

**Uma Arquitetura para Agentes Inteligentes
baseada na Sociedade da Mente**

Mauren Fernanda Meira Brenner

Dissertação de Mestrado

Uma Arquitetura para Agentes Inteligentes baseada na Sociedade da Mente

Mauren Fernanda Meira Brenner¹

Dezembro de 1996

Banca Examinadora:

- Prof. Dr. Heloísa Vieira da Rocha (Orientadora)
- Prof. Dr. Ariadne M. B. R. Carvalho (Co-orientadora)
- Prof. Dr. Jacques Wainer
- Prof. Dr. Marcus Vinicius Poggi de Aragão (PUC-RJ)
- Prof. Dr. Luiz Eduardo Buzato (Suplente)

¹A autora é Engenheira de Computação pela Universidade Estadual de Campinas.



UNIDADE	BC
N.º CHAMADA:	
T.º	Unicamp
	B751a
V.	Ex.
T.º MGO BC/	32.220
PROC.	281191
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	25/11/97
N.º CPD	

CM-00103003-3

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Brenner, Mauren Fernanda Meira

B751a Uma arquitetura para agentes inteligentes baseada na sociedade da mente / Mauren Fernanda Meira Brenner -- Campinas, [S.P. :s.n.], 1996.

Orientadora : Heloisa Vieira da Rocha

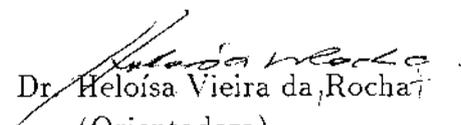
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Inteligência artificial. 2. Inteligência artificial distribuída. I. Rocha, Heloisa Vieira. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Uma Arquitetura para Agentes Inteligentes baseada na Sociedade da Mente

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Mauren Fernanda Meira Brenner e apro-
vada pela Banca Examinadora.

Campinas, 12 de Dezembro de 1996.

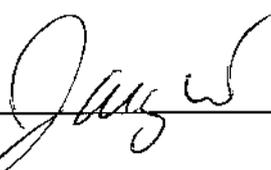

Prof. Dr. Heloisa Vieira da Rocha
(Orientadora)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

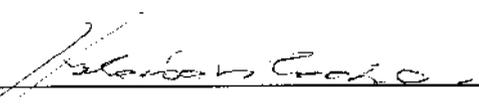
Tese de Mestrado defendida e aprovada em 12 de dezembro de 1996 pela Banca Examinadora composta pelos Professores Doutores



Prof^o. Dr^o. Marcus V. Poggi Aragão



Prof^o. Dr^o. Jacques Wainer



Prof^a. Dr^a. Heloisa Vieira da Rocha

© Mauren Fernanda Meira Brenner, 1997.
Todos os direitos reservados.

Agradecimentos

Às professoras Heloísa e Ariadne pela orientação.

Ao CNPq e FAPESP pelo apoio financeiro.

Resumo

A Sociedade da Mente é o nome dado a um conjunto de propostas de estruturas e mecanismos subjacentes à mente e à inteligência. Estas propostas foram desenvolvidas por Marvin Minsky e apresentadas principalmente durante a década de 80. Entre elas está a noção de que a mente é constituída por unidades desprovidas de inteligência chamadas agentes, unidades estas que podem ser identificadas em diversos níveis de abstração, sendo o mais “baixo” deles o nível equivalente ao neural. A noção de agente é usada para definir os conceitos e mecanismos relativos ao funcionamento integrado da mente como um todo, ou seja, relativos à coordenação dos agentes. O conceito central é o de estado mental parcial, que corresponde ao estado de um subconjunto dos agentes da mente. Estados mentais parciais constituem o meio de comunicação entre grupos de agentes, que na Sociedade da Mente são chamados agências, e são controlados através de agentes especiais que implementam determinados mecanismos de coordenação.

Neste trabalho, a Sociedade da Mente é usada como fundamento para uma arquitetura para agentes inteligentes. Uma vez que na Sociedade da Mente a mente é composta por agentes, segue-se que o próprio agente também deve ser constituído de agentes menores, coordenados mediante os mecanismos propostos na Sociedade da Mente. Assim, projetamos uma arquitetura para agentes inteligentes baseada na Sociedade da Mente como uma arquitetura de um sistema multi-agentes (um sistema de Inteligência Artificial Distribuída), onde a comunicação e coordenação entre os agentes é feita de uma forma inspirada no conceito de estados mentais parciais e nos mecanismos de coordenação da Sociedade da Mente.

Descrevemos o processo de desenvolvimento no qual a arquitetura foi transformada desde um mapeamento direto entre agentes e agências da Sociedade da Mente e agentes da arquitetura multi-agentes, até a sua versão final baseada no modelo de *blackboard*, o qual constitui um paradigma bem conhecido em Inteligência Artificial Distribuída. Na arquitetura resultante, os mecanismos de coordenação da Sociedade da Mente servem como modelo para objetos organizados nas diferentes seções do *blackboard*. Esses objetos são “ativos,” no sentido de que não constituem apenas informações a serem manipuladas, mas possuem também funcionalidades específicas, segundo as quais atuam como os mecanismos de coordenação da Sociedade da Mente.

Mencionamos algumas possibilidades de aplicações usando a arquitetura desenvolvida e descrevemos em detalhe a implementação de uma dessas aplicações, que consiste em um agente que conta uma história. Esta implementação constituiu em uma experiência com o intuito de testar não somente a viabilidade da arquitetura proposta, mas também a própria metodologia deste trabalho, que foi a de tomar o modelo teórico da Sociedade da Mente como ponto de partida em lugar de projetar uma arquitetura para realizar alguma tarefa específica.

Abstract

The Society of Mind is a collection of proposals concerning structures and mechanisms underlying mind and intelligence. These proposals were developed by Marvin Minsky and presented mostly during the 80s. Among them, there is the notion that the mind is made up of mindless units called agents. Such units can be identified at many levels, the lowest of them being the equivalent to the neural level. The Society of Mind relies upon the notion of agent to define the concepts and mechanisms concerning the working of the mind as a whole, i.e. concerning the coordination of agents. The central concept is the partial mental state, which corresponds to the description of the states of some agents of the mind. Groups of agents, which are called agencies in the Society of Mind, communicate through partial mental states, which are controlled by special agents that implement certain coordination mechanisms.

In this work, the Society of Mind is taken as a foundation for an intelligent agent architecture. Since in the Society of Mind the mind itself comprises many agents, it follows that the intelligent agent should be made up from smaller agents, coordinated through the mechanisms of the Society of Mind. Thus, we have designed an intelligent agent architecture based on the Society of Mind as an architecture of a multi-agent system (a Distributed Artificial Intelligence system), where the communication and coordination between agents is done in a way which is inspired on the concept of partial mental states and on the coordination mechanisms of the Society of Mind.

We have described the design process through which the architecture was transformed from a direct mapping between agents and agencies of the Society of Mind to its final version based on the blackboard model, which is a well-known Distributed Artificial Intelligence paradigm. The coordination mechanisms of the Society of Mind work as models for objects placed in the different sections of the blackboard. These objects are “active” in the sense that they do not only contain information to be operated upon, but also have specific functionalities according to which they act like the coordination mechanisms of the Society of Mind.

We have mentioned some possibilities of applications using the architecture we developed, and described in some detail the implementation of one of those applications: a story-telling agent. This implementation was an experience whose objective was to test

the feasibility of the proposed architecture as well as the methodology of this work itself, which consisted of taking the theoretical model of the Society of Mind as a starting point instead of designing an architecture to perform some particular task.

Conteúdo

Agradecimentos	iv
Resumo	v
Abstract	vii
1 Introdução	1
2 A Sociedade da Mente	8
2.1 Introdução	8
2.2 Agentes	9
2.3 Agentes e agências	10
2.4 As abordagens simbólica e conexionista na Sociedade da Mente	12
2.5 Mecanismos de coordenação na Sociedade da Mente	13
2.5.1 Estados mentais parciais	13
2.5.2 K-Lines	14
2.5.3 Polynemes	17
2.5.4 Micronemes	19
2.5.5 Pronomes e paranomes	19
2.5.6 Frames	20
2.5.7 Reconhecedores	24
2.5.8 Supressores e censores	24
2.5.9 Agências administrativas	24
2.6 Aprendizado e desenvolvimento na Sociedade da Mente	25
2.7 A Sociedade da Mente como fundamento para uma arquitetura para agentes inteligentes	26
2.8 Conclusão	27
3 Inteligência Artificial Distribuída	29
3.1 Introdução	29

3.2	O projeto de um sistema de IAD	31
3.2.1	Os agentes do sistema	31
3.2.2	Cooperação	32
3.2.3	Controle	33
3.2.4	Conflitos	34
3.2.5	Comunicação e interação	34
3.2.6	Organização	36
3.2.7	A arquitetura interna de um agente em um sistema de IAD	36
3.3	Arquiteturas de Inteligência Artificial Distribuída	38
3.3.1	A arquitetura de Rede de Contratos	38
3.3.2	A arquitetura de blackboard	41
3.4	Conclusão	44
4	O desenvolvimento de uma arquitetura para agentes inteligentes baseada na Sociedade da Mente	45
4.1	Introdução	45
4.2	O modelo da Sociedade da Mente e a arquitetura de um agente	46
4.3	Desenvolvimento da arquitetura	47
4.3.1	Uma sociedade de agentes	47
4.3.2	Agrupando agentes de coordenação	51
4.3.3	Modificando o esquema de comunicação	53
4.3.4	A versão final da arquitetura	54
4.4	Aplicações	57
4.4.1	Um agente de senso comum	58
4.4.2	Um agente de busca bibliográfica	61
4.4.3	Um agente que conta uma história	64
4.4.4	Sociedades de agentes baseados na Sociedade da Mente	65
4.5	Comparação com a arquitetura M	65
4.6	Conclusão	67
5	A implementação de um agente	68
5.1	Introdução	68
5.2	A organização interna do agente	71
5.2.1	A organização das informações	71
5.2.2	As conexões entre as informações	73
5.2.3	O mecanismo de evocação de informações relacionadas	74
5.2.4	O mecanismo de evocação de assuntos	74
5.2.5	O processo de evolução do foco de atenção	75
5.2.6	Os agentes internos	77

5.3	Implementação	78
5.3.1	Os objetos do blackboard	78
5.3.2	O blackboard	80
5.3.3	Os agentes internos	84
5.4	Resultados e extensões	85
5.5	Conclusão	92
6	Conclusão	93
6.1	Discussão	93
6.2	Direções futuras	100
6.3	Conclusão	104
	Bibliografia	106

Lista de Figuras

2.1	Conexão entre agentes através de um barramento.	10
2.2	Exclusão cruzada.	10
2.3	Agentes e agências.	11
2.4	Uma agência com estrutura hierárquica.	12
2.5	Uma <i>k-line</i>	14
2.6	Faixas de níveis.	15
2.7	A memória de um evento: uma <i>k-line</i> conectada a outras <i>k-lines</i>	16
2.8	Camadas de <i>k-lines</i> em torno de uma agência.	16
2.9	Um <i>polyneme</i> representando uma maçã.	17
2.10	Um agente reconhecedor.	18
2.11	O <i>loop</i> de <i>reminding</i> ou rememoração.	18
2.12	Um <i>frame</i>	20
2.13	Um <i>trans-frame</i>	21
2.14	<i>Frames</i> de imagens.	22
2.15	<i>Interaction-square array</i> e <i>direction-nemes</i>	23
3.1	A arquitetura de um agente de uma Rede de Contratos.	37
3.2	Outra arquitetura de um agente.	37
4.1	Um sistema fictício.	51
4.2	O sistema anterior na segunda versão da arquitetura.	52
4.3	O sistema anterior na terceira versão da arquitetura.	53
4.4	A versão final da arquitetura.	54
4.5	Um tipo de reconhecedor.	56
4.6	Outro tipo de reconhecedor.	56
4.7	Uma <i>k-line</i>	56
4.8	Representação de um livro.	63
4.9	Agentes de busca bibliográfica e de senso comum integrados.	64
5.1	Reconhedores conectando itens de dados, <i>k-lines</i> e palavras-chave.	75
5.2	A hierarquia de classes correspondentes aos objetos do <i>blackboard</i>	79

5.3	A hierarquia de classes correspondentes aos agentes.	84
-----	--	----

Capítulo 1

Introdução

Este trabalho descreve o desenvolvimento de uma arquitetura para agentes inteligentes baseada na Sociedade da Mente, um modelo teórico-computacional da mente proposto por Marvin Minsky na década de 80 [13]. A proposta foi de partir deste modelo conceitual e projetar uma arquitetura genérica para agentes inteligentes incorporando algumas das características do modelo. Isto foi feito através da adequação dos mecanismos básicos da Sociedade da Mente ao paradigma de Inteligência Artificial Distribuída (IAD) ou sistemas multi-agentes, o qual consideramos como sendo o paradigma mais adequado para a arquitetura do agente, uma vez que na Sociedade da Mente a mente é composta por agentes menores.

A idéia de projetar uma arquitetura para agentes inteligentes baseada na Sociedade da Mente surgiu a partir da concepção de um ambiente computacional totalmente baseado em agentes, com o conceito de agente substituindo o conceito de aplicação, com um estilo de interação baseado mais na delegação de tarefas para estes agentes do que no modelo de manipulação direta, e amparada no princípio de que o usuário pode obter virtualmente tudo o que quiser através da combinação das funcionalidades dos agentes que estão à sua disposição, dada a sua interoperabilidade. Estes e outros princípios constituem os fundamentos do que chamamos de “domínios multi-agentes.” Ambientes computacionais desta natureza devem compreender agentes de diversos tipos, dos mais simples aos mais sofisticados. Decidimos então explorar possíveis arquiteturas de agentes que pudessem algum dia integrar um ambiente computacional deste tipo.

A arquitetura de um agente deve ser entendida como composta de duas partes: uma parte relativa à funcionalidade e outra relativa à interação. A parte de interação define como o agente interage com outros agentes e com o ambiente, enquanto a parte de funcionalidade está relacionada com o funcionamento interno do agente. Reunindo as duas partes da arquitetura, temos a arquitetura de um agente que pode interagir com outros agentes ou com um ambiente. A parte de interação é definida a partir da estrutura do

ambiente e da forma pela qual os agentes deste ambiente interagem uns com os outros; já a parte de funcionalidade depende apenas de cada agente. Uma mesma arquitetura funcional pode ser adaptada para diversos tipos de interação e desta forma um mesmo agente básico pode ser adaptado a vários ambientes.

Neste trabalho, uma arquitetura funcional para agentes inteligentes é desenvolvida a partir de um modelo teórico, a Sociedade da Mente, em lugar de constituir uma proposta de arquitetura para resolver um problema específico, como é mais comum. Isto traz as vantagens de que a arquitetura resultante é potencialmente aplicável a uma classe de problemas em vez de um problema particular, e de que se dispõe do próprio modelo teórico como uma indicação de quais sejam as potencialidades da arquitetura e sua adequação para tratar de diferentes problemas.

