mensagem contém o formato definido pelo protocolo de comunicação. Cabe ao processo de filtro de mensagens verificar se as mensagens são “socialmente aceites”, ou seja, se se enquadram numa lógica de comunicação.

O processo de execução da intenção é estendido do *AgentSpeak*, fazendo distinção entre ações ambientais, objetivos a alcançar, objetivos de teste, notas mentais, ações internas e expressões.

O ciclo de execução da plataforma *Jason* encontra-se representado na figura 2.4.

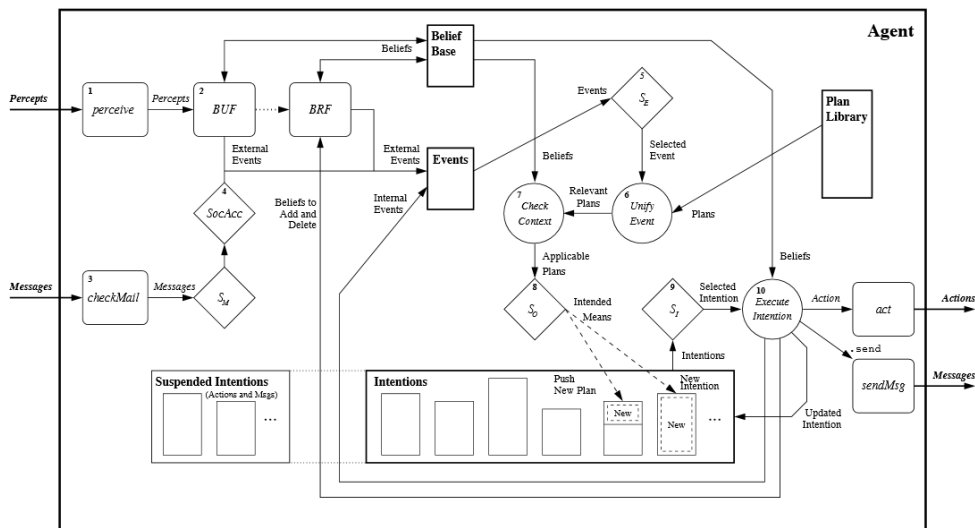


Figura 2.4: Ciclo de execução do *Jason* [Bordini et al., 2007].

2.5.5 Principais Características da Plataforma *Jason*

A comunicação é baseada nas diretivas *AgentSpeak*, sendo estendida de forma a permitir mais funcionalidades.

O mecanismo de recepção e envio de mensagens é assíncrono. As mensagens são gravadas numa caixa de correio e processadas no início do ciclo de raciocínio. Uma função de seleção de mensagens escolhe as mensagens a serem processadas, função essa que pode ser definida pelo programador. De seguida, as mensagens processadas passam por um filtro, no qual se verifica: (i) a validade da mensagem; (ii) se esta pode ou deve ser processada; (iii) o seu emissor (se tem ou não autorização para enviar a mensagem); (iv) o que fazer com a mensagem.

Em cada instante, o agente armazena a informação das mensagens recebidas, enviadas e intenções a aguardar resposta. As mensagens recebidas têm o formato (mid, id, ilf, cnt) , em que *mid* corresponde ao id da mensagem, *id* à informação do remetente da mensagem, *ilf* à força elocucionária e *cnt* ao conteúdo. A definição de força elocucionária pode ser encontrada em 3.1.1.

Aos elementos característicos dos planos (definidos pelo *AgentSpeak*), são adicionados dois parâmetros: (i) a fonte (*source*); (ii) a lista de acesso (*access specifier*). O primeiro determina a origem do plano (qual o agente que forneceu o plano), enquanto o segundo identifica o conjunto de agentes com o qual o plano pode ser partilhado. A partilha pode ser pública (para todos os agentes), privada (não é partilhada), ou apenas para agentes confiáveis.

A gestão do que fazer, quando não existe um plano para satisfazer um evento, recai sobre o programador ao definir se o interpretador deve descartar o evento ou colocá-lo no final da fila.

2.5.6 Outros Aspetos da Plataforma *Jason*

Mais complexa que o PRS, a plataforma *Jason* é uma plataforma multi-agente bem consolidada pela sua larga utilização.

Distinguem-se as seguintes características da arquitetura da plataforma *Jason*:

- é possível criar anotações para os planos, permitindo estender as suas funcionalidades;
- quando uma ação falha, ou não existe nenhum plano para o subobjetivo, o plano é removido da intenção;
- é possível executar ações internas (começam com um ”.”) e são codificadas em *Java*. No entanto, existem ações internas que são fornecidas de origem (como, por exemplo, um módulo de comunicação entre agentes);
- existe a possibilidade de verificar os desejos e intenções e largá-los sob qualquer circunstância.

Esta arquitetura aproxima-se mais das ideias deste trabalho, no que respeita à sintaxe e semântica da programação dos planos. Porém, a possibilidade dos planos conterem metainformação relevante para a definição do plano, para além de aumentar a entropia da programação de planos, contraria o processo mental natural da sua elaboração ao exigir um esforço adicional que se desvia da simples ordenação de tarefas e desejos a atingir, o que dificulta a sua programação.

Outro aspeto contrastante entre a plataforma *Jason* e as ideias deste trabalho reside na supersimplificação da semântica. O raciocínio é efetuado sobre um suporte simbólico, pelo que o tipo de simbolismo é pouco relevante para o bom funcionamento do raciocínio. Todavia, a simplificação da semântica poderá complicar a programação e compreensão dos planos definidos, pelo que, por não influenciar nenhum aspeto do raciocínio se defende a adoção de uma semântica de maior compreensão pelo programador.

Tal como o PRS, o *Jason* é uma plataforma BDI orientada a eventos, ou seja, os seus planos são ativados pelo despoletar de eventos no sistema. As desvantagens da dificuldade de filtragem encontram-se já abordadas anteriormente.

2.6 Conclusão

Neste capítulo foram estudadas arquiteturas de agentes reativos e deliberativos dando especial ênfase à arquitetura BDI. O estudo da arquitetura BDI foi aprofundado com o recurso à avaliação de duas plataformas BDI já existentes: a *Procedural Reasoning System* e a *Jason*.

O próximo capítulo visa o estudo de sistemas multi-agente que permitirão a integração da arquitetura de agente num contexto social. Os temas da comunicação entre agentes, do problema da coordenação e da interação social serão abordados, sendo este último tema aprofundado pelo estudo de três métodos de interação social: (i) raciocínio social, (ii) estruturas organizacionais (iii) e licitação.

CAPÍTULO 3

Sistemas Multi-Agente

A Inteligência Artificial Distribuída (IAD) [Weiss, 1999] compreende o estudo, construção e aplicação de sistemas multi-agente e o seu objetivo a longo prazo é desenvolver mecanismos e métodos que permitem aos agentes interagir tão bem como os humanos (ou ainda melhor) e compreender a interação entre entidades inteligentes, sejam elas computacionais, humanas, ou ambas.

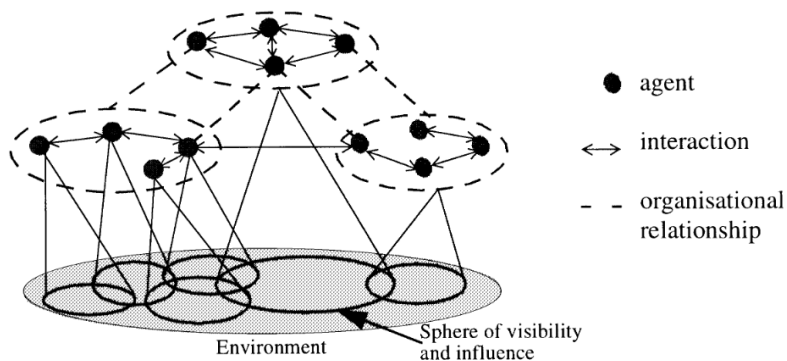


Figura 3.1: Sistema multi-agente padrão [Jennings, 1999].

Um sistema multi-agente é um sistema composto por mais do que um agente inteligente, os quais interagem socialmente (fig. 3.1). A coordenação orientada pelo objetivo, ou tarefa, é um dos elementos chave da interação em sistemas multi-agente [Weiss, 1999].

Relativamente aos sistemas centralizados, os sistemas multi-agente apresentam as seguintes vantagens [Moulin e Cloutier, 1994]:

- maior eficiência na resolução de problemas através da exploração de paralelismo;
- maior flexibilidade na resolução de problemas, permitindo que agentes com diferentes capacidades se coliguem dinamicamente;
- maior robustez em caso de falha de um, ou mais agentes, pela possibilidade de outros assumirem as suas responsabilidades de forma dinâmica.

A implementação de sistemas multi-agente permite também abstrair diversos paradigmas de programação (fig. 3.2). Esta abstração permite a independência de linguagens de programação e a integração em sistemas não baseados em agentes.

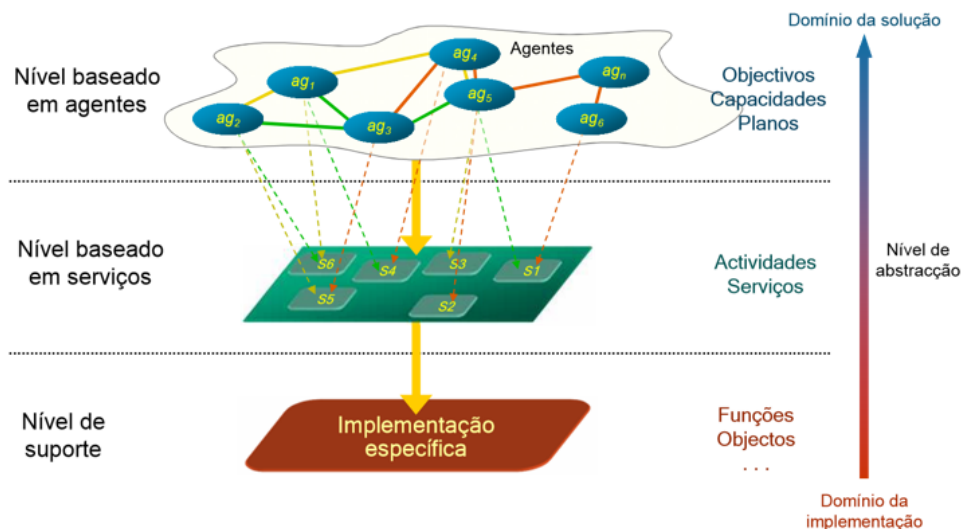


Figura 3.2: Integração de níveis de abstração [Morgado, 2014].

No entanto, os sistemas multi-agente centrados em agentes têm como

desvantagens [Jennings, 1999]:

- os padrões e os resultados das interações podem ser imprevisíveis;
- prever o comportamento total do sistema baseado nos seus componentes pode ser difícil (por vezes impossível) devido à possibilidade de comportamento emergente (eventualmente não desejado).

A **segurança** das aplicações pode ficar comprometida com a possibilidade de todos os agentes comunicarem com os restantes. Desta forma, é fácil um agente infiltrar-se no sistema, comprometendo o seu propósito [Ferber et al., 2004].

3.1 Comunicação entre Agentes

A comunicação assume um papel importante na interação entre agentes no âmbito de um sistema multi-agente, ao permitir a transmissão de informação entre agentes, a qual seria difícil, ou impossível, obter única e exclusivamente através da perceção.

3.1.1 Atos de Discurso

A teoria dos atos do discurso [Austin, 1962] e [Searle, 1969] tem como premissa que certos elementos elocutórios do discurso produzem alterações no mundo, da mesma forma que outro tipo de ações. Assume-se que as ações de discurso são efetuadas por agentes, da mesma forma que outras ações [Wooldridge, 2002].

Entende-se por **elementos elocutórios** todos os componentes relativos à forma da enunciação de um raciocínio através de uma forma de linguagem.

Um **performativo** é um verbo cujo enunciado realiza a ação que significa.

As condições necessárias para que haja conclusão com êxito de performativos são: (i) haver uma convenção aceite para o performativo, e as circunstâncias e os agentes têm de ser especificados no procedimento; (ii) o

procedimento tem de ser executado corretamente; (iii) o ato tem de ser fidedigno e qualquer condição tem de ser verificada, tanto quanto possível.

A identificação das categorias possíveis de performativos permite criar categorias de classificação de tipos possíveis de atos de discurso. Searle define as seguintes categorias de performativos [Searle, 1969]:

- **representativa:** compromete o locutor à verdade de uma proposição expressa (e.g. informar);
- **diretiva:** tentativa por parte do locutor de levar o recetor a efetuar alguma coisa (e.g. pedir);
- **comissiva:** comprometer o locutor com um curso de ação (e.g. prometer);
- **expressiva:** exprimir um estado psicológico (e.g. agradecer);
- **declarativa:** efetuar alterações em estados de relações institucionais (e.g. declarar guerra).

3.1.2 Linguagens de Comunicação

A *Knowledge Query and Manipulation Language* (KQML) e a *Knowledge Interchange Format* (KIF) são duas linguagens de comunicação baseadas/influenciadas pela teoria dos atos de discurso. Criadas no início da década de 90 pela *Knowledge Sharing Effort* - fundada pelo *Defence Advanced Research Projects Agency* (DARPA) -, estas linguagens são uma referência clássica à problemática da comunicação entre agentes.

A KIF [Genesereth e Fikes, 1992] é uma linguagem de comunicação cujo objetivo é a criação de um padrão de expressão de propriedades de um domínio em particular. Não versa a forma como a comunicação é efetuada, mas sim, a forma como o conteúdo da mensagem é enviado. Através da KIF os agentes poderão exprimir: (i) propriedades de objetos num domínio; (ii) relações entre objetos num domínio; (iii) propriedades gerais de um domínio [Wooldridge, 2002].

A KIF fornece um vocabulário para os agentes definirem objetivos em

particular e algumas funções padrão como, por exemplo, relações de ordem e operações aritméticas. A sua sintaxe é muito idêntica à da linguagem de programação LISP.

A KQML é uma linguagem baseada em mensagens para as quais define um formato comum. A uniformização do formato da mensagem serve de camada intermediária que permite interligar diversos sistemas com linguagens distintas. Cada mensagem é um objeto que contém um performativo e parâmetros com valores atribuídos que poderão ser, ou não, obrigatórios [Wooldridge, 2002].

Esta linguagem baseia-se na ideia de que os agentes que estabelecem comunicação com recurso a este protocolo podem estar implementados em diferentes linguagens e paradigmas. Por consequência, a representação interna do mundo de cada agente é mantida, possivelmente, em diferentes formas. De modo a permitir aos agentes a comunicação, é criada uma base virtual de conhecimento -*virtual knowledge base* - que não é mais do que uma “interface” de conhecimento que serve de interligação entre linguagens.

Os parâmetros principais de uma mensagem KQML são:

- **content:** conteúdo da mensagem;
- **force:** informa se o remetente da mensagem irá negar o conteúdo da mensagem;
- **reply-with:** informa se o remetente espera uma resposta e em caso afirmativo define um identificador para a resposta;
- **in-reply-to:** referência ao parâmetro anterior;
- **sender:** remetente da mensagem;
- **receiver:** recetor da mensagem;
- **language:** linguagem em que está definido o conteúdo;
- **ontology:** define a ontologia em uso na mensagem.

Diversas versões da linguagem KQML foram definidas, cada uma com os seus performativos específicos, mencionando-se a título exemplificativo:

- **achieve:** pedido para o recetor concretizar um objetivo;

- **ask-all:** pedido de todas as respostas para determinada questão;
- **forward:** pedido para o recetor reencaminhar uma mensagem para outro agente;
- **ready:** informação de que o agente se encontra disponível para responder a uma questão efetuada anteriormente.

O aprofundar da temática da comunicação poderá ter o trabalho de Vieira, Moreira, Wooldridge e Bordini [Vieira et al., 2007] como bom ponto de partida.

3.2 O Problema da Coordenação Multi-Agente

O problema da coordenação multi-agente consiste na necessidade de dois, ou mais agentes, orientarem uma ação conjunta, podendo este problema ter duas vertentes. A primeira, num ponto de vista de gestão de conflitos onde os agentes poderão ter o mesmo objetivo, mas não o poderão concretizar em simultâneo. A segunda abordagem é feita do ponto de vista de concretização de objetivos, onde um agente, por não ter a capacidade de concretizar um determinado objetivo, necessita de o delegar, tendo de decidir sobre qual o agente que apresenta as melhores garantias de concretização.

Os métodos de coordenação multi-agente fornecem soluções para o problema da coordenação multi-agente, ficando a escolha do melhor método dependente do problema em específico.

3.2.1 Métodos de Coordenação Multi-Agente

Existem quatro métodos principais de coordenação multi-agente [Sichman, 2003]: (i) o agrupamento, (ii) a especialização, (iii) a resolução distribuída de problemas e (iv) a coordenação de ação.

O método do **agrupamento** é bastante frequente em sistemas multi-agente compostos por agentes reativos. A simples aproximação dos agentes diminui a prevalência de problemas individuais, suplantada pela interação

com os restantes.

Através do método da **especialização** dos seus agentes em tarefas específicas e estruturadas dentro de um sistema multi-agente, é possível otimizar o comportamento geral, fomentando uma maior eficiência do sistema.

O método da **resolução distribuída** de problemas consiste na concretização de um objetivo por diversos agentes, onde cada agente concretiza parte do objetivo, fazendo convergir o sistema para a concretização do objetivo geral. A resolução distribuída de problemas é composta pelas seguintes fases [Smith e Randall, 1980]:

- decomposição do problema em subproblemas menores, sendo os subproblemas decompostos sucessivamente até um agente conseguir dar resposta ao subobjetivo;
- resolução dos problemas de forma isolada, ou seja, cada agente resolve o seu problema;
- sintetização da solução pela integração das soluções.

Existem dois tipos de abordagem para a decomposição de tarefas: (i) **centralizada**, através da existência de um agente especializado na subdivisão de tarefas, posteriormente distribuídas aos restantes agentes; ou (ii) **descentralizada**, podendo qualquer agente do sistema requisitar ou oferecer serviços.

Independentemente do tipo de abordagem escolhido, pode ser utilizada uma de duas técnicas para a distribuição/requisição de tarefas: (i) **técnicas informadas**, onde os agentes conhecem os restantes agentes e as suas capacidades e (ii) **técnicas não informadas**, caracterizadas por mecanismos de “licitação”, onde um agente questiona os demais sobre quem pode e se encontra disponível para realizar determinada tarefa.

O método da **coordenação de ação** compreende a realização de atividades suplementares que não seriam necessárias caso um único agente estivesse isoladamente a tentar concretizar um objetivo [Malone, 1988]. Este método pode ser aplicado através de:

- **sincronização:** encadeamento de ações realizadas pelos agentes do sistema;
- **planeamento de ações:** raciocinar sobre o conjunto de todas as ações a serem realizadas para atingir um objetivo e escolher o melhor plano. Um plano pode ser elaborado por apenas um agente (centralizado), por vários agentes em cooperação (descentralizado), ou por vários agentes interessados nos seus objetivos, os quais apenas cooperam para atenuar conflitos (descentralizado de planos distribuídos);
- **coordenação reativa:** os comportamentos reativos dos agentes estão estruturados de forma a coordenarem as suas ações;
- **coordenação por regulamentação:** implementação de regras sociais *a priori*, de forma a evitar potenciais conflitos.

O modelo de resolução cooperativa de problemas consiste em quatro fases [Wooldridge e Jennings, 1994]:

1. **deteção do potencial de cooperação:** inferência sobre se os objetivos e ações dos outros são complementares aos seus;
2. **formação de coligação:** organização dinâmica dos agentes destinada a concretizar um objetivo;
3. **formação de plano:** decisão comum sobre como atingir os objetivos;
4. **ação coligada:** execução do plano pelos diversos agentes.

Este modelo de resolução cooperativa de problemas serve de base aos métodos de interação social estudados a seguir, tendo para cada um dos métodos graus de elaboração distintos.

3.3 Interação Social

A **interação social** consiste no processo através do qual agentes se relacionam uns com os outros, sendo o resultado de relações objetivas existentes entre os elementos de uma sociedade (relações de dependência e poder) [Conte e Castelfranchi, 1995a]. Esta definição revela implicitamente que os elementos de uma sociedade não necessitam de ter conhecimento das suas relações com os demais, bastando que os objetivos, capacidades e planos de cada agente gerem relações de dependência e poder.

Podem ser identificados dois tipos de interação social: (i) a interação social positiva, que resulta da partilha do esforço de resolução de objetivos entre os agentes, e (ii) a interação social negativa, onde os agentes impedem outros de atingirem os respetivos objetivos.

A interação social positiva pode acontecer sob duas formas: (i) cooperação, quando os agentes possuem o mesmo objetivo, mas apenas o conseguem atingir se partilharem as suas capacidades; e (ii) troca social, quando os agentes possuem diferentes objetivos que não conseguem atingir por si só, trocam-nos e comprometem-se a concretizar o objetivo do agente parceiro em troca da concretização, por parte deste, de um objetivo seu.

No que se refere aos métodos de interação social, por serem os mais comuns, serão estudados os métodos: (i) do raciocínio social, (ii) da licitação e (iii) das estruturas organizacionais.

3.4 Raciocínio Social

O raciocínio social consiste no processo de determinação de relações de dependência e poder, bem como no processo de adoção de objetivos de outros agentes [Sichman et al., 1993]. Este raciocínio necessita de uma representação do mundo constituída pela representação dos outros agentes e do próprio agente, que poderá ser eventualmente falsa ou até incompleta. Neste modelo interno, um agente é representado pelos seus objetivos, planos, capacidades e recursos.

A determinação de relações de dependência e poder forma redes de dependência e é efetuada de acordo com a teoria da dependência.

3.4.1 Redes de Dependência

A teoria da dependência tem como objetivo dar resposta a dois problemas da interação social [Castelfranchi, 1990]: o problema da sociabilidade (porquê interagir socialmente) e o problema da adoção (como fazer com que um problema se torne comum). Estes dois problemas são abordados através das noções de dependência e poder, respetivamente.

As relações de dependência são inferidas tendo por base dois axiomas:

1. Um agente tem o poder de g , se este pode atingir o objetivo “g”.
2. Se um agente “A” tem a capacidade de ajudar, ou impedir um agente “B” de atingir um objetivo, diz-se que o agente “A” tem poder sobre “B”.

No contexto da teoria da dependência, a noção de **dependência** é definida da seguinte forma [Conte e Castelfranchi, 1995b]: um agente “D” depende de um agente “C” se: (i) um agente “C” tem a capacidade de realizar uma determinada ação que aumenta a probabilidade de um determinado estado do mundo ser atingido e (ii) o estado do mundo provável de atingir é um objetivo de um outro agente “D” que é incapaz de o concretizar.

As dependências podem ser: (i) **unilaterais**, quando apenas um agente depende de outro, ou (ii) **bilaterais**, quando os agentes dependem um do outro. As dependências bilaterais ainda podem ser divididas em dependência: (i) **mútua**, quando os agentes contribuem com as suas capacidades para atingir o mesmo objetivo, ou (ii) **recíproca**, quando cada agente precisa que o outro lhe concretize determinado objetivo, sendo os objetivos diferentes. A dependência mútua favorece a cooperação, enquanto a recíproca propicia a troca social.

A incapacidade de concretizar um objetivo poderá advir da falta de: (i) capacidades, (ii) recursos, (iii) ou ambos [Sichman et al., 1993].

A base motivacional de um agente na adoção de objetivos de outros agentes permite determinar o comportamento cooperativo deste. Existem quatro motivações para a adoção de um objetivo:

1. **adoção instrumental:** o objetivo adotado é considerado como forma de atingir um dos seus objetivos;
2. **adoção terminal:** o objetivo adotado é já um objetivo de alto nível;
3. **adoção cooperativa:** o objetivo é desejado por diversos agentes;
4. **benevolência:** o objetivo é adotado incondicionalmente.

O raciocínio social efetuado com base na teoria da dependência versa a elaboração de redes de dependência através da identificação de relações de dependência/poder e, nos casos em que se aplica, identificar e convencer os respetivos agentes de que ambos dependem um do outro. Poderão existir quatro situações de dependência:

- **LBMD - *locally believed mutual dependence*:** apenas o agente acredita que existem condições para uma cooperação;
- **MBMD - *mutually believed mutual dependence*:** ambos os agentes acreditam que existem condições para uma cooperação;
- **LBRD - *locally believed reciprocal dependence*:** apenas o agente acredita que é possível efetuar uma troca social para determinados objetivos;
- **MBRD - *mutually believed reciprocal dependence*:** ambos os agentes acreditam que é possível efetuar uma troca social para determinados objetivos.

O modelo de interação social que tem por base redes de dependência é um modelo bastante completo que permite a qualquer agente atingir os seus objetivos de uma forma eficiente. Contudo, devido à necessidade de elaborar um raciocínio com base em outros agentes, este tipo de raciocínio é caracterizado por uma grande complexidade, o que aumenta exponencialmente com a entrada de novos agentes no sistema.

O raciocínio social hierárquico apresenta-se como solução mitigadora da

elevada complexidade do raciocínio social ao permitir uma abstração dos agentes e respetivas características, no exercício do raciocínio.

3.4.2 Raciocínio Social Hierárquico

O raciocínio social hierárquico [Morgado e Gaspar, 2000] pretende minimizar o peso computacional do raciocínio social através da elaboração de redes de dependência a diferentes níveis de abstração. É proposta a criação e o agrupamento dos agentes constituintes do sistema multi-agente em coligações, que simplificam o processo de raciocínio ao abstraírem o conjunto de agentes numa coligação. Desta forma, o agente não necessita de conhecer e saber o que cada agente é capaz de fazer, necessitando apenas de saber quais as características da coligação (fig. 3.3).

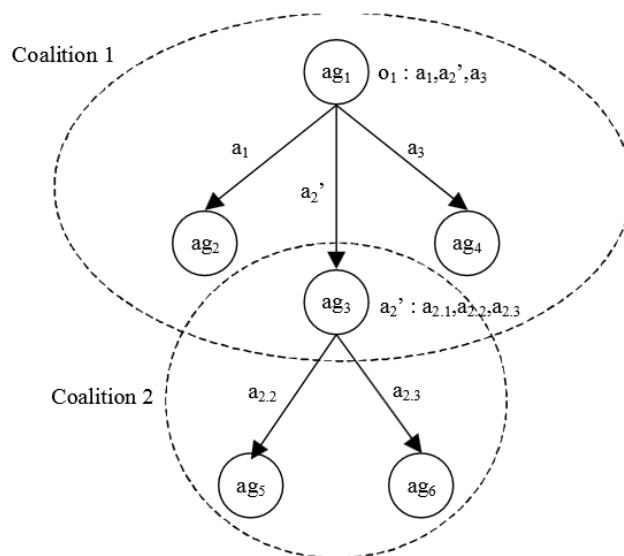


Figura 3.3: Exemplificação de coligações [Morgado e Gaspar, 2000].

Diz-se que um agente adota uma perspetiva local quando elabora o raciocínio tendo por base as coligações. Quando é adotada uma perspetiva local, a utilização dos recursos de processamento do agente diminui sem colocar em causa a qualidade das coligações criadas.

O raciocínio social hierárquico permite uma diminuição da carga de processamento dos processos internos do agente ao lidar com:

- **formação e seleção de coligações:** a existência de uma representação explícita das coligações permite a referência destas últimas como a composição de todos os seus elementos constituintes, tanto no processo de seleção, como no de negociação;
- **limitação do âmbito do raciocínio dos agentes:** ao facilitar o processo de raciocínio do agente pelo facto de não necessitar nem de saber, nem raciocinar sobre todos os agentes envolvidos e ações necessárias a realizar para atingir o seu objetivo;
- **redução do número de comunicações:** pela redução do número de agentes a contactar aquando da formação de coligação ao existirem “representantes” de cada coligação que servem de ponto de contacto.

Este método de raciocínio social apresenta uma diminuição da utilização de recursos computacionais de um agente inserido num ambiente multi-agente. No entanto, é feita com repercussões noutros agentes do sistema que têm de reencaminhar mensagens aos restantes elementos da coligação, os quais, ao serem o ponto de contacto com a coligação, adquirem essa responsabilidade. Ainda assim, torna-se compensatória pela razão entre a diminuição da utilização dos recursos computacionais utilizados no raciocínio social de um agente e o acréscimo da utilização de recursos no reencaminhamento de mensagens.

3.5 Estruturas Organizacionais

As estruturas organizacionais [Jennings, 1999] apresentam um paradigma de programação multi-agente cujo centro é o sistema multi-agente, representado pelo conceito de organização, ao invés do agente.

Neste contexto, entende-se por organização um conjunto de agentes com papéis específicos que têm determinadas relações uns com os outros e que fazem parte de padrões sistemáticos institucionalizados de interações entre agentes com diferentes papéis [Wooldridge et al., 2000].

As características principais de uma organização são [Ferber et al., 2004]:

- uma organização é constituída por agentes que têm comportamentos;
- a organização geral pode ser dividida em partições que se sobrepõem;
- os comportamentos dos agentes estão relacionados com a atividade geral da organização;
- os agentes têm relações dinâmicas que podem ser definidas usando uma taxonomia de papéis, tarefas ou protocolos;
- os tipos de comportamentos relacionam-se pelas relações entre papéis, tarefas e protocolos.

Uma organização multi-agente não é considerada em termos de estados mentais, mas apenas em termos de capacidades e restrições na ótica da organização, como os papéis (funções ou posições), grupos (ou comunidades), tarefas (ou atividades) e protocolos de interação (ou estruturas de diálogo), além do que relaciona a estrutura da organização com o comportamento observável dos seus agentes [Ferber et al., 2004].

Uma organização é caracterizada pela sua definição estrutural e dinâmica [Grossi et al., 2005].

A estrutura de uma organização é constituída por duas vertentes: (i) estrutura das partes e (ii) estrutura de papéis. A estrutura das partes define como os agentes são integrados em grupos e como estes últimos se relacionam uns com os outros. A estrutura de papéis é definida para cada grupo e corresponde a um conjunto de papéis e respetivas relações, definindo também as restrições que os agentes devem satisfazer para assumir os papéis e os respetivos benefícios.

A dinâmica de uma organização compreende os padrões de interação da instituição (que são definidos nos papéis). A dinâmica também define: (i) os protocolos de integração e abandono dos agentes; (ii) de que forma são aplicados os protocolos e controladas as obrigações e permissões; (iii) como a estrutura da organização está relacionada com os comportamentos dos agentes.

O **papel** é a descrição de uma posição/função que compõe uma organização. Descreve as restrições, os benefícios e os padrões de interação que um agente terá ao desempenhar a função.

3.5.1 Modelo AGR

O modelo AGR (*Agent, Group, Role*) [Ferber et al., 2004] define um sistema multi-agente centrado na organização.

Segundo este modelo (fig. 3.4), um **agente** é um membro ativo e comunicante que assume papéis dentro de grupos, podendo um agente assumir vários papéis dentro de vários grupos.

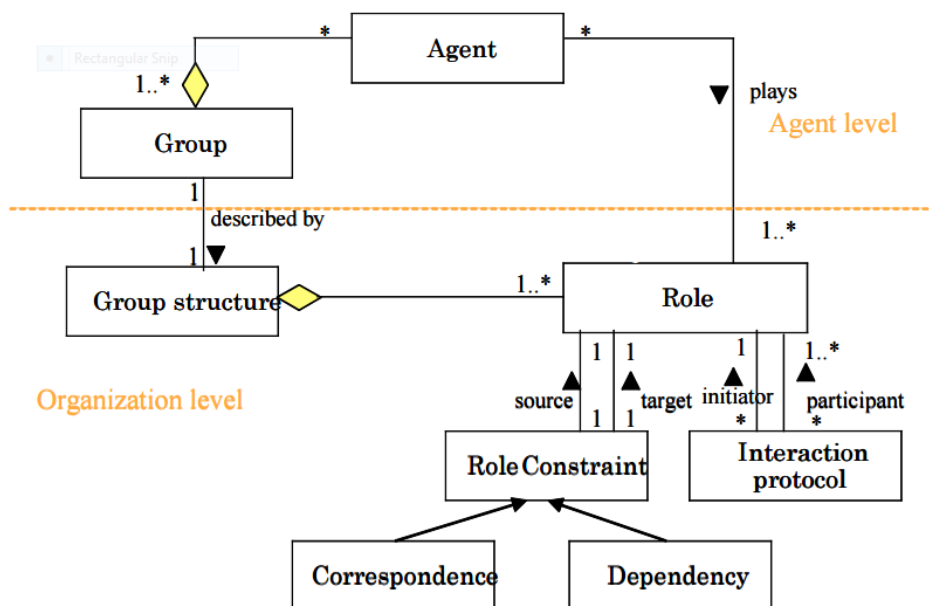


Figura 3.4: Metamodelo UML do AGR [Ferber et al., 2004].

Um **grupo** é um conjunto de agentes que partilham características comuns e é usado para particionar organizações.

Um **papel** é uma representação abstrata de uma posição funcional de um agente num grupo.

Segundo o modelo AGR:

- qualquer agente é membro de, pelo menos, um grupo;
- dois agentes apenas podem comunicar se forem membros do mesmo grupo;
- todos os agentes assumem um papel no grupo;
- um agente é um membro do grupo no qual assume um papel;
- cada papel é característico da estrutura do grupo.

Princípios do Modelo AGR:

1. O nível organizacional identifica a estrutura dos padrões de comportamento dos agentes, mas não como estes se comportam;
2. O nível organizacional não deve ter conceitos mentais como crenças, desejos, intenções, objetivos, ou outros. Deve fornecer apenas descrições de comportamentos esperados;
3. Os grupos estabelecem fronteiras cujo funcionamento interno apenas é conhecido pelos agentes que o integram.

O estabelecimento explícito de redes de dependência pela caracterização dos grupos de uma organização na sua estrutura, funcionamento interno e ligação com outros grupos, permite a um agente enquadrado na organização interagir socialmente sem despendar recursos computacionais na formação e manutenção destas redes de dependência.

Organizações estruturadas em pirâmide, para além da redução da utilização dos recursos computacionais referidos anteriormente, permitem ainda uma especialização dos seus agentes em diversos níveis de resolução do problema, proporcionando uma focagem dos recursos computacionais.

3.6 Licitação

A licitação é um método de interação social que consiste na distribuição de tarefas através de um método de oferta e procura. O agente que pre-

tende delegar uma tarefa anuncia-a a todos os potenciais interessados, sendo delegada ao agente que oferecer as melhores condições para a execução da tarefa.

3.6.1 Protocolo *Contract Net*

O *Contract Net* é um protocolo que concretiza o método da licitação e compreende três passos:

1. **anúnciação da tarefa:** um agente anuncia a existência de uma tarefa. Os restantes agentes verificam se são elegíveis para essa tarefa;
2. **licitação:** os agentes elegíveis licitam as suas contrapartidas para a tarefa em questão;
3. **atribuição:** o agente anunciador atribui a tarefa a um determinado agente, anunciado-lha.

O protocolo *Contract Net* é um modelo de interação social bastante simples, pois não aplica nenhum tipo de raciocínio social. Não há nem a necessidade dos agentes terem uma representação interna dos restantes, nem a de elaborar raciocínios sobre quais é que poderão ser bons parceiros para a conclusão de um objetivo (assumindo que todos os agentes licitam de boa fé).

As desvantagens deste protocolo advêm do número elevado de mensagens trocadas comparativamente com o modelo de redes de dependência [Sichman, 2003]. Este elevado número de mensagens pode comprometer a eficiência do sistema, sobretudo se a comunicação for um ponto crítico.

3.7 Conclusão

No presente capítulo, foi abordado o tema dos sistemas multi-agente concretizado na abordagem à temática da comunicação e a da coordenação entre agentes, com um maior aprofundamento nos métodos de interação social.

Verificou-se que o raciocínio social está presente em quase todos os métodos de interação social, fornecendo grandes vantagens a nível da qualidade da coordenação de ação. No entanto, o raciocínio social é pesado computacionalmente, seja em termos de carga de processamento, seja em termos de memória. Para minimizar a utilização dos recursos computacionais derivados do raciocínio social, existem diversos métodos de coordenação e interação social, dos quais se estudou: o raciocínio social hierárquico e as estruturas organizacionais.

A eficiente operação de sistemas multi-agente depende não só da adequação dos métodos de coordenação e interação social, como também da eficiência do funcionamento interno de cada um dos agentes. Sendo o agente o elemento base que compõe um sistema multi-agente, é necessário dotá-lo de características que lhe permitam operar em tempo real e com as suas limitações de recursos.

No próximo capítulo é efetuado um estudo sobre o problema da limitação de recursos em agentes inteligentes, com especial ênfase dada aos obstáculos que surgem da respetiva limitação em agentes modelados por processos cognitivos. A regulação cognitiva será apresentada como um dos métodos para lidar com a racionalidade limitada, sendo abordado o Modelo Emocional de Fluxo que servirá como ferramenta à implementação de uma regulação cognitiva de base emocional.

CAPÍTULO 4

O Problema da Limitação de Recursos

A temática de agentes inseridos em ambientes de recursos limitados pode ser abordado de duas perspectivas. A primeira prende-se com a limitação de recursos do próprio agente. A segunda segue o estudo de arquiteturas de agente que competem entre si, de forma a adquirirem recursos necessários à concretização dos seus objetivos

O estudo de arquiteturas que tentam tirar o melhor proveito de um agente com recursos limitados tem como objetivo aproximá-las do mundo real, onde agentes inteligentes têm de operar com capacidade de resposta em tempo real e, confinados às suas arquiteturas, têm de realizar as tarefas que permitem concluir os seus objetivos. Esta limitação de recursos poderá ter diferentes formas, e.g. a energia disponível para a operação do sistema, a velocidade de processamento, a capacidade de memória e a própria capacidade de raciocínio.

É na linha do estudo de agentes com recursos limitados que se pretende desenvolver este trabalho, sobretudo na limitação da capacidade de processamento e de memória, sendo este capítulo orientado nesse sentido.

4.1 Coordenação de Agentes com Recursos Limitados

Quando um agente pretende delegar uma tarefa necessita de utilizar um método de coordenação de ação, implementando o modelo de resolução cooperativa de problemas apresentado na secção 3.2. Neste modelo, a deteção do potencial de cooperação e a formação de coligação são, tipicamente, problemas do tipo NP-completo em termos de complexidade computacional [Inverno et al., 1997].

Para limitar a complexidade deste tipo de problemas poderão ser utilizadas diversas técnicas ou arquiteturas como, por exemplo, metarraciocínio [Anderson e Oates, 2005], raciocínio social hierárquico (subsecção 3.4.2), estruturas organizacionais (secção 3.5), ou um sistema baseado em licitação (secção 3.6).

Independentemente do método de coordenação escolhido, na eventualidade de mais do que um agente poder concretizar a tarefa pretendida, o agente coordenador terá sempre o problema de decidir qual o melhor agente a quem delegar uma tarefa. Este problema designa-se por problema da delegação de tarefas.

4.1.1 O Problema da Delegação de Tarefas

O problema da delegação de tarefas surge quando um agente necessita de realizar determinada tarefa para concretizar um objetivo e não possui as capacidades para a executar, necessitando de delegar essa tarefa. A figura 4.1 representa um exemplo deste problema, no qual um agente ag_1 possui o objetivo o_1 , para o qual existe um plano que envolve a realização de três tarefas: a_1 , a_2 e a_3 . Como o agente ag_1 não tem as capacidades necessárias para a realização das respetivas tarefas, delega-as aos agentes ag_2 , ag_3 e ag_4 .

Para delegar tarefas, o agente coordenador necessita de executar as quatro fases do modelo de resolução cooperativa de problemas definido na secção 3.2. É no momento da formação da coligação - ponto dois deste modelo - que o

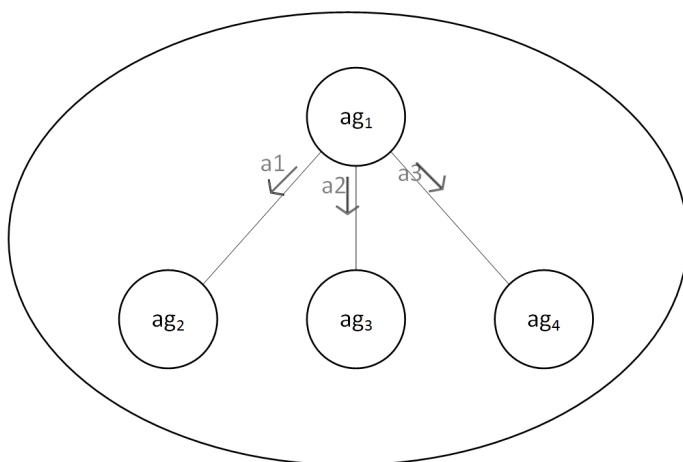


Figura 4.1: Delegação de tarefas.

agente se poderá defrontar com o facto de mais do que um agente subordinado ser capaz de realizar a tarefa que pretende delegar. A forma como o agente coordenador consegue determinar quais os agentes subordinados capazes de realizar a tarefa depende do método de coordenação definido.

Na figura 4.2 exemplifica-se o problema da delegação de tarefas. O agente coordenador ag_1 pretende delegar a tarefa a_2 e sabe que os agentes ag_2 e ag_3 são capazes de a realizar.

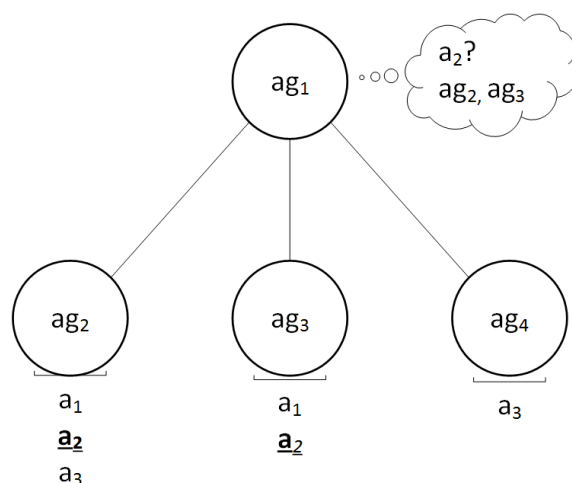


Figura 4.2: Problema da delegação de tarefas.

O agente ag_1 deve agora delegar a tarefa ao agente subordinado que a con-

segue realizar de forma mais eficiente, garantindo assim a máxima eficiência do sistema. A solução para este problema não é fácil, sobretudo quando resolvida por agentes com limitação de recursos, devido ao grande número de variáveis que o influenciam como, por exemplo, o número e o respetivo custo de execução de tarefas já delegadas aos subordinados ou o potencial de execução que estes últimos possuem.

4.2 Racionalidade Limitada

A racionalidade limitada é uma consequência da limitação de recursos e caracteriza-se pela não garantia de produção de soluções ótimas por consequência da incapacidade do agente em processar toda a informação sobre um determinado problema.

A racionalidade limitada também pode resultar de informação incompleta sobre o mundo, devido: (i) à limitação dos sensores, ou (ii) à reduzida capacidade de manipulação de volumes de informação, que forçam o agente a limitar a informação sobre a qual recairá o processo de raciocínio. Esta limitação na manipulação de volumes de informação, pode dever-se à capacidade de processamento, memória, energia, ou uma combinação das anteriores.

Neste contexto, o processo de raciocínio é limitado podendo não ser capaz de fornecer uma solução ótima por não ser alcançável com os recursos existentes, ou seja, o processo de decisão deve ser orientado para um nível de satisfação subótimo dos objetivos do agente, tendo de ser encontrado para cada problema o rácio de satisfação vs. otimização dos objetivos do agente.

Nesse sentido, a capacidade de lidar com a complexidade associada aos processos envolvidos na concretização dos objetivos de um agente é um aspeto essencial no âmbito da racionalidade limitada.

Os principais métodos para lidar com a complexidade são a abstração, a modularização, a aproximação e a focagem [Morgado, 2013].

A **abstração** corresponde à redução do detalhe, que resulta na diminuição da complexidade de representação. É um método não completo que pode

apresentar problemas na qualidade da solução.

Entende-se por **método não completo** todo o método que pode não encontrar solução, não obstante a sua existência.

A **modularização** obtém-se pela redução da complexidade por decomposição combinatória, ao dividir um problema em problemas mais pequenos e por composição da solução dos vários problemas na solução final.

A **aproximação** consiste na utilização de métodos aproximados, nomeadamente heurísticos, na resolução de um problema. Nesse sentido, embora permitam resolver um problema com menor complexidade computacional, têm a imprecisão como custo.

A **focagem** corresponde ao processamento seletivo de informação ou processamento relevante para a resolução de um problema. A focagem pode ser realizada na dimensão: (i) espacial, por limitação de memória ou por mecanismos de atenção, ou (ii) temporal, feita por limitação de tempo de processamento.

A escassez de recursos computacionais de um agente e a sua necessidade de adaptação ao ambiente exigem-lhe uma adaptação na utilização desses mesmos recursos, por exemplo, encaminhando-os para determinados processos, em detrimento de outros, de acordo com o seu estado. A regulação cognitiva é um dos métodos que permitem ao agente realizar esta adaptação na utilização dos seus recursos, lidando assim com a sua escassez.

4.3 Regulação Cognitiva

A regulação cognitiva é uma abordagem ao problema da racionalidade limitada que permite adaptar a utilização dos recursos utilizados pelos processos internos do agente, face ao seu estado atual. A regulação cognitiva pode ter uma base metacognitiva ou emocional.

4.3.1 Regulação Cognitiva por Metacognição

A metacognição é uma das abordagens à regulação cognitiva do agente (fig. 4.3). Nesta abordagem existe um processo responsável por monitorizar e regular todos os processos internos do agente.

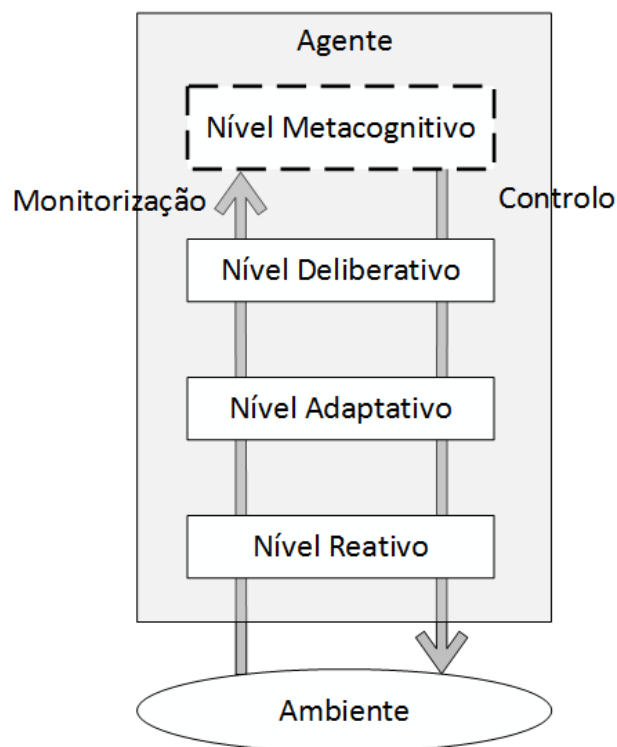


Figura 4.3: Modelo geral metacognitivo.

Sinais de monitorização resultantes da perceção e dos processos internos servem de representação do estado atual. Esta representação serve de base a um metarraciocínio que produz sinais de controlo para os processos internos.

Este método é computacionalmente complexo ao necessitar de uma perspetiva global e um conhecimento extensivo tanto dos processos internos do agente, como do meio envolvente.

4.3.2 Regulação Cognitiva de Base Emocional

A regulação cognitiva de base emocional [Morgado e Gaspar, 2005, 2011] pode ser utilizada de forma isolada ou como complemento à regulação com base em metacognição.

Numa arquitetura de agente de regulação cognitiva apenas de base emocional, é utilizada uma componente adaptativa (*Base Mechanisms* - fig. 4.4) capaz de regular os processos cognitivos através de mecanismos afetivos e emocionais.

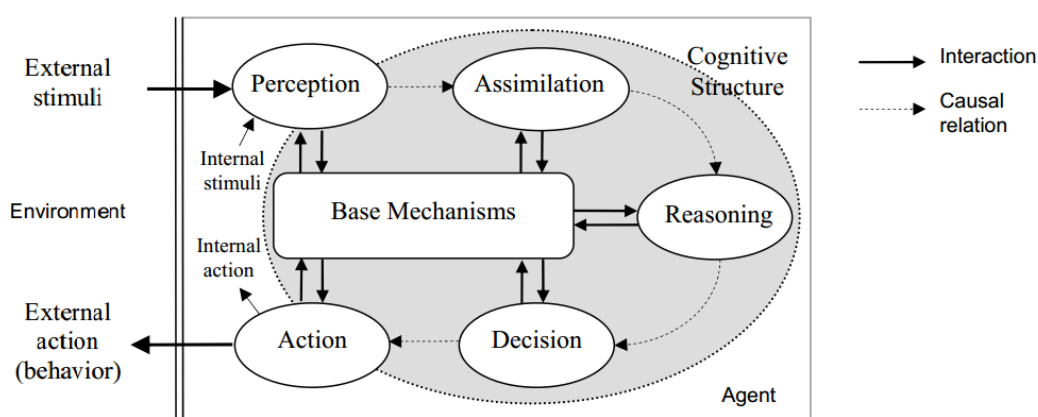


Figura 4.4: Arquitetura genérica de agente de modelo de fluxo [Morgado e Gaspar, 2007].

Este método utiliza menos recursos computacionais ao ter uma perspectiva local dos processos internos do agente e ao não necessitar de um conhecimento geral complexo do agente, nem do mundo em que opera.

4.4 O Modelo Emocional de Fluxo

O Modelo Emocional de Fluxo [Morgado, 2005] é um modelo de emoção onde os fenómenos emocionais são modelados através das dinâmicas da atividade cognitiva, de modo a suportar a regulação cognitiva de base emocional. De forma a preservar a natureza contínua e a dinâmica dos fenómenos emocionais, estas dinâmicas baseiam-se em dois aspetos principais: (i) a relação

entre o agente e o ambiente e (ii) a capacidade de um agente de lidar com a situação em que se encontra.

A relação entre o agente e o ambiente é determinada pela relação entre o potencial do agente lidar com a situação (potencial de concretização) e a condutância no acoplamento com o ambiente (condutância de concretização). O **potencial de concretização** (P) corresponde ao potencial do agente em realizar as alterações do mundo necessárias para atingir os seus objetivos, enquanto a **condutância de concretização** (C) representa o nível de resistência ou favorecimento do ambiente às alterações. Da relação entre o potencial e a condutância de concretização resulta um **fluxo de concretização** (F) que corresponde às alterações do mundo efetuadas conducentes à concretização das motivações do agente, podendo ser determinado pela relação entre o potencial de concretização e a condutância de concretização.

Dois dinâmicas base suscetíveis de influenciar o agente podem ser identificadas: (i) a variação temporal do potencial de concretização (δP) e (ii) a variação temporal do fluxo de concretização (δF):

$$\delta P = \frac{dP}{dt} \quad e \quad \delta F = \frac{dF}{dt} \quad (4.1)$$

Estas duas dinâmicas servem de base à geração de disposições emocionais.

4.4.1 Disposições Emocionais

Através das equações descritas em 4.1, é possível identificar padrões básicos de evolução do processo de concretização das motivações.