A Sociedade da Mente é essencialmente uma coleção de idéias que formam um modelo coerente da mente. Como modelo da mente e da inteligência, ela apresenta as seguintes características:

- a mente é descrita como sendo composta por uma coleção de componentes desprovidos de inteligência chamados “agentes.” Trata-se, portanto, de um modelo descentralizado que evita assim o problema do “homúnculo,” que consiste em supor que existe algum elemento central que concentra a inteligência, levando à regressão infinita;
- os agentes são organizados em grupos chamados “agências,” que podem ser identificados com funções mais ou menos específicas. O modelo impõe, portanto, uma organização à mente;
- a descrição da mente em termos de agentes e agências é aplicada em diversos níveis de abstração ou de detalhe: no nível mais elementar, os agentes são componentes puramente reativos que interagem através do envio de sinais uns para os outros, sendo comparáveis aos componentes de uma rede neural; estes agentes são agrupados em agências, e estas por sua vez podem ser vistas como agentes que são agrupados formando agências mais complexas em um nível mais alto de abstração. A distinção entre agentes e agências significa essencialmente que um agente é uma unidade fechada ou encapsulada no nível de descrição considerado, ao passo que uma agência é um conjunto de agentes que interagem entre si de acordo com determinados padrões;
- o conceito que estabelece um relacionamento entre os níveis descritivos é o de “estado mental parcial,” que é definido como sendo uma descrição conjunta dos estados de alguns agentes da mente mas não necessariamente todos (daí o qualificativo “parcial”). Estados mentais parciais são mecanismos de comunicação entre agências: enquanto agentes se comunicam através de troca de mensagens, que no nível mais

elementar correspondem a sinais enviados ao longo de conexões, agências se comunicam através do estabelecimento e detecção de estados mentais parciais;

- é proposta uma série de mecanismos que atuam na detecção e estabelecimento de estados mentais parciais. É essencialmente por meio destes mecanismos que as agências da mente se comunicam de modo que o funcionamento da mente como um todo seja coordenado e coerente. Estes mecanismos são chamados de mecanismos de coordenação, e constituem em sua maioria agentes especiais, sendo por isso também chamados de agentes de coordenação;
- a inteligência emerge do funcionamento integrado e coerente dos agentes da mente, que influenciam uns aos outros, e está relacionada ao fato de que esses agentes (ou agências) operam cada um segundo estratégias e representações próprias, sendo esta multiplicidade que garante que a mente seja capaz de enfrentar de uma forma ou de outra virtualmente qualquer situação que lhe seja imposta pelo mundo real. Por isso mesmo, o tipo de inteligência que a Sociedade da Mente enfatiza é a inteligência genérica em contraste com aspectos mais especializados da inteligência, como por exemplo o raciocínio lógico.

Assim, com base nas características enumeradas, este trabalho explora o modelo da Sociedade da Mente como um fundamento para a arquitetura de um agente inteligente no que se refere à parte da funcionalidade. Chamamos este agente de “inteligente,” não pretendendo com isto sugerir que o agente possua estados mentais reais, mas sim que ele é inspirado em um modelo teórico-computacional da inteligência.

As características da Sociedade da Mente nos levaram naturalmente a conceber um agente que fosse, ele próprio, um sistema multi-agentes. Esta é provavelmente a denominação mais comum atualmente para sistemas de Inteligência Artificial Distribuída (IAD) e denota sistemas descentralizados nos quais diversos componentes, também chamados “agentes,” interagem e cooperam uns com os outros com o intuito de resolver problemas, executar tarefas etc. A Inteligência Artificial Distribuída se preocupa com os seguintes aspectos:

- quais são os agentes do sistema, se eles são homogêneos ou heterogêneos, reativos (atuam de acordo com um esquema de estímulo e resposta) ou cognitivos (suas ações são mediadas por processos internos mais ou menos complexos) e qual vai ser a funcionalidade ou o papel de cada um deles;
- como os agentes cooperam, ou seja, como o sistema como um todo deve funcionar;
- como os agentes interagem e/ou se comunicam;

- qual agente atua em qual momento (controle);
- a possibilidade de ocorrerem conflitos;
- como o sistema é organizado, isto é, quais as relações de subordinação entre os agentes, sejam elas permanentes ou temporárias, explícitas (hierarquias) ou implícitas (contratação de serviços) etc., existindo inclusive abordagens que buscam aplicar noções da teoria das organizações humanas nos sistemas de IAD, ao mesmo tempo que sistemas multi-agentes podem ser usados para modelar e simular aspectos das organizações e sociedades humanas;
- arquiteturas de sistemas de IAD, que são especificações abstratas de sistemas no que se refere aos aspectos mencionados.

Um agente baseado no modelo da Sociedade da Mente deve ser, portanto, um sistema constituído por diversos componentes autônomos. Estes componentes podem ser, eles próprios, considerados como sendo agentes, embora sua característica não seja mais uma funcionalidade sofisticada mas sim o fato de pertencerem a um sistema multi-agentes no qual os agentes devem atuar de forma coordenada. Assim, o agente baseado na Sociedade da Mente é, ele próprio, um sistema multi-agentes, e a arquitetura para agentes inteligentes baseada na Sociedade da Mente descrita neste trabalho é uma arquitetura de um sistema multi-agentes caracterizado principalmente por empregar mecanismos de coordenação inspirados naqueles do modelo da Sociedade da Mente.

É importante reafirmar que este trabalho tem a finalidade de projetar uma arquitetura baseada na Sociedade da Mente que seja genérica, ou seja, sem estar vinculada a nenhuma aplicação em particular. Desta forma, uma característica essencial da metodologia deste trabalho é o fato de que, em lugar de partir de um problema e propor uma solução para o mesmo, tomamos como ponto de partida um modelo conceitual. Escolhemos o modelo da Sociedade da Mente de Minsky porque constitui um modelo de inteligência genérica. A escolha da IAD é decorrente da escolha do modelo da Sociedade da Mente, uma vez que neste modelo a mente é formada por unidades menores.

O agente pode ser imaginado como composto por um núcleo central—a sua funcionalidade—e uma camada de interação. A arquitetura desenvolvida neste trabalho é a arquitetura do núcleo central. A camada de interação depende muito do núcleo central, constituindo, por assim dizer, uma espécie de “intérprete” que permite que o agente se comunique com outros agentes do ambiente. A camada de interação constitui portanto a interface do agente, seja com o usuário, seja com outros agentes, ou com todos estes. O núcleo central é um sistema de IAD, e o agente como um todo—ou seja, o núcleo central mais a camada de interação—pode fazer parte de um ambiente multi-agentes que também é um sistema de IAD. Entretanto, é importante notar que no contexto deste trabalho não estamos nos

preocupando com o ambiente multi-agentes, ou seja, o sistema de IAD maior no qual o agente pode estar imerso.

Em certo sentido, o problema foi o de identificar estruturas e mecanismos da Sociedade da Mente que pudessem ser utilizadas no contexto de um sistema multi-agentes, e transformá-las em estruturas e mecanismos passíveis de serem realizadas neste tipo de sistema. Assim, a abordagem que adotamos consiste em dar ênfase aos mecanismos de coordenação da Sociedade da Mente, que são propostos a partir do conceito de “estado mental parcial” como mecanismo de comunicação entre agências. Estes mecanismos de coordenação passam a ser vistos como os mecanismos de coordenação de um sistema de IAD. Esta abordagem preserva a essência do que é a inteligência no modelo da Sociedade da Mente, e que vem a ser o funcionamento integrado e coerente (coordenado) das múltiplas agências da mente, que também é a característica essencial dos sistemas de IAD.

Do ponto de vista da IAD, este trabalho constitui uma tentativa de utilizar um modelo de coordenação e comunicação entre agentes como base para os reais mecanismos de coordenação e comunicação de um sistema de IAD. Neste sentido, podemos fazer uma analogia com a arquitetura de Rede de Contratos, que se baseia em um modelo de coordenação e comunicação no qual os agentes são como indivíduos que oferecem seus serviços de execução de tarefas e/ou contratam serviços de outros agentes a fim de realizar as tarefas que lhes foram atribuídas. Neste caso, o sistema implementa um protocolo de comunicação que permite estabelecer esses contratos. No caso da arquitetura baseada na Sociedade da Mente, o modelo de coordenação e comunicação é o de estados mentais parciais coordenados através de uma série de mecanismos, que são implementados através dos objetos de um *blackboard* [6, 17, 16], no qual os estados mentais parciais são estabelecidos por meio da funcionalidade desses objetos.

Os conceitos de agentes, agências e coordenação através de estados mentais parciais e mecanismos ou agentes especiais de coordenação que detectam e evocam estados mentais parciais constituíram os fundamentos para a arquitetura do agente. Entre agentes, agências e agentes especiais de coordenação, selecionamos dois tipos de agentes para serem realizados na arquitetura: agentes que correspondem a funções de alto nível (em comparação com os agentes elementares), comparáveis a programas, procedimentos ou funções na terminologia usual de computação, e agentes de coordenação, que incluem: reconhecedores, *k-lines*, *pronomes*, *frames* e *paranomes*. Por realizarem funções de alto nível, os primeiros seriam descritos como agências em um nível mais detalhado. No entanto, eles correspondem a agentes de IAD, e portanto não são, em princípio, passíveis de serem divididos em seus agentes constituintes; por isso, são chamados “agências atômicas” ou simplesmente agentes do sistema. Os agentes de coordenação foram ultimamente transformados em objetos “ativos” que residem nas diferentes seções de um *blackboard*, o qual também contém os objetos de dados propriamente ditos que o agente manipula. Referimo-

nos àqueles como objetos de coordenação ou, às vezes, agentes de coordenação.

A transformação das duas categorias de agentes em diferentes elementos decorre principalmente do fato de que são agentes de granularidades e funcionalidades diferentes, como descreveremos no capítulo 4. Estados mentais parciais acabaram por corresponder precisamente a estados de objetos do *blackboard*, incluindo objetos de coordenação e os objetos de dados propriamente ditos, ou seja, estados parciais do *blackboard*.

Para os estados dos objetos do *blackboard*, decidimos manter os estados descritos na Sociedade da Mente. Estes estados são basicamente dois, ativo e inativo, mas pode-se empregar também toda uma gama intermediária de estados, ou ainda acrescentar novos estados, como o estado “hiperativo” que usamos no exemplo de aplicação descrito no capítulo 5. Esta decisão vem da conveniência de se utilizar um mesmo conceito de estado para *todos* os objetos do *blackboard*, que sua vez se deve aos princípios de funcionamento dos agentes de coordenação. O significado desses estados fica em aberto para ser definido por cada aplicação.

A arquitetura que desenvolvemos apresenta as seguintes características: ela define que os agentes do sistema são aqueles que convencionamos chamar de agências atômicas. São agentes de uma complexidade intermediária, isto é, não são nem simples agentes reativos, nem agentes cognitivos capazes de raciocinar, realizar diálogos e negociações sofisticados etc., mas agentes capazes de realizar algum processamento específico. Estes agentes serão provavelmente heterogêneos na grande maioria das aplicações, mesmo porque a heterogeneidade das agências é um princípio importante do modelo da Sociedade da Mente. Sugerimos também que as funcionalidades desses agentes sejam definidas preferencialmente com base em uma decomposição comportamental [3] da funcionalidade do sistema como um todo. A arquitetura reúne características do modelo de *blackboard*, onde este constitui o principal meio de comunicação entre os agentes do sistema e ao mesmo tempo a “área de trabalho” sobre a qual os agentes operam de forma interativa e cooperativa, e os mecanismos de coordenação da Sociedade da Mente, implementados como objetos reativos nesse mesmo *blackboard*. A arquitetura proposta não estabelece um esquema de controle obrigatório. Assim, a atuação dos agentes pode se dar de forma oportunística, estratégia de acordo com a qual cada agente atua quando reconhece uma oportunidade para tal; pode depender de algum estado do *blackboard*, que neste caso pode ser considerado também como o meio de controle; pode também ocorrer de acordo com uma ordem pré-definida ou como resultado de algum outro tipo de escalonamento externo aos agentes. Os agentes de coordenação devem em geral atuar de forma concorrente, havendo no entanto a necessidade do controle de concorrência a fim de evitar a ocorrência de estados ou situações inconsistentes ou desprovidos de significado. No que se refere à organização do sistema, em princípio os agentes são autônomos, ou seja, não existe nenhuma relação de subordinação explícita.

A abordagem que utilizamos no sentido de definir uma arquitetura genérica deixa muita coisa a ser definida caso se pretenda pensar em aplicações específicas, mas isto constitui antes uma flexibilidade esperada. Mostramos que o fato de a arquitetura proposta ser genérica permite pensar em aplicações bastante diversas, e que por ter sido baseada em um modelo teórico da mente e da inteligência que é ao mesmo tempo um modelo claramente computacional a torna uma alternativa adequada para uma ou mais classes de aplicações. Para ilustrar isto, descrevemos algumas possibilidades de aplicações, que são um agente de busca bibliográfica e um agente para navegação em ambientes estruturados, e sugerimos que um agente baseado na Sociedade da Mente pode ser uma abordagem adequada para o modelamento do senso comum. Entre estes, selecionamos um caso particular de agente para navegação em um ambiente estruturado para implementação. O caso escolhido foi o de um agente que conta uma história, onde a própria história constitui um conjunto de informações estruturado.

Concluimos este trabalho discutindo a abordagem utilizada e apontando as vantagens (e desvantagens) de se projetar uma arquitetura genérica para agentes inteligentes baseada em um modelo teórico.

A estrutura deste trabalho é a seguinte: no capítulo 2, descrevemos o modelo da Sociedade da Mente. Embora esta seja uma descrição geral, ela coloca ênfase nas estruturas e mecanismos de coordenação que pretendemos utilizar, em lugar de adotar uma perspectiva mais teórica ou conexionista. O capítulo 3 descreve os princípios gerais da Inteligência Artificial Distribuída. No capítulo 4, descrevemos o processo de desenvolvimento da arquitetura e discutimos brevemente algumas possibilidades de aplicações. No capítulo 5 apresentamos com mais detalhes um exemplo de aplicação que escolhemos para ser implementado com o propósito de testar a viabilidade da arquitetura desenvolvida. O capítulo 6 apresenta a conclusão deste trabalho.

Capítulo 2

A Sociedade da Mente

2.1 Introdução

Marvin Minsky, um dos fundadores da IA, desenvolveu ao longo da década de 80 uma série de idéias a respeito da inteligência, todas baseadas em um mesmo princípio: o de que a mente é organizada como uma coleção de pequenas unidades desprovidas de inteligência, e a inteligência e os demais fenômenos mentais *emergem* do funcionamento do conjunto. A estas idéias ele chamou de *Sociedade da Mente* [13]. Assim, a mente seria organizada como uma “sociedade” dessas pequenas unidades, que recebem o nome de “agentes.” Neste contexto, “sociedade” significa uma coleção de unidades que interagem, e não uma organização social, uma comunidade, uma sociedade humana. Isto porque os membros de uma sociedade humana são entidades muito complexas por si sós, e cada um dos membros da sociedade é potencialmente capaz de se comunicar com cada um dos demais. Na Sociedade da Mente, ocorre justamente o contrário: seus membros são entidades extremamente simples, e que se comunicam de uma forma bastante restrita.

Os agentes da Sociedade da Mente também não podem ser comparados aos chamados *personal assistants*, *softbots*, agentes inteligentes, agentes convincentes etc. Um agente da Sociedade da Mente é muito mais simples, muito menos complexo do que um desses agentes. No entanto, um agente inteligente pode ser construído de modo a ter uma arquitetura interna semelhante à que Minsky propõe para uma mente na abordagem da Sociedade da Mente. Neste caso, este agente inteligente é estruturado como uma coleção de agentes menores. O objetivo deste trabalho é precisamente este: apresentar uma arquitetura para agentes inteligentes baseada na Sociedade da Mente. Este capítulo e o próximo apresentam os fundamentos para esta arquitetura. Neste capítulo, falaremos do fundamento teórico, conceitual: a Sociedade da Mente.

Começaremos por descrever os “blocos básicos” da Sociedade da Mente, que são os agentes, como estes se comunicam, e como os agentes são organizados em agrupamentos

denominados “agências.” Discutiremos então como a organização da Sociedade da Mente em agentes e agências e a interação entre eles reflete a maneira com que esta abordagem tenta combinar as abordagens simbólica e conexionista. Em seguida passaremos a descrever os mecanismos da Sociedade da Mente que coordenam o funcionamento dos agentes e das agências. Depois falaremos dos processos de aprendizado e desenvolvimento na Sociedade da Mente e finalmente concluiremos mostrando como a Sociedade da Mente pode servir como fundamento para uma arquitetura para agentes inteligentes.

2.2 Agentes

As unidades mais elementares da Sociedade da Mente são semelhantes às unidades de uma rede neural. Na Sociedade da Mente, estas unidades são chamadas *agentes*. Mais adiante, mostraremos que, no contexto da Sociedade da Mente, o termo *agente* possui na verdade um significado mais amplo mas, por enquanto, vamos nos ater aos agentes mais elementares.

Um agente possui um *nível de atividade* que varia de inativo a ativo. Cada agente possui conexões com outros agentes. Através destas conexões, o agente pode ser ativado ou inibido, e pode ativar ou inibir outros agentes. O nível de atividade do agente é modificado em resposta à ativação ou inibição realizada por outros agentes, e define a intensidade com a qual o agente exerce sua própria ação de ativar ou inibir outros agentes.