Uma variação positiva do potencial de concretização ($\delta P > 0$) significa que o agente está a conseguir lidar com a situação, pelo facto de o seu potencial de concretização se encontrar a crescer. Neste caso, dois padrões de evolução podem ser identificados:

1. o fluxo de concretização é também positivo ($\delta F > 0$), significando que para além do agente se encontrar a lidar bem com a situação, esta última encontra-se a evoluir favoravelmente. Este padrão de evolução

encontra-se associado a disposições emocionais de *alegria/felicidade*;

2. o fluxo de concretização é negativo ($\delta F < 0$), o agente consegue lidar com a situação, mas esta evolui desfavoravelmente. Este padrão de evolução encontra-se associado a disposições emocionais de *frustração* e *zanga*.

Quando o potencial de concretização evolui desfavoravelmente ($\delta P < 0$), o agente encontra-se numa situação com a qual não consegue lidar, pois o seu potencial de concretização encontra-se a decrescer. Neste caso, outros dois padrões de evolução podem ser identificados:

1. o fluxo de concretização é também negativo ($\delta F < 0$), significando que para além do agente não conseguir lidar com a situação, esta última encontra-se a evoluir também negativamente. Este padrão encontra-se associado a disposições emocionais de *apreensão/medo*, por ser a pior situação para o agente;
2. o fluxo de concretização é positivo ($\delta F > 0$), iniciando uma situação de recuperação na qual o agente não consegue lidar bem com a situação, mas esta encontra-se a evoluir favoravelmente. Este padrão de evolução está associado a disposições emocionais de *desânimo* ou *tristeza*.

A disposição emocional (ED) representa as dinâmicas emocionais geradas pelo potencial de concretização e pelo fluxo de realização em forma de vetor.

$$ED \equiv (\delta P, \delta F) \quad (4.2)$$

A disposição emocional (ED) representa as dinâmicas emocionais geradas pelo potencial de realização e pelo fluxo de realização em forma de vetor.

$$ED \equiv (\delta P, \delta F) \quad (4.3)$$

A noção de disposição emocional é definida como uma tendência reguladora de ação, não sendo por isso, uma emoção propriamente dita. Neste modelo as emoções surgem do efeito da disposição emocional pelos diversos níveis de organização que geram fenómenos emocionais cada vez mais ricos [Morgado e Gaspar, 2008].

Num determinado instante $t = \tau$ o vetor disposição emocional (ED) é definido por uma orientação e uma intensidade:

$$Quality(ED) \equiv arg(ED) \quad (4.4)$$

$$Intensity(ED) \equiv ED \quad (4.5)$$

Cada quadrante do espaço dimensional ($\delta P \times \delta F$) pode ser diretamente relacionado a um tipo específico de disposição emocional qualitativa (fig. 4.5.b) de acordo com os padrões e evolução descritos anteriormente.

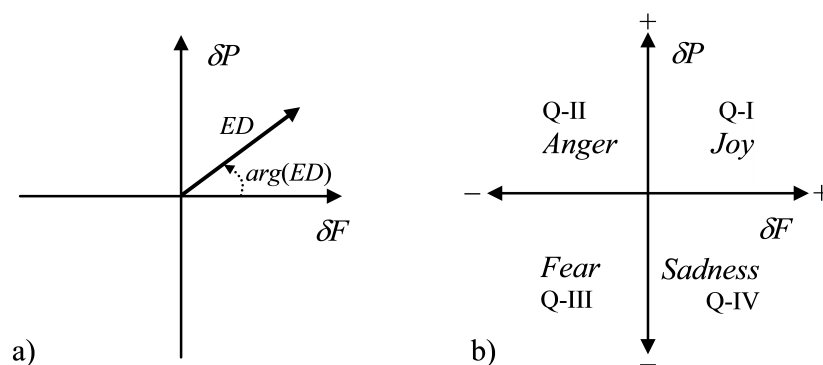


Figura 4.5: Vector ED como função de δP e δA (a); relação entre os quadrantes ED e a tendência qualitativa emocional. [Morgado e Gaspar, 2008]

4.4.2 Modelo de Agente

Para modelar agentes sob o Modelo Emocional de Fluxo é necessária uma representação interna que define a estrutura cognitiva do agente.

A **estrutura cognitiva** de um agente é o suporte através do qual um agente mantém um estado interno que reflete a evolução da interação entre o agente e o ambiente ao longo do tempo. Para efeitos de modelação das dinâmicas cognitivas e emocionais, a evolução ao longo do tempo dos elementos constituintes da estrutura cognitiva é descrita num espaço abstrato multidimensional, designado espaço cognitivo.

O **espaço cognitivo** é um espaço de sinais multidimensional, onde cada dimensão descreve um elemento cognitivo [Morgado, 2005]. Um elemento

cognitivo pode ser visto como o análogo da noção clássica de estado mental [Bratman, 1987], devido à sua existência limitada no tempo e inerentemente transiente [Morgado, 2005].

No Modelo Emocional de Fluxo, três tipos principais de elementos cognitivos são identificados: (i) observações, (ii) motivadores e (iii) mediadores.

As **observações** representam o estado atual observado pelo agente (externo e interno). Os **motivadores** representam o estado que o agente pretende atingir. Os **mediadores** representam os meios de que o agente dispõe para atingir os motivadores.

O processo de realização da motivação, do qual surgem as disposições emocionais, pode ser descrito como a relação entre a situação atual (observação) e o estado pretendido (motivador). Internamente, o agente irá tentar maximizar as alterações que levam à redução da distância entre a observação e os motivadores através do uso de mediadores.

No espaço cognitivo, o processo de concretização de motivadores pode ser descrito pela movimentação dos elementos cognitivos, onde motivadores e observações correspondem a posições no espaço cognitivo e os mediadores definem direções de movimento tal como ilustrado na figura 4.6 (φ_1 e φ_2 representam os sinais de dimensão base que definem o espaço e $\sigma(\tau)$ representa um elemento cognitivo em determinado instante).

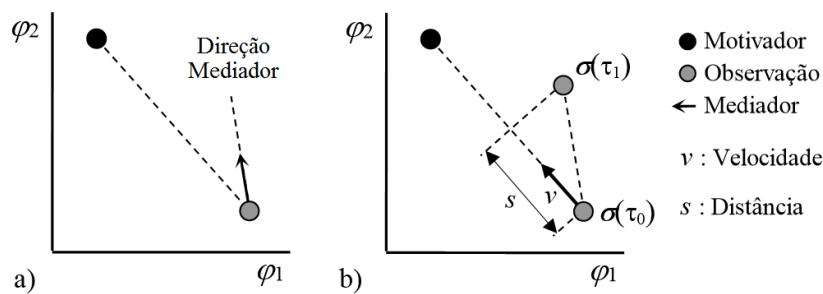


Figura 4.6: Realização de um motivador num espaço cognitivo bidimensional [Morgado e Gaspar, 2008].

O sinal das dinâmicas δP e δF representa a direção do movimento: positivo, se for um movimento em direção ao motivador, ou negativo, caso

contrário. Inserido num ambiente dinâmico onde, em determinado instante, diversos fatores podem contribuir para variação do potencial e fluxo de concretização do agente de forma positiva ou negativa, caracteriza-se a dinâmica geral de δP e δF por:

$$\delta P = \delta P^+ - \delta P^- \quad e \quad \delta F = \delta F^+ - \delta F^- \quad (4.6)$$

As dinâmicas instantâneas δP e δF poderão ser acumuladas, dando origem a uma disposição emocional cumulativa.

4.4.3 Disposição Emocional Cumulativa

A disposição emocional cumulativa (ED_c) corresponde à acumulação das disposições emocionais ao longo do tempo, caracterizando dinâmicas emocionais de longa duração. ED_c pode ser descrito por:

$$ED_c \equiv (P, F) \quad (4.7)$$

$$P = \int_t (\gamma_p^+ \delta P^+ - \gamma_p^- \delta P^-) \quad (4.8)$$

$$F = \int_t (\gamma_f^+ \delta F^+ - \gamma_f^- \delta F^-) \quad (4.9)$$

onde γ_p^+ , γ_p^- , γ_f^+ and γ_f^- são coeficientes de sensibilidade. Os coeficientes de sensibilidade determinam o grau de influência de cada um dos elementos da dinâmica geral de δP e δF , ou seja, o valor destes coeficientes permitem regular a influência das disposições emocionais em termos da sensibilidade ao carácter positivo ou negativo dessas disposições emocionais.

À disposição emocional cumulativa podem ser aplicados os mesmos padrões de evolução identificados para a disposição emocional, servindo tanto a disposição emocional, como a disposição emocional cumulativa, de base para a geração de estados motivacionais centrais.

4.4.4 Estados Motivacionais Centrais

As disposições emocionais servem de base a fenómenos emocionais mais complexos e ao comportamento adaptativo dos agentes, principalmente com

Quadrante ED	Situação	Emoção	Comportamento
Q-I	Favorável	<i>Alegria</i>	Avançar
Q-II	Adverso	<i>Zanga</i>	Reagir
Q-III	Perigo	<i>Medo</i>	Afastar
Q-IV	Perda	<i>Tristeza</i>	Recuperar

Tabela 4.1: Situação, emoção e comportamento.

restrições de recursos. Em particular, as disposições emocionais cumulativas originam estados que predispõem a determinados tipos de ação, no sentido de lidar com as situações de concretização de forma antecipatória. Esses estados são por isso designados *estados motivacionais centrais*, na medida em que têm um efeito global no processamento interno do agente, que determina o seu comportamento.

Considerando que a disposição emocional qualitativa caracteriza padrões de dinâmicas de concretização dos motivadores, é possível identificar padrões de comportamentos associados aos quatro quadrantes da disposição emocional, tal como representado na tabela 4.1.

Numa situação favorável (Q-I), o agente deve manter o comportamento originado no sentido da situação observada, uma vez que esta é favorável. Numa situação adversa (Q-II), o agente necessita de mobilizar recursos no sentido de ultrapassar a situação observada, uma vez que esta é desfavorável. Quando se encontra numa situação de perigo (Q-III), o agente deve parar e proteger os seus recursos, afastando-se da situação. Confrontado com uma situação de perda (Q-IV), o agente necessita de recuperar para voltar a atingir o seu potencial de concretização.

O Modelo Emocional de Fluxo caracteriza as disposições emocionais numa forma vetorial. Esta representação vetorial, com qualidade e intensidade, possibilita uma modelação da disposição emocional que preserva a sua continuidade, ou seja, a transição entre as diversas disposições emocionais dá-se de uma forma continuada no tempo pela definição da disposição emocional atual face à situação do agente no momento, ponderada pela disposição emocional do momento anterior (disposição emocional acumulada).

A disposição emocional, gerada por este modelo, serve de base à regulação dos processos internos da arquitetura e política de delegação de tarefas propostas. O seu carácter vetorial, permitirá orientar a regulação dos processos internos da arquitetura e a política de delegação de tarefas ao fornecer informações não só da qualidade da disposição emocional, como da sua intensidade.

4.5 Conclusão

Serviu o presente capítulo para abordar o problema da limitação de recursos em arquiteturas de agentes inteligentes no qual, do ponto de vista do sistema multi-agente, se estudou que a coordenação de agentes poderá ser afetada por esta limitação, necessitando de uma abordagem especializada para este tipo de problema. Singularmente, estudou-se que a racionalidade limitada pode ser gerida através de regulação cognitiva de base metacognitiva ou emocional.

O próximo capítulo propõe uma arquitetura de agente híbrida capaz de interagir socialmente em ambientes de recursos limitados. A inclusão na arquitetura de disposições emocionais irá permitir a regulação dos processos internos, permitindo focar os recursos computacionais em processos prioritários. A criação de uma estrutura organizacional em pirâmide versátil irá permitir uma adaptação do próprio sistema ao ambiente em que se insere e diminuir os recursos computacionais necessários à coordenação de ação.

CAPÍTULO 5

Arquitetura de Agente Proposta

O presente capítulo propõe uma arquitetura de agente capaz de suportar interação social em ambientes de recursos limitados. A operação eficiente neste tipo de ambientes exige uma arquitetura de agente com diversos níveis de decisão e o seu funcionamento tem de se adaptar às condições em que se encontra o agente. A dinâmica interna do sistema multi-agente tem de proporcionar uma interação social cuja prioridade é a diminuição dos recursos computacionais dos seus elementos, sem comprometer, de forma crítica, a qualidade da solução.

A arquitetura de agente proposta é dotada de um subsistema reativo capaz de implementar mecanismos de reação rápida, equipando-a com capacidade de resposta em tempo real em situações de emergência.

Pela sua complexidade interna, a arquitetura de agente é dotada de uma componente adaptativa que gera disposições emocionais. As disposições emocionais terão duas finalidades: (i) a regulação dos processos cognitivos internos e (ii) permitir uma distribuição de tarefas de acordo com o método de coordenação baseado em mecanismos emocionais, proposto neste trabalho.

Ao nível deliberativo, a arquitetura de agente é dotada de um motor BDI cujo processo deliberativo tem em consideração a limitação de recursos do próprio agente ao analisar constantemente a viabilidade das suas capacidades. A inserção numa estrutura organizacional, cuja coordenação é efetuada através de um planeamento centralizado, exige que da deliberação possa resultar a determinação de realização de diversas tarefas em simultâneo, possibilitando aos agentes coordenadores a realização de ações e/ou a delegação de tarefas em simultâneo e aos agentes subordinados a execução de diversas tarefas em simultâneo.

É escolhido um motor BDI para concretizar a componente deliberativa do agente por permitir uma adequada integração entre diferentes níveis cognitivos, nomeadamente a integração com os mecanismos emocionais, definidos no nível adaptativo.

A componente social da arquitetura de agente mantém o agente enquadrado na estrutura organizacional proposta, suportando a delegação e adoção de tarefas.

A arquitetura social proposta é composta por uma estrutura organizacional hierárquica, cuja coordenação é efetuada através de planeamento centralizado que se inicia no topo da hierarquia. Esta solução permite uma especialização de agentes, cujos recursos ficam exclusivamente dedicados a uma tarefa em particular. Determinados agentes ficam adstritos ao planeamento e distribuição de tarefas, enquanto outros ficam dedicados à sua execução. Esta restrição de funções permite que os agentes desenvolvam soluções de maior qualidade, ou com menor exigência de recursos.

A estrutura organizacional possui capacidade de adaptação, ao permitir a entrada e saída dinâmicas de agentes no sistema e uma política de delegação de tarefas que tenha em conta as capacidades que cada agente possui no momento. Estas características permitem que o sistema multi-agente, que opera em ambientes de recursos limitados, fique mais robusto à perda de capacidades individuais, à inoperacionalidade ou à ineficiência geral dos seus agentes.

5.1 Organização Geral da Arquitetura de Agente Proposta

A arquitetura proposta é constituída por cinco subsistemas (fig. 5.1): (i) executivo, (ii) reativo, (iii) adaptativo, (iv) deliberativo e (v) social.

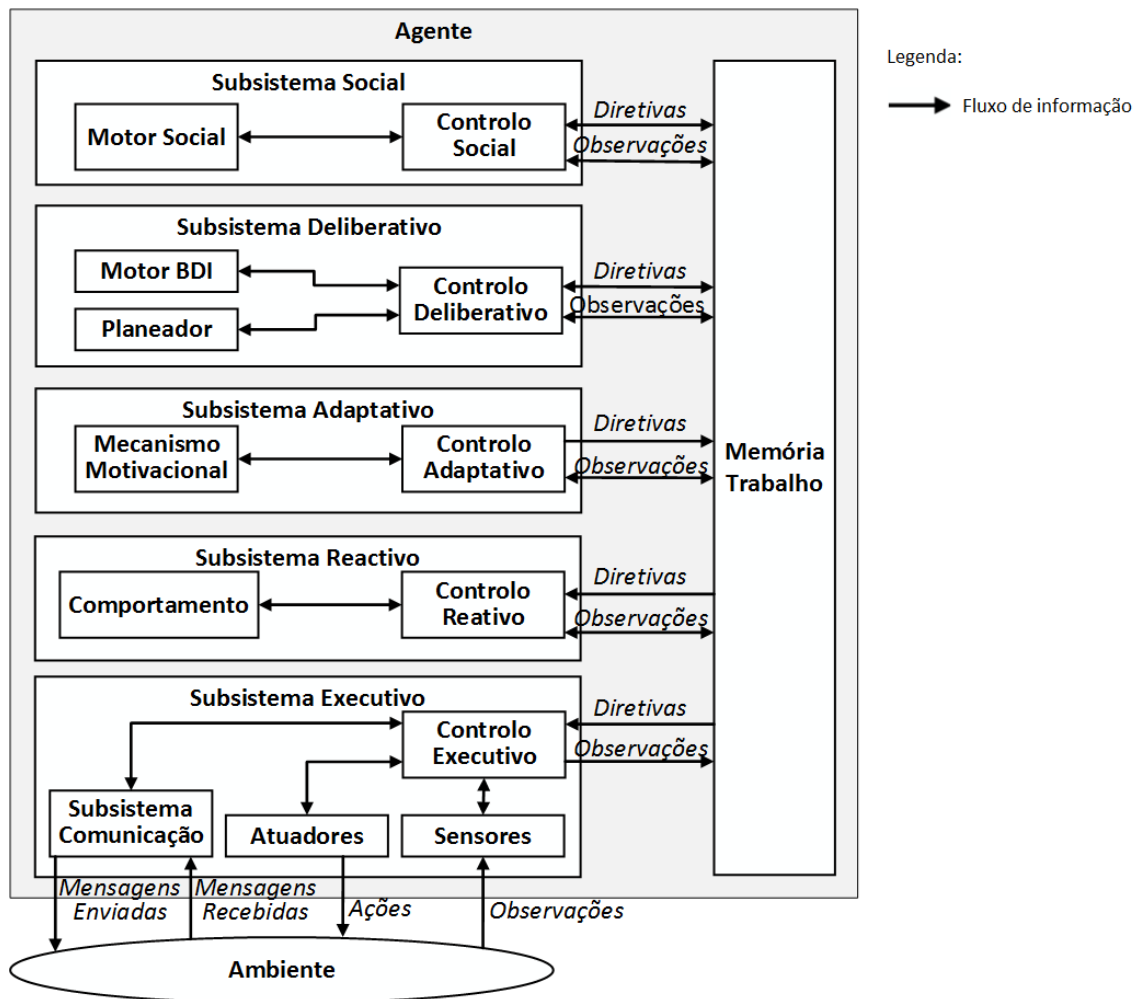


Figura 5.1: Arquitetura de subsistemas da arquitetura de agente proposta.

O subsistema executivo é a camada responsável pela interação com o mundo, ou seja, percebe, executa ações, recebe e envia mensagens.

O subsistema reativo é a camada responsável pela componente reativa do agente, gerando comportamentos do tipo *estímulo-resposta*.

O subsistema adaptativo gera os sinais de base emocional do agente para regulação cognitiva.

O subsistema deliberativo é responsável pela componente deliberativa do agente, ou seja, decide o que fazer e como fazê-lo.

O subsistema social tem como funções a integração do agente na organização, a receção e delegação de objetivos, a receção e informação das disposições emocionais.

Todos os subsistemas trabalham de forma independente e comunicam entre si através de diretivas e observações que são armazenadas numa memória de trabalho que assume a forma de *blackboard*. Esta solução, embora utilize mais memória, permite eliminar redundância na passagem de informação entre camadas, libertando capacidade de processamento para outras tarefas e permitindo que todos os processos trabalhem de forma independente.

Em função dos seus processos gerais, a arquitetura de agente é representada pela figura 5.2.

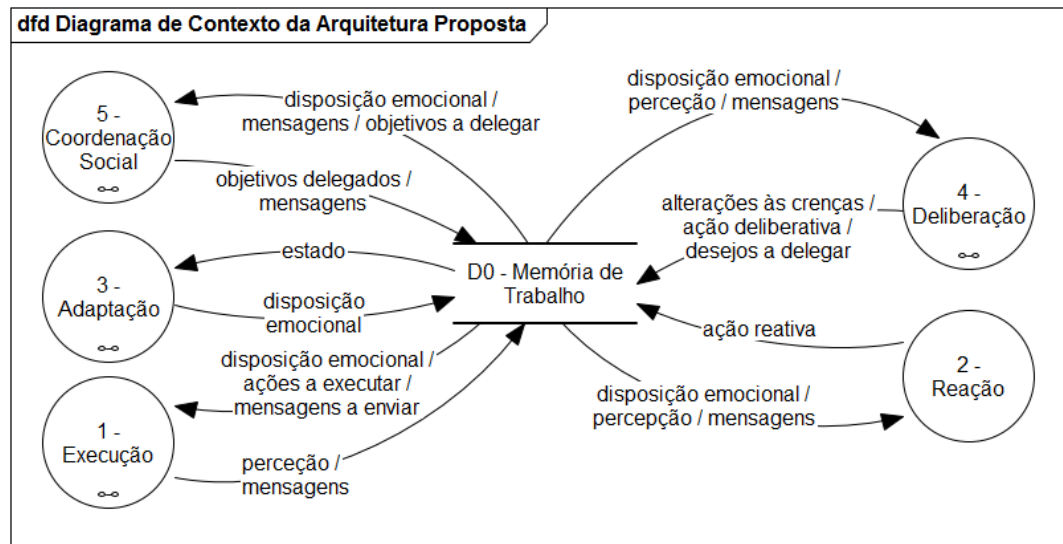


Figura 5.2: Diagrama de contexto da arquitetura proposta.

5.2 Subsistema Executivo

O subsistema executivo é composto pelo processo de execução que é responsável por todo o contacto com o ambiente tal como representado na figura 5.3.

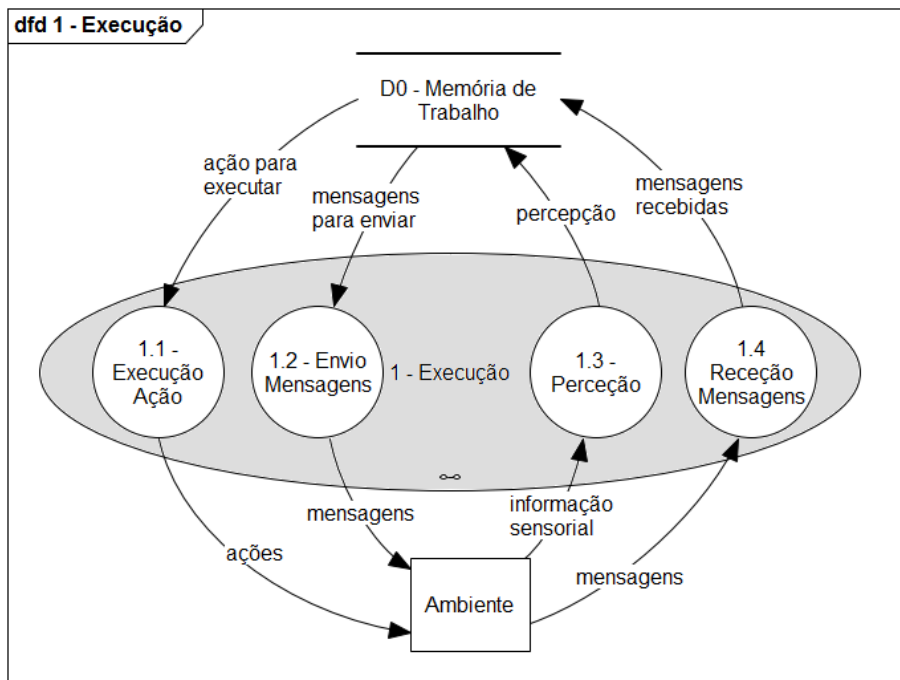


Figura 5.3: Processo Executar.

O subsistema executivo é constituído por quatro processos:

1. **Execução de Ação:** concretização no mundo, através dos atuadores, das ações a realizar. É utilizado um mecanismo de coordenação de ação que seleciona quais as ações a realizar (se as provenientes do subsistema reativo, ou do subsistema deliberativo). As ações provenientes do subsistema deliberativo são acompanhadas de um atributo de prioridade que guiará a seleção por parte do mecanismo de coordenação de ação;
2. **Envio de Mensagens:** envio de mensagens para outros agentes;
3. **Percepção:** recolha da informação sensorial e interpretação, gerando a percepção;
4. **Receção de mensagens:** receção e armazenamento de mensagens.

5.3 Subsistema Reativo

O subsistema reativo é composto por um processo de reação que gera um comportamento de acordo com a percepção. O comportamento é característico do problema, podendo ser definido qualquer tipo de arquitetura reativa.

A figura 5.4 exemplifica um comportamento reativo caracterizado por uma hierarquia de subsunção.

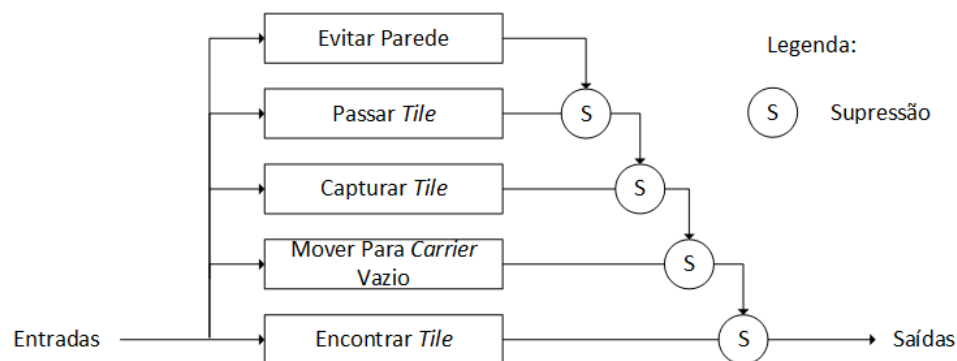


Figura 5.4: Exemplificação de um comportamento reativo.

Neste exemplo, a ativação de um dos comportamentos de nível superior suprime os comportamentos de nível inferior.

5.4 Subsistema Adaptativo

O subsistema adaptativo tem como função a geração dos sinais de base emocional, sendo composto por um único processo que implementa os mecanismos de geração de disposições emocionais e utiliza o estado geral dos restantes processos internos para gerar disposições emocionais.

As disposições emocionais servirão de elemento regulador de todo o processo interno do agente e servirão de base à política de delegação de tarefas do coordenador do agente.

Os mecanismos emocionais do agente terão por base o Modelo Emocional de Fluxo estudado na secção 4.4. A disposição emocional será gerada de

acordo com o definido na secção 6.5.2.

Todos os subsistemas contribuem com informação para a geração da disposição emocional. O subsistema executivo contribui com a indicação do número de mensagens pendentes. O subsistema reativo indica se o seu comportamento se encontra ativo, estando conseqüentemente a serem geradas respostas reativas (de viabilidade). O subsistema deliberativo contribui com o número de desejos ainda não resolvidos e não delegados. O subsistema social informa da disposição emocional dos seus agentes subordinados, refletindo na sua disposição emocional, a disposição emocional dos seus subordinados.

Caberá ao subsistema adaptativo determinar as capacidades de execução máximas relativas aos elementos cognitivos informados pelos restantes subsistemas, que dependerá do problema no qual a arquitetura opera.

5.5 Subsistema Deliberativo

O subsistema deliberativo é composto por um processo de deliberação que compreende a arquitetura BDI do agente, sendo composto pelos processos de atualização do modelo interno, reconsideração, deliberação e planeamento (fig. 5.5).

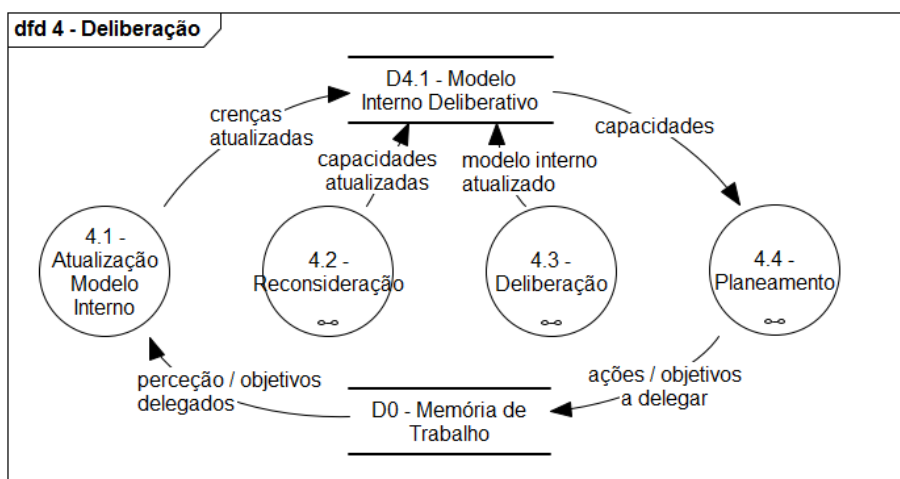


Figura 5.5: Processo de deliberação.

Os três primeiros processos compõem o motor BDI e o seu funcionamento interno é aprofundado na subsecção 5.5.1.

O processo de planeamento (fig. 5.6) visa converter o resultado do processo de deliberação numa ou mais ações, podendo compreender processos de planeamento e/ou seleção de ação, consoante o resultado da deliberação efetuada pelo motor BDI.

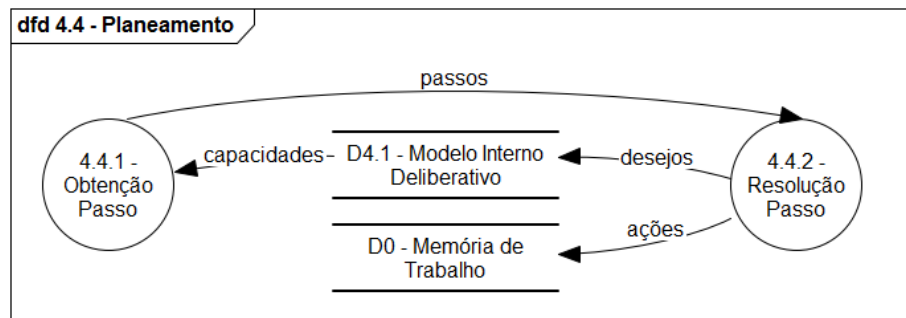


Figura 5.6: Processo de planeamento.

Este processo é composto pelos processos de obtenção de passo e resolução de passo.

Obtenção de passo é o processo que determina quais os passos dos planos a executar. Para cada intenção, verifica qual o passo correspondente ao estado de execução que é depois enviado para o processo de resolução de passo.

Para cada passo enviado, o processo de resolução de passo verifica: (i) se é um desejo, adicionando-o à base de desejos; (ii) se é uma ação, entregando-a à camada executiva para realização; ou (iii) se corresponde a um objetivo passível de planeamento, entregando-o ao planeador para resolver.

5.5.1 O Motor BDI

A componente deliberativa da arquitetura proposta é definida pelo motor BDI. O processamento da componente deliberativa consome muitos recursos computacionais (processamento e memória), pelo que a sua estrutura e dinâmica terão uma grande influência na capacidade de adaptação do agente

à escassez de recursos computacionais e, conseqüente, na capacidade de resposta em tempo real.

Para o motor BDI trabalhar em consonância com as definições do sistema social definidas, necessita de ter em consideração (i) as alterações de capacidades provenientes da interação social, (ii) os objetivos delegados e (iii) a possibilidade de carregamento dinâmico de papéis. As crenças serão geradas através da percepção e das alterações às crenças constantes no *blackboard* da organização. Todo o processo deliberativo terá de ter em consideração a possibilidade dos objetivos poderem ser delegados, sendo fator preferencial para a sua resolução.

5.5.2 Componentes do Motor BDI

Uma **crença** assume a forma de predicado e é armazenada na base de crenças. Todas as crenças são acompanhadas de uma etiqueta que pode ser utilizada para identificar a origem e o tipo de crença.

Um **desejo** é caracterizado por um nome, valor de utilidade, indicador de manutenção e estado. O nome do desejo corresponde ao facto do mundo que o agente pretende ver concretizado. A utilidade é um valor numérico real que determina uma ordem de escolha entre desejos. O indicador de manutenção é um valor booleano o qual identifica os desejos que, após a sua conclusão, são mantidos, ou descartados. O estado indica em que situação se encontra o desejo.

Consoante a representação do modelo interno e o ciclo de execução da arquitetura, um desejo poderá encontrar-se em diferentes estados:

- **Avaliação:** o desejo encontra-se a aguardar o enquadramento com as capacidades e a base de crenças;
- **Pendente:** de acordo com a base de crenças, existe pelo menos um plano que consegue concretizar o desejo;
- **Espera:** de acordo com a base de crenças, não existe nenhum plano que consegue concretizar o desejo;

- **Aguardar Realização Externa:** o agente não possui nenhum plano que concretize o desejo e o desejo foi delegado;
- **Execução:** o agente encontra-se a executar um plano que irá concretizar o desejo.

O modelo da dinâmica de um desejo é representado na figura 5.7.

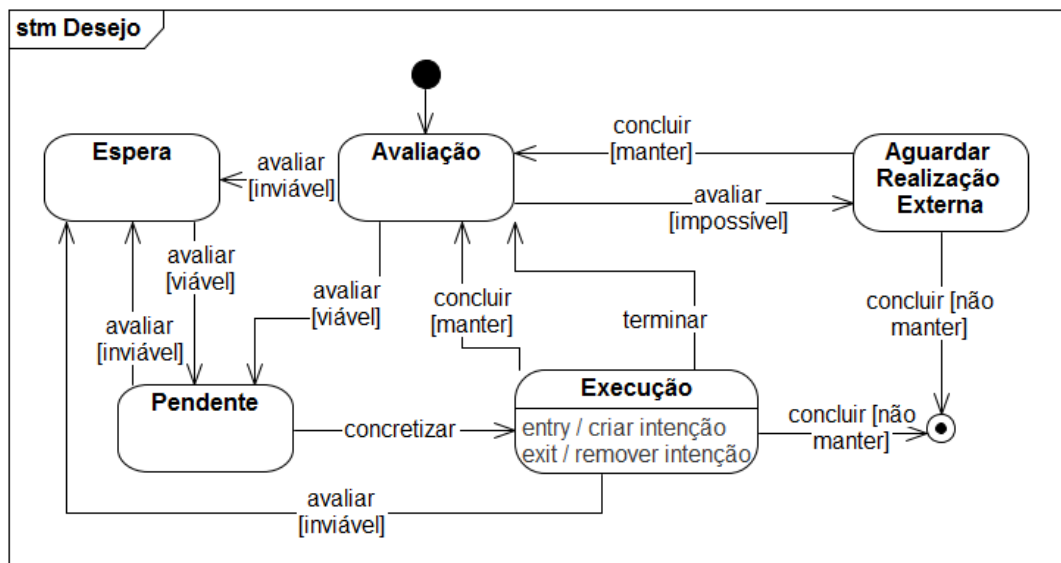


Figura 5.7: Modelo da dinâmica de um desejo.

A existência do estado “Avaliação” justifica-se pela necessidade de distinguir desejos que nunca foram avaliados de acordo com a base de crenças, dos que já o foram. Estes últimos já não precisam de ser avaliados por todas as crenças, mas apenas pelas alterações às crenças.

Designa-se por **capacidade** uma ação ou um plano que o agente é capaz de realizar.

As **ações** são descrições do resultado da ação dos atuadores do agente no mundo e são caracterizadas por um nome, pré-condições, pós-condições e um estado. O nome corresponde a uma descrição da ação (interna ao motor BDI). As pré-condições são o conjunto de crenças que têm de ser verdadeiras para que as ações possam ser realizadas. As pós-condições são as alterações ao estado do mundo que a ação produz. O estado indica o nível de operacionalidade da ação e é definido pela máquina de estados representada

pela figura 5.8:

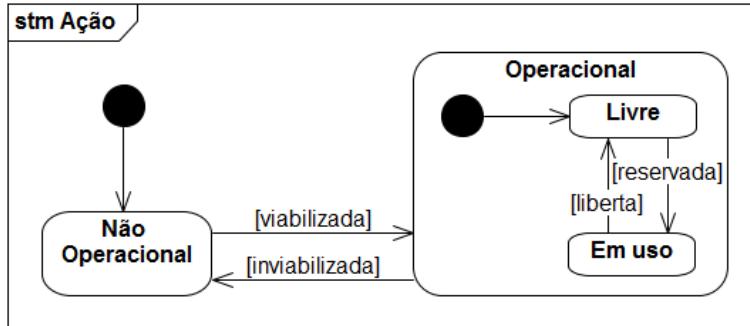


Figura 5.8: Modelo de dinâmica de uma ação.

Uma ação pode assumir um dos seguintes estados:

- **Não Operacional:** não se encontram reunidas as condições, segundo a base de crenças, para a afetação da ação;
- **Livre:** a ação encontra-se disponível para ser afetada a uma intenção;
- **Em Uso:** a ação encontra-se atribuída a uma intenção.

Um **plano** é caracterizado por um objetivo, utilidade, pré-condições, contexto, corpo e estado. O desejo que o plano concretiza é identificado pelo objetivo. A utilidade determina uma ordem de escolha entre planos que concretizam o mesmo desejo. Ao grupo de factos sobre o mundo, que têm de ser verdadeiros para que o plano seja exequível, dá-se o nome pré-condições. O contexto representa o conjunto de factos sobre o mundo que têm de ser verdadeiros para que o plano se possa manter válido durante a sua execução. O corpo do plano é constituído pelas ações e/ou objetivos a serem executados/concretizados. O corpo do plano permite a definição de execução/concretização de múltiplas ações/desejos em simultâneo com a seguinte sintaxe:

- *<nome da ação>*: ação a ser realizada;
- *achieve(<nome do desejo>, <valor de utilidade>)*: desejo a ser concretizado e respetivo valor de utilidade;
- *forall(<crença>)*: colocado no início do corpo do plano. O plano é instanciado tantas vezes quantas as unificações possíveis de *<crença>*;

- *forany*(<crença>): colocado no início do corpo do plano. O plano é instanciado com a primeira unificação de <crença>;
- [*<ação1/desejo1>*, ..., *<açãoN/desejoN>*]: realização em simultâneo das ações/desejos.

Um plano é caracterizado por três dinâmicas que correspondem à dinâmica das pré-condições, contexto e ações. Cada uma das dinâmicas é definida pelo diagrama de transição de estados representado na figura 5.9.

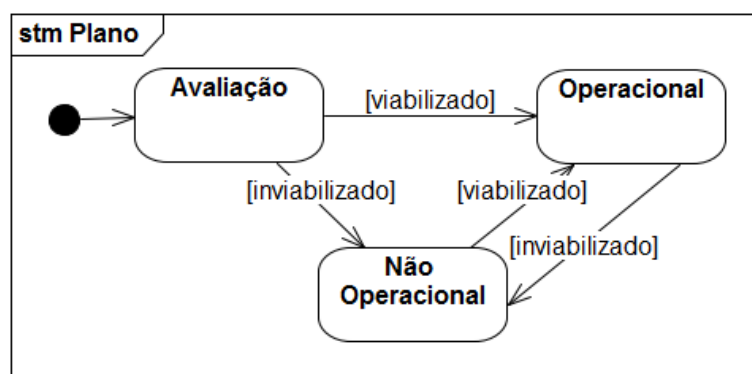


Figura 5.9: Modelo de dinâmica de um plano.

A caracterização destas três dinâmicas permite identificar qual o estado do plano relativamente a cada um dos seus componentes (pré-condições, contexto e ações). Por exemplo, um plano que se encontre no estado (*Operacional*, *Não Operacional*, *Operacional*) tem todas as suas pré-condições verificadas como verdadeiras na base de crenças e as ações do seu corpo encontram-se todas disponíveis para serem utilizadas.

A persistência de estado dos desejos, ações e planos permite a definição de processos internos mais modulares e coesos e também possibilita a validação dos componentes do motor BDI através das alterações às crenças.

Uma **intenção** é caracterizada por um objetivo e um ou mais planos com o respetivo estado de execução. O objetivo identifica o estado do mundo que a intenção se propõe concretizar. Cada plano identifica o conjunto de tarefas que têm de ser executadas para o objetivo ser concretizado.

5.5.3 Dinâmica do Motor BDI

Em problemas caracterizados por um grande volume de crenças, a avaliação constante de todas as crenças poderá colocar em risco a capacidade de resposta em tempo real do agente. Esta avaliação permanente pode não ser necessária, por motivos vários como a grande abrangência de informação proveniente de uma percepção insignificante para o problema, ou o baixo dinamismo dos elementos que compõem o problema. Este problema agrava-se pela utilização de regras de inferência no processo deliberativo, onde cada inferência tem por base outras inferências e/ou crenças sobre o mundo. Se durante o processo deliberativo uma inferência tiver de ser avaliada mais do que uma vez, são desperdiçados recursos computacionais caso se tenha de avaliar sempre todas as condições dessa regra.

A mesma linha de raciocínio se aplica aos restantes componentes do modelo BDI, por exemplo: um plano para ser considerado operacional necessita que as ações do seu corpo estejam todas operacionais. Uma ação é considerada operacional se as pré-condições para a utilização dessa ação forem verdadeiras e a ação já não se encontrar em uso. Numa perspectiva de economia de recursos computacionais, se mais do que um plano utilizar a mesma ação, não faz sentido estar constantemente a avaliar essa ação e respetivas pré-condições.

A solução deste problema parte da caracterização da dinâmica dos componentes BDI (pré-condição, ação, plano, desejo e intenção) e da avaliação das dinâmicas de transição de estado através de alterações, e.g. uma ação a_1 para ser considerada operacional, tem de ver verificadas determinadas pré-condições como verdadeiras e, suponha-se que em determinado instante, apenas parte destas é que o são, pelo que a ação se encontra no estado “Não Operacional”. Na iteração seguinte, as pré-condições que restavam como falsas passaram a ser verdadeiras e o motor BDI, no momento de avaliação da ação a_1 irá apenas verificar que as pré-condições que eram falsas passaram a verdadeiras, transitando a ação a_1 para o estado “Livre”.

Apenas os componentes BDI que nunca foram alvo de avaliação é que terão de ser avaliados pelo conteúdo completo da base de informação.

A movimentação do agente pelos diversos grupos da organização podendo, inclusive, assumir qualquer papel, exige por parte do motor BDI a capacidade de dinamicamente alterar os seus planos e ações. A capacidade do motor BDI receber e delegar objetivos também tem de ser considerada, bem como a receção de crenças e capacidades provenientes da estrutura organizacional.

Assim sendo, foi definida a dinâmica do motor BDI que se encontra resumida na figura 5.10.

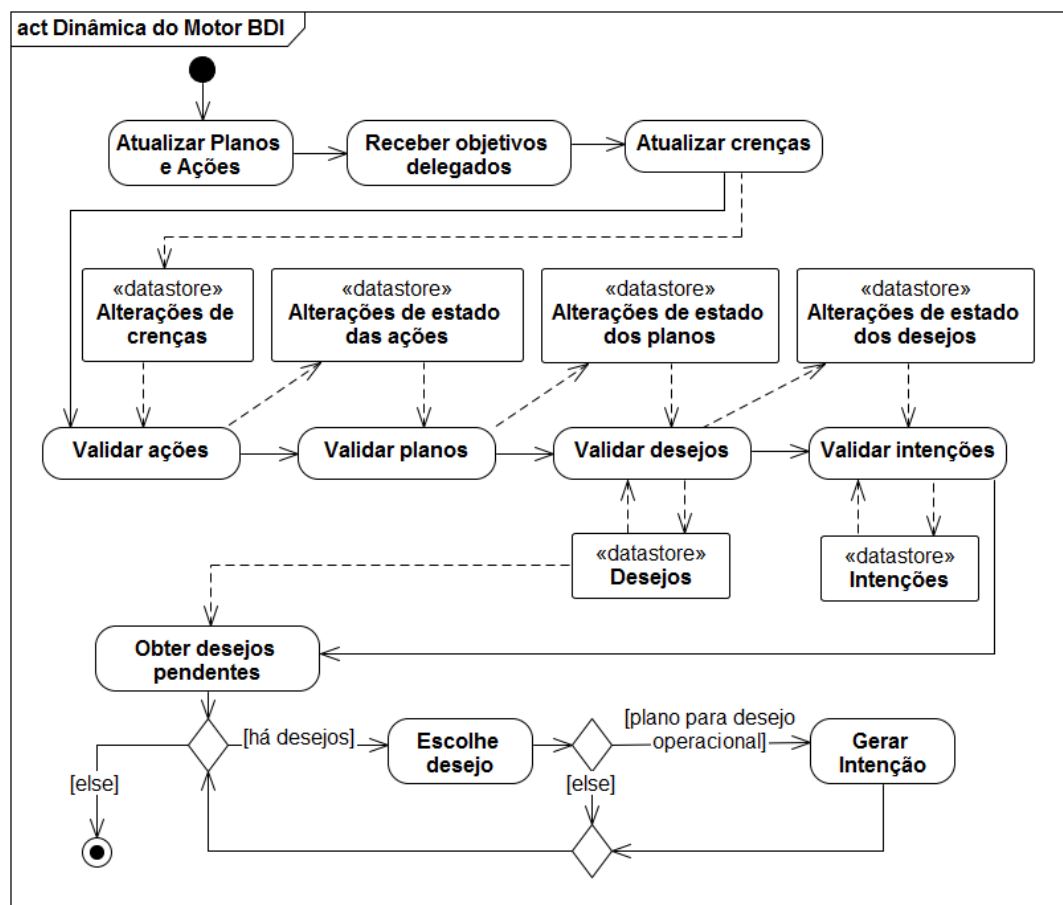


Figura 5.10: Resumo da dinâmica do motor BDI.

O processo de atualização do modelo interno inicia-se pela atualização de planos e ações (caso haja alteração do papel na organização), segue pela receção das alterações de capacidades dos seus subordinados, pela receção dos objetivos delegados e termina com a atualização das crenças e respetiva verificação de quais é que foram alteradas.

O processo de reconsideração compreende a validação dos componentes do modelo BDI: ações, planos, desejos e intenções.

A validação de ações é feita pela verificação da verdade das suas pré-condições.

A validação de planos é feita pela verificação da verdade das suas pré-condições e contexto, e da operacionalidade das ações do corpo pela avaliação do estado das suas ações.

A validação de desejos é efetuada, numa primeira fase, pela verificação da existência de planos que os concretizem. Na eventualidade de existirem planos que concretizem um desejo, a sua validação é efetuada pela verificação dos estados dos planos.

No final da validação de desejos, se existir pelo menos um desejo no estado “Pendente” é necessário reconsiderar.

A validação das intenções compreende as seguintes verificações:

- o desejo para o qual a intenção foi gerada se encontra concretizado;
- as ações atuais dos planos foram concluídas, avançando-os para o passo seguinte;
- os planos se encontram concluídos;
- a validade dos planos que compõem a intenção, verificando se o estado do plano para o contexto ainda se mantém operacional;
- a existência de desejos com maior utilidade que os desejos das intenções atuais, mas para os quais existam planos que se encontrem operacionais relativamente às suas pré-condições, mas inoperacionais relativamente às suas ações, devido a estas se encontrarem reservadas ou em uso por estas intenções.

Os planos que se encontram concluídos são removidos da intenção e esta termina quando já não contém nenhum plano.

Quando o estado de um plano de uma intenção para o seu contexto é “Não Operacional”, a intenção é terminada automaticamente.

Na eventualidade de existirem desejos com maior utilidade, cujos planos não possam ser utilizados devido à utilização de ações por parte de outras intenções, estas últimas são terminadas, as suas ações libertadas e o respetivo desejo transita para o estado “Espera”, tal como apresentado na figura 5.7. Os desejos com maior utilidade transitam para o estado “Pendente”.

Em consonância com o modelo BDI, o processo de deliberação é composto pelos processos de geração de opções e seleção de opções. O processo de geração de opções identifica todos os desejos possíveis de realizar, ou seja, seleciona todos os desejos no estado “Pendente”, entregando-os ao processo de seleção de opções. O processo de seleção de opções seleciona desejos por ordem decrescente de utilidade e gera, para cada um deles, uma intenção. Na eventualidade de dois ou mais desejos necessitarem da mesma ação para serem concretizados, apenas será gerada uma intenção para o desejo de maior utilidade, permitindo a concretização do máximo de desejos possíveis em simultâneo sem que estes colidam nas necessidades de ação.

No final do processo de deliberação pode: (i) não existir nenhuma intenção; ou (ii) existir uma ou mais intenções compostas, cada uma, por um ou mais planos.

5.6 Subsistema Social

A coordenação de ação do sistema multi-agente é assegurada por uma estrutura organizacional do tipo hierárquico. Esta escolha enquadra-se nos modelos de estruturas organizacionais apresentados na secção 3.5.

A grande carga computacional exigida pelo raciocínio social no estabelecimento das redes de dependência e a grande carga de comunicação inerente aos processos de negociação dos modelos de licitação limitam a escolha destes modelos de coordenação em sistemas multi-agente inseridos em ambientes de recursos limitados.

A implementação de uma estrutura organizacional fundamenta-se pelo aproveitamento do melhor dos dois modelos anteriores. O estabelecimento das relações de dependência e poder fica claro na definição dos grupos e papéis

da organização. As relações de dependência, determinadas pela estrutura da organização e não pela dependência de objetivos, planos e capacidades, evitam uma grande carga de processamento no estabelecimento e manutenção destas relações. Através da criação de protocolos de propagação de objetivos de uma forma hierárquica e da obrigatoriedade em atendê-los, evita-se o excedente de comunicação existente no processo de licitação.

É proposta uma arquitetura social em pirâmide com múltiplos níveis de coordenação e um agente coordenador de todo o sistema. De forma a garantir a robustez do sistema, o coordenador pode ser substituído a qualquer momento (subsecção 5.6.2). O coordenador é responsável por decompor os desejos de alto nível em subdesejos, que são transmitidos ao seu nível inferior, que decompõem o subdesejo em outros subdesejos, até subdesejos diretamente decompostos em ações que são executadas. A delegação de desejos é efetuada com base na disposição emocional dos agentes subordinados.

Para aumentar a capacidade de adaptação do sistema multi-agente à operação com recursos limitados e a sua robustez, a arquitetura social é dotada de protocolos que permitem a entrada e saída dinâmica de agentes no sistema e uma política de delegação de tarefas que tem em conta as capacidades que o agente possui naquele momento.

5.6.1 Definições da Arquitetura Social

A caracterização da estrutura social é composta pelas definições dos seguintes elementos:

- **grupo:** conjunto de elementos devidamente enquadrados e estruturados;
- **coordenador geral:** agente que se encontra no topo da hierarquia;
- **subordinado:** agente ou grupo, que se encontra definido na estrutura de um grupo. Cada subordinado assume um papel dentro do grupo e é responsável por executar as determinações do seu coordenador;
- **coordenador:** elemento responsável por:
 - implementar a estrutura do grupo que representa ao atribuir papéis aos agentes que pretendem fazer parte do grupo;

- informar o seu coordenador das capacidades gerais do grupo;
 - receber e garantir que se concretizam os objetivos atribuídos ao grupo;
 - concretizar os desejos do grupo.
- **papel:** função dentro de um grupo. Os papéis são característicos de cada grupo e são constituídos pela definição de desejos, capacidades necessárias para assumir o papel e planos que utiliza.