As conexões entre os agentes podem ser feitas, por exemplo, utilizando um esquema semelhante ao proposto por Calvin Mooers em 1946 [13]. Neste esquema, conforme mostra a figura 2.1, diversos agentes são conectados a um “barramento” composto por diversos “fios.” Cada agente transmissor é ligado a um subconjunto destes fios. Agentes receptores recebem suas entradas de agentes que “detectam” a ativação dos fios apropriados. Evidentemente, a noção de agentes transmissores e agentes receptores é apenas local: tipicamente, agentes desempenham tanto o papel de transmissores como o papel de receptores. Deste modo, cada agente vai estar conectado a vários destes barramentos.

Na Sociedade da Mente, cada agente está conectado a alguns poucos outros agentes, em contraste com outros esquemas conexionistas caracterizados por uma rede densa de interconexões. Este é um aspecto no qual a abordagem da Sociedade da Mente difere da abordagem conexionista.

Um padrão importante de conexão entre agentes é o chamado esquema de *exclusão cruzada* (*cross-exclusion*). Neste esquema, mostrado na figura 2.2, agentes são conectados de forma a enviarem sinais *inibitórios* uns aos outros. Agentes assim conectados constituem um *grupo de exclusão cruzada*. Um grupo de exclusão cruzada tende a ter apenas um agente ativo, pois quando um dos agentes no grupo é ativado, ele passa a exercer mais intensamente sua ação inibitória sobre os demais, fazendo com que a capacidade

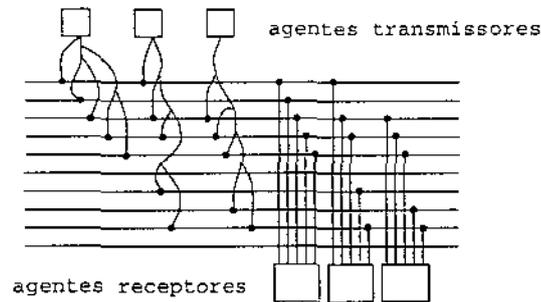


Figura 2.1: Conexão entre agentes através de um barramento.

inibitória destes decresça. Outra propriedade que um grupo de exclusão cruzada apresenta é que a existência de conflitos dentro do grupo tende a inibir o grupo inteiro. Um conflito, neste caso, é simplesmente a ativação de mais de um agente, que passam a se inibir mutuamente. Esta propriedade constitui o exemplo mais elementar do *Princípio do Não-Compromisso (Noncompromise)*. De acordo com este princípio, quanto mais persistirem conflitos internos em um grupo de agentes, mais “fraco” este se torna diante de seus próprios competidores. Finalmente, um grupo de exclusão cruzada apresenta uma terceira propriedade: na ausência de conflitos, ou seja, quando há apenas um agente ativo, o grupo tende a manter sua configuração. Esta propriedade de “estabilidade” faz com que um grupo de exclusão cruzada possa ser comparado a uma “memória” a nível de *hardware*, semelhante a um *latch*.

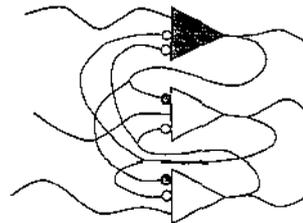


Figura 2.2: Exclusão cruzada.

2.3 Agentes e agências

Os agentes elementares da Sociedade da Mente descritos anteriormente dificilmente podem ser individualmente identificados com alguma função, ou representação, ou algo do gênero.

Neste nível mais elementar, funções, representações etc. são *distribuídas* sobre grupos de agentes. Estes grupos de agentes são chamados *agências*.

Uma agência pode ser identificada com uma função, um comportamento, um conhecimento, um conceito, uma representação etc. De um modo geral, as agências mais elementares, ou seja, aquelas que são compostas pelos agentes elementares descritos na seção anterior, só podem ser identificadas com funções, representações etc. extremamente simples. Por isso, um dos princípios mais poderosos da Sociedade da Mente é o que sugere que o agrupamento de agentes em agências seja feito em níveis sucessivos de abstração. As agências elementares são agrupadas em agências mais elaboradas, e assim por diante. Desta forma, as agências que compõem uma outra agência passam a ser vistas como os *agentes* que a compõem, conforme mostrado na figura 2.3. No contexto da Sociedade da Mente, o termo *agente* é usado para fazer referência a algo que tem alguma função, por trivial que seja, mas que deve ser visto como uma “caixa-preta”, isto é, quando a estrutura interna é irrelevante, enquanto o termo *agência* é usado quando se pretende levar em consideração a estrutura e o funcionamento internos [20].

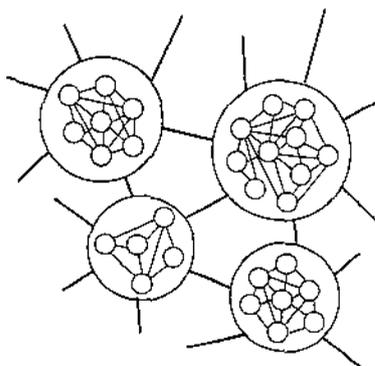


Figura 2.3: Agentes e agências.

Uma agência elementar possui uma estrutura conexionalista, mas agências em níveis mais altos de abstração se aproximam cada vez mais do processamento simbólico. Estas agências correspondem, então, a funções e representações mais familiares, ou seja, que podemos identificar mais facilmente como tais; e podem representar conceitos, procedimentos etc.

Algumas agências da mente possuem uma estrutura hierárquica na qual os agentes mais próximos do topo da hierarquia estão relacionados com metas e objetivos, enquanto os agentes no nível mais baixo são responsáveis pelas atividades sensorial e motora [12, 13]. Os agentes relacionados com as metas e objetivos de uma agência em um certo sentido “sumarizam” o que a agência faz, isto é, sua função ou comportamento. Uma agência

com uma tal estrutura hierárquica é mostrada na figura 2.4.

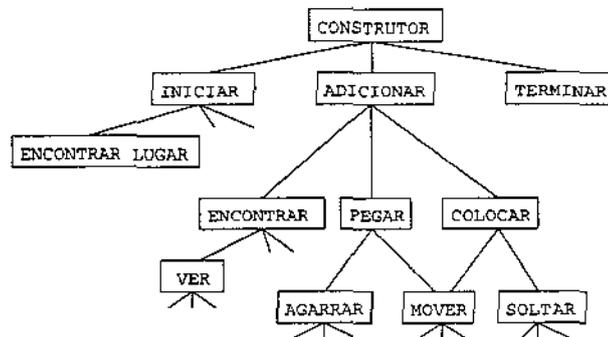


Figura 2.4: Uma agência com estrutura hierárquica.

2.4 As abordagens simbólica e conexionista na Sociedade da Mente

A Sociedade da Mente tenta combinar as abordagens simbólica e conexionista com o intuito de superar as deficiências de ambas [14, 15]. A abordagem simbólica ou cognitivista, na qual computadores executam programas que manipulam símbolos de modo a replicar os processos cognitivos de uma mente natural, é capaz de duplicar apenas processos muito restritos em domínios bastante limitados. A abordagem conexionista, cujo fundamento consiste na duplicação da estrutura neural que suporta os processos mentais, enfatizando seu caráter maciçamente paralelo, também só é capaz de duplicar certos processos mentais, em geral perceptuais. Assim, embora ambas as abordagens tenham servido de base para construir aplicações úteis, como sistemas especialistas e redes neurais para reconhecimento de padrões, ainda nenhuma delas explica ou duplica aspectos importantes do comportamento inteligente, como o senso comum, por exemplo.

Na Sociedade da Mente, o uso de representações particulares, que caracteriza a abordagem simbólica mas que também traz certa falta de flexibilidade, é substituído pelo uso de múltiplas representações e processos em diferentes partes da mente, que operam em paralelo e que podem influenciar umas às outras. O grande número de interconexões característico da abordagem conexionista clássica é substituído pela divisão da mente em agrupamentos que possuem muitas conexões internas mas poucas conexões com outros agrupamentos. Estes agrupamentos correspondem às agências elementares, que possuem uma organização conexionista, mas podem ser identificadas com funções ou representações específicas. A possibilidade de agrupar agências elementares para formar agências mais

sofisticadas faz com que em um certo nível estas agências possam ser identificadas com processamento simbólico. Deste modo, a Sociedade da Mente reúne aspectos simbólicos e conexionistas, tentando contornar os pontos fracos de ambas as abordagens.

2.5 Mecanismos de coordenação na Sociedade da Mente

Até agora, descrevemos os “blocos básicos” da Sociedade da Mente, que são os agentes e as agências. Dissemos também que a Sociedade da Mente tenta contornar a falta de flexibilidade das abordagens simbólicas clássicas permitindo que diferentes agências façam uso de diferentes representações e estratégias. Mas como todos estes agentes e agências operando em paralelo podem dar origem a um comportamento inteligente? A resposta é que estes agentes e agências interagem e influenciam uns aos outros através de mecanismos que levam a um funcionamento integrado, “coerente”, coordenado.

Nesta seção, descreveremos os mecanismos da Sociedade da Mente que permitem aos agentes e agências funcionarem de uma maneira coordenada. Quando falamos de coordenação, portanto, não estamos nos referindo a uma coordenação centralizada, como a coordenação de um projeto ou de uma escola, mas de um outro conceito de coordenação: a coordenação que permite a uma pessoa dirigir um carro, ou que permite aos músicos de uma banda tocarem juntos.

Começaremos explicando o que são *estados mentais parciais*, pois vários dos mecanismos que vamos apresentar baseiam-se neste conceito. Em seguida, descreveremos as *k-lines*, que constituem o mecanismo mais básico e que correspondem às memórias; os *polynemes*, *micronemes* e *pronomes*, que são tipos de *k-lines* com funções diferentes; os *frames*, que são basicamente mecanismos de estruturação; os reconhecedores, que são agentes que reconhecem configurações de outros agentes; os *supressores* e *censores*, que são agentes que tentam impedir que agências repitam ações que levaram ao fracasso no passado; e a noção de uma “mente administrativa” (*B-brain*) que controla tudo o que a mente “principal” faz.

2.5.1 Estados mentais parciais

Um *estado mental* (total) é definido como sendo a especificação do nível de atividade de cada um dos agentes da mente, ou seja, a indicação de quais agentes estão ativos e quais agentes estão inativos. Um *estado mental parcial* corresponde a uma especificação de *alguns* dos agentes que estão ativos, sem dizer nada sobre o nível de atividade dos demais. Estados mentais parciais são ambíguos por definição. Uma mente sempre vai estar em muitos estados parciais ao mesmo tempo, pois cada um destes estados parciais é uma possível descrição (parcial) do nível de atividade dos agentes. Um estado mental parcial

pode corresponder, por exemplo, a uma idéia, a um passo na resolução de um problema, a uma experiência real e muitas outras possibilidades.

Em geral, fala-se do estado mental parcial relativo a uma única agência, ou a algumas poucas agências, mas pode-se também falar de um estado mental parcial que abrange agentes espalhados pelos cantos mais remotos da mente. Conflitos podem surgir quando são induzidos estados mentais parciais referentes a uma mesma parte da mente, ou a partes que se sobrepõem parcialmente. O esquema de exclusão cruzada descrito anteriormente pode ser considerado como um mecanismo de resolução de conflitos deste tipo a nível de grupos de exclusão cruzada.

2.5.2 K-Lines

Uma *k-line* (*knowledge-line*) é um tipo especial de agente que é capaz de memorizar e tornar a induzir um estado mental parcial [12]. Quando um estado mental parcial deve ser memorizado, uma *k-line* nova é criada ou alocada e conectada a alguns dos agentes que estavam ativos naquele momento. Posteriormente, a ativação da *k-line* causará a ativação dos agentes aos quais ela está conectada, reproduzindo um estado mental parcial presente no instante da formação da *k-line*. Em outras palavras, a ativação de uma *k-line* induz a estes agentes níveis de atividade idênticos aos de quando ocorreu a memorização do estado mental parcial (figura 2.5).

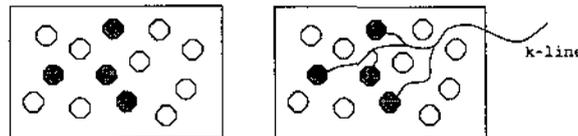


Figura 2.5: Uma *k-line*.

K-lines correspondem, portanto, a memórias, mas não a memórias “declarativas” (relativas a fatos) ou “proposicionais,” e sim ao que Minsky chamou de *memórias disposicionais*, porque induzem em uma ou mais agências uma *disposição*, uma *tendência* a atuar de modo semelhante a um instante passado.

A formação de uma *k-line* é causada por uma outra agência, que “reconhece” a presença de um estado mental parcial que deve ser memorizado. Em geral, uma *k-line* memoriza um estado parcial relativo a uma única agência ou a poucas agências, próximas umas das outras. A *k-line* induz uma disposição nesta(s) agência(s) através do reestabelecimento do nível de atividade de um *conjunto relevante* de agentes dentro da(s) agência(s). Os agentes que fazem parte deste conjunto dependem do estímulo que causou a formação

da *k-line*. Por exemplo, no caso de uma agência com uma estrutura hierárquica semelhante à mostrada na figura 2.4, se uma *k-line* se conectasse aos agentes do nível sensorial, sua ativação causaria uma “alucinação” do evento original. Se a *k-line* se conectasse aos agentes dos níveis mais altos, metas e planos apropriados à situação original mas não à atual poderiam se impor, e possivelmente causar conflitos com objetivos e planos atuais. Por isso, não convém que a *k-line* se conecte a agentes em níveis muito altos ou muito baixos. Uma memória deve causar uma disposição, mas mantendo a sensibilidade da agência à situação e aos dados atuais. Isto é feito estabelecendo uma *faixa de níveis (level-band)* intermediária na agência, dentro da qual estão localizadas as conexões *mais intensas* da *k-line*. A posição desta faixa está relacionada com o estímulo que causou a formação da *k-line*.

O chamado *Princípio da Faixa de Níveis* descrito anteriormente não estabelece uma fronteira rígida entre esta faixa e os demais níveis; a faixa intermediária corresponde apenas à região na qual as conexões da *k-line* são mais intensas. A *k-line* também se conecta a agentes em regiões acima e abaixo desta faixa, que constituem a *aba superior (upper fringe)* e a *aba inferior (lower fringe)*, respectivamente, mas estas conexões tendem a ser tanto mais fracas quanto mais distantes se encontram da faixa principal (figura 2.6).

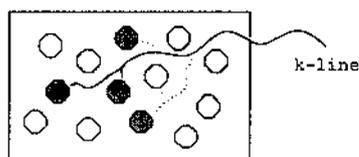


Figura 2.6: Faixas de níveis.

Como as conexões na aba superior e na aba inferior são fracas, elas não resistem a conflitos, sendo facilmente “encobertas” pela situação presente. Deste modo, os agentes da aba inferior, que representam detalhes da lembrança, atuam como “suposições *default*.” Já os agentes da aba superior estão relacionados com aspectos mais subjetivos, como metas, planos e intenções.

Outro princípio importante relativo às *k-lines* é o chamado *Princípio da K-Recursão*. De acordo com este princípio, *k-lines* podem se conectar também a outras *k-lines*. As memórias formadas conectando *k-lines* a outras *k-lines* têm um efeito um pouco diferente de uma conexão direta. Elas podem servir tanto para armazenar algo que foi *reconhecido* em vez de *experienciado*, como para *estruturar* outras memórias. A figura 2.7 mostra como um evento pode ser memorizado através de conexões com as *k-lines* que representam os aspectos do evento. O Princípio da Faixa de Níveis se aplica igualmente às *k-lines* que se conectam a outras *k-lines*.

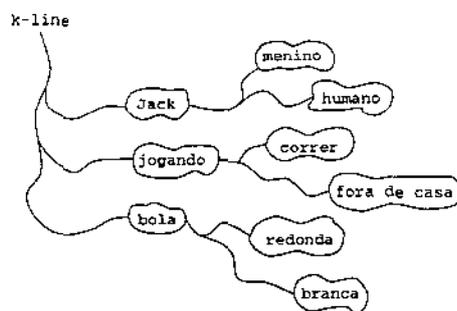


Figura 2.7: A memória de um evento: uma *k-line* conectada a outras *k-lines*.