5.6.2 Arquitetura do Sistema Social

A arquitetura proposta implementa uma estrutura organizacional definida de forma hierárquica piramidal.

A caracterização de um grupo é especificada por: (i) designação do grupo; (ii) agentes/subgrupos que o compõem; (iii) desejos; (iv) capacidades; (v) capacidades necessárias para ser representante do grupo.

Um papel é caracterizado por: (i) designação do grupo a que pertence o papel; (ii) designação do papel; (iii) capacidades; (iv) capacidades necessárias para assumir o papel.

Iniciação da Estrutura Organizacional

O coordenador geral é determinado pelo programador. A estrutura organizacional só começará a formar-se quando o coordenador geral se encontrar ativo. No momento em que é iniciado, o coordenador geral cria um *blackboard* e carrega-o com as definições de todos os grupos. De seguida, carrega para o seu modelo interno social as características do grupo coordenador e termina com o registo no *blackboard* da sua informação como coordenador geral.

Integração de um Novo Agente

Todos os agentes são dotados do contacto do coordenador geral e quando iniciados apresentam-se a este indicando as suas capacidades. O coordenador

geral enquadra as capacidades apresentadas com as necessidades do grupo e subgrupos e decide se o agente fica seu subordinado, ou se será integrado num subgrupo.

No momento em que é informado da decisão do coordenador geral sobre a sua integração, o agente recebe um número (identificador do agente) e a informação de acesso ao *blackboard*. O coordenador geral pode decidir que o agente: (i) integra o seu grupo, tornando-se seu subordinado; ou (ii) integra um subgrupo coordenado por si.

Ao tornar-se um elemento do grupo, o agente assume o papel que o coordenador lhe atribui, consulta o *blackboard* carregando para o seu modelo interno os planos e desejos da função que irá ocupar e fica a aguardar a atribuição de objetivos. Em ato contínuo, o coordenador analisa as capacidades do agente, contrapõe com as capacidades do grupo e verifica se, com a adição das capacidades do novo agente, são geradas novas capacidades. As capacidades geradas ficam disponíveis para deliberação.

Na eventualidade de o agente ser destinado a um subgrupo, é-lhe indicada a sua designação. O agente consulta o *blackboard* e verifica se já existe um coordenador. Na eventualidade de já existir um coordenador, o agente apresenta-se indicando as suas capacidades. O coordenador deliberará sobre se lhe atribui um papel, ou se o integra num subgrupo, repetindo-se o ciclo até o agente se tornar elemento de um grupo.

Se nenhum agente assumiu as funções de coordenação, o agente verifica se pode assumir esse papel. Ao ter as capacidades necessárias para coordenar o grupo, o agente assume as funções de coordenação e: (i) carrega os planos e desejos do grupo; (ii) informa o seu coordenador das funções que se encontra a desempenhar; (iii) informa os agentes já registados no grupo do seu papel e atribui-lhes um. Ao verificar que não tem as capacidades de coordenação, o agente regista-se no *blackboard* como elemento do grupo e aguarda o aparecimento de um coordenador.

O coordenador é responsável por registar e manter atualizada a informação no *blackboard* referente aos elementos do seu grupo, quer seja o registo dos agentes que o incorporam, quer sejam as capacidades do grupo.

Abandono de um Agente

Se durante um intervalo de tempo pré-definido um elemento não enviar nenhuma comunicação ao seu coordenador, é automaticamente considerado inativo por este.

Na eventualidade de o agente não necessitar de comunicar com o representante durante o intervalo de tempo pré-definido, envia uma mensagem de apresentação. Esta mensagem tem como resposta uma outra mensagem de apresentação.

Quando um agente é considerado inativo pelo seu coordenador é descartado do modelo interno deste último e do *blackboard*. No caso de o agente inativo ser coordenador de um subgrupo, os agentes subordinados, ao não receberem a mensagem de apresentação do seu coordenador, irão considerá-lo inativo e tentar assumir a coordenação do grupo. Na eventualidade de mais do que um agente pretender assumir a coordenação do grupo o desempate dá-se pela antiguidade. Se nenhum dos agentes tiver capacidades para coordenar o grupo irão apresentar-se ao coordenador do seu ex-coordenador.

Informação de Capacidades

Os agentes informam o seu coordenador das alterações às suas capacidades e este mantém atualizado o seu modelo interno e o *blackboard* com essa informação.

Esta informação de capacidades é elevada até ao topo da hierarquia, permitindo que os níveis superiores consigam, logo à partida, saber se são capazes de concretizar determinado desejo/objetivo. Sem esta informação, os coordenadores teriam de determinar os objetivos/ações aos seus subordinados sabendo apenas *a posteriori* se podem, ou não, ser concretizados depois de informados pelos subordinados da possibilidade de concretização.

A persistência das capacidades ativas de um agente no *blackboard* permite um processo de substituição de um coordenador mais simples. Um agente ao assumir a coordenação de um grupo já em funcionamento, não necessita

de questionar os seus novos subordinados sobre as suas capacidades, nem de informar o seu coordenador de todas as suas capacidades novamente. Após os devidos ajustes provenientes da sua entrada no grupo, o coordenador apenas informa as alterações às capacidades previamente definidas.

Atribuição de Objetivos

Esta arquitetura dá primazia à resolução de objetivos pela via social, ou seja, se um coordenador conseguir delegar um objetivo/ação, fá-lo-á. Desta forma, é garantida a máxima disponibilidade para a concretização de outros objetivos que daí poderão advir.

No momento da determinação de um objetivo, o subordinado acusa a sua receção informando se é possível, ou não, concretizá-lo. Embora o coordenador tenha no seu modelo interno as capacidades dos seus subordinados, durante o intervalo de tempo entre o envio/receção da mensagem e deliberação sobre a resolução do objetivo, as capacidades dos subordinados podem ser alteradas e inviabilizar a concretização do objetivo, pelo que, a confirmação sobre a possibilidade de resolução é necessária.

Quando um coordenador é informado da impossibilidade de concretização de um objetivo, verifica se outro subordinado pode concretizá-lo, sendo-lhe delegado em caso afirmativo. Na eventualidade de não ser possível delegar o objetivo e ser um subobjetivo de um objetivo delegado, são cancelados os restantes subobjetivos gerados pelo objetivo e é informado o seu coordenador da não possibilidade de concretização.

Cancelamento de Objetivos

O cancelamento do objetivo é determinado pelo coordenador que poderá tomar esta decisão por: (i) determinação superior; (ii) verificar depois de já ter atribuído objetivos que será impossível o objetivo ser concretizado; (iii) sobrecarga do subordinado, indicada pela disposição emocional.

À determinação do cancelamento de objetivos, é sempre retornada uma resposta afirmativa.

Transmissão de Crenças

As crenças de todos os elementos são armazenadas no *blackboard*. Todos os agentes adicionam no *blackboard* as alterações às suas crenças. Assim, qualquer agente poderá gerar as suas crenças com base nas crenças de terceiros, sem as questionar.

O processo de informação de crenças permite um compromisso constante dos recursos despendidos por cada agente, sendo assim mais fáceis de controlar. Ainda que este protocolo exija aos agentes a utilização dos seus recursos na manutenção das suas crenças no *blackboard* - as quais poderão nem ser usadas -, a outra opção, questionar a cadeia de comando, poderia exigir ao sistema a utilização de recursos incontroláveis.

Por exemplo, um coordenador de um grupo pretende saber se determinada crença é verdadeira. Como não tem os recursos para o saber pela sua perceção, questiona os seus subordinados. Nenhum dos subordinados responde afirmativamente à existência da referida crença, o que não significa que não seja verdadeira. O coordenador vê-se obrigado a questionar o seu coordenador que também não consegue obter a resposta na sua base de crenças e ramifica o pedido para os elementos do seu grupo. Este ciclo pode percorrer toda a estrutura e, para além da latência do tempo de resposta, prejudica os elementos da estrutura ao utilizarem a sua capacidade de processamento para responder a uma questão.

Cada crença armazenada no *blackboard* contém a identificação do agente que permitirá, a quem consultar o *blackboard*, filtrar as crenças da forma que melhor lhe convier (por agente(s) e/ou grupo(s)).

Concretização de Desejos e Objetivos

A concretização de desejos e objetivos pode ser efetuada de quatro formas:

1. resolução a título individual das ações determinadas pelo plano da intenção;
2. distribuição das ações e/ou objetivos do plano da intenção aos seus

- inferiores hierárquicos;
- 3. subdivisão do objetivo em subobjetivos e respetiva distribuição pelos inferiores hierárquicos;
- 4. combinação das hipóteses anteriores.

Integração da Arquitetura Social na Arquitetura de Agente

Ao nível do agente, a arquitetura social é implementada pelo subsistema social que é composto por um processo de coordenação social, o qual, por sua vez, é composto pelos processos de receção de disposições emocionais, envio da disposição emocional, delegação, receção de delegações, enquadramento hierárquico e atualização de crenças (fig. 5.11).

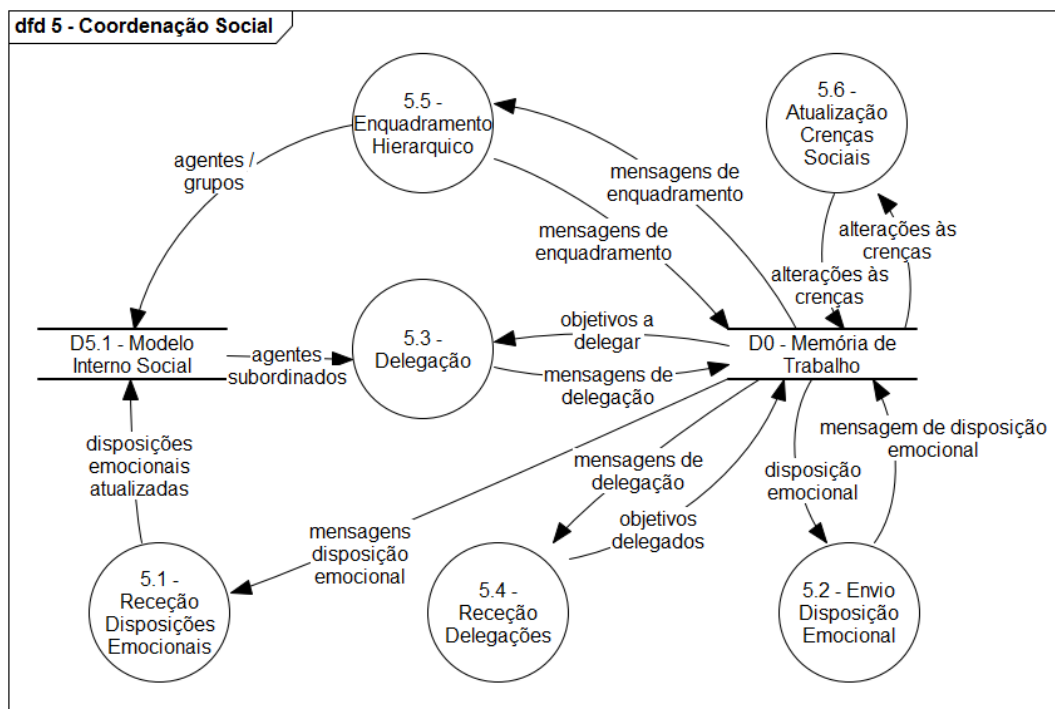


Figura 5.11: Processo Coordenação Social.

Os processos de receção e envio de disposições emocionais processam a informação relativa às disposições emocionais.

O processo de delegação delega os objetivos determinados pelo subsistema deliberativo de acordo com a política de delegação de tarefas definida na subsecção 6.5.1.

O processo de receção de delegações processa as mensagens que delegam objetivos ao agente encaminhando-os para o subsistema deliberativo.

O processo de enquadramento hierárquico implementa os protocolos de integração na estrutura hierárquica, fazendo: (i) a integração e manutenção do agente no grupo; (ii) a integração e manutenção de agentes como seus subordinados; (iii) a informação do coordenador das suas capacidades; (iv) a receção das capacidades dos seus subordinados e avaliação das capacidades coletivas.

O processo de atualização de crenças sociais tem como finalidades: (i) a receção, filtro e adição à memória de trabalho das alterações às crenças inseridas no *blackboard* e (ii) o envio para o *blackboard* das alterações às crenças geradas pelo subsistema deliberativo.

5.7 Conclusão

Neste capítulo foi apresentada uma arquitetura de agente híbrida que permite interagir socialmente pelo enquadramento numa estrutura organizacional hierárquica de planeamento centralizado.

A arquitetura tem como objetivo uma interação social simples, através da utilização do menor número possível de recursos computacionais. Da proposta resulta um sistema social hierárquico onde os desejos são decompostos sucessivamente, desde o topo até ao fundo da hierarquia, permitindo um foco dos recursos computacionais utilizados em níveis de abstração que otimizam a qualidade da solução.

A arquitetura social é projetada de forma a permitir uma maior adaptação

ao meio envolvente (pela entrada e saída de agentes no sistema) e poupança de recursos individuais de cada agente ao estabelecer dependências fixas e bem determinadas, como também a existência de níveis de abstração de capacidades, entre outros.

Para fazer face à limitação de recursos computacionais, a arquitetura de agente é dotada de diversos níveis de decisão regulados por uma componente cognitiva de base emocional.

O próximo capítulo serve de base ao fundamento da arquitetura proposta neste trabalho ao apresentar a concretização de sistemas multi-agente, suportados por diversas arquiteturas de agente e uma política de delegação de tarefas de base emocional.

CAPÍTULO 6

Concretização e Resultados Experimentais

A arquitetura proposta no capítulo anterior fundamenta-se pela concretização experimental apresentada neste capítulo. O primeiro caso prático concretiza e testa o potencial de interação social de uma arquitetura de agente reativo. O segundo, estuda a potencialidade de interação social com recurso ao planeamento descentralizado, suportado por uma arquitetura de agente deliberativa, composta por um motor BDI e cujo planeamento é feito de uma forma descentralizada. O terceiro, concretiza uma arquitetura de agente híbrida, que se integra numa estrutura organizacional piramidal e interage dentro desta através de planeamento centralizado e distribuição de tarefas. As arquiteturas de agente concretizadas e testadas resultam da instanciação a vários níveis da arquitetura proposta.

Para além do estudo das potencialidades de interação social das arquiteturas existentes, é demonstrada a potencialidade da utilização de políticas de delegação de tarefas baseadas em mecanismos emocionais como alternativa ao problema da coordenação de tarefas em ambientes de recursos limitados, através da concretização de uma política de delegação de tarefas de base emocional enquadrada num problema de coordenação específico.

O estudo das potencialidades de cooperação das arquiteturas de agente é efetuado em igual contexto experimental, pelo que este capítulo se inicia pela sua descrição.

6.1 Contexto Experimental

As potencialidades de cooperação das arquiteturas de agente concretizadas são testadas num problema do tipo *Tile World*, adaptado para exigir cooperação entre agentes. Existem dois tipos de agente: *captor* e *carrier*. Os agentes do tipo *captor* são responsáveis por obter e entregar blocos (*tiles*) aos agentes do tipo *carrier* que, por sua vez, têm como missão transportar os blocos para as bases. Nenhum agente poderá carregar mais do que um bloco em simultâneo. Detalhes sobre a plataforma de simulação poderão ser encontrados no anexo B.

A simulação será composta por dois agentes do tipo *carrier* e dois agentes do tipo *captor*, um ambiente sem obstáculos, com apenas uma base e cinco blocos com posição aleatória.

O objetivo do sistema multi-agente será conseguir obter e entregar na base todos os blocos, demonstrando assim a capacidade de coordenação de cada uma das arquiteturas.

6.2 Interação Social com Base em Arquiteturas Reativas

As arquiteturas reativas são excelentes arquiteturas de agente para operar em ambientes de recursos limitados, pois ao implementarem comportamentos do tipo estímulo-resposta, os processos que compõem estas arquiteturas não exigem um grande esforço computacional.

Na secção 3.2 estudou-se que os sistemas multi-agente compostos por agentes reativos poderão coordenar-se através da estruturação dos seus comportamentos, ou seja, a interação social de um sistema multi-agente composto

por sistemas reativos é definida pelos comportamentos dos respectivos agentes. Caberá à definição dos comportamentos estabelecer a correta relação entre a percepção e a ação, fazendo com que todo o sistema opere de forma consistente na concretização dos seus objetivos.

Esta arquitetura tem por objetivo aproveitar a potencialidade de cooperação das arquiteturas reativas as quais, devido ao baixo consumo de recursos, se tornam ideais para ambientes de recursos limitados. Da arquitetura de agente proposta é instanciado o subsistema reativo que é responsável por interagir com o ambiente através da percepção e execução de ações e determinar, de acordo com a percepção, qual o comportamento adequado.

Ao nível da sua arquitetura, o agente é composto por um controlo reativo que, por sua vez, é composto por um comportamento. O comportamento poderá ser definido como uma reação ou uma reação composta, ou seja, um conjunto de reações (fig 6.1).

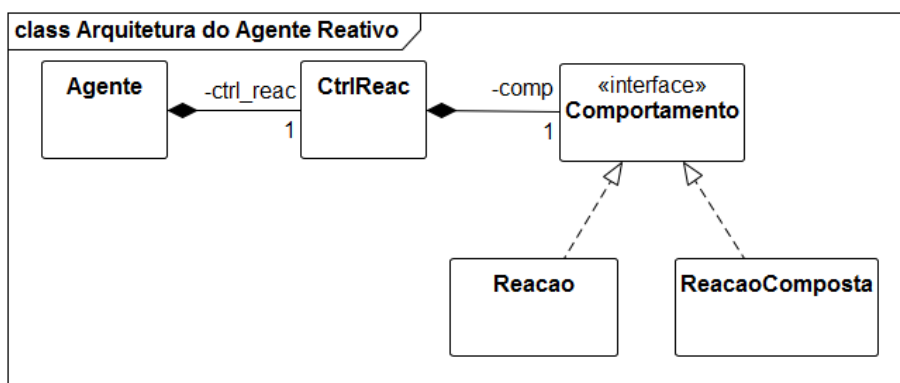


Figura 6.1: Arquitetura do Agente Reativo.

6.2.1 Comportamentos Reativos

O comportamento de um agente do tipo *captor* é definido numa arquitetura reativa de subsunção e encontra-se representado na figura 6.2.

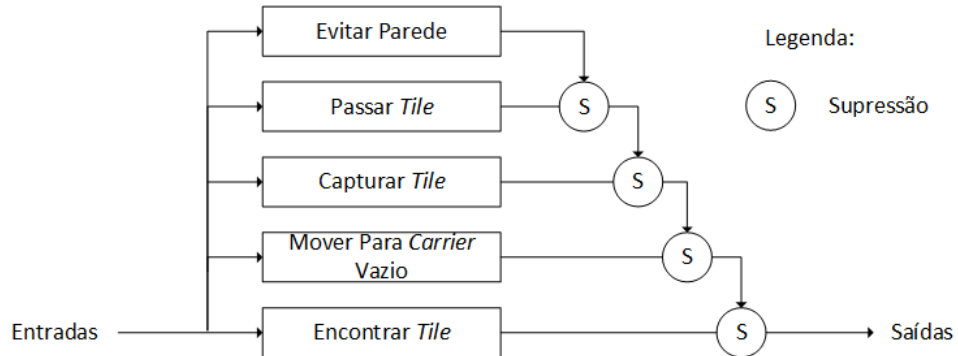


Figura 6.2: Comportamento de um agente tipo *captor*.

O agente: (i) procura blocos quando não percebe nenhum, navegando aleatoriamente pelo mapa; (ii) quando percebe pelo menos um bloco e se encontra no estado vazio (sem bloco) obtém o bloco mais próximo; (iii) no momento em que se encontra no estado cheio (com bloco) desloca-se para um agente do tipo *carrier* no estado vazio para lhe entregar o bloco; (iv) em qualquer momento, se o agente se depara com uma parede desvia-se.

O comportamento de um agente do tipo *carrier* é definido numa arquitetura reativa de subsunção e encontra-se representado na figura 6.3.

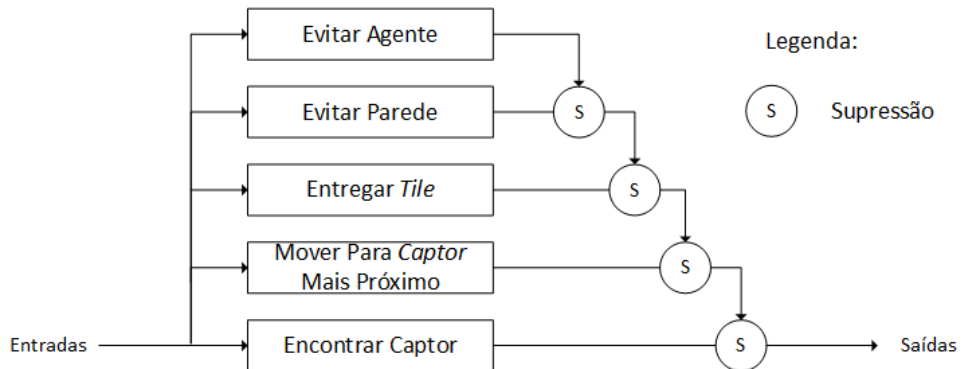


Figura 6.3: Comportamento de um agente tipo *carrier*.

O agente: (i) procura agentes do tipo *captor* quando não é percebido nenhum, navegando aleatoriamente pelo mapa; (ii) move-se para o *captor* mais próximo quando o próprio se encontra no estado vazio; (iii) entrega o bloco, movendo-se para a base e largando-o quando o próprio agente se encontra no estado cheio; (iv) em qualquer momento, se o agente se depara

com um obstáculo (agente e parede) desvia-se.

6.2.2 Resultados Experimentais e Análise

Executando os agentes na plataforma experimental, verifica-se que são capazes de entregar todos os blocos (fig. 6.4).

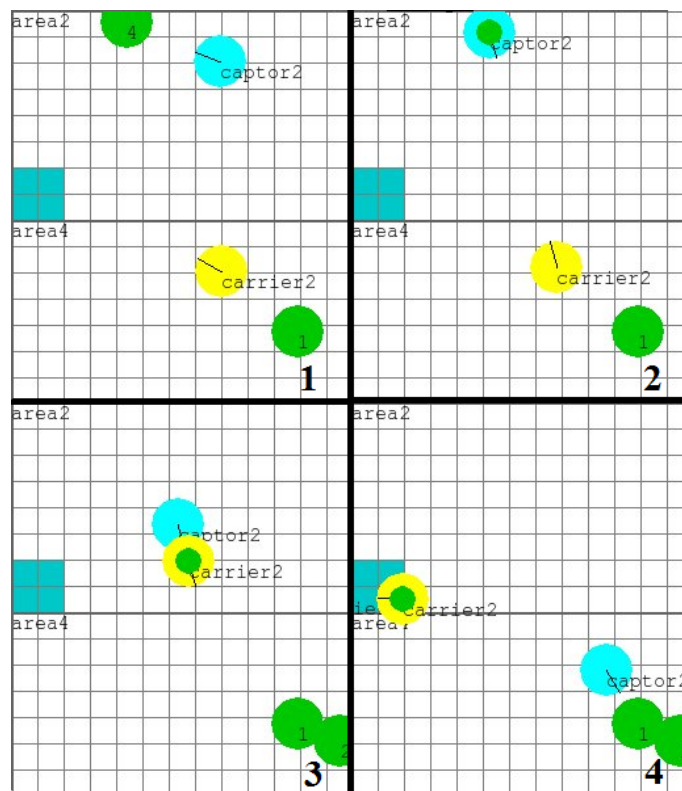


Figura 6.4: Exemplo das trajetórias dos agentes reativos.

Na figura 6.4 identificam-se os seguintes momentos: **1-** O agente *captor2* dirige-se para o bloco 4; **2-** O agente *captor2* obtém o bloco 4 e dirige-se para o agente *carrier2*. O agente *carrier2* vai ao encontro do agente *captor2*. **3-** Após aproximação, o agente *captor2* transfere o bloco para o agente *carrier2*; **4-** O agente *carrier2* entrega o bloco na base.

Porém, verifica-se a inexistência de coordenação entre agentes do mesmo tipo, e.g. existem situações em que os dois agentes *captor* se dirigem para

o mesmo bloco por ser o mais próximo, culminando esse comportamento apenas quando um dos agentes o obtém, tal como representado na figura 6.5.

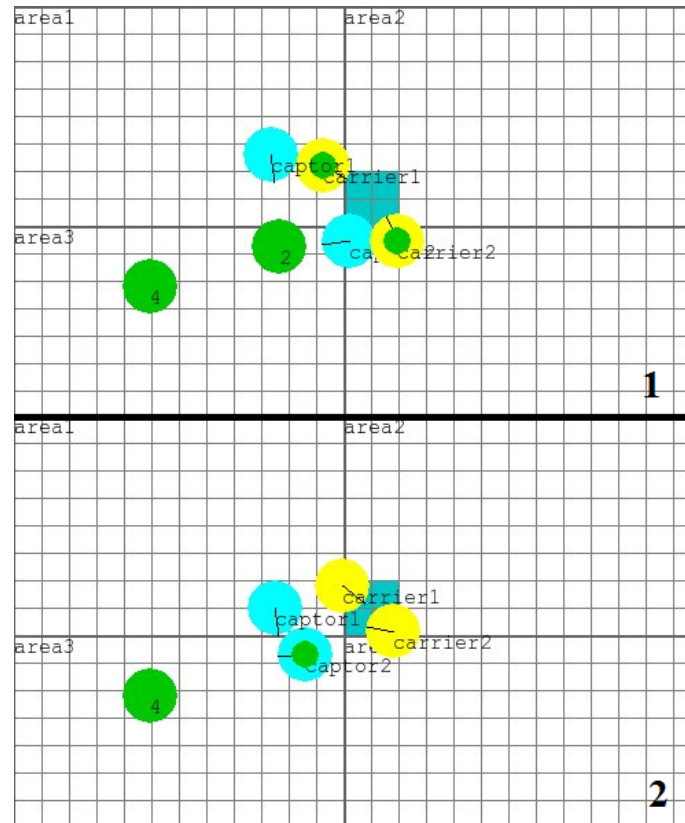


Figura 6.5: Exemplo do problema da falta de coordenação dos agentes reativos.

Na figura 6.5 verificam-se os seguintes momentos: **1-** Os agentes *captor1* e *captor2* dirigem-se para o bloco 2; **2-** O agente *captor2* obteve o bloco 2.

Em problemas complexos de interação, o comportamento poderá também ter de ser complexo para satisfazer os requisitos da solução. Esta possibilidade agrava-se em sistemas multi-agente compostos por agentes heterogêneos aos quais, para além da complexidade na definição do comportamento de cada um, acresce a complexidade da coordenação entre eles. Toda esta complexidade na definição de comportamentos pode dificultar, ou até mesmo tornar impossível, a sua definição de forma reativa.

Outro problema das arquiteturas reativas encontra-se na não otimização

das suas ações, que se agrava nos sistemas multi-agente, sobretudo naqueles cujo custo de falha é elevado, pois a convergência de um agente para uma solução subótima poderá fazer com que o sistema encontre também uma solução subótima, comprometendo assim todo o sistema.

A falta de coordenação demonstrada pela experimentação é uma outra falha das arquiteturas reativas quando servem de base a sistemas multi-agente. Como os comportamentos das arquiteturas reativas não têm por base qualquer representação do mundo, conceitos relevantes para a coordenação como “já se dirige para” não podem ser utilizados.

No sentido de colmatar as falhas desta arquitetura em termos individuais e coletivos, o próximo caso prático estuda a capacidade de cooperação de uma arquitetura deliberativa cuja coordenação é efetuada com recurso a planeamento descentralizado.

6.3 Interação Social com Base em Arquiteturas Deliberativas de Planeamento Descentralizado

A resolução distribuída de problemas através de um planeamento descentralizado é um método de interação social no qual um problema coletivo é decomposto em sucessivos subproblemas. Este tipo de planeamento caracteriza-se pelo facto de não existir um agente responsável pela divisão do problema, encontrando-se essa responsabilidade repartida pelos diversos agentes.

As arquiteturas deliberativas proporcionam ao agente uma característica adaptativa face ao estado do mundo que não é exequível nas arquiteturas reativas, evitando assim comportamentos subótimos que poderão prejudicar o sistema.

A cooperação em agentes deliberativos pode ser efetuada de forma descentralizada, ou seja, através dos seus planos cada agente resolve a sua parte do problema, fazendo evoluir o sistema multi-agente para uma solução final.

Neste caso prático, é instanciada a componente executiva e deliberativa da arquitetura proposta que visa colmatar as falhas da componente reativa instanciada na secção interior.

O subsistema executivo é responsável pela interação com o ambiente e é composto por um processo de execução de ação e outro de perceção.

O subsistema deliberativo implementa a componente de raciocínio do agente e é composto pelos processos de atualização de crenças, atualização do estado interno, deliberação e planeamento.

A organização funcional da arquitetura encontra-se representada na figura 6.6.

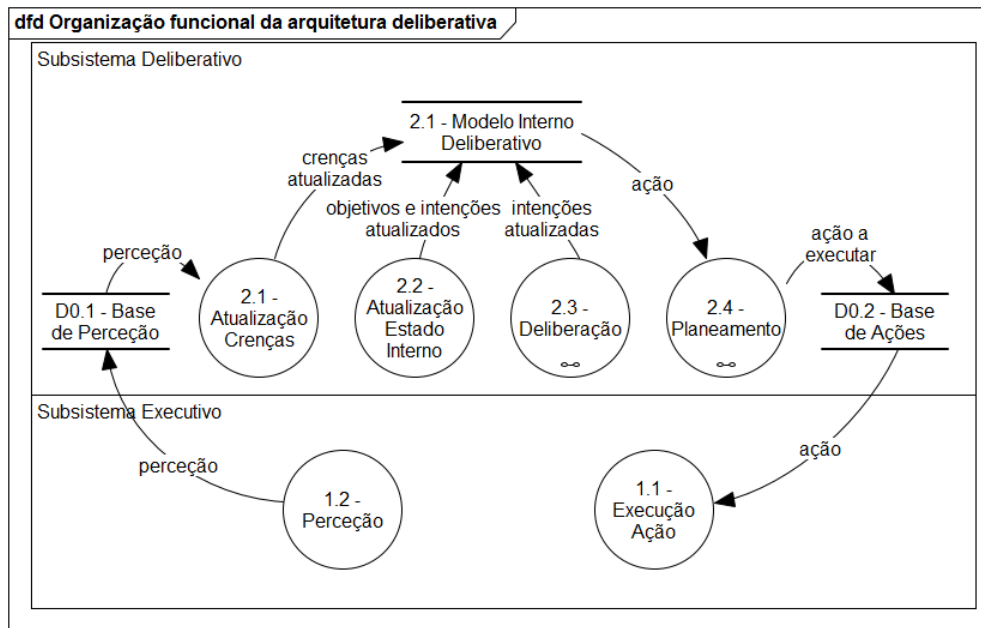


Figura 6.6: Organização funcional da arquitetura deliberativa.

Ao nível da sua arquitetura, o agente é composto por um controlo executivo e outro deliberativo. O controlo executivo implementa o subsistema executivo e agrega sensores e atuadores. O controlo deliberativo implementa o subsistema deliberativo e é composto pelo motor BDI proposto e por um planeador do tipo A*.

A figura 6.7 representa a arquitetura do agente deliberativo.

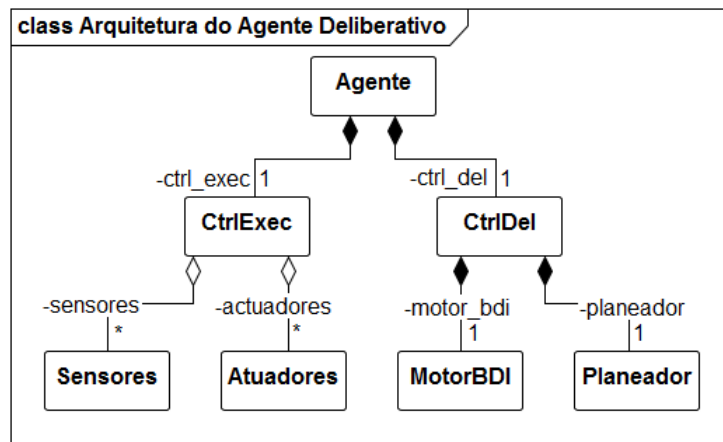


Figura 6.7: Arquitetura do agente deliberativo.

Tanto os agentes do tipo *captor*, como os do tipo *carrier*, são dotados de um conjunto de regras de inferência que se encontram descritas no apêndice C.1.

Os agentes do tipo *captor* são dotados dos desejos, planos e ações identificados no anexo C.2. Este tipo de agente tem como desejo a obtenção do bloco mais próximo e depois de obtido deseja livrar-se dele. Para se livrar do bloco, desloca-se para o *carrier* mais próximo e entrega-o.

Os agentes do tipo *carrier* são dotados dos desejos, planos e ações identificados no anexo C.3. Este tipo de agente tem como desejo transportar o bloco do agente do tipo *captor* mais próximo. Este desejo apenas é ativado quando existe um agente do tipo *captor* no estado cheio. Após a receção do bloco, é gerado o desejo de largar o bloco que ativa o plano de deslocação para a base mais próxima onde o larga.

6.3.1 Resultados Experimentais e Análise

Desenvolvida a simulação na plataforma experimental com os planos e regras de inferência anteriormente descritos, verifica-se que os agentes são capazes de entregar todos os blocos (fig. 6.8).

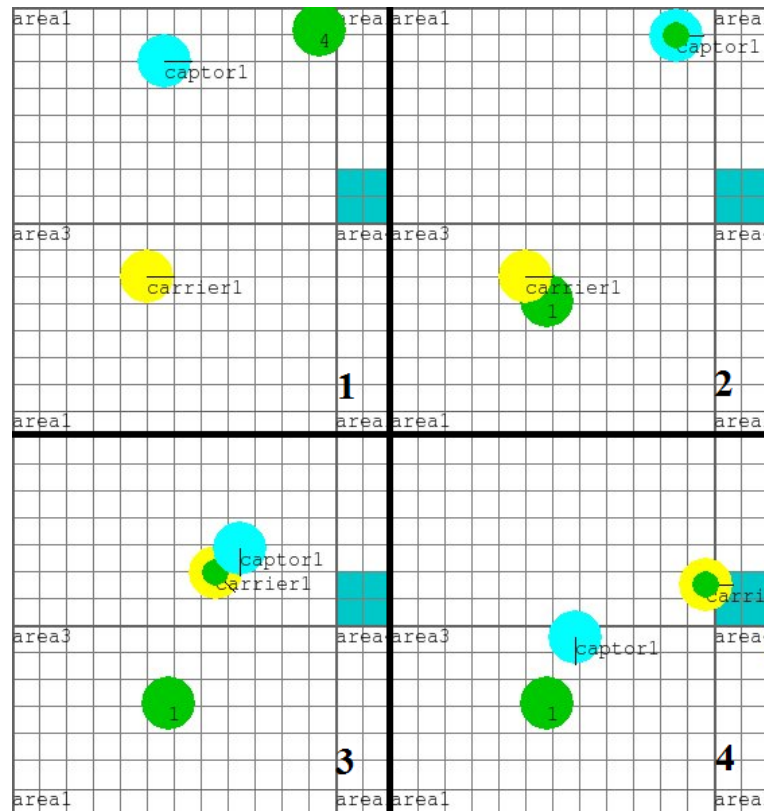


Figura 6.8: Exemplo das trajetórias dos agentes deliberativos de planejamento descentralizado.

Na figura 6.8 identificam-se os seguintes momentos: **1-** O agente *captor1* dirige-se para o bloco 4; **2-** O agente *captor1* obtém o bloco 4 e dirige-se para o agente *carrier1*. O agente *carrier1* vai ao encontro do agente *captor1*. **3-** Após aproximação, o agente *captor1* transfere o bloco para o agente *carrier1*; **4-** O agente *carrier1* entrega o bloco na base.

Não obstante, mantém-se a falta de coordenação de agentes já observada no exercício anterior, representada na figura 6.9, onde se verifica que os agentes *captor1* e *captor2* se dirigem para o mesmo bloco.

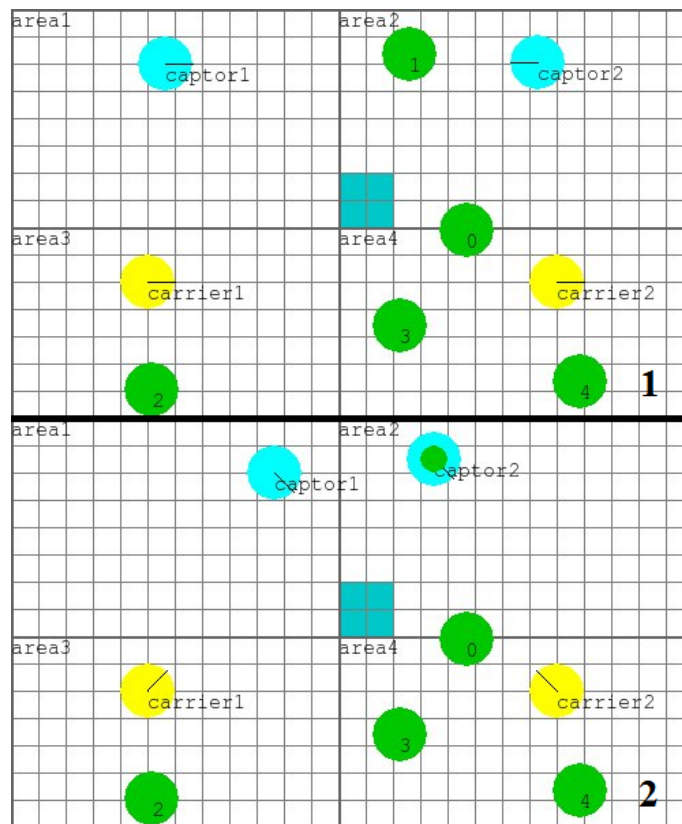


Figura 6.9: Exemplo do problema da falta de coordenação de agentes deliberativos de planeamento descentralizado.

Na figura 6.9 verificam-se os seguintes momentos: **1-** Os agentes *captor1* e *captor2* dirigem-se para o bloco 1; **2-** O agente *captor2* obteve o bloco 1.

Esta falta de coordenação deve-se à inexistência de regras de inferência mais complexas que permitam ao agente saber, por exemplo, se algum agente do tipo *captor* já se encontra a deslocar para um determinado bloco.

A descentralização do planeamento num contexto de interação social simples pode ser o ideal, ao não ser necessário um nível dedicado à coordenação (que consome recursos). No entanto, esta arquitetura apresenta dois problemas que podem determinar a sua inoperacionalidade em cenários mais complexos.

A descentralização do planeamento obriga a que os agentes necessitem de um modelo do mundo mais complexo, pois precisam de efetuar deliberação de

âmbito geral (objetivos de alto nível e respetiva decomposição), e específico (objetivos de baixo nível e respetiva concretização). A coordenação entre agentes, efetuada através de planeamento descentralizado, como verificado nos resultados experimentais, também pode ser um fator de exigência de modelos do mundo mais complexos.

As desvantagens desta arquitetura são minimizadas numa arquitetura de agente que permita a interação social com planeamento centralizado pois permite aos agentes um foco dos seus recursos em atividades de níveis específicos de abstração. O modelo do mundo terá a complexidade necessária a cada nível de abstração, fazendo com que os agentes não necessitem de despender tantos recursos na deliberação.

6.4 Interação Social com Base em Arquiteturas Deliberativas de Planeamento Centralizado

Uma arquitetura que implemente uma resolução distribuída de problemas através de um planeamento descentralizado pode enfrentar dificuldades quando enquadrada em cenários complexos. A coordenação eficiente dos elementos do sistema multi-agente através de um planeamento descentralizado é difícil, pois é elevada a complexidade da coordenação de agentes através dos planos e crenças acerca do mundo. Outro problema surge na quantidade de informação e níveis de decisão que cada agente, a nível individual, é obrigado a tratar, tendo este de efetuar decisões de coordenação em diversos níveis de detalhe e de concretizar ações, não lhe sendo possível especializar e focar os seus recursos num pequeno conjunto de tarefas perdendo, conseqüentemente, qualidade na solução final.

A elevada complexidade e a exigência de recursos computacionais para manter agentes coordenados através dos seus planos e respetivas pré-condições de ativação justificam a concretização de uma arquitetura de agente que cria um sistema multi-agente coordenado por meio de planeamento centralizado.

Esta solução permite uma especialização de agentes, cujos recursos ficam

exclusivamente dedicados a uma tarefa em particular. Determinados agentes ficam adstritos ao planeamento e distribuição de tarefas, enquanto outros ficam dedicados à execução de tarefas. Este enfoque de recursos permite o desenvolvimento de soluções com maior qualidade, ou com menor exigência de recursos.

É instanciada a arquitetura proposta no capítulo anterior, sendo o subsistema social do agente simplificado, pois apenas delega e recebe objetivos enquadrado num sistema multi-agente piramidal de estrutura fixa. A delegação de objetivos é feita de acordo com uma política de delegação de tarefas uniforme, ou seja, os objetivos são delegados em igual número a cada agente.

A arquitetura de agente instanciada é composta por um controlo executivo, um controlo reativo, um controlo deliberativo e um controlo social (fig. 6.7).

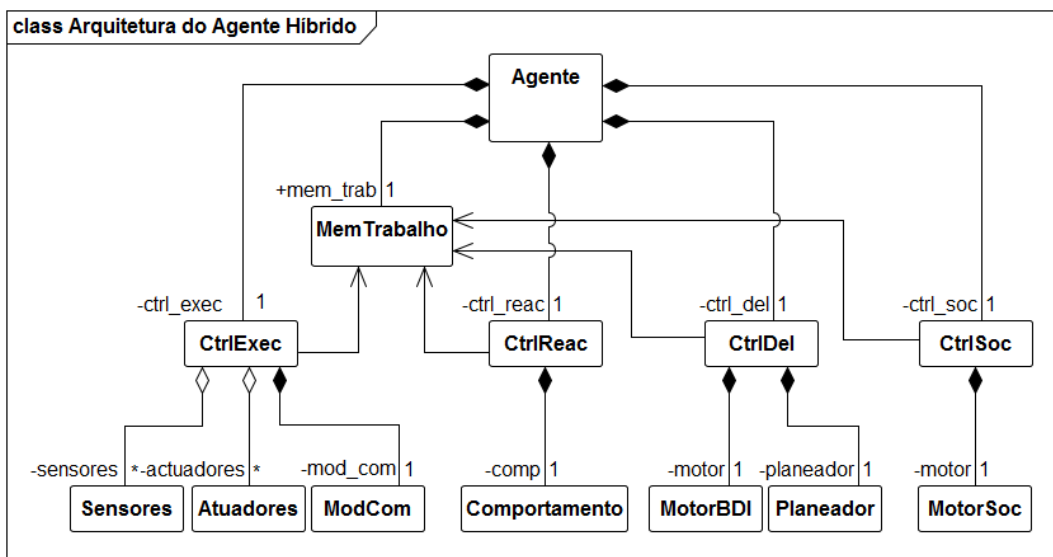


Figura 6.10: Arquitetura do agente híbrido.

Cada controlo implementa o subsistema correspondente e todos partilham uma memória de trabalho para comunicarem. O planeador do controlo deliberativo é do tipo A*.

Neste caso prático é introduzido um agente coordenador cuja missão é

realizar o planeamento geral. Todos os agentes são dotados de um conjunto de regras de inferência que se encontram descritas no apêndice C.1.

O agente coordenador é dotado de um conjunto de desejos e planos identificados no anexo D.2. Este agente tem como objetivo limpar todas as áreas de blocos e, para cada área com pelo menos um bloco, gera o desejo de a limpar. Os objetivos de limpeza de uma área são delegados aos agentes do tipo *captor*.

Os agentes do tipo *captor* são dotados dos planos e das ações identificados no anexo D.3. O agente do tipo *captor*, depois de delegada a área de limpeza, dá indicação ao seu colega de equipa (*carrier*) para se deslocar para a área determinada enquanto se desloca para lá também. Após a chegada da equipa à área, o *captor* inicia a obtenção de blocos que vai entregando ao seu colega de equipa.

Os agentes do tipo *carrier* são caracterizados pelos planos e ações identificados no anexo D.4. O agente do tipo *carrier* recebe e cumpre o objetivo de se deslocar para determinada área. De seguida, aproxima-se do seu coordenador, quando este lhe delega esse objetivo, e quando se encontra junto a este aguarda que lhe seja entregue o bloco, seguindo de seguida para a base.

6.4.1 Resultados Experimentais e Análise

Executados os agentes na plataforma experimental com os planos e regras de inferência apresentados, verifica-se que os agentes são capazes de entregar todos os blocos de forma coordenada com a limpeza feita por equipas e por área

Os resultados obtidos mostram que é possível obter-se coordenação num sistema multi-agente, utilizando poucos recursos computacionais através do estabelecimento de relações de dependência e poder, definidas numa estrutura organizacional hierárquica. A definição desta estrutura permite também que os agentes que nela operam otimizem a utilização dos seus recursos ao atribuir, a cada agente, um nível de abstração e função específicos.

A figura 6.11 apresenta uma exemplificação da dinâmica da limpeza de uma área.

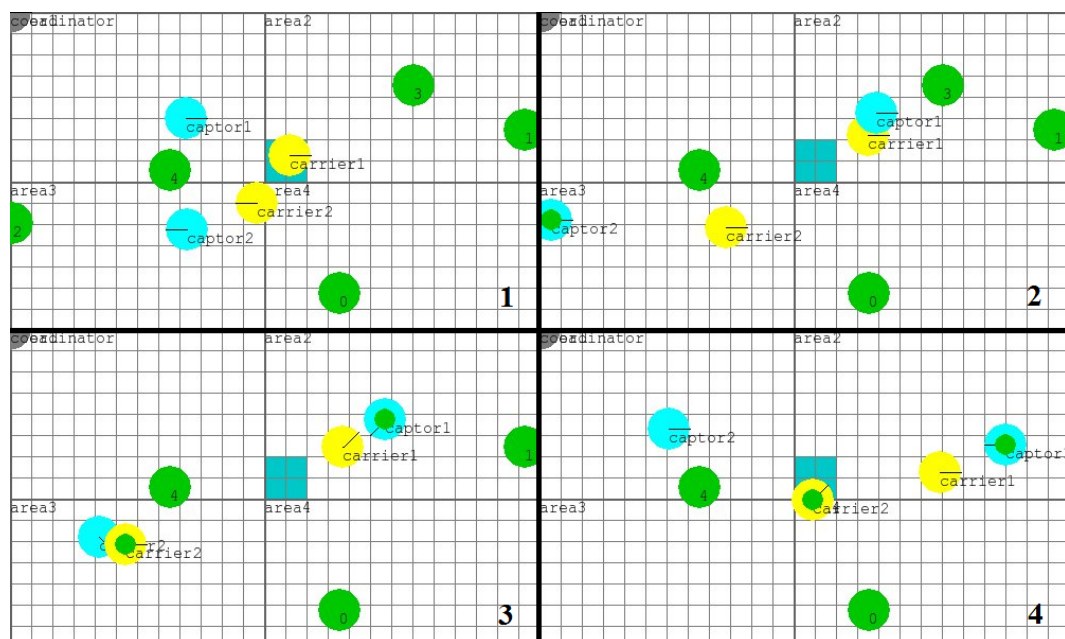


Figura 6.11: Exemplo das trajetórias dos agentes deliberativos de planeamento descentralizado.

Na figura 6.11 identificam-se os seguintes momentos: **1-** O agente *captor2* e *carrier2* chegaram à área 3. O agente *captor2* dirige-se para o bloco 2; **2-** O agente *captor2* obtém o bloco 2 e dirige-se para o agente *carrier2*. O agente *carrier2* vai ao encontro do agente *captor2*. **3-** Após aproximação, o agente *captor2* transfere o bloco para o agente *carrier2*; **4-** O agente *carrier1* entrega o bloco na base. O agente *captor2* como não possui mais nenhum bloco para obter na área 3, desloca-se para a área 1 onde aguarda pela chegada de *captor2* para iniciar a limpeza da área.

Terminado o estudo das potencialidades de coordenação de arquiteturas multi-agente, segue-se uma demonstração das potencialidades da utilização dos mecanismos emocionais na interação social nomeadamente no que concerne o problema de delegação de tarefas.

6.5 Interação Social com Base em Mecanismos Emocionais

Para ilustrar o papel dos mecanismos de base emocional no contexto da coordenação em agentes com recursos limitados, será considerado um cenário onde a coordenação é conseguida através de uma estrutura organizacional fixa composta por um agente coordenador e alguns agentes subordinados dotados de características de execução e condições de execução heterogêneas. Os agentes subordinados informam o agente coordenador da sua disposição emocional, tal como representado na figura 6.12. A disposição emocional resulta da situação do agente subordinado perante as tarefas que lhe foram delegadas. O agente coordenador irá utilizar a respetiva disposição emocional dos agentes subordinados para decidir quantas tarefas delegar a cada agente, na respetiva iteração.

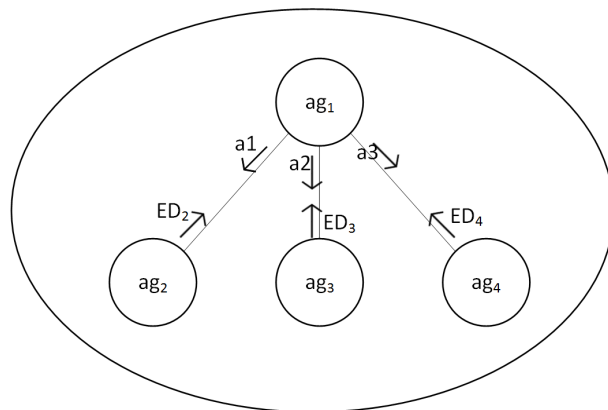


Figura 6.12: Comunicação da disposição emocional no contexto da delegação de tarefas.

Este tipo de organização é consistente com o que acontece por exemplo em sociedades humanas, nas quais as emoções assumem um papel importante na coordenação de interações sociais [Keltner e Haidt, 2001]. Os estados emocionais de um agente são formados a nível individual como meio de regulação cognitiva. Contudo, se ao agente forem comunicadas as disposições emocionais dos restantes agentes, estas terão também influência na formação da sua disposição emocional e, desta forma, permitirá uma regulação cognitiva que inclui as condições de concretização dos motivadores dos restantes agentes

do sistema.

6.5.1 Delegação de Tarefas de Base Emocional

A política de delegação de tarefas do agente coordenador determina a eficiência global do sistema. Porém, é necessário ter em consideração que determinadas políticas, especialmente as baseadas em modelos de raciocínio, poderão não ser compatíveis com as limitações de recursos existentes. De uma forma alternativa, uma política fixa de delegação de tarefas pode ser aplicada, na qual um número fixo de tarefas é delegado aos restantes agentes independentemente das condições de execução. Embora a sua execução não seja pesada computacionalmente, esta política de delegação de tarefas não é adaptável às alterações das condições de execução.

Uma política de delegação de tarefas baseada em emoção serve de alternativa às anteriores, mantendo uma diminuta complexidade computacional e fornecendo adaptabilidade dinâmica. O Modelo Emocional de Fluxo servirá de base para a definição das dinâmicas emocionais dos agentes e para a criação dos padrões de comportamento do agente coordenador face à disposição emocional apresentada pelos agentes.