Como foi dito anteriormente, a maior parte das *k-lines* se conecta a agentes em uma ou em poucas agências. Logo, uma *k-line* estabelece conexões predominantemente locais, e as diversas *k-lines* se acumulam em camadas em torno de coleções de agências. As camadas de *k-lines* que se acumulam em torno das agências apresentam a seguinte estrutura, esquematizada na figura 2.8: toda agência substancial possui unidades de “micromemórias,” que são *k-lines* capazes de memorizar ou reinduzir rapidamente o estado de vários dos agentes da agência; cada agência possui também unidades de “memória de curto prazo” que podem, por sua vez, memorizar ou restaurar o estado das micromemórias; e há ainda a camada de “memórias de longo prazo.” As memórias de curto prazo podem ser “transferidas” de alguma forma para as memórias de longo prazo; caso contrário, se forem reutilizadas, seu conteúdo é perdido.

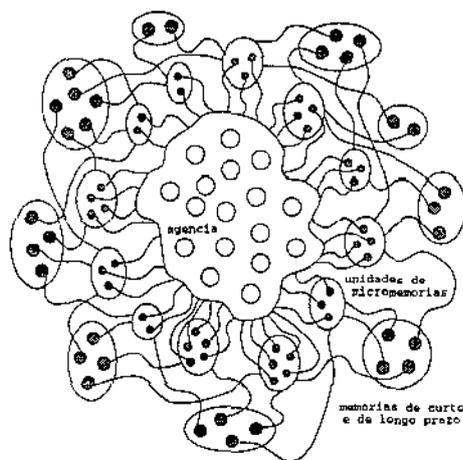


Figura 2.8: Camadas de *k-lines* em torno de uma agência.

2.5.3 Polynemes

Agentes cuja denominação toma o sufixo *-neme*, como *polynemes* e *micronemes* (que serão descritos na próxima seção), são agentes cuja principal função é causar a reprodução de algum estado mental parcial. Em particular, um *polyneme*, como o prefixo *poly-* indica, é um agente capaz de reinduzir um estado mental parcial em *diversas* agências.

Um *polyneme* é um tipo de *k-line*. Mais precisamente, um *polyneme* é uma *k-line* permanente que mantém memórias de longo prazo. Um *polyneme* atinge diversas agências. Assim, é como se o *polyneme* enviasse uma “mensagem” para cada uma destas agências. A mensagem é a mesma para todas as agências, mas cada uma delas “responde” de uma maneira diferente. A maneira de responder a um *polyneme* deve ter sido anteriormente aprendida. Portanto, a mensagem que o *polyneme* envia na verdade ativa uma memória de longo prazo—isto é, uma *k-line*—que armazenou a maneira de responder à mensagem desse *polyneme* e que corresponde a reinduzir um certo estado parcial na agência.

O processo de formação de um *polyneme* envolve conectá-lo a uma *k-line* para cada agência que ele atinge, e fazer com que cada uma destas memorize o estado parcial adequado. Este processo pode ocorrer da seguinte forma: uma agência detecta a necessidade de formar um *polyneme*; esta agência causa a atribuição de uma unidade de memória (*k-line*) em cada agência envolvida; finalmente, a mesma agência causa a criação ou alocação de um *polyneme* e o conecta a cada uma das unidades de memória atribuídas no passo anterior.

Um *polyneme* pode ser usado para armazenar um conceito, ou a noção de um objeto. Em geral, um objeto possui propriedades. Uma maçã, por exemplo, tem um tamanho mais ou menos determinado, uma certa textura, um certo odor, um formato redondo, e pode ser vermelha ou verde. A figura 2.9 mostra como um *polyneme* pode representar uma maçã.

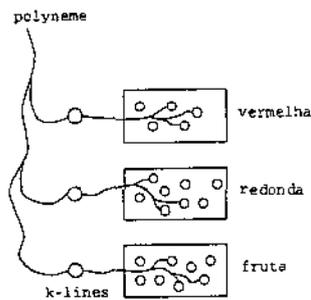


Figura 2.9: Um *polyneme* representando uma maçã.

Polynemes participam do processo chamado *reminding* ou “rememoração” através

do qual palavras ou impressões parciais evocam impressões mais completas e vívidas. Na figura 2.11, *reconhecedores* são agentes que ativam outro(s) agente(s) a partir da ponderação de um conjunto de entradas, onde os pesos podem ser positivos ou negativos. Desta forma, como o próprio nome indica, um reconhecedor reconhece uma configuração de outros agentes, sendo que a atribuição de pesos às entradas tem o efeito de tornar este reconhecimento flexível, podendo-se dizer que um agente deste tipo “pondera evidências” (figura 2.10). Agentes reconhecedores podem receber suas entradas de agentes sensores, que recebem impressões sensoriais, ou de agentes memorizadores, que induzem estados mentais parciais.

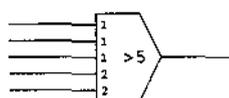


Figura 2.10: Um agente reconhecedor.

Os agentes reconhecedores, por sua vez, ativam *polynemes*, que colocam agências em estados mentais parciais que evocam uma idéia, um objeto etc. Estes estados mentais parciais afetam os reconhecedores, e assim completa-se uma configuração em *loop* (laço). Esta configuração permite que impressões parciais acabem por evocar impressões mais completas, denominadas *simulus*.

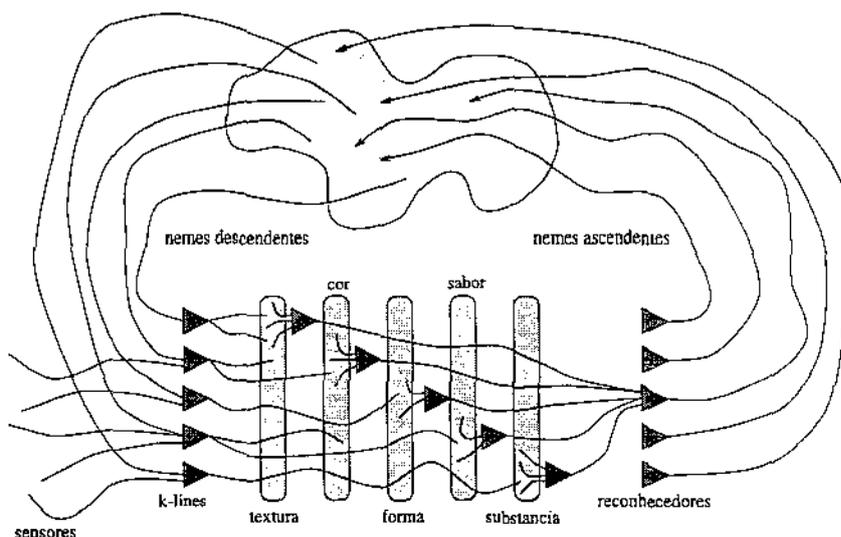


Figura 2.11: O loop de reminding ou rememoração.

Nesta configuração em *loop*, os *polynemes* ativados pelos reconhecedores recebem o nome de *nemes ascendentes*, e as *k-lines* ativadas pelos *polynemes*, que fecham o *loop*, recebem o nome de *nemes descendentes*.

2.5.4 Micronemes

Micronemes, como o sufixo *-neme* indica, também reinduzem um estado mental parcial anterior. Mas enquanto *polynemes* atingem algumas agências e podem evocar fortemente uma idéia completa, *micronemes* atingem um número muito grande de agências e não chegam a evocar nenhuma impressão. *Micronemes* exercem efeitos de ativação e de inibição muito fracos, insuficientes para evocar impressões, mas que contribuem para estabelecer tendências, facilitando ou dificultando a ativação de outros agentes. Isto pode servir para, por exemplo, desfazer ambigüidades, pois os *micronemes* podem “empurrar” a interpretação para um lado e não para o outro.

2.5.5 Pronomes e paranomes

O sufixo *-nome* designa agentes cujo efeito consiste em afetar outras agências de uma mesma maneira—em contraste com *polynemes*, que afetam diversas agências mas cada uma de uma maneira particular. Agentes deste tipo recebem o nome genérico de *isonomes*, pois *iso-* significa “o mesmo”.

Um *pronome*—que não deve ser confundido com a categoria gramatical “pronome”—é um tipo de *k-line* temporária, isto é, que a cada instante pode estar conectada a agentes diferentes. Um *pronome* causa a ativação ou atribuição de uma unidade de memória de curto prazo em cada agência que atinge.

Pronomes podem ser usados para, por exemplo, estabelecer focos de atenção, “apontando” para coisas que ocupam a atenção. Estas coisas podem ser quaisquer: podem ser idéias, ações, imagens mentais e outras, representadas por qualquer um dos mecanismos que estamos descrevendo. Um outro uso muito importante dos *pronomes* é para associar coisas a papéis (*roles*). Neste caso, *pronomes* são usados juntamente com *frames*; por isso, este tipo de uso é descrito na próxima seção.

Os mecanismos descritos até agora—*k-lines*, *polynemes*, *micronemes*, *pronomes* e *isonomes*—podem ser considerados também como mecanismos de comunicação entre agências similares à passagem de parâmetros. Estados mentais parciais provocados pela atuação destes agentes podem ser diretamente usados ou podem influenciar o que se passa em outras agências.

Tem-se também *paranomes*, que são *pronomes* que estabelecem uma “ligação” entre diversos “domínios” mentais. Isto porque virtualmente qualquer idéia ou impressão vai

evocar memórias que cobrem uma diversidade de domínios. A ativação simultânea de estados mentais em diferentes domínios pode levar a estabelecer correlações, que constituem, em parte, a origem das *analogias* e das *metáforas*.

2.5.6 Frames

Até agora, todos os mecanismos de coordenação que descrevemos são na verdade tipos de *k-lines*. *Frames* são mecanismos completamente diferentes. Um *frame* é uma coleção de agentes que realizam uma operação de *conjunção* (AND) sobre duas entradas. Uma destas entradas é proveniente de um *pronome* e, portanto, é ativada toda vez que o *pronome* for ativado. A outra entrada é ativada sempre que o próprio *frame* for ativado (figura 2.12).

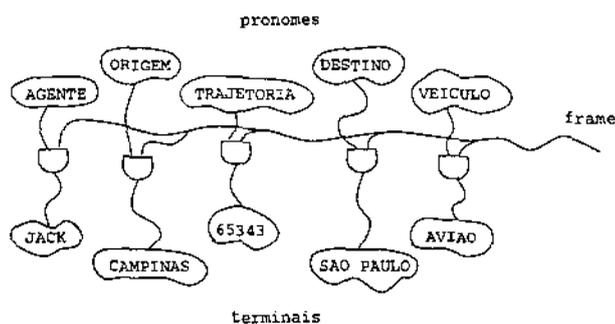


Figura 2.12: Um *frame*.

Frames são os mecanismos da Sociedade da Mente responsáveis pela *estruturação* de impressões, ações, memórias etc. (experiências passadas). Um *frame* impõe uma estrutura ao conjunto dos objetos apontados pelos *pronomes*. *Frames* equivalem às conhecidas estruturas de representação de conhecimento também chamadas *frames* [11, 1]. Os agentes-AND mostrados na figura 2.12 são chamados *terminais*, e correspondem aos *slots* dos *frames* tradicionais. Como estes, os *frames* da Sociedade da Mente podem ser vistos como “fôrmas” com espaços a serem preenchidos. Os *pronomes* ligados aos terminais do *frame* correspondem aos “valores” que preenchem os *slots*.

Os terminais de um *frame* podem ser pontos de conexão não somente para *pronomes*, mas também para *k-lines*, *polynemes*, *scripts* ou mesmo outros *frames*, dependendo do que o terminal representa. Os terminais podem possuir ligações *default*, que correspondem a memórias de longo prazo de experiências passadas, em geral da experiência que originou o *frame*; as ligações *default* têm a característica de serem muito fracas, de modo que as evidências da experiência presente facilmente predominam.

A formação de um *frame* é causada por alguma outra agência, que detecta possivelmente algum “padrão” que sugere uma estruturação. Esta agência “sabe” o que pretende

colocar no *frame*, por isso ela é capaz de estabelecer as ligações *default*.

A associação dos *pronomes* ligados aos terminais com coisas adequadas também tem que ser feita por outra agência, que “administra” o *frame*, por assim dizer. O *frame* é ativado por agentes reconhedores que detectam a ativação dos seus terminais.

Um *frame* pode impor virtualmente qualquer estrutura sobre conteúdos mentais. Portanto, existe uma infinidade de tipos ou formatos de *frames*. Entre estes, porém, podemos destacar alguns que são mais importantes porque representam estruturas muito usadas, e que são descritos a seguir.

Trans-frames

Trans-frames constituem um tipo particular de *frames* que representam transições, transferências, mudanças de estado, diferenças (“antes → depois”) e toda uma série de noções similares. A estrutura que um *trans-frame* estabelece corresponde a um encadeamento entre uma origem e um destino. O nome *trans-frames* foi inspirado no termo “*trans*” das Dependências Conceituais de Roger Schank.

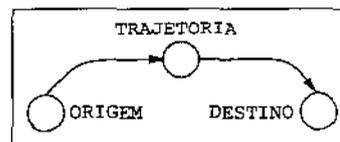


Figura 2.13: Um *trans-frame*.

Os terminais, ou *slots*, de um *trans-frame* correspondem aos papéis de “origem” ou “antes,” “destino” ou “depois,” e “trajetória” ou “diferença.” *Pronomes* apontam para coisas que preenchem estes papéis. Por exemplo, a “origem” pode ser uma localização física, o “destino” pode ser outra localização diferente, e a “trajetória” pode ser uma seqüência de ações representada por outro *frame*.

Uma característica importante dos *trans-frames* é que eles podem ser encadeados, de modo que, por exemplo, a localização física que faz o papel de “destino” de um *trans-frame* pode fazer o papel da “origem” de outro, e assim por diante. O encadeamento de *trans-frames* constitui um tipo de *raciocínio*, de acordo com o qual, por exemplo, os dois fragmentos de conhecimento “João deu o livro para Maria” e “Maria deu o livro para Paulo” são encadeados resultando na conclusão de que o livro que era originalmente de João acabou ficando com Paulo.

Frames de imagens

Outro tipo importante de *frame* são os chamados *frames de imagens* (*picture-frames*). *Frames* de imagens são utilizados para o processamento de informação espacial, e seus terminais correspondem a regiões do espaço físico.

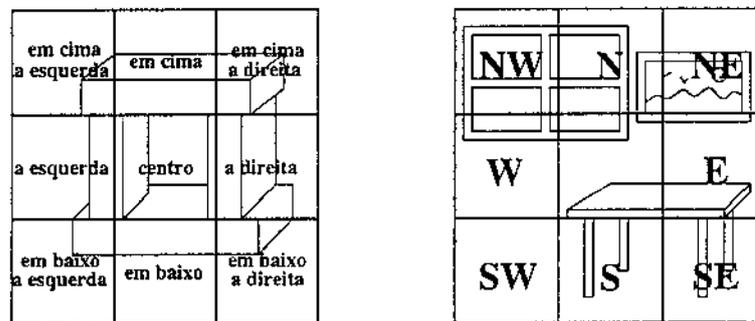


Figura 2.14: *Frames* de imagens.

Os terminais de um *frame* de imagem são conectados a agentes semelhantes a *pronomes*, denominados *direction-nemes*. Os *direction-nemes* podem ser considerados “memórias direcionais” ou “espaciais”. Os *direction-nemes*, por sua vez, são controlados pelos chamados *interaction-square arrays*, que são agentes que tratam certos tipos de interações estruturadas através de estruturas em forma de matriz (figura 2.15). Um exemplo de interação estruturada é a que ocorre entre movimentos e coisas dispostas no espaço. Neste caso, *interaction-square arrays* tridimensionais refletem a estrutura do espaço. A figura 2.15 mostra um exemplo bidimensional.

Frames de linguagem

Frames de linguagem (*language-frames*) são *frames* envolvidos com a compreensão e o uso da linguagem. Existem vários tipos de *frames* de linguagem. *Frames de sentenças* (*sentence-frames*) são *frames* que representam sentenças. *Frames* de sentenças contém outros *frames*, como *frames de substantivos* e *frames de verbos*. Quando uma sentença é lida ou ouvida, ela é acomodada em diversos *frames* ao mesmo tempo. Estes *frames* associam uma estrutura gramatical (sintática) à sentença, e também indicam o seu conteúdo semântico. Por isso, uma sentença não é meramente uma cadeia de símbolos armazenados na mente; uma sentença evoca idéias, conceitos, impressões, imagens mentais de objetos, experiências passadas. Na Sociedade da Mente, as palavras não possuem significado por si sós; elas não “denotam,” “representam” ou “designam,” mas causam a ativação de estados mentais.

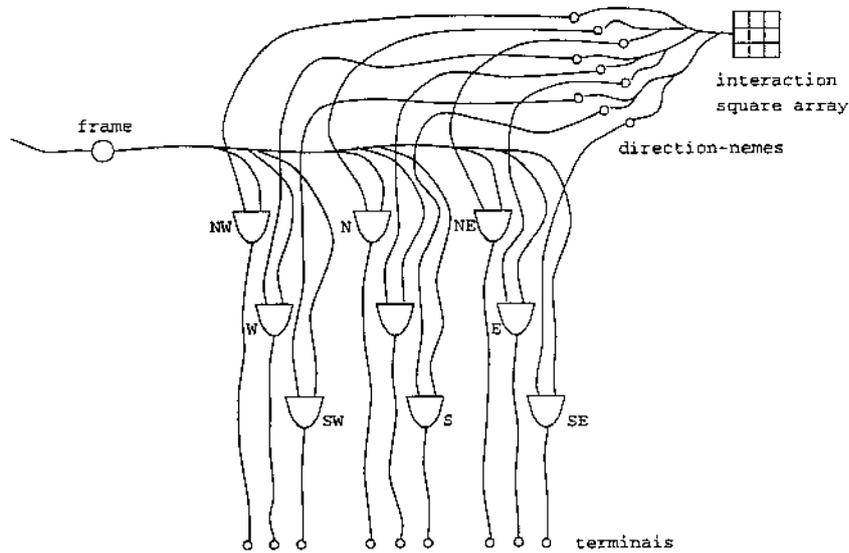


Figura 2.15: *Interaction-square array e direction-nemes.*

Scripts

Scripts são *frames* que estruturam ações, isto é, organizam ações em uma seqüência (*script*). Um exemplo de *script* é mostrado a seguir:

```

DESENHAR CASA
  desenhar paredes
  desenhar telhado
  desenhar janelas
  desenhar porta
  
```

Um *script* possui muitos de seus terminais conectados a *k-lines* que causam com que certas agências realizem ações. Evidentemente, um *script* pode ser muito complexo, envolvendo *pronomes*, *polynemes* etc.

Frame-arrays

Frame-arrays são coleções de *frames* que utilizam os mesmos terminais. *Frame-arrays* podem ser usados, por exemplo, para possibilitar o reconhecimento de objetos vistos de diferentes posições. Cada um dos *frames* do *frame-array* é associado com uma direção no espaço através de *direction-nemes* controlados por *interaction-square arrays*. Assim, à medida que os olhos da pessoa se movem e vêem o objeto de uma maneira diferente, os *frames* vão sendo substituídos, causando a chamada “ilusão de imanência,” que faz

com que pensemos que estamos vendo o mesmo objeto apesar de as impressões sensoriais serem diferentes.

2.5.7 Reconhecedores

Até agora, dissemos que agentes reconhecedores “ponderam evidências” para detectar a presença de alguma impressão mental, seja sensorial ou interna. No entanto, o reconhecimento de alguma coisa pode ser muito complexo. Para reconhecer um objeto, por exemplo, não basta reconhecer suas partes: em geral, é necessário que existam determinadas relações entre estas partes. Assim, os agentes reconhecedores também podem ser, na verdade, agências constituídas por agentes que reconhecem as partes e as relações entre elas, sendo que estes agentes podem ser também agências, e assim por diante.

Existem agentes reconhecedores que ficam à espera de determinadas configurações e então se “auto-ativam” para intervir realizando alguma ação. Estes agentes são chamados *demons*.

2.5.8 Supressores e censores

Supressores e *censores* são agentes cuja atividade é essencialmente inibitória. Ambos são usados para reconhecer e prevenir cursos de pensamento inefetivos ou deletérios. Os supressores atuam reconhecendo um “estado” que precedeu ações prejudiciais no passado, e impedindo as agências envolvidas de realizarem estas ações. Os censores reconhecem estados que *precedem* o estado a partir do qual as ações prejudiciais foram realizadas, e intercedem antes mesmo que este estado seja atingido.

Assim como os agentes reconhecedores, os agentes supressores e censores podem corresponder a agências bastante sofisticadas, e “estado” deve ser entendido em um sentido bem amplo, como sendo alguma “indicação” de que existe a necessidade de uma ação de supressão ou censura por parte destes agentes.

O uso de supressores apresenta a desvantagem de poder exigir retrocesso (*backtracking*) a um estado anterior, bloqueando temporariamente o fluxo do pensamento. Em lugar de bloquear o pensamento, censores *defletem* o pensamento para uma direção aceitável. No entanto, a desvantagem do uso de censores é que eles sequer permitem *conceber* a possibilidade de ações alternativas às ações proibidas, uma vez que não permitem atingir o estado onde elas ocorreriam.

2.5.9 Agências administrativas

Ao descrever os mecanismos de coordenação da Sociedade da Mente, várias vezes foram mencionadas agências que determinam quando um estado mental parcial deve ser memo-

rizado, quando um *polyneme* deve ser formado, quando e a que um *pronome* deve ser associado, e assim por diante. Estas agências podem ser consideradas como “agências administrativas,” porque administram, ou seja, gerenciam outros agentes e agências, não possuindo conexões diretas com as atividades sensorial e motora. Na Sociedade da Mente, o conjunto destas agências administrativas é chamado *B-brain*, em analogia com o *A-brain*, que seria composto pelo conjunto das agências que interagem com o mundo exterior. Poderia ser possível até mesmo distinguir diversos níveis de gerenciamento, de modo que um *C-brain* gerenciasse a atividade do *B-brain* e assim por diante.

As agências administrativas podem, por exemplo, detectar padrões repetitivos de atividade e reconhecê-los como uma evidência de que as agências envolvidas estão em um impasse e intervir. Outro tipo de possibilidade de atuação das agências administrativas está relacionado a atingir metas e realizar intenções. Para isto, agências administrativas podem utilizar as chamadas *difference engines*, que são agências capazes de manter descrições de metas e situações e identificar as suas diferenças.

Em geral, uma agência deste tipo vai identificar diferenças entre uma situação presente e uma situação que se pretende atingir. Se as configurações correspondentes a cada uma das situações tivessem que ser representadas simultaneamente a fim de serem comparadas, poderia ser necessário um número muito grande de agências “extras” só para este fim. Por isso, uma possibilidade é adotar uma estratégia denominada *time blinking*, na qual a *difference engine* compara os estados mentais parciais ativados sucessivamente e rapidamente por duas *k-lines*, uma representando a situação atual e a outra, a situação desejada.

2.6 Aprendizado e desenvolvimento na Sociedade da Mente

O aspecto de aprendizado e desenvolvimento é um aspecto-chave em qualquer teoria da inteligência. Na Sociedade da Mente, distingue-se diversos tipos de aprendizado: aprender um conceito não é o mesmo que aprender como fazer alguma coisa, ou como *não* fazer alguma coisa.

Um tipo de aprendizado e ao mesmo tempo de desenvolvimento ocorre dentro das próprias agências. As agências se desenvolvem através do acréscimo de novos agentes que *gerenciam* grupos de agentes já existentes. Este tipo de aprendizado corresponde ao chamado *Princípio de Papert*, segundo o qual o desenvolvimento mental consiste não apenas em adquirir novas habilidades, mas também em aprender novas maneiras de administrar as que já se possui.

Já o aprendizado de conceitos ocorre através de dois processos principais, denominados

acumulação e *uniframing* (“uniestruturação”). O processo de acumulação se dá quando instâncias do conceito se acumulam. Estas instâncias não precisam ser necessariamente “semelhantes” ou “compatíveis” em todos os aspectos, desde que representem o conceito considerado. Na verdade, a utilidade da estratégia de acumulação está precisamente em reunir instâncias nas quais não é possível identificar propriedades comuns.

O processo de *uniframing* ou “uniestruturação” consiste em reunir diversas instâncias em uma única descrição a partir de suas características em comum, que podem ser propriedades, funções etc. Tais descrições recebem a denominação de *uniframes*, que são *frames* que estabelecem uma “estrutura” comum a um conjunto de instâncias. As características comuns entre as instâncias, ou a ausência delas, podem ser identificadas como sendo *obrigatórias* (*enforcement*), *proibidas* (*prevention*) ou *indiferentes* (*tolerance*). A identificação de todas estas características é feita por agências administrativas, e depende muito de para que propósito o conhecimento assim adquirido vai ser usado. Normalmente, acumulações e *uniframes* são usados em conjunto, com os *uniframes* correspondendo à “regra” e as acumulações às “exceções à regra,” por assim dizer. Isto porque, de acordo com o chamado *Princípio da Exceção*, não vale a pena modificar um conceito que quase sempre se aplica apenas por causa de algumas poucas exceções: é melhor manter o conceito (*uniframe*) e complementá-lo com uma coleção (acumulação) de exceções.

Outro mecanismo de aprendizado é o de *reformulação*, e consiste em modificar um conceito adotando uma descrição diferente. Um exemplo de reformulação é o que ocorre quando uma descrição baseada em uma decomposição do todo em certas partes é substituída por uma descrição que decompõe o todo em outras partes.

2.7 A Sociedade da Mente como fundamento para uma arquitetura para agentes inteligentes

Como dissemos no início do capítulo, é possível conceber um agente inteligente cuja arquitetura interna seja semelhante à que Marvin Minsky chama de Sociedade da Mente. Mas a Sociedade da Mente é uma arquitetura conceitual, uma arquitetura teórica. Para efetivamente projetar e implementar um agente deste tipo, seria conveniente partir de uma arquitetura computacional. Um agente deste tipo seria na verdade um sistema composto por agentes menores; logo, esta seria uma arquitetura de IAD e, conforme vimos no capítulo anterior, teria que definir essencialmente como seriam estes agentes menores e como estes se comunicariam uns com os outros. No caso de uma arquitetura baseada na Sociedade da Mente, a atividade de definir como são os agentes se divide em duas partes:

- definir como são os agentes elementares; e

- definir como são os agentes que correspondem a mecanismos de coordenação, ou seja, *k-lines*, *polynemes* etc.

Nos próximos capítulos, descreveremos uma arquitetura computacional baseada na Sociedade da Mente. Uma característica importante desta arquitetura é que ela não supõe que os agentes elementares sejam de natureza conexionista. Na seção 2.3, dissemos que um conjunto de agentes pode ser agrupado em uma agência, e que esta agência pode então ser considerada como um agente que faz parte de uma agência de nível mais alto e, portanto, tanto agentes como agências podem ser vistos a partir de diversos níveis de abstração, sendo que a característica distintiva é que, no nível de abstração adotado, a estrutura interna de uma agência é relevante e a estrutura interna de um agente não é. Em outras palavras, pode-se adotar descrições em diversos níveis de abstração. Assim, a partir de um nível de abstração suficientemente alto, um agente pode ser comparado a um procedimento ou processo computacional. Na arquitetura que vamos descrever, estes serão os agentes elementares.

Como já foi dito no primeiro capítulo, acreditamos que uma arquitetura deste tipo representa uma etapa importante no processo de facilitar o projeto e a implementação de tais agentes; e ainda que uma arquitetura baseada na Sociedade da Mente em particular parece ser particularmente adequada para agentes dedicados a tratar de problemas relacionados com o conhecimento e raciocínio de senso comum em um ambiente de agentes.

2.8 Conclusão

Neste capítulo, apresentamos o fundamento conceitual para a arquitetura para agentes inteligentes que será descrita nos próximos capítulos: a abordagem da Sociedade da Mente, proposta por Marvin Minsky [13]. Esta abordagem modela a mente como uma “sociedade” de “agentes,” que são entidades muito simples, de natureza conexionista. Descrevemos como são estes agentes, como se comunicam, e como são organizados em agrupamentos denominados agências. Em seguida, passamos a descrever os mecanismos de coordenação da Sociedade da Mente, que permitem que todos estes agentes e agências funcionem de forma coordenada, integrada. Começamos por definir o que é um estado mental parcial, e descrevemos o mecanismo básico de memorização, que são as *k-lines*, cujo funcionamento consiste em memorizar e reinduzir um estado mental parcial. Então descrevemos diversos outros mecanismos que na verdade são tipos particulares de *k-lines*: os *polynemes*, que são *k-lines* permanentes, associadas a memórias de longo prazo e que representam conceitos, objetos, impressões mentais; os *micronemes*, que são *k-lines* caracterizadas por não evocar idéias ou impressões mas sim estabelecer contextos e tendências; e os *pronomes*, que são *k-lines* temporárias que conectam memórias a outros mecanismos. Depois descrevemos

um outro tipo de mecanismo, que são os *frames* e que servem para impor estruturas ao conteúdo da mente. Discutimos brevemente alguns dos tipos mais comuns e mais importantes de *frames*: os *trans-frames*, que representam transições e transferências de diversos tipos; os *frames de imagens*, usados para representação de conhecimento espacial; os *frames de linguagem*, que são *frames* de vários tipos envolvidos com a compreensão e o uso da linguagem (natural); e os *scripts*, que são *frames* que estruturam conjuntos de ações. Falamos ainda dos *frame-arrays*, que são coleções de *frames* que utilizam os mesmos terminais. Em seguida, descrevemos outros tipos de agentes envolvidos na coordenação dos agentes e agências em geral: agentes reconhecedores, que reagem a configurações dos outros agentes, agentes supressores e agentes censores, que atuam inibindo outros agentes. Discutimos a noção de uma “mente administrativa” composta de agências que administram ou gerenciam o funcionamento das agências “comuns.” Descrevemos brevemente alguns processos de aprendizado e desenvolvimento na Sociedade da Mente: o acréscimo de agentes administrativos dentro das agências, e os processos relacionados de *uniframing* ou uniestruturação, acumulação e reformulação. Finalmente, discutimos como a Sociedade da Mente pode ser usada como fundamento a arquitetura computacional para agentes inteligentes que será descrita no capítulo 4. No próximo capítulo, falaremos da Inteligência Artificial Distribuída, que trata essencialmente do projeto e implementação computacional de sistemas de IA organizados como sistemas distribuídos de agentes.

Capítulo 3

Inteligência Artificial Distribuída

3.1 Introdução

Inteligência Artificial Distribuída (IAD) é a parte da Inteligência Artificial que se preocupa com o estudo e o desenvolvimento de sistemas compostos por coleções descentralizadas de componentes que interagem entre si. Os componentes de um sistema de IAD são chamados *agentes* e podem ser desde módulos altamente especializados, como um agente que analisa sinais sonoros para extrair fonemas, até agentes comparáveis a indivíduos, como um “agente professor” de um ambiente educacional, passando por toda a gama intermediária, onde encontramos agentes cujas capacidades ou especialidades podem ser mais ou menos restritas assim como suas capacidades de interação. A característica fundamental é que os agentes sejam unidades ou entidades independentes, embora possam precisar da *colaboração* uns dos outros.

Os primeiros sistemas de IAD que surgiram foram chamados sistemas de resolução distribuída de problemas (RDP ou DPS, *Distributed Problem Solving*), e seus agentes eram comumente denominados Ks (*knowledge sources*: fontes de conhecimento) ou nós. Atualmente, sistemas de IAD recebem o nome de sistemas multi-agentes, embora alguns autores prefiram reservá-la para sistemas nos quais os próprios agentes são capazes de se organizar diante da tarefa com as quais são confrontados [26].

Sistemas de IAD podem ser projetados ou empregados por diversos motivos. Um sistema deste tipo pode ser utilizado na resolução de problemas ou tratamento de tarefas naturalmente distribuídas como, por exemplo, a integração dos dados de sensores geograficamente distribuídos. Em um sistema deste tipo, agentes associados a sensores em diferentes localizações geográficas se comunicam com agentes que integram os dados referentes a uma certa área, que por sua vez se comunicam com outros agentes que sintetizam as informações de uma área maior e assim por diante, até obter informações globais coerentes e significativas [4]. De forma análoga, podemos ter sistemas de controle das partes

de um robô ou outro mecanismo complexo, sistemas de controle industrial, sistemas de alarmes etc. Outra possibilidade está relacionada à utilização de estratégias distribuídas de resolução de problemas. O exemplo clássico é o modelo de *blackboard* para resolução de problemas, que inspirou a estrutura de *blackboard* usada em sistemas de IAD dos mais variados tipos. Este foi o modelo utilizado naquele que pode ser considerado o primeiro sistema de IAD, o sistema HEARSAY-II para reconhecimento de sentenças faladas. Neste sistema, cada um dos agentes analisa a informação em um certo nível de abstração: um agente analisa sinais sonoros para extrair fonemas, outro agente agrupa fonemas em palavras e assim por diante. Outro exemplo de uso de uma estratégia distribuída consiste em aplicar diversas heurísticas a fim de obter uma solução satisfatória para um problema onde, como no modelo de *blackboard*, os agentes atuam oportunisticamente, cada um aplicando a sua heurística no momento que julgar adequado. Uma terceira possibilidade é a integração de estratégias complementares para resolver um problema. Neste caso, cada agente contribui com uma abordagem própria, e espera-se obter uma solução combinando as diferentes abordagens. Cada um dos agentes pode ser, por exemplo, um sistema especialista ou, de modo mais geral, um sistema baseado em conhecimento [18]. Finalmente, é possível explorar o aspecto sob o qual um sistema multi-agentes constitui uma sociedade de agentes, sendo que a analogia é válida tanto para sociedades humanas como para sociedades animais. Assim, por um lado, pode-se tentar obter um determinado comportamento do sistema como um todo a partir de um grande número de agentes extremamente simples, em geral homogêneos, que muitas vezes recebem o nome de agentes “reativos” em contraste com agentes “cognitivos” [26]. Exemplos incluem coleções de robôs minúsculos que coletam sujeira à medida que se deslocam a esmo, onde o resultado obtido pelo sistema considerado como um todo é limpar um local [3], e sistemas que simulam sociedades de abelhas, formigas etc. Nestes casos, podemos dizer que o comportamento do sistema como um todo é um comportamento emergente. Por outro lado, pode-se modelar sociedades nas quais os agentes são capazes de conversar, dialogar, negociar, buscar um consenso etc. Neste tipo de sistema, os agentes possuem habilidades e capacidades significativas que em grande parte das vezes diferem ou complementam as de outros agentes. Exemplos incluem sistemas de planejamento multi-agentes em que os agentes constroem um plano de atuação comum; simulação de organizações, estruturas de mercado, grupos, times etc.; modelamento de cooperação e raciocínio social [27]; ambientes educacionais que permitem ao usuário interagir com agentes que lhe fornecem informações, solicitam explicações etc.; aplicações de entretenimento, como jogos, realidades virtuais habitadas por agentes, e outros.

Esta enumeração dos propósitos, motivações e áreas de aplicação da IAD mostra a sua abrangência, que vai desde domínios essencialmente pertencentes à engenharia, como integração de dados de sensores, até o estudo de fenômenos que pertencem aos domínios das

ciências biológicas ou humanas como biologia, etologia, psicologia, sociologia e economia. Desta forma, a IAD pode desde estar preocupada com os princípios de coordenação e controle dos componentes altamente especializados de um sistema, até se dedicar a identificar características relacionadas ao comportamento dos agentes dentro de sociedades.

De modo geral, os aspectos importantes a serem definidos na concepção de um sistema de IAD incluem definir quais são os agentes do sistema, como eles cooperam, como interagem e se comunicam e os aspectos de controle, que definem qual agente atua em qual momento. Todos esses aspectos constituem a *arquitetura* do sistema. Em particular, podemos ter sistemas que envolvam princípios de IAD em diferentes níveis, onde um agente que faz parte de uma sociedade de agentes pode ser, ele próprio, organizado internamente como um sistema multi-agentes.

Na próxima seção, discutiremos os aspectos do projeto de um sistema de IAD, ou seja, a definição de quais são os agentes do sistema, como esses agentes cooperam, como é feito o controle de qual agente atua em qual instante, como podem ser tratados os eventuais conflitos, comunicação, interação, a organização do sistema e a arquitetura interna de um agente. Em seguida, discutiremos o conceito de arquitetura em IAD, ilustrando-o com as arquiteturas clássicas de Rede de Contratos e *blackboard*.

3.2 O projeto de um sistema de IAD

O projeto de um sistema de IAD envolve as seguintes etapas: decidir quais serão os agentes do sistema; definir quais as estratégias de cooperação a serem empregadas; definir como os agentes vão interagir entre si e com o ambiente e como vão se comunicar entre si; definir os aspectos de controle e a organização global do sistema.

3.2.1 Os agentes do sistema

Se o objetivo do sistema for modelar uma sociedade ou organização, os agentes estão automaticamente definidos, uma vez que são os seus membros ou componentes. Por outro lado, se o sistema tem como meta resolver um problema ou desempenhar uma tarefa, é necessário definir quem são os agentes e o que cada um deles vai fazer, o que pode ser feito a partir de decisões mais ou menos arbitrárias.

Os agentes de um sistema de IAD podem ser homogêneos ou heterogêneos. Agentes homogêneos são agentes que possuem as mesmas capacidades, habilidades etc. e agentes heterogêneos são agentes que possuem capacidades ou “especialidades” diferentes. Agentes homogêneos podem ser usados, por exemplo, em sistemas fisicamente distribuídos onde cada agente é associado a uma posição geográfica, em sistemas compostos por um grande número de agentes simples, e outros. Entretanto, em muitos sistemas a própria



heterogeneidade dos agentes constitui a sua característica mais importante, como no caso de um sistema no qual os agentes aplicam diferentes heurísticas a fim de convergir para uma solução satisfatória de um problema.

Se os agentes forem heterogêneos, existem diversas possibilidades: cada um deles pode ser especializado em desempenhar uma tarefa específica, ou vários agentes podem ser capazes de desempenhar a mesma tarefa usando diferentes táticas, ou cada agente pode ser capaz de tratar determinadas situações mas não outras e assim por diante. De acordo com a “especialidade” de cada agente, eles podem atuar de uma forma pré-determinada como, por exemplo, como se fosse uma linha de montagem, onde cada agente atua no instante adequado; podem atuar todos juntos em paralelo; ou ainda podem agir sob demanda, ou seja, cada agente entra em ação quando percebe uma oportunidade de atuar, ou quando outro agente solicita sua participação etc.; finalmente, diversos agentes podem ter que reunir seus esforços de forma coordenada ou sincronizada. Desta forma, a caracterização dos agentes conduz diretamente a outros aspectos do projeto do sistema, concernentes a como os agentes atuam cooperativamente, como os agentes sabem quando atuar, como os agentes interagem e se comunicam etc.

3.2.2 Cooperação

Uma vez definido quem são os agentes e o que cada um deles faz, é necessário definir como eles cooperam. Em determinados sistemas, os agentes podem nem mesmo ter que tomar conhecimento uns dos outros; no caso do sistema em que os agentes aplicam diferentes heurísticas, por exemplo, é necessário apenas que cada agente seja capaz de reconhecer as oportunidades adequadas para aplicar a sua heurística na solução do problema, sem haver necessidade do agente ter conhecimento da existência dos demais agentes. No extremo oposto, podemos imaginar sistemas nos quais os agentes tenham que agir de forma estritamente coordenada e sincronizada. A maior parte dos sistemas está evidentemente entre estes dois extremos.

Sob a abordagem de RDP foram identificados dois paradigmas básicos de cooperação entre agentes: compartilhamento de tarefas (*task-sharing*) e compartilhamento de resultados (*result-sharing*) [30].

O paradigma de compartilhamento de tarefas se baseia no princípio de que a tarefa que o sistema como um todo deve realizar pode ser decomposta em subtarefas que podem ser resolvidas por um ou mais agentes do sistema e que os resultados obtidos podem ser integrados a fim de se obter o resultado final. Em geral, a decomposição da tarefa que o sistema deve realizar em subtarefas é feita quando o sistema é projetado; no entanto, a decomposição e a distribuição de tarefas também podem ser feitas em tempo de execução, como em sistemas baseados na arquitetura de Rede de Contratos, nos quais a divisão de uma tarefa é tipicamente feita por um agente que “contrata” outros agentes para resolver

as subtarefas e depois realiza a síntese dos resultados.

O paradigma de compartilhamento de resultados é mais adequado para situações nas quais os resultados obtidos por um agente são utilizados como entrada (*input*) para outro agente, ou onde os resultados obtidos por um agente influenciam ou restringem a atuação de outros agentes, ou quando a solução para o problema é construída de forma incremental através da atuação dos agentes, ou quando o compartilhamento de resultados leva a uma convergência em direção a uma solução correta ou satisfatória. O esquema de compartilhamento de resultados não exclui a possibilidade de que exista uma divisão de tarefas entre os agentes, mas esta tem que ser feita *a priori*. Além disso, é necessário que se tenha algum meio através do qual os resultados possam ser compartilhados. Este pode tomar a forma de uma representação uniforme que permita aos agentes trocar resultados e/ou uma estrutura de memória compartilhada distribuída como, por exemplo, um *blackboard*, na qual os resultados possam ficar disponíveis para os agentes.

Do ponto de vista mais amplo dos sistemas multi-agentes, estes dois paradigmas nem sempre são suficientes, embora eles em geral o sejam quando o propósito do sistema consiste em resolver um problema ou desempenhar alguma tarefa. Para sistemas que não se encaixam nesta definição, podemos pensar em outros paradigmas como, por exemplo, um paradigma de compartilhamento de informações, que corresponde a sistemas multi-agentes nos quais os agentes essencialmente fornecem informações uns aos outros de modo a aumentar a informação que cada um deles possui individualmente e/ou aumentar a informação total que o grupo como um todo possui através da integração de fragmentos de informação anteriormente isolados. Esquemas híbridos também podem ser adotados, combinando aspectos dos diferentes paradigmas de cooperação.

3.2.3 Controle

O aspecto do controle corresponde a definir qual agente atua em que momento. Em sistemas nos quais a cooperação é feita através de compartilhamento de tarefas, por exemplo, o controle é tipicamente orientado por metas (*goal-directed*), enquanto em sistemas que utilizam compartilhamento de resultados o controle é em geral orientado por dados (*data-directed*), o que significa que o rumo do processamento é determinado pelos dados (resultados intermediários) disponíveis.

O controle pode ser feito localmente em cada agente, ou seja, cada agente possui todo o conhecimento necessário para decidir quando atuar. Em um esquema de compartilhamento de tarefas, o caso ideal seria aquele no qual as metas do sistema como um todo—às vezes chamadas “metas globais”—podem ser totalmente traduzidas em metas locais para os agentes do sistema, de modo que a situação na qual todos os agentes atingiram suas metas locais equivale à situação na qual o sistema considerado como um todo atingiu sua meta global. No esquema de compartilhamento de resultados, o caso ideal seria aquele no

qual a simples inspeção dos resultados disponíveis permite ao agente tomar uma decisão autônoma a respeito de que ações executar.

Em outros casos, a ordem na qual os agentes atuam pode ser importante, e assim pode ser necessário impor uma meta de mais alto nível a um agente ou a um conjunto de agentes etc. Nestes casos, o controle pode ser exercido através de um escalonador (*scheduler*) que define a ordem de atuação dos agentes ou através de uma organização estática ou dinâmica dos agentes de modo a que uns fiquem subordinados a outros.

Um aspecto importante do controle é a definição de critérios de parada ou condições de término, isto é, a definição de quando os agentes devem cessar suas atividades. Em um sistema que tem um objetivo bem definido, a condição de término é simplesmente a realização deste objetivo. Em sistemas cuja meta é obter uma solução satisfatória, no entanto, a condição de término pode ser relacionada ao tempo de execução, taxa de convergência etc. Em qualquer caso, podemos incorporar estes critérios a cada um dos agentes, ou podemos ter um agente que realiza as verificações adequadas e faz com que o sistema pare. Em sistemas cujo único objetivo é a interação entre os agentes, como por exemplo um sistema educacional interativo, pode não haver um critério de término senão a decisão do usuário.

3.2.4 Conflitos

Um outro aspecto que deve ser citado é a possibilidade de ocorrerem conflitos entre os agentes do sistema. Conflitos em um sistema de IAD ocorrem quando agentes discordam a respeito de alguma coisa. Agentes podem discordar quanto a um resultado, por exemplo, ou quanto a qual agente deve fazer alguma coisa em um determinado instante. Um conflito não envolve necessariamente uma contradição, pois pode envolver também uma relação de complementaridade [22]; por exemplo, agentes podem discordar quanto a seus resultados, mas estes podem ser complementares, podendo ser integrados em um resultado mais completo. É importante notar que também podem existir inconsistências ou contradições no sistema sem que estas causem conflitos, pois os agentes podem nunca chegar a confrontá-las.

Conflitos podem ser resolvidos por um subconjunto dos agentes envolvidos (um único agente, vários ou todos) ou por um agente especializado em resolver conflitos que atua como um “juiz” ou “árbitro.”

3.2.5 Comunicação e interação

Os mecanismos de comunicação incluem todos aqueles através dos quais os agentes podem trocar informações por meio do envio e recebimento de mensagens. Em alguns casos, é conveniente fazer uma distinção entre comunicação e interação, onde a primeira envolve

todos os atos relacionados à troca de informações e a segunda abrange o restante das formas de interação entre os agentes. Suponhamos, por exemplo, que temos um sistema que simula o tráfego de automóveis em uma cidade, onde os agentes são os veículos. Neste caso, acidentes, ultrapassagens, a percepção da distância e da posição de outros veículos etc. são exemplos de interação mas não de comunicação.

Os dois paradigmas fundamentais de comunicação entre agentes em IAD são troca de mensagens e memória compartilhada distribuída, que correspondem aos dois mecanismos análogos de comunicação entre processos. A diferença é que em lugar de processos temos agentes e as informações trocadas possuem estruturas e significados específicos.

A definição de como vai ser a comunicação entre os agentes em um sistema de IAD envolve a especificação de formatos e representações das informações que transitam pelo sistema e que possam ser entendidas pelos agentes apropriados. No caso do esquema de troca de mensagens, isto inclui a definição de um protocolo que especifica o formato das mensagens. Um exemplo é o Protocolo de Rede de Contratos (*Contract Net Protocol*), no qual a representação das informações trocadas entre os agentes é definida pela *Common Internode Language* [4, 29].

O esquema de troca de mensagens admite todos os diferentes tipos de difusão de mensagens: para um único destinatário, para vários destinatários (*multicast*) e para todos os agentes do sistema (*broadcast*). É importante notar que um sistema de IAD é um tipo particular de sistema de computação distribuída, de modo que, em princípio, qualquer mecanismo de comunicação para sistemas distribuídos como, por exemplo, comunicação de grupos [33] pode ser usado como suporte para implementação de um esquema de comunicação em um sistema de IAD.

O esquema de comunicação através de memória compartilhada distribuída se caracteriza pela presença de algum mecanismo que possa ser visto pelos agentes como se fosse uma região de memória compartilhada, ou seja, que pode ser acessada por meio de operações equivalentes às de leitura e escrita em memória. O esquema recebe o nome de memória compartilhada distribuída porque permite o acesso pelos componentes de um sistema distribuído, no qual tipicamente não há compartilhamento de memória física. Isto contrasta com o mecanismo tradicional de comunicação entre processos através de memória compartilhada, que pressupõe o compartilhamento de memória física. Diferentes propostas de implementação de memória compartilhada distribuída podem ser encontradas na literatura de sistemas distribuídos [32, 31]. O paradigma de memória compartilhada distribuída constitui o fundamento do modelo de *blackboard*, no qual os agentes se comunicam exclusivamente através de uma estrutura de dados compartilhada denominada *blackboard*, que geralmente contém dados organizados de forma hierárquica [6, 17, 16].

Tanto o esquema de troca de mensagens como o esquema de memória compartilhada distribuída podem ser usados para suportar qualquer uma das estratégias de cooperação,

isto é, compartilhamento de tarefas, compartilhamento de resultados, compartilhamento de informações etc. e suas combinações.

3.2.6 Organização

A organização do sistema designa as relações entre os agentes. Em geral estas relações são de dois tipos: subordinação ou igualdade. Uma relação de subordinação entre dois agentes é uma relação assimétrica na qual um agente dá ordens para o outro, definindo quando ele vai atuar ou o que ele vai fazer, ou na qual um agente apresenta seus resultados a outro, ou ambas. Relações de subordinação podem ser fixas ou podem ser estabelecidas e alteradas em tempo de execução. Uma relação de igualdade entre agentes é simplesmente a ausência de relações de subordinação entre eles.

Um conjunto de agentes com relações de igualdade é chamado grupo, time, comitê ou heterarquia. Um conjunto de agentes organizados através de relações de subordinação é denominado hierarquia ou organização. Um tipo particular de organização é a que recebe o nome de mercado. Em uma organização de mercado, as relações de subordinação são estabelecidas dinamicamente através da compra de serviços, resultados etc.

3.2.7 A arquitetura interna de um agente em um sistema de IAD

Um agente em um sistema de IAD consiste essencialmente em uma entidade computacional independente que possui capacidades de processamento e interação com outros agentes. Um agente pode ser um processo, uma *thread*, um procedimento, um objeto etc. que, além de possuir alguma capacidade de processamento em geral especializada, também é capaz de interagir com outros agentes de modo a trocar informações e estabelecer vínculos cooperativos. Nesta seção mostraremos dois exemplos de arquiteturas internas de agentes de sistemas de IAD.

A arquitetura de um agente de uma Rede de Contratos é mostrado na figura 3.1 [29]. O *processador de comunicações*, o *processador de contratos* e o *processador de tarefas* podem atuar de forma concorrente, e são responsáveis, respectivamente, pela troca de mensagens de baixo nível, pela decisão de a quais contratos o agente vai se candidatar e/ou quais candidatos vai escolher para contratar, e pela execução das tarefas para as quais o agente foi contratado.

Uma arquitetura um pouco mais complexa é mostrada na figura 3.2 [18]. O agente apresenta dois componentes nitidamente separados, o chamado *sistema inteligente* que contém a funcionalidade do agente e uma *camada de cooperação* composta por um *módulo de comunicação de alto nível* para troca de informações com outros agentes, um *módulo de cooperação e coordenação* para gerenciar protocolos de cooperação com outros agentes

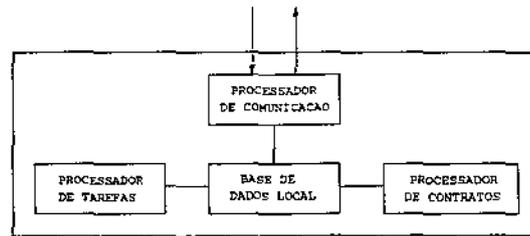


Figura 3.1: A arquitetura de um agente de uma Rede de Contratos.

e todas as atividades internas, um modelo do que o próprio agente pode fazer e do que ele precisa dos outros para fazer (*Self-Model*), um modelo do que outros agentes podem fazer e que pode interessar ao agente (*Agent Acquaintance Model*) e um monitor que agenda as atividades do sistema inteligente. Os componentes da camada de cooperação trocam informações entre si através de um *blackboard* dividido em duas partes, uma parte com dados e outra com metas. Este *blackboard* não é usado para comunicação entre agentes e sim para comunicação entre componentes internos do agente.

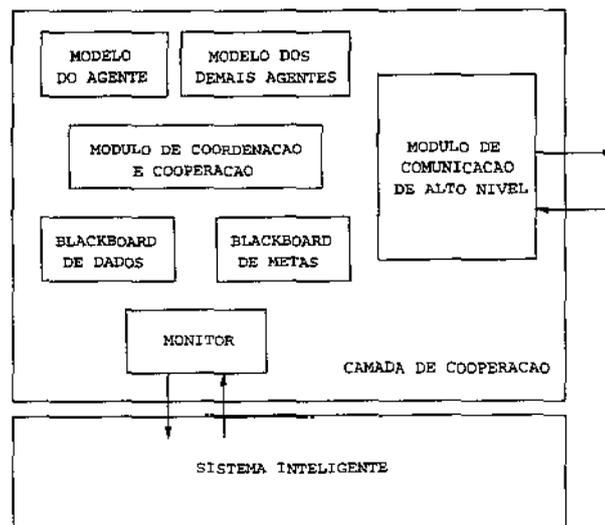


Figura 3.2: Outra arquitetura de um agente.

3.3 Arquiteturas de Inteligência Artificial Distribuída

Na seção anterior, discutimos todos os aspectos que devem ser considerados ao se projetar um sistema de IAD. Estes aspectos se referem a quais são os agentes do sistema, como as tarefas e o conhecimento do sistema são distribuídos entre os diversos agentes, como estes agentes são capazes de agir cooperativamente, como definir qual agente atua em que momento e quando o sistema deve cessar suas atividades, o que pode ser feito na presença de conflitos e como realizar as decisões concernentes a todos estes aspectos através de mecanismos de comunicação, interação e da organização do sistema. Finalmente, descrevemos dois exemplos de arquitetura interna de um agente de um sistema de IAD.

A possibilidade de abstrair as decisões relacionadas aos aspectos anteriormente enumerados—ou seja, a possibilidade de separar as decisões quanto à escolha de estratégias e paradigmas de um problema ou sistema particular—conduz à definição de arquiteturas de IAD. Desta forma, o próprio uso da denominação é flexível: podemos nos referir à arquitetura de um sistema específico como abstração de suas características, ao mesmo tempo que podemos ter uma arquitetura formada pelas características comuns a vários sistemas, ou uma arquitetura desenvolvida *a priori*, ou ainda uma arquitetura que é uma versão estendida ou restrita de alguma outra, e assim por diante.

Nesta seção, vamos descrever duas arquiteturas de IAD: a arquitetura de Rede de Contratos (*Contract Net*) e a arquitetura de *blackboard*, também chamada de modelo de *blackboard*. Estas duas arquiteturas foram escolhidas por diversos motivos. Primeiro, porque ilustram o conceito de uma arquitetura (abstrata) de IAD de forma especialmente adequada. Segundo, porque são arquiteturas bastante diferentes uma da outra, e portanto refletem decisões diferentes concernentes aos aspectos enumerados na seção anterior. Finalmente, porque são arquiteturas que, apesar de terem sido propostas nos primórdios da IAD, continuam sendo usadas como fundamento ou ponto de partidas para o desenvolvimento de sistemas e de novas arquiteturas. Em particular, o modelo de *blackboard* constitui o fundamento da arquitetura desenvolvida neste trabalho e que vamos descrever no próximo capítulo.

3.3.1 A arquitetura de Rede de Contratos

A arquitetura de Rede de Contratos [4, 29] utiliza uma estratégia de cooperação através de compartilhamento de tarefas na qual as três etapas—decomposição do problema em subtarefas, resolução das subtarefas e síntese do resultado—são feitas continuamente em tempo de execução pelos agentes do sistema. Subtarefas podem ser divididas sucessivamente, mas esta arquitetura pressupõe que em algum momento sejam obtidas subtarefas elementares o suficiente para serem resolvidas por agentes individuais.

Quando um agente de uma Rede de Contratos se depara com uma tarefa, ele pode

resolvê-la, se for capaz disto, ou dividi-la e delegar cada uma das subtarefas resultantes a um agente que possa resolvê-la. O problema de encontrar um agente adequado para resolver uma sub tarefa foi chamado pelos criadores da arquitetura de Rede de Contratos de *problema da conexão*. A idéia da arquitetura consiste em resolver o problema da conexão dinamicamente, ou seja, determinar o agente mais apropriado para resolver uma sub tarefa em tempo de execução. Assim, a arquitetura de Rede de Contratos consiste essencialmente em um protocolo de comunicação de alto nível denominado Protocolo de Rede de Contratos (*Contract Net Protocol*) através do qual os agentes do sistema realizam *negociações* com o propósito de resolver o problema da conexão. Este problema pode ser visto de duas perspectivas: de um lado, tarefas têm que ser executadas, e para isto é necessário encontrar agentes adequados e disponíveis; de outro lado, agentes ociosos procuram tarefas que sejam capazes de executar. O Protocolo de Rede de Contratos especifica o formato das mensagens que permitem aos agentes anunciarem tarefas ou disponibilidade de serviços, selecionarem (contratarem) agentes para executar tarefas e transmitirem resultados. O formato de algumas mensagens é mostrado na figura a seguir.

TO:	TO:	TO:
FROM:	FROM:	FROM:
TYPE: TASK ANNOUNCEMENT	TYPE: BID	TYPE: AWARD
CONTRACT:	CONTRACT:	CONTRACT:
TASK ABSTRACTION:	NODE ABSTRACTION:	TASK SPECIFICATION:
ELIGIBILITY SPECIFICATION:		
BID SPECIFICATION:		
EXPIRATION TIME:		

A mensagem de anúncio de tarefa (*task announcement*) permite a um agente divulgar uma tarefa ou sub tarefa que precisa ser executada. Esta mensagem pode ser enviada em *broadcast* ou no modo chamado *focused addressing*, que pode ser um *broadcast* restrito ou ponto-a-ponto. O campo (*slot*) de especificação de elegibilidade (*eligibility specification*) deve indicar quais os requisitos que um agente deve ter para poder se candidatar ao contrato de execução desta tarefa. O campo de descrição da tarefa (*task abstraction*) serve para os demais agentes avaliarem seu interesse em se candidatarem. O campo de especificação de proposta (*bid specification*) indica quais informações sobre o agente candidato devem ser enviadas na proposta (*bid*) para que o contratante possa avaliar qual o melhor candidato. Os agentes que pretenderem se candidatar a um contrato respondem à mensagem de anúncio de tarefa com uma mensagem do tipo proposta (*bid*). O campo descrição do nó (*node abstraction*) contém as informações sobre o candidato que foram especificadas no campo de especificação da proposta (*bid specification*) da mensagem de anúncio de tarefa. Quando o contratante escolhe um candidato para estabelecer um

contrato, ele envia uma mensagem de anúncio de contratação (*announced award*) para o agente escolhido. O campo de especificação da tarefa (*task specification*) contém as informações necessárias para dar início à execução da tarefa.

Se um agente que pretende anunciar uma tarefa já sabe qual agente deverá executá-la, é utilizada a mensagem de contrato direcionado (*directed award*). O agente que recebe esta mensagem pode responder com uma mensagem de aceitação (*acceptance*) ou rejeição (*refusal*).

A partir do estabelecimento do contrato, passa a existir uma relação entre contratante (*manager*) e contratado (*contractor*). A mensagem de informação (*information*) pode ser usada para troca de informações entre contratante e contratado. Esta mensagem também pode ser usada, juntamente com a mensagem de pedido (*request*), para troca de informações entre agentes quaisquer, isto é, sem que haja um contrato estabelecido. A mensagem de relatório intermediário (*interim report*) é usada para o agente contratado enviar resultados parciais, e a mensagem de relatório final (*final report*), para enviar os resultados completos. Se, por algum motivo, o agente contratante tiver que cancelar o contrato, ele envia uma mensagem de término (*termination*), fazendo com que o contratado pare de executar a tarefa e/ou cancele todos os subcontratos correspondentes.

A mensagem de nó disponível (*node available*) permite a um agente indicar que está ocioso e, portanto, disponível para executar tarefas. O campo de especificação de elegibilidade (*eligibility specification*) indica os requisitos nos quais uma tarefa deve se enquadrar para poder ser executada por este agente. O campo de descrição do nó (*node abstraction*) descreve as características do agente ocioso.

Todas as mensagens possuem um cabeçalho com a identificação do remetente e do(s) destinatário(s) e outros dados de verificação. O conteúdo dos campos (*slots*) é codificado em uma linguagem denominada *Common Internode Language*. A maior parte do conteúdo desta linguagem é dependente da aplicação e portanto deve ser definida durante o projeto do sistema, mas todos os agentes devem ser capazes de entendê-la.

Através do esquema de contratação, estabelecem-se relações de subordinação temporárias entre os agentes. Estes formam uma hierarquia que reflete a hierarquia de decomposição sucessiva de tarefas.

A arquitetura de Rede de Contratos especifica um protocolo de troca de mensagens para compartilhamento de tarefas e controle dos agentes, mas não especifica um particionamento particular do problema. Os agentes do sistema podem possuir diferentes especialidades, sendo portanto capazes de executar tarefas de certos tipos e não de outros, mas isto é apenas uma possibilidade. Um outro exemplo pode ser observado em uma aplicação específica desta arquitetura: um sistema de monitoramento de tráfego de veículos em uma região [4]. Agentes espalhados em uma ampla região geográfica possuem capacidades de sensoriamento e de processamento. Um agente processador é responsável

por produzir um mapa final a partir da integração dos dados de todos os demais. Para isto, este agente contrata os serviços de outros agentes processadores que produzem mapas parciais de áreas menores. Estes agentes, por sua vez, contratam serviços de agentes que provêm dados de sensoriamento relativos a partes da área e dados relativos a veículos particulares, que são obtidos contratando os serviços dos agentes sensores. Desta forma, agentes que apresentam as mesmas capacidades são escolhidos para executar determinadas tarefas em função de sua localização geográfica.

3.3.2 A arquitetura de blackboard

A arquitetura de *blackboard* surgiu a partir do modelo de *blackboard* para resolução de problemas. Este modelo especifica como organizar os passos de raciocínio e o conhecimento do domínio para resolver um problema, e se caracteriza pela aplicação “oportunistica”—ou seja, quando parecer conveniente—de geração de possíveis soluções a partir de hipóteses (*forward-reasoning*) e geração de hipóteses a partir de possíveis soluções (*backward-reasoning*) por componentes independentes e disjuntos denominados KSs (*knowledge sources*) que se comunicam exclusivamente através do compartilhamento de resultados parciais ou preliminares depositados em uma estrutura de dados global chamada *blackboard*.

Este modelo de resolução de problemas foi utilizado no sistema HEARSAY-II [17], de cujos princípios de funcionamento a arquitetura de *blackboard* foi originalmente abstraída. O sistema HEARSAY-II foi desenvolvido na década de 70 e pode ser considerado o precursor dos sistemas de IAD. O objetivo do HEARSAY-II consiste em reconhecer sentenças faladas. A principal característica do problema de reconhecimento de sentenças faladas é que palavras são pronunciadas de formas diferentes não somente quando são proferidas por pessoas diferentes, mas também pela mesma pessoa em ocasiões diferentes. A consequência disto é que o conhecimento que permite ao sistema mapear seqüências de sinais sonoros em sentenças é necessariamente aproximado, pois não pode acomodar todas as variações possíveis. A fim de lidar com este fato, foram adotados os seguintes princípios de funcionamento:

- o conhecimento do domínio do problema foi dividido em níveis de abstração, de modo que em um nível ocorre o mapeamento de sinais sonoros em segmentos fonêmicos, no nível seguinte ocorre o agrupamento de segmentos fonêmicos em sílabas, depois sílabas são reunidas em palavras e assim por diante;
- este particionamento do conhecimento foi mapeado sobre agentes separados—chamados KSs, *knowledge sources* ou fontes de conhecimento—capazes de operar independentemente uns dos outros, de modo que uma KS toma sinais sonoros como entrada e

os mapeia em segmentos fonêmicos que são devolvidos como saída, outra KS toma segmentos fonêmicos como entrada e devolve sílabas etc.;

- os dados foram colocados em um *blackboard* dividido em seções que refletem os tipos de dados que as KSs manipulam, ou seja, sinais sonoros, segmentos fonêmicos, sílabas, palavras etc.

Inicialmente o *blackboard* contém apenas os dados de entrada do sistema, que são sinais sonoros. A partir de então, cada KS espera por uma situação na qual possa contribuir para a solução do problema, que visa obter uma sentença escrita plausível correspondente aos sinais sonoros de entrada. Quando uma KS detecta uma tal situação, ela lê do *blackboard* os dados sobre os quais vai operar e escreve nele seus dados de saída. Assim, ao iniciar a execução do sistema, a KS que mapeia sinais sonoros em segmentos fonêmicos processa os sinais originais e escreve no *blackboard* os segmentos que conseguiu reconhecer, que por sua vez servem de dados de entrada para outra KS, e assim a execução prossegue.

As características de cooperação e comunicação descritas anteriormente correspondem ao modelo de *blackboard* em sua forma original. O *blackboard* é um repositório de elementos de um espaço de soluções que inclui soluções completas e incompletas, dados de entrada, soluções temporárias, hipóteses etc. A arquitetura de *blackboard* prescreve uma organização hierárquica para estes elementos. Esta hierarquia pode ser de diversos tipos: de abstrações, todo-parte etc. O sistema HEARSAY-II, por exemplo, utiliza uma hierarquia de abstrações. Os elementos do *blackboard* servem como dados de entrada para agentes ou KSs. Um *blackboard* pode conter diversas hierarquias; neste caso, cada uma delas é chamada *painel (blackboard panel)*. O *blackboard* pode ser visto como um espaço sobre o qual a solução para o problema é gradativamente construída, seja juntando partes, seja através de refinamentos sucessivos. Dados de controle, isto é, usados pelo componente de controle do sistema, também podem ser armazenados no *blackboard*.

O conhecimento do domínio do problema é particionado em agentes ou KSs independentes. Um agente ou KS pode ser visto como sendo composto de duas partes: uma parte que identifica ou descreve situações nas quais o agente ou KS pode contribuir para a construção da solução do problema, e uma parte que especifica qual o acréscimo ou modificação que o agente ou KS efetua no *blackboard*. A primeira parte recebe às vezes o nome de *pré-condição*, e a segunda, de *ação*. Um agente ou KS pode tomar a forma de um procedimento, um conjunto de regras etc.

O particionamento do conhecimento do domínio do problema em agentes ou KSs está relacionado com a estruturação do *blackboard*. Tipicamente um agente ou KS processa dados de um nível do *blackboard* e contribui com dados em outro nível. Por exemplo, no sistema HASP [17] para identificação de plataformas marítimas (navios, submarinos etc.) a partir de sinais captados por hidrofones, a maior parte das KSs realiza processamento

bottom-up ou *data-driven*, combinando sinais para formar hipóteses sobre a presença de plataformas, enquanto outras KSs atuam no sentido *top-down* ou *model-driven*, gerando hipóteses quanto aos sinais a serem captados na presença de determinadas plataformas.

A arquitetura de *blackboard* foi originalmente proposta para ambientes de computação serial, onde seu caráter distribuído se limita à distribuição do conhecimento em agentes ou KSs independentes e que por isso podem atuar em qualquer ordem, segundo uma estratégia oportunista. Por isso, a arquitetura de *blackboard* inclui um componente de controle que é responsável pela atuação oportunista dos agentes ou KSs. Este componente de controle é composto por um ou mais *monitores* e um *escalonador (scheduler)*. Os monitores detectam *eventos* que indicam a possibilidade de atuação de algum agente ou KS. Em geral, eventos correspondem a modificações no *blackboard*. Por exemplo, quando um agente acrescenta novos dados em um determinado nível do *blackboard*, os agentes que utilizam estes dados como entrada podem reconhecer nestes dados material interessante para ser processado neste momento. Eventos também podem ser de outros tipos, como a passagem de um intervalo de tempo. O escalonador é responsável pela escolha do próximo agente ou KS a atuar. Esta escolha pode ser feita a partir de informações fornecidas pelos próprios agentes (por exemplo, quais agentes estão aptos a atuar), ou a partir de informações no *blackboard* (por exemplo, qual “ilha” de soluções ou hipóteses deve ser explorada), ou ambas. A função do controle também pode ser vista como a definição de um *foco de atenção* para o sistema (como um todo) a cada instante.

Em um ambiente de computação distribuída, que pode ser desde um único computador com um sistema operacional multitarefa até uma rede de computadores, os agentes podem operar de forma realmente concorrente. Cada agente ou KS poderia monitorar o *blackboard* e outros eventos, reconhecendo as oportunidades de atuar e realizando as ações apropriadas. Neste caso, não haveria necessidade de monitores e escalonador, mas as leituras e escritas concorrentes transformariam o *blackboard* em uma região crítica que teria que ter o acesso devidamente controlado através de mecanismos de exclusão mútua.

Finalmente, pode-se ter algum critério de término que indica quando o processo deve ser detido. Em geral não há um critério bem definido, pela própria natureza das aplicações às quais a arquitetura de *blackboard* se adequa; por isso, este critério pode ser encontrar uma solução “aceitável” ou “satisfatória”, ou mesmo a impossibilidade de prosseguir.

Atualmente, o modelo de *blackboard* se caracteriza pelo fato de o *blackboard* ser um repositório de dados estruturado, em geral organizado hierarquicamente, e usado como meio exclusivo de comunicação entre agentes. O uso do *blackboard* para conter resultados intermediários não constitui uma característica essencial do modelo; ao contrário, o *blackboard* também pode ser utilizado sob uma estratégia de compartilhamento de tarefas.

3.4 Conclusão

Neste capítulo, descrevemos sistemas de Inteligência Artificial Distribuída (IAD) como sendo sistemas nos quais múltiplos agentes atuam juntos, e discutimos os aspectos que devem ser definidos ao se projetar um sistema de IAD, ou seja, quais são os agentes, como as tarefas do sistema e o conhecimento são divididos entre os agentes, como estes cooperam, como é feito o controle do sistema, a possibilidade de ocorrerem conflitos e a realização das decisões concernentes a todos estes aspectos através de mecanismos de comunicação e da organização do sistema. Discutimos também a arquitetura interna de um agente de um sistema de IAD. Em seguida, foram discutidos o conceito de uma arquitetura de IAD, que corresponde a uma abstração das características do sistema no que se refere aos aspectos anteriormente apresentados; foram apresentadas como exemplos as arquiteturas de Rede de Contratos e *blackboard*. A Inteligência Artificial Distribuída é o paradigma computacional que utilizamos para projetar a arquitetura para agentes inteligentes baseada na Sociedade da Mente, que será descrita no próximo capítulo. Os aspectos da IAD que discutimos aqui correspondem aos aspectos que definimos a partir da adequação de conceitos da Sociedade da Mente.

Capítulo 4

O desenvolvimento de uma arquitetura para agentes inteligentes baseada na Sociedade da Mente

4.1 Introdução

No primeiro capítulo, dissemos que a arquitetura de um agente pode ser dividida em duas partes: uma parte relativa à funcionalidade do agente, e a outra relativa à sua interação com os demais agentes do ambiente. Destas duas partes, a primeira depende apenas do que queremos que o agente faça, enquanto a segunda depende da arquitetura do ambiente multi-agentes como um todo. Se pensarmos em um ambiente multi-agentes aberto, isto é, no qual seja possível acrescentar agentes quando se queira, só faz sentido especificar a parte de interação; cada agente individual terá sua funcionalidade própria, e portanto esta parte da arquitetura simplesmente não pode ser definida sem que se saiba qual é esta funcionalidade. Por outro lado, se pensarmos em um agente sem levar em conta o ambiente no qual ele poderá estar imerso, não faz sentido especificar a parte de interação, somente a de funcionalidade. As arquiteturas de agentes desenvolvidas de acordo com cada um destes dois pontos de partida se completam quando pensamos em um ambiente multi-agentes povoado com agentes, ou seja, quando pensamos em agentes inseridos em um ambiente multi-agentes ou em um ambiente no qual existe um único agente computacional e um agente humano (o usuário). Neste trabalho, como estamos pensando em um agente isolado e não em um ambiente multi-agentes, o que entendemos por “arquitetura” se refere somente à parte de funcionalidade do agente.

Neste capítulo descreveremos a arquitetura de um agente inteligente como uma arquitetura de IAD baseada no modelo da Sociedade da Mente. Na próxima seção mostraremos quais são os aspectos do modelo da Sociedade da Mente que vamos utilizar. Na seção 4.3

descreveremos o processo de desenvolvimento da arquitetura. Finalmente, discutiremos algumas possibilidades de aplicações, e faremos uma comparação com a arquitetura M de agentes integrados [21], que é a arquitetura de um assistente que auxilia um usuário em um *Virtual Meeting Room* e também utiliza idéias da Sociedade da Mente.

4.2 O modelo da Sociedade da Mente e a arquitetura de um agente

A nossa intenção não é simular o modelo da Sociedade da Mente. Apesar de estarmos nos referindo a ele como um modelo, na verdade a Sociedade da Mente consiste antes em uma coleção de idéias inter-relacionadas. Nosso objetivo consiste em traduzir algumas destas idéias para uma arquitetura genérica de IAD.

A tradução de idéias da Sociedade da Mente para uma arquitetura de IAD não é direta, porque a Sociedade da Mente é um modelo conexionista composto por agentes de granularidade fina que realizam um tipo de processamento maciçamente paralelo. Em IAD, agentes correspondem a programas ou partes de programas que se comunicam entre si; portanto, em lugar de uma abordagem conexionista, temos uma abordagem “simbólica” ou tradicional; em lugar de agentes de granularidade fina, temos agentes de granularidade maior; em lugar de processamento maciçamente paralelo, temos processamento distribuído. Ao mesmo tempo, conforme visto no capítulo 2, o modelo da Sociedade da Mente pode ser descrito em diferentes níveis de abstração. Assim, aquelas diferenças foram contornadas partindo de um nível de descrição mais alto do modelo da Sociedade da Mente, no qual existem dois tipos de agentes: agentes de coordenação, como *k-lines*, reconhecedores etc., e agentes que realizam tarefas específicas de alto nível. Estes últimos correspondem a agências de um nível de descrição mais baixo, mas como estamos partindo de um nível acima deste, estas agências não podem ser vistas como grupos de agentes mas somente como “agências atômicas,” ou seja, agências que não podem ser subdivididas. Agências atômicas e agentes de coordenação constituem, portanto, os dois componentes do modelo da Sociedade da Mente que vamos traduzir em componentes de uma arquitetura de IAD.

Parece-nos plenamente justificado traduzir o modelo da Sociedade da Mente em uma arquitetura de IAD partindo de um nível de descrição arbitrário, porque consideramos que este “estilo” descritivo peculiar da Sociedade da Mente constitui mais do que um estilo meramente expositivo. Ele reflete o fato de que o modelo leva em consideração o fenômeno de “computação emergente,” ou seja, aquilo que um agente realiza em um nível corresponde à computação emergente que uma agência, ou seja, um grupo de agentes, realiza em um nível inferior, e constitui também uma estratégia para dominar a complexi-

dade do modelo através de abstração, semelhante à programação estruturada, permitindo encapsular uma agência e vê-la como um agente.

4.3 Desenvolvimento da arquitetura

Nesta seção descrevemos como os conceitos de agências atômicas e agentes de coordenação foram progressivamente traduzidos em aspectos de uma arquitetura de IAD. Deste modo, em lugar de descrever a arquitetura em sua forma acabada, adotamos uma perspectiva que mostra o seu desenvolvimento e as diferentes possibilidades em que pensamos para realizar aquela tradução.

Neste processo, a correspondência entre uma agência atômica e um agente do sistema de IAD permaneceu invariante, enquanto a correspondência entre agentes de coordenação e componentes do sistema de IAD foi sendo modificada. Apontamos também para o que pode ser visto como estados mentais parciais e como um *B-brain*—isto é, um conjunto de componentes dedicado a coordenar outros componentes da mente—ao longo do processo.

Embora o processo de desenvolvimento da arquitetura esteja dividido em quatro etapas para fins expositivos, ele não descreve quatro arquiteturas, mas o desenvolvimento gradual de uma única arquitetura.

4.3.1 Uma sociedade de agentes

Ao lado das agências atômicas e dos agentes de coordenação, um terceiro aspecto do modelo da Sociedade da Mente que deve ser traduzido em algum aspecto de uma arquitetura de IAD é o conceito de estado mental parcial, que constitui originalmente um conceito de baixo nível e de natureza conexionista, mas cujo significado é o de comunicação entre agências. Agências se comunicam umas com as outras evocando e detectando estados mentais parciais dentro e fora delas mesmas. Evocar um estado mental parcial dentro de uma agência pode ser considerado equivalente a invocar uma “função” ou “procedimento” desta agência, enquanto evocar um estado mental parcial fora de uma agência pode ser visto como “passagem de parâmetros” ou a definição dos dados a serem processados pela agência. Assim, começamos considerando cada agente de coordenação como um agente do sistema de IAD que envia mensagens para outros agentes indicando basicamente quais funções executar sobre quais dados, embora isto só tenha que ser feito explicitamente no caso das agências atômicas.

Cada um dos agentes do sistema de IAD corresponde portanto a um programa ou processo que recebe entradas, realiza algum processamento e envia saídas para outros agentes. Entradas e saídas correspondem a mensagens, e também podem ser consideradas como “conexões” entre os agentes. Cada agente deve saber para quais agentes enviar suas

saídas, e pode enviá-las sempre para os mesmos agentes, ou para agentes diferentes, assim como pode enviar mensagens diferentes para cada um ou a mesma mensagem para vários agentes. As saídas de um agente correspondem às entradas de outros agentes, exceto no caso dos agentes que constituem a “interface” do sistema, ou seja, aqueles que se comunicam com elementos externos ao agente, que podem ser desde elementos de uma interface gráfica até agentes de um ambiente multi-agentes.

Existem basicamente quatro tipos de mensagens:

Ativação: tem o efeito de ativar o agente que a recebe, sendo que o verdadeiro significado da “ativação” depende do agente ativado;

Inibição: tem o efeito contrário, ou seja, de inibir o agente que a recebe. Da mesma forma, o efeito real depende do agente receptor;

Informação: carrega algum tipo de conteúdo para ser processado por uma agência atômica;

Administrativa: carrega informações que modificam o comportamento de um agente de coordenação.

Destes, os dois últimos podem ser divididos em subtipos. A totalidade dos tipos de mensagens permite definir um protocolo para o sistema. Os diferentes tipos de agentes são caracterizados por receberem e enviarem diferentes tipos de mensagens e também pela maneira com que estas são processadas. Agentes de coordenação podem ser reunidos em três grupos que abrangem agentes com comportamentos semelhantes:

Agentes memorizadores e agentes de ligação: *k-lines*, *polynemes*, *micronemes*, *pronomes* e *paranomes*;

Agentes reconhecedores: reconhecedores, *demons*, supressores e censores;

Agentes de frames: *frames* e terminais de *frames*.

Agentes memorizadores são agentes capazes de restaurar estados mentais parciais no modelo da Sociedade da Mente; são, portanto, agentes que enviam sempre as mesmas mensagens para outros agentes quando ativados, mensagens estas que devem ter sido anteriormente armazenadas, isto é, transmitidas através de uma mensagem administrativa. Agentes deste tipo respondem a mensagens de ativação enviando mensagens previamente definidas para outros agentes, também anteriormente definidos, e respondem a mensagens administrativas modificando o conteúdo das mensagens a serem enviadas e/ou seus destinatários.

O tipo básico de agente memorizador é a *k-line*. Uma *k-line* nova é formada do seguinte modo: primeiro, um agente já existente cria um novo agente do tipo *k-line*, enviando em seguida uma mensagem administrativa indicando quais as mensagens que esta deve enviar para quais agentes toda vez que for ativada. O mesmo agente pode “conectar” outros agentes a esta *k-line*, fazendo com que estes possam enviar-lhe mensagens de ativação, ou ele mesmo pode vir a fazê-lo no futuro. A partir de então, quando a nova *k-line* recebe uma mensagem de ativação, ela responde imediatamente enviando as mensagens pré-definidas para seus respectivos destinatários.

Os outros tipos de agentes memorizadores são variantes de *k-lines*: um *polyneme* é uma *k-line* que envia mensagens de ativação para outras *k-lines*, e um *microneme* é um tipo de *k-line* que envia mensagens de informação para diversos agentes espalhados pela mente. Os chamados agentes de ligação são agentes que estabelecem ligações entre outros agentes, e incluem *pronomes* e *paranomes*.

Agentes reconhedores são agentes que recebem mensagens de informação e analisam-nas