O objetivo desta abordagem é manter cada agente subordinado na situação de concretização mais favorável possível, de forma a maximizar a eficiência de execução dos agentes subordinados e, conseqüentemente, de todo o sistema. Dessa forma, o agente coordenador irá recorrer à disposição emocional dos agentes subordinados comunicada por estes, e adaptar a sua política de delegação de tarefas de acordo com a tabela 6.1.

A tabela 6.1 concretiza a tabela 4.1 no problema da delegação de tarefas, onde define uma política de delegação de tarefas de acordo com o comportamento esperado e definido em 4.1.

Quadrante ED	Situação	Emoção Observada	Política de Delegação
Q-I	Favorável	<i>Alegria</i>	Aumentar o número de tarefas delegadas
Q-II	Adversa	<i>Zanga</i>	Diminuir o número de tarefas delegadas
Q-III	Perigo	<i>Medo</i>	Cancelar tarefas delegadas e diminuir o número de tarefas a delegar
Q-IV	Perda	<i>Tristeza</i>	Parar a delegação de tarefas

Tabela 6.1: Política de delegação de tarefas baseada em emoção.

Um agente numa situação favorável e com tendência a melhorar irá apresentar uma disposição emocional correspondente ao quadrante Q-I (*alegria*), pelo que é viável aumentar o número de tarefas a delegar.

Um agente que apresente uma disposição emocional no quadrante Q-II (*zanga*) indica que se encontra numa situação favorável mas com tendência a piorar, pelo que é necessário aliviar o número de tarefas a delegar.

Um agente, numa situação de perigo, apresenta uma disposição emocional enquadrada no quadrante Q-III (*medo*), indicando que não está a conseguir lidar com a situação, a qual se deteriora. Nestas situações deve cancelar-se tarefas delegadas e diminuir o número de tarefas a delegar.

Numa situação de perda, um agente apresenta uma disposição emocional no quadrante Q-IV (*tristeza*), indicando que a situação é desfavorável mas com tendência a melhorar. Nestas condições deve ser interrompida a delegação de tarefas, de forma a permitir ao agente recuperar.

6.5.2 Disposição Emocional dos Agentes Subordinados

Desenvolvida a política de delegação de tarefas é necessário definir a estrutura cognitiva do agente de acordo com a subsecção 4.4.2, ou seja, observações, motivadores e mediadores.

A capacidade de execução (CE) corresponde ao número de tarefas que o agente, numa iteração e em condições normais, consegue realizar. Assim, o espaço cognitivo de cada agente é composto apenas por uma dimensão - φ_{CE} - que representa a informação CE medida em número de tarefas disponíveis, como exemplificado na figura 6.13.

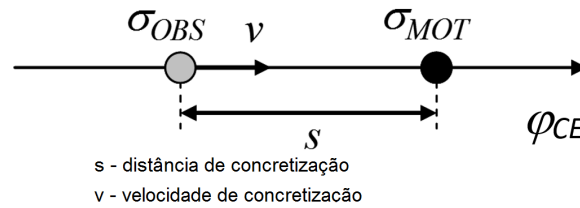


Figura 6.13: Representação do espaço cognitivo de um agente subordinado.

A motivação do agente é a de atingir a sua capacidade de execução máxima e as observações correspondem à distância relativamente à capacidade de execução, em número de tarefas.

6.5.3 Resultados Experimentais e Análise

A simulação e a extração de resultados são efetuadas com recurso à plataforma *Emotional Task Delegation* (anexo A) e são compostas por um agente coordenador com 400 tarefas para distribuir por dois agentes subordinados (ag_1 e ag_2). Os dois agentes subordinados têm como missão a concretização das tarefas e têm uma capacidade de execução (CE) de 20 tarefas, significando que conseguem resolver 20 tarefas por iteração. O agente ag_1 tem condições adversas de execução, pode não conseguir atingir a sua capacidade de execução podendo, inclusive, não conseguir resolver nenhuma tarefa. Por outro lado, o agente ag_2 tem condições de execução favoráveis e pode, inclusive, exceder a sua capacidade de execução até ao limite de 40 iterações. A variação das condições de execução são simuladas por distribuições aleatórias uniformes.

São utilizadas três políticas de delegação de tarefas: (i) política uniforme, (ii) política de base emocional e (iii) política ótima. Por cada política de delegação de tarefas foram efetuadas 100 simulações.

Na política de delegação de tarefas uniforme, o agente coordenador delega 50 tarefas a cada agente por iteração. Na política de delegação de tarefas baseada na emoção, o agente coordenador adapta o número de tarefas a ser delegado a cada agente de acordo com os padrões de comportamento definidos na tabela 6.1, começando com 20 tarefas a delegar por agente. Aplicando a política de delegação de tarefas ótima, a cada iteração, o agente coordenador antecipa as condições de concretização que os agentes subordinados irão enfrentar na iteração seguinte, atribuindo-lhes o número de tarefas estritamente necessário. Esta política ótima serve apenas para efeito de comparação, pois a antecipação das condições de concretização, numa situação prática não é realizável, pois implica a capacidade de prever o futuro.

O objetivo do agente coordenador é orientar a execução de tarefas de forma a que a sua resolução seja efetuada o mais rápido possível sendo, conseqüentemente, o desempenho do processo de execução de tarefas medido pelo número de iterações necessárias para resolver o conjunto de tarefas.

A figura 6.14 apresenta os resultados comparativos obtidos pelas três políticas de delegação de tarefas definidas: uniforme, baseada em emoção e ótima.

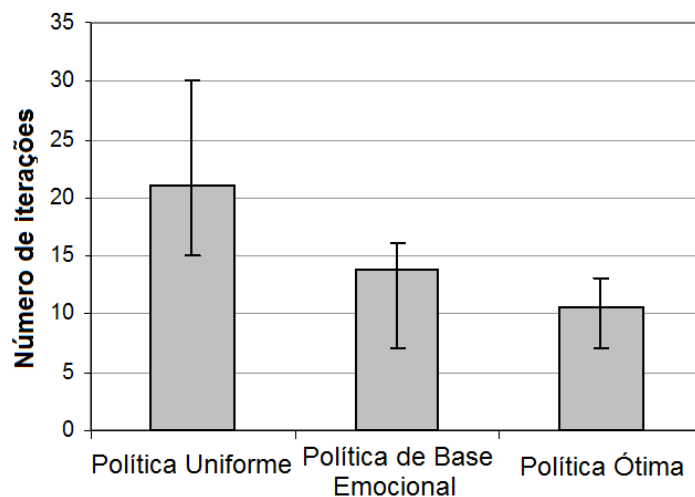


Figura 6.14: Resultados para as políticas de distribuição de tarefas uniforme, baseada na emoção e ótima.

No gráfico da figura, é possível observar que a política de delegação de tarefas baseada em emoção melhora significativamente o desempenho do processo de execução de tarefas relativamente à política de delegação uniforme, aproximando-se do desempenho obtido pela política ótima.

As alterações que justificam a diferença entre as políticas uniforme e a de base emocional podem ser observadas na perspectiva do agente. Sob a política uniforme, ambos os agentes são mantidos quase todo o tempo nas disposições emocionais correspondentes ao *medo* (Q-III) e *tristeza* (Q-IV), como se verifica na figura 6.15. Estas disposições emocionais resultam do facto de os agentes não conseguirem executar as tarefas delegadas imediatamente, ficando a acumular-se em lista de espera. Neste problema em concreto, se um agente apresenta uma disposição emocional correspondente ao quadrante Q-III (*medo*) significa que lhe foi delegado, ou tem vindo a ser delegado, um número de tarefas superior ao que tem conseguido concretizar, diminuindo o potencial e o fluxo de concretização que convergem para valores negativos, significando que a situação tende a deteriorar-se.

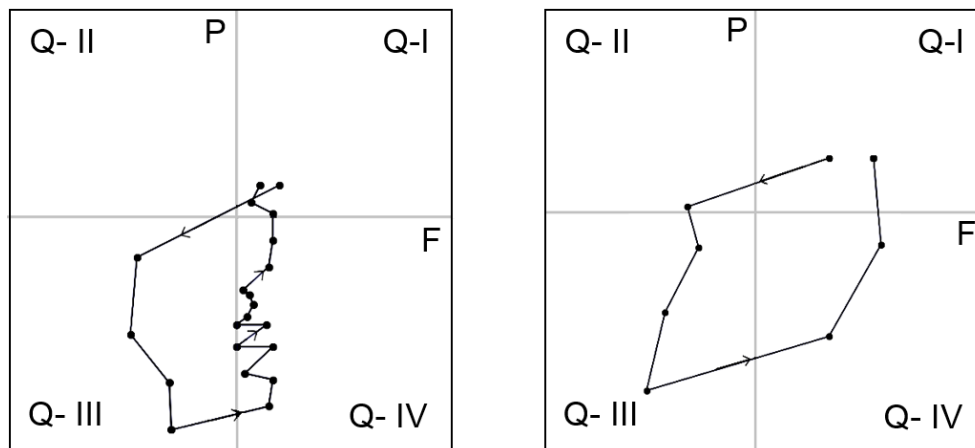


Figura 6.15: Disposição emocional dos agentes ag_1 (esquerda) e ag_2 (direita) sob a política de delegação de tarefas uniforme.

Nos agentes com disposição emocional de *medo* (Q-III) e *tristeza* (Q-IV), as tarefas mantidas em espera pelo agente não podem ser delegadas a outros agentes, causando uma degradação geral do desempenho. Este cenário pode ser constatado na figura 6.15 onde ambos os agentes passam do Q-III para o Q-IV (param de receber tarefas). Após a entrada no Q-IV, o agente ag_2

completa todas as suas tarefas em duas iterações, enquanto o agente ag_1 demora 15 iterações.

A figura 6.16 mostra a evolução das disposições emocionais dos agentes subordinados sob a política de distribuição de tarefas baseada na emoção. Como pode ser observado, ambos os agentes mantêm-se principalmente nas disposições emocionais correspondentes aos quadrantes Q-I (*alegria*) e Q-II (*zanga*).

O potencial de concretização positivo indica que os agentes estão a operar dentro da sua capacidade de execução disponível. Todavia, no caso do agente ag_1 que se depara com condições de execução adversas, em alguns casos, são observadas disposições emocionais enquadradas no quadrante Q-III (*medo*). Apesar disso, o agente consegue recuperar graças ao comportamento adaptativo do coordenador que considera a disposição emocional comunicada pelos agentes subordinados.

Outro aspeto relativo à disposição emocional dos agentes subordinados que pode ser observado na figura 6.16 é a estabilidade da sua evolução, demonstrando uma adaptação do comportamento de delegação do coordenador às capacidades disponíveis dos agentes subordinados.

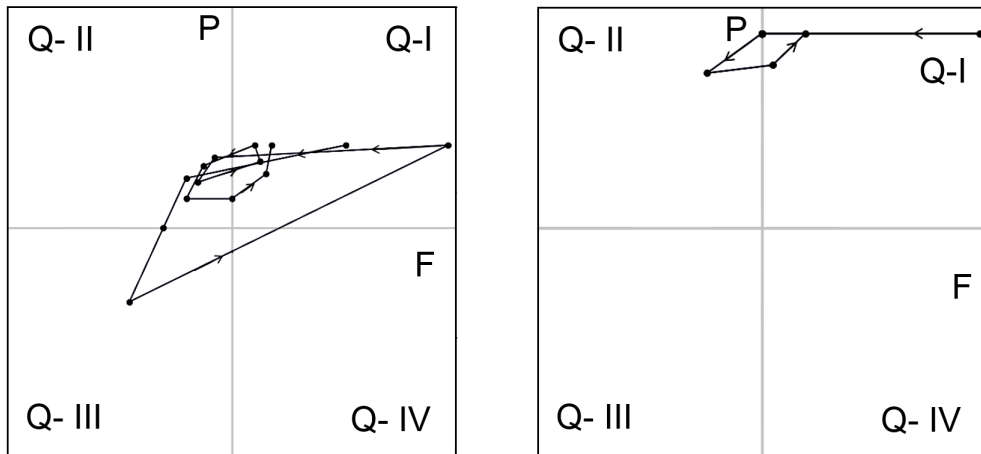


Figura 6.16: Disposição emocional dos agentes ag_1 (esquerda) e ag_2 (direita) sob a política de delegação de tarefas baseada em emoções.

É importante salientar que se os agentes subordinados apenas comunicassem ao coordenador um único componente da disposição emocional como

P ou F , ou mesmo comunicassem apenas a informação referente ao número de tarefas pendentes, o coordenador não teria informação suficiente para selecionar um comportamento adequado e.g. a situação é favorável ($\delta P > 0$), mas tende a deteriorar-se ($\delta F < 0$). Por outras palavras, as disposições emocionais fornecem informação integrada, respeitante à situação de concretização de um agente e que não se encontra disponível nos seus componentes individuais isolados.

6.6 Conclusão

No presente capítulo foram efetuados vários testes de arquiteturas de agentes e respetivas características de interação social, que serviram de suporte às opções tomadas na arquitetura proposta. Foi testado e demonstrado que as arquiteturas de agentes reativos possibilitam a criação de sistemas multi-agente que consomem baixos recursos, mas que podem apresentar resultados subótimos. Foi também testada a capacidade de interação social de uma arquitetura deliberativa cuja coordenação era obtida através de planeamento descentralizado, arquitetura essa que provou ser uma solução viável para a criação de um sistema multi-agente que opere em ambientes de recursos limitados. Contudo, a coordenação deste sistema torna-se difícil devido à descentralização do planeamento. Por último, foi desenvolvida e testada uma arquitetura de agente híbrida capaz de operar dentro de uma estrutura organizacional em pirâmide e cuja coordenação resulta de um planeamento centralizado e respetiva delegação de tarefas.

Foi igualmente proposta e demonstrada a utilidade de uma política de delegação de tarefas de base emocional na resolução do problema da delegação de tarefas enquadrado em ambientes de recursos limitados.

CAPÍTULO 7

Conclusão

O presente trabalho teve como objetivo o estudo da interação social entre agentes centrada na limitação de recursos.

Foi efetuado o enquadramento teórico das arquiteturas de agentes inteligentes reativas e deliberativas, com especial foco na arquitetura BDI, tendo sido estudadas implementações desta arquitetura.

O estudo do tema da interação social comportou aspetos da comunicação entre agentes e de métodos de interação social concretizados pelos métodos do raciocínio social, estruturas organizacionais e licitação.

As arquiteturas de agente e os métodos de interação social estudados foram enquadrados no problema da limitação de recursos, estudando-se o efeito da racionalidade limitada e da possibilidade de gestão destas limitações através da regulação cognitiva. Concluiu-se que a regulação cognitiva de base emocional é uma boa opção para a regulação dos processos internos de um agente, sobretudo quando enquadrada em problemas de limitação de recursos. Foi estudado o Modelo Emocional de Fluxo, com vista à sua integração na arquitetura de agente como modelo de geração de disposições emocionais.

No seguimento do estudo, foi proposta e demonstrada a potencialidade de uma política de delegação de tarefas de base emocional. No entanto, estudos mais aprofundados podem ser realizados no sentido de aprimorar a política de delegação de tarefas proposta.

O estudo de outras aplicações dos fenómenos emocionais no contexto da interação social também pode ser efetuado, nomeadamente a sua utilização no raciocínio social.

Neste trabalho foi proposta uma arquitetura de agente híbrida, capaz de operar socialmente em ambientes de recursos limitados, sendo para o efeito dotada de mecanismos de regulação cognitiva e capaz de se enquadrar e coordenar numa estrutura social hierárquica.

A estrutura organizacional hierárquica proposta permite aos agentes coordenarem-se através de planeamento centralizado. Para tornar o sistema multi-agente mais robusto a falhas individuais dos agentes foram definidos protocolos de entrada e saída na estrutura organizacional.

A proposta da arquitetura fundamenta-se em testes efetuados em arquiteturas de agentes reativos e deliberativos. Os agentes reativos demonstraram limitações na capacidade de coordenação, atingindo níveis subótimos de coordenação. As arquiteturas deliberativas de planeamento descentralizado apresentam melhorias na coordenação, relativamente às arquiteturas reativas. Porém, a complexidade da definição de planos e regras de inferência torna este tipo de solução menos atrativa para a operação com recursos limitados. Por fim, uma versão simplificada da arquitetura proposta foi testada, aferindo-se as vantagens da utilização de um planeamento centralizado para a coordenação em recursos limitados.

Não foi constatada a influência dos parâmetros de sensibilidade nas políticas de delegação de tarefas definidos no Modelo Emocional de Fluxo, pelo que o seu estudo é também um dos aspetos a considerar como trabalho futuro.

Fica a curiosidade da possibilidade do modelo proposto permitir o estudo, ou até mesmo a validação, de modelos de interação social já propostos, pela investigação da influência de determinadas emoções que surgem nos agentes

a nível individual e a sua influência na qualidade final da solução apresentada pelo sistema.

As arquiteturas apresentadas, sobretudo as deliberativas, poderão ser aperfeiçoadas no sentido de melhorar a sua eficiência interna.

Pelo aprofundamento do presente trabalho surgiu o interesse pelo desenvolvimento de uma linguagem de programação de agentes com ênfase declarativa, contrastando com as linguagens procedimentais existentes.

APÊNDICE A

Plataforma *Emotional Task Delegation*

A plataforma *Emotional Task Delegation* permite desenvolver políticas de delegação de tarefas, ao simular o problema da delegação de tarefas.

Esta plataforma permite criar estruturas hierárquicas de agente sem limitação do número de níveis da hierarquia.

Cada agente é caracterizado por: (i) nome, (ii) número de tarefas pendentes, (iii) número de tarefas que consegue resolver por iteração (*niter*), (iv) percentagem de variação positiva e negativa do número de tarefas que consegue resolver por iteração (*np*, *nn*), (vi) agentes aos quais pode distribuir tarefas e (vii) valores dos coeficientes de acumulação do potencial e fluxo de concretização (4.4.3).

A cada iteração, o simulador percorre o sistema multi-agente de forma descendente em toda a sua hierarquia executando o processamento interno de cada agente. O processo interno de um agente inclui os processos: (i) resolução das tarefas pendentes, (ii) determinação da sua disposição emocional e (iii) informação ao coordenador sobre a sua disposição emocional.

O processo de resolução das tarefas pendentes difere consoante o tipo de agente. No caso de ser um agente coordenador (tem agentes a quem distribuir tarefas), qualquer política de delegação de tarefas pode ser utilizada, não havendo limites para o número de tarefas distribuídas. Um agente executor (sem agentes a quem distribuir tarefas) simula a resolução das suas tarefas pendentes em número igual a $niter - (nn \times niter) + (np \times niter)$.

Uma simulação encontra-se concluída quando nenhum dos agentes que compõem o sistema multi-agente possui tarefas por resolver. No final de cada iteração, são apresentados e gravados os resultados da simulação em tabelas e gráficos.

A.1 Arquitetura da Plataforma *Emotional Task Delegation*

Na figura A.1 encontra-se uma representação da arquitetura da plataforma *Emotional Task Delegation*.

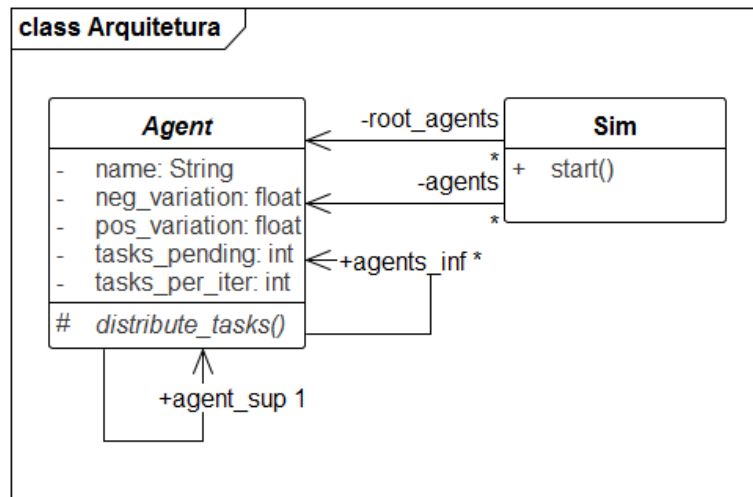


Figura A.1: Arquitetura da plataforma *Emotional Task Delegation*.

Esta arquitetura é composta por um simulador e por um agente. A definição do agente é abstrata, devendo ser generalizada caso se pretenda implementar um agente coordenador. A implementação de um agente que resolve

tarefas é feita pela instanciação de um objeto da classe *Agent*.

A dinâmica desta plataforma encontra-se representada na figura A.2.

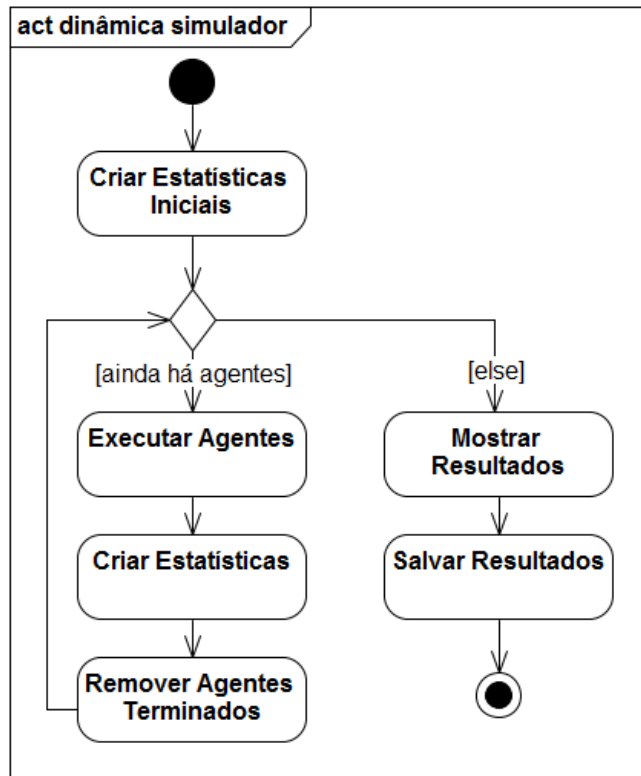


Figura A.2: Dinâmica da plataforma de simulação de delegação de tarefas.

Quando é iniciado, o simulador cria as estatísticas iniciais e, enquanto houver agentes, executa o seu processo interno, cria as estatísticas da ronda e remove os agentes do topo da hierarquia que já não têm tarefas a delegar/resolver. Quando não há mais agentes com tarefas por resolver, o simulador dá a simulação por terminada, mostra e salva os resultados.

Durante a simulação, o topo da hierarquia é dinâmico e deste são removidos os agentes que o compõem no momento em que já não têm mais tarefas a delegar. Quando são removidos do topo da hierarquia, é dado o lugar do agente removido aos seus agentes subordinados.

O ciclo de execução do agente encontra-se caracterizado na figura A.3.

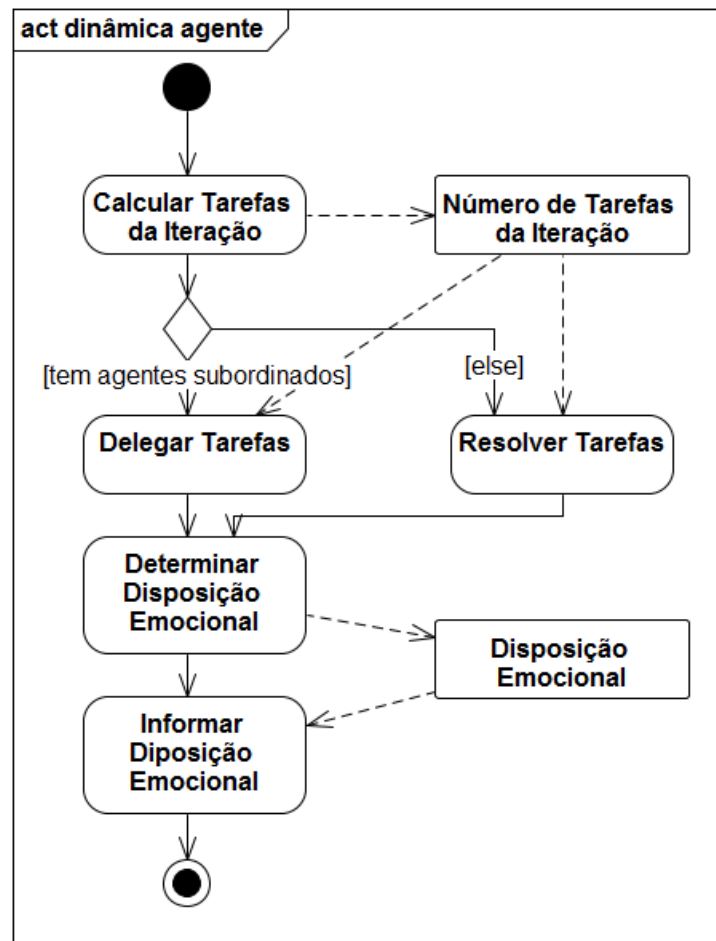


Figura A.3: Dinâmica de um ciclo de execução do agente.

Por cada chamada do método de execução do agente é definido e calculado o número de tarefas que o agente conseguirá resolver na iteração. De seguida, o agente delega tarefas se for coordenador, ou resolve-as caso contrário. Após a atividade de delegação ou resolução de tarefas, o agente determina a sua disposição emocional e informa-a ao seu coordenador.

APÊNDICE B

Plataforma *Tile World Multi-Agent Sim*

A plataforma *Tile World Multi-Agent Sim* (twmas) tem por base o problema *tile world*, no qual um agente tem como missão obter e transportar os blocos (*tiles*) para uma base.

Para proporcionar um ambiente onde seja necessária a cooperação entre agentes, o problema *tile world* é estendido para um cenário onde existem dois tipos diferentes de agente: *captor* e *carrier*.

Um agente do tipo *captor* tem como missão obter os blocos e entregá-los a um agente *carrier*. Este tipo de agente é dotado das capacidades:

- **move:** locomoção pelo mundo, com ou sem bloco carregado;
- **pass:** entrega do bloco a um agente do tipo *carrier*.

Um agente do tipo *carrier* recebe os blocos do agente do tipo *captor* e entrega-os na base, sendo dotado das capacidades:

- **move:** locomoção pelo mundo, com ou sem bloco carregado;
- **drop:** entrega do bloco na base.

A figura B.1 representa a dinâmica da captura de um bloco e respectiva entrega na base.

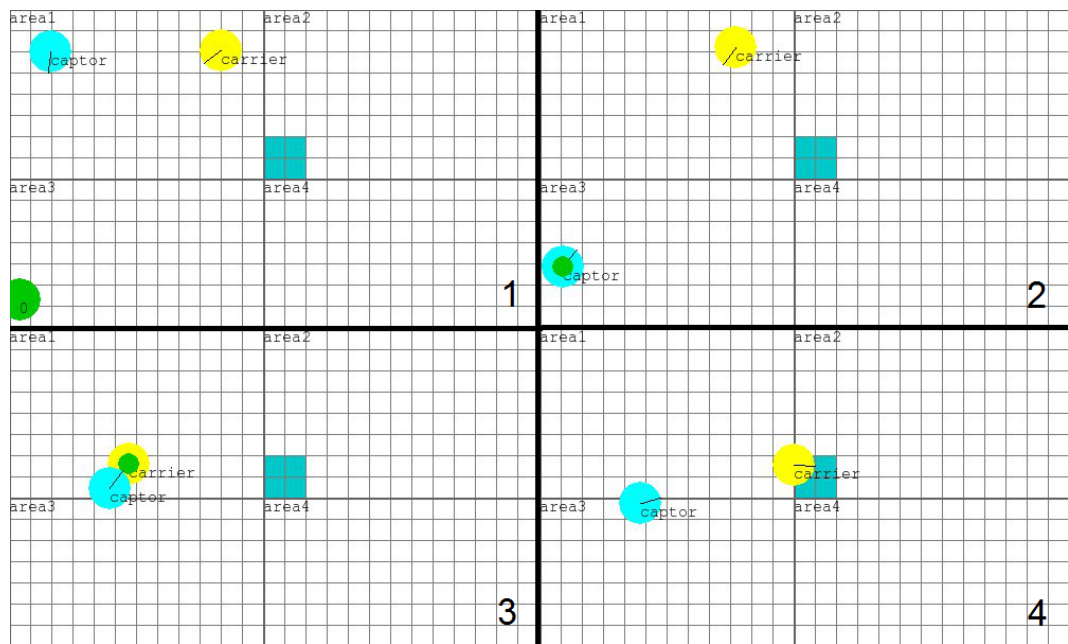


Figura B.1: Demonstração da obtenção e entrega de um bloco.

A dinâmica de captura de um bloco é composto por quatro momentos. O momento 1 representa o início de um hipotético cenário, onde um *captor*, um *carrier* e um bloco encontram-se no cenário. No momento 2 o agente do tipo *captor* dirige-se para o bloco e captura-o. Após a captura do bloco o agente do tipo *captor* entrega-o ao agente do tipo *carrier* - momento 3. Finalmente no momento 4, o agente do tipo *carrier* dirige-se à zona da base onde entrega o bloco.

Os elementos que compõem o mundo onde os agentes navegam são estáticos, com exceção dos blocos que aparecem e desaparecem em intervalos de tempo aleatórios dentro de um espaço de tempo pré-determinado.

B.1 Arquitetura da Plataforma *Tile World MultiAgent Sim*

Esta plataforma simula uma variante do problema *tile world* onde dois tipos de agente, *captor* e *carrier*, têm de se coordenar de forma a obter os blocos e entregá-los no local respetivo. Pretende-se que esta arquitetura permita a utilização de qualquer arquitetura de agente, independentemente da linguagem em que é definido o agente.

O modelo de domínio da arquitetura desta plataforma encontra-se definido na figura B.2.

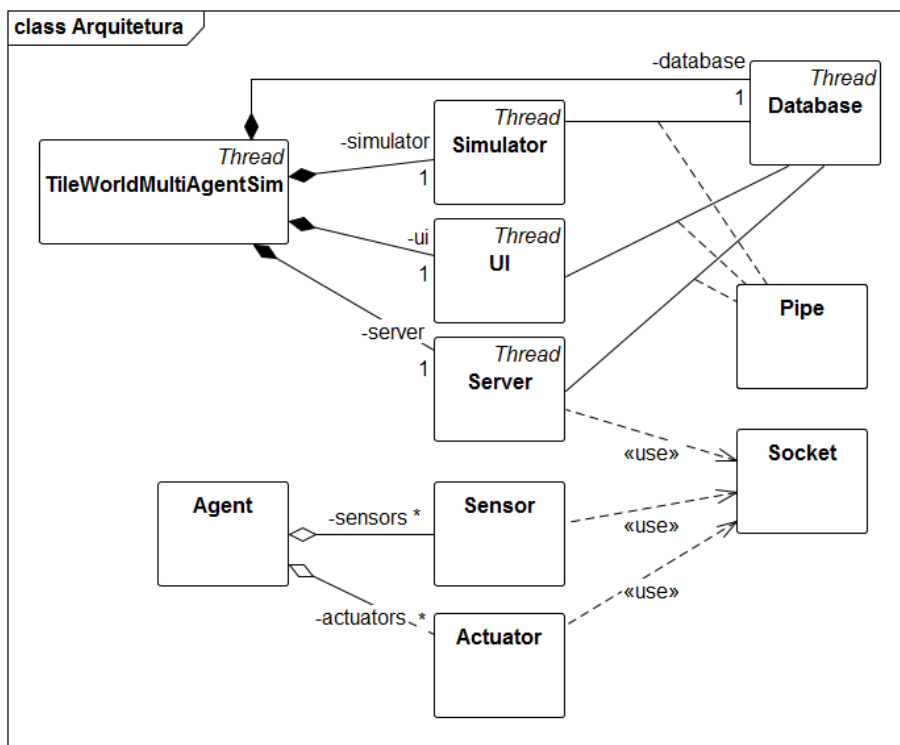


Figura B.2: Arquitetura da plataforma *Tile World MultiAgent Sim*.

A plataforma de simulação é composta por quatro componentes principais: um simulador, uma interface, um servidor e uma base de dados.

A base de dados é o componente que armazena toda a informação de simulação como a informação do mapa, dos agentes e dos blocos. Os restantes

componentes comunicam com esta de forma a modificar ou consultar a sua informação.

O simulador é o componente responsável pela simulação. Consulta a informação dos agentes na base de dados e atualiza a posição destes de acordo com a sua posição e vetor de movimento (fig. B.3).

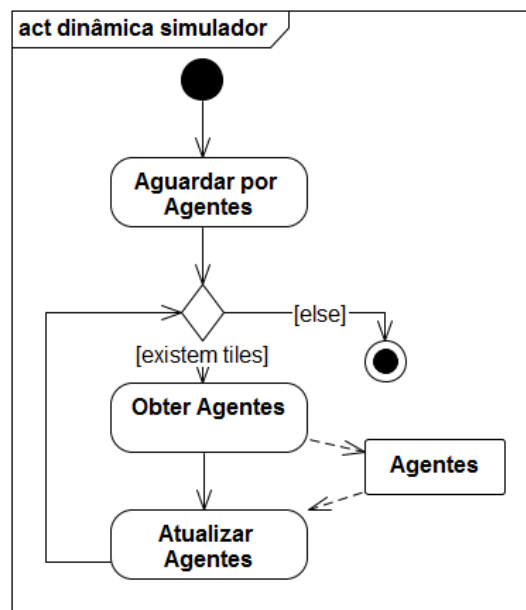


Figura B.3: Dinâmica do simulador.

A interface é o componente que apresenta o resultado da simulação numa interface e grava as imagens em disco.

O servidor é o ponto de comunicação dos agentes com o simulador. Através de *sockets*, os agentes podem consultar ou atualizar a sua informação.

Cada um destes componentes funciona de forma independente, em *threads* separadas.

APÊNDICE C

Regras de Inferência, Desejos, Planos e Ações 1

C.1 Regras de Inferência

```
closest_tile(TileName) :-  
    findall(Dist, bel(tile(_, _, Dist, _)), Z),  
    list_min(Z, Min),  
    bel(tile(TileName, _, Min, _)),  
    !.
```

```
closest_agent(AgentName, AgentType, State) :-  
    findall(Dist,  
            bel(agent(  
                AgentName, _, AgentType, _,  
                Dist, -, -, State, -))),  
            Z),  
    list_min(Z, Min),  
    bel(agent(  
        AgentName, _, AgentType, _,  
        Min, -, -, State, -)),  
    !.
```

```

closest_base(CX, CY) :-
    findall(Dist, bel(
        cell(-, -, base, -, -, -, Dist, -, -)),
        Z),
    list_min(Z, Min),
    bel(cell(-, -, base, CX, CY, -, Min, -, -)),
    !.

in_capture_range(TileName) :-
    bel(tile(TileName, -, -, 1)).

very_near_to_agent(AgentName) :-
    bel(agent(AgentName, -, -, -, -, -, -, -, 1)).

inside_cell(X, Y) :-
    bel(cell(-, -, -, X, Y, -, -, 1, -)).

empty :-
    bel(state(empty)).

full :-
    bel(state(full)).

list_min([L|Ls], Min) :-
    list_min(Ls, L, Min).

list_min([], Min, Min).
list_min([L|Ls], Min0, Min) :-
    Min1 is min(L, Min0),
    list_min(Ls, Min1, Min).

```

C.2 Captor

```

goal(capture_nearest_tile, 50, maintain).

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = capture_nearest_tile,
    Utility = 50,

```

```

PreCond = [ closest_tile(TileName), empty ],
Context = [],
Body = [
    achieve(in_capture_range(TileName), 100),
    capture_tile(TileName),
    achieve(empty, 100)
].

```

```

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = capture_nearest_tile,
    Utility = 50,
    PreCond = [ in_capture_range(TileName), empty ],
    Context = [],
    Body = [
        capture_tile(TileName),
        achieve(empty, 100)
    ].

```

```

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = in_capture_range(TileName),
    Utility = 50,
    PreCond = [ closest_tile(TileName) ],
    Context = [ closest_tile(TileName) ],
    Body = [ move_to_tile(TileName) ].

```

```

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = empty,
    Utility = 50,
    PreCond = [
        closest_agent(AgentName, carrier, empty),
        full
    ],
    Context = [
        closest_agent(AgentName, carrier, empty),
        full
    ],
    Body = [
        move_to_agent(AgentName),
        pass_tile(AgentName)
    ].

```

```

action(Name, PosCond) :-
    Name = capture_tile(-),
    PosCond = [full].

action(Name, PosCond) :-
    Name = move_to_tile(TileName),
    PosCond = [in_capture_range(TileName)].

action(Name, PosCond) :-
    Name = move_to_agent(AgentName),
    PosCond = [very_near_to_agent(AgentName)].

action(Name, PosCond) :-
    Name = pass_tile(-),
    PosCond = [empty].

```

C.3 Carrier

```

goal(transport_nearest_tile, 50, maintain).

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = transport_nearest_tile,
    Utility = 50,
    PreCond = [closest_agent(AgentName, captor, full),
empty],
    Context = [closest_agent(AgentName, captor, full)],
    Body = [achieve(very_near_to_agent(AgentName), 100),
achieve(empty, 100)].

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = very_near_to_agent(AgentName),
    Utility = 50,
    PreCond = [closest_agent(AgentName, captor, full)],
    Context = [closest_agent(AgentName, captor, full)],
    Body = [move_to_agent(AgentName)].

plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = full,
    Utility = 50,
    PreCond = [very_near_to_agent(AgentName),

```

```
        closest_agent (AgentName, captor, full)],
Context = [very_near_to_agent (AgentName),
           closest_agent (AgentName, captor, full)],
Body = [].
```

```
plan(Goal, Utility, PreCond, Context, Body) :-
    Goal = empty,
    Utility = 50,
    PreCond = [closest_base (X,Y), full],
    Context = [closest_base (X,Y), full],
    Body = [move_to_cell (X,Y), drop_tile].
```

```
action(Name, PosCond) :-
    Name = drop_tile,
    PosCond = [bel(empty)].
```

```
action(Name, PosCond) :-
    Name = move_to_cell (X,Y),
    PosCond = [bel(inside_cell (X,Y))].
```

```
action(Name, PosCond) :-
    Name = move_to_agent (AgentName),
    PosCond = [bel(very_near_to_agent (AgentName))].
```


APÊNDICE D

Regras de Inferência, Desejos, Planos e Ações 2

D.1 Regras de Inferência

```
area_has_tile(AreaName) :-  
    bel(area(AreaName, XI, YI, XF, YF, -, -, -)),  
    bel(tile(-, XT, YT, -)),  
    XI =< XT,  
    XT =< XF,  
    YI =< YT,  
    YT =< YF.
```

```
tile_on_area(Tile, AreaName) :-  
    bel(area(AreaName, XI, YI, XF, YF, -, -, -)),  
    bel(tile(Tile, XT, YT, -)),  
    XI =< XT,  
    XT =< XF,  
    YI =< YT,  
    YT =< YF.
```

```
area(AreaName) :-
```

```

        bel(area(AreaName, -, -, -, -, -, -, -)).

team_member(Agent) :-
    bel(agent(Agent, Team, -, -, -, -, -)),
    bel(team(Team)).

team_member(Agent, Type) :-
    bel(agent(Agent, Team, Type, -, -, -, -)),
    bel(team(Team)).

closest_carrier(Agent) :-
    findall(Dist, bel(distance(_, Dist)), Distances),
    min_list(Distances, Dist),
    bel(distance(Agent, Dist)).

on_carrier(Agent) :-
    bel(distance(Agent, Dist)),
    bel(agent(Agent, -, carrier, -, -, -, -)),
    Dist <= 1.5.

on_area(Area) :-
    bel(area(Area, -, -, -, -, -, -, 1)).

distance(Agent, Dist) :-
    bel(agent(Agent, -, -, X, Y, -, -)),
    bel(pos(X1, Y1)),
    Dist is sqrt((X-X1)**2 + (Y-Y1) **2).

```

D.2 Coordinator

```

des(all_areas_cleared, 50, keep).

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = clear_all_areas,
    Des = all_areas_cleared,
    Utility = 50,
    PreCond = [area_has_tile(Area)],
    Context = [],
    Body = [
        forall(area(Area)),

```

```

        achieve(area_cleared(Area), 100, not_keep)
    ].

```

D.3 Captor

```

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = clear_area,
    Des = area_cleared(Area),
    Utility = 50,
    PreCond = [area_has_tile(Area)],
    Context = [],
    Body = [
        [
            move(area, Area),
            achieve(team_on_area(Area), 200, not_keep)
        ],
        achieve(area_has_no_tile(Area), 300, not_keep)
    ].

```

```

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = move_team_to_area,
    Des = team_on_area(Area),
    Utility = 50,
    PreCond = [],
    Context = [],
    Body = [
        forall(team_member(Agent)),
        achieve(on_center_area(Agent, Area), 200, not_keep)
    ].

```

```

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = clear_area,
    Des = area_has_no_tile(Area),
    Utility = 50,
    PreCond = [],
    Context = [tile_exists(Tile)],
    Body = [
        forall(tile_on_area(Tile, Area)),
        achieve(tile_captured(Tile), 400, not_keep)
    ].

```

```

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = capture_tile,
    Des = tile_captured(Tile),
    Utility = 50,
    PreCond = [],
    Context = [],
    Body = [
        move(tile, Tile),
        capture(Tile),
        achieve(free, 500, not_keep)
    ].

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = get_free,
    Des = free,
    Utility = 50,
    PreCond = [full],
    Context = [],
    Body = [
        forany(closest_carrier(Carrier)),
        [
            move(agent, Carrier),
            achieve(close_to(Carrier), 600, not_keep)
        ],
        pass(Carrier)
    ].

act(Name, PreCond, PosCond) :-
    Name = move(area, Area),
    PreCond = [],
    PosCond = [on_center_area(Area)].

act(Name, PreCond, PosCond) :-
    Name = move(tile, Tile),
    PreCond = [],
    PosCond = [on_tile(Tile)].

act(Name, PreCond, PosCond) :-
    Name = move(agent, Agent),
    PreCond = [],

```

```
PosCond = [close_to(Agent)].
```

```
act(Name, PreCond, PosCond) :-
    Name = capture(_),
    PreCond = [free],
    PosCond = [full].
```

```
act(Name, PreCond, PosCond) :-
    Name = pass(_),
    PreCond = [],
    PosCond = [free].
```

D.4 Carrier

```
plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = get_closer,
    Des = close_to(_),
    Utility = 50,
    PreCond = [],
    Context = [],
    Body = [
        forany(coordinator(Coordinator)),
        move(agent, Coordinator),
        achieve(full, 100, not_keep),
        achieve(free, 200, not_keep)
    ].
```

```
plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = go_to_area,
    Des = on_center_area(_, Area),
    Utility = 50,
    PreCond = [],
    Context = [],
    Body = [move(area, Area)].
```

```
plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = get_free,
    Des = free,
    Utility = 50,
    PreCond = [],
```

```

Context = [],
Body = [
    forany(cell(XC, YC, base)),
    move(cell, XC, YC),
    drop
].

plan(PlanName, Des, Utility, PreCond, Context, Body) :-
    PlanName = capture_tile,
    Des = full,
    Utility = 50,
    PreCond = [],
    Context = [],
    Body = [
        wait_for_pass
    ].

act(Name, PreCond, PosCond) :-
    Name = move(area, Area),
    PreCond = [],
    PosCond = [on_center_area(Area)].

act(Name, PreCond, PosCond) :-
    Name = move(cell, XC, YC),
    PreCond = [],
    PosCond = [on_cell(XC, YC)].

act(Name, PreCond, PosCond) :-
    Name = move(agent, Coordinator),
    PreCond = [],
    PosCond = [close_to(Coordinator)].

act(Name, PreCond, PosCond) :-
    Name = drop,
    PreCond = [],
    PosCond = [free].

act(Name, PreCond, PosCond) :-
    Name = wait_for_pass,
    PreCond = [],
    PosCond = [full].

```

Bibliografia

- [Anderson e Oates, 2005] Anderson, M. e Oates, T. (2005). *Metacognition in Computation*. AAAI Spring Symposium, Technical Report SS-05-04, AAAI Press.
- [Arkin, 1998] Arkin, R. C. (1998). *Behaviour-Based Robotics*. MIT Press.
- [Austin, 1962] Austin, J. L. (1962). *How to do Things with Words*. Oxford: Clarendon Press.
- [Bordini et al., 2007] Bordini, R., Hubner, J. F., e Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley and Sons Ltd.
- [Bratman, 1987] Bratman (1987). *Intentions, Plans and Practical Reason*. Harvard University Press.
- [Brooks, 1985] Brooks, R. A. (1985). *A Robust Layered Control System For a Mobile Robot*. Massachusetts Institute Of Technology.
- [Castelfranchi, 1990] Castelfranchi, C. (1990). *Social Power: A missing point in Multi-Agent, DAI and HCP*. in Y. Demazeau and J. P. Muller (eds.) *Decentralized A. I.*, Elsevier Science Publishers B. V. 1990. p. 49-62.

- [Conte e Castelfranchi, 1995a] Conte, R. e Castelfranchi, C. (1995a). *Cognitive and Social Action*. UCL Press.
- [Conte e Castelfranchi, 1995b] Conte, R. e Castelfranchi, C. (1995b). *Cognitive and Social Action*. UCL Press.
- [Ferber et al., 2004] Ferber, J., Gutknecht, O., e Michel, F. (2004). *From Agents to Organizations: an Organizational View of Multi-Agent Systems*. University of Montpellier II.
- [Genesereth e Fikes, 1992] Genesereth, M. R. e Fikes, R. E. (1992). *Knowledge Interchange Format Version 3 Reference Manual*. Logic-92-1, Stanford University Logic Group.
- [Grossi et al., 2005] Grossi, D., Dignum, F., Dastani, M., e Royakkers, L. (2005). *Foundations of Organizational Structures in Multiagent Systems*. In: AAMAS'05, Proceedings of Fourth International Conference on Autonomous Agents and Multiagent Systems, pp. 690-697. ACM Press, New York (2005).
- [Ingrand et al., 1996] Ingrand, F. F., Chatila, R., Alami, R., e Robert, F. (1996). *PRS: A High Lever Supervision and Control Language for Autonomous Mobile Robots*. Proceedings of the 1996 IEEE International Conference on Robotics and Automation (Minneapolis, USA).
- [Inverno et al., 1997] Inverno, M., Luck, M., e Wooldridge, M. (1997). *Cooperation Structures*. Proceedings of the 15th International Joint Conference on Artificial Intelligence.
- [Jennings, 1999] Jennings, N. R. (1999). *On agent-based software engineering*.
- [Jennings et al., 2004] Jennings, N. R., Sierra, C., Sonenberg, L., e Tambe, M. (2004). *CooAgentSpeak: Cooperation in AgentSpeak through plan exchange*. ACM Press.
- [Keltner e Haidt, 2001] Keltner, D. e Haidt, J. (2001). *Social functions of emotions*. In T. Mayne & G. A. Bonanno (Eds.), *Emotions: Current issues and future directions*, Guilford Press.

- [Laplante e Ovaska, 2011] Laplante, P. A. e Ovaska, S. J. (2011). *Real-Time Systems Design and Analysis - Tools for the Practitioner 4th ed.*
- [Machado e Bordini, 2002] Machado, R. e Bordini, R. H. (2002). *Running AgentSpeak(L) agents on SIM_AGENT.*
- [Malone, 1988] Malone, T. (1988). *What is Coordination Theory.* MIT, Boston, USA: MIT Sloan School of Management.
- [Morgado, 2013] Morgado, L. (2013). *Apresentação: Racionalidade Limitada.* ISEL.
- [Morgado, 2014] Morgado, L. (2014). *Apresentação: Sistemas Multi-Agente.* ISEL.
- [Morgado e Gaspar, 2000] Morgado, L. e Gaspar, G. (2000). *A Social Reasoning Mechanism Based on a New Approach for Coalition Formation.* 2nd Int. Symposium "From Agent Theory to Agent Implementation", 15th European Meeting on Cybernetics and Systems Research".
- [Morgado e Gaspar, 2007] Morgado, L. e Gaspar, G. (2007). *A Signal Based Approach to Artificial Agent Modeling.* Advances in Artificial Life, 1050-1059.
- [Morgado e Gaspar, 2008] Morgado, L. e Gaspar, G. (2008). *Towards Background Emotion Modeling for Embodied Virtual Agents.* Proc. 7th Int. Conf. on Autonomous Agents and Multiagent Systems.
- [Morgado, 2005] Morgado, L. F. G. (2005). *Integração de Emoção e Raciocínio em Agentes Inteligentes.* Faculdade de Ciências da Universidade de Lisboa.
- [Moulin e Cloutier, 1994] Moulin, B. e Cloutier, L. (1994). *Collaborative Work based on Multi-Agent Architectures: A Methodological Perspective.* In Aminzadeh, F., and Jamshidi, M., eds., *Soft Computing: Fuzzy Logic, Neural Networks and Distributed Artificial Intelligence.* Prentice Hall. 261-296.
- [Murphy, 2000] Murphy, R. R. (2000). *Introduction to AI Robotics.* A Bradford Book.

- [Rae e Georgeff, 1995] Rae, A. S. e Georgeff, M. P. (1995). *BDI Agents: From Theory to Practice*.
- [Russel e Norvig, 2009] Russel, S. e Norvig, P. (2009). *Artificial Intelligence: A Modern Approach, 3rd Edition*. Prentice Hall.
- [Searle, 1969] Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.
- [Sichman et al., 1993] Sichman, J., Demazeau, Y., Conte, R., e Castelfranchi, C. (1993). *A Social Reasoning Mechanism Based On Dependence Networks*.
- [Sichman, 2003] Sichman, J. S. (2003). *Raciocínio Social e Organizacional em Sistemas MultiAgentes: Avanços e Perspectivas*. Escola Politécnica da Universidade de São Paulo.
- [Smith e Randall, 1980] Smith, R. G. e Randall, D. (1980). *Frameworks for Cooperation in Distributed Problem Solving*. IEEE Transactions on Computers, v. 29, n.12.
- [Vieira et al., 2007] Vieira, R., Álvaro Moreira, Wooldridge, M., e Bordini, R. H. (2007). *On the Formal Semantics of Speech-Act Based Communication in an Agent-Oriented Programming Language*. Journal of Artificial Intelligence Research 29.
- [Weiss, 1999] Weiss, G. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press.
- [Wooldridge, 2002] Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems*. John Wiley and Sons.
- [Wooldridge e Jennings, 1994] Wooldridge, M. e Jennings, N. R. (1994). *Towards a Theory of Cooperative Problem Solving*. Pre-proceedings of the 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World.
- [Wooldridge e Jennings, 1995] Wooldridge, M. e Jennings, N. R. (1995). *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review.

-
- [Wooldridge et al., 2000] Wooldridge, M., Jennings, N. R., e David, K. (2000). *The Gaia Methodology for Agent-Oriented Analysis and Design*. Journal of Autonomous Agents and Multi-Agent Systems.
- [Wooldridge e Parsons, 1999] Wooldridge, M. e Parsons, S. (1999). *Intention Reconsideration Reconsidered*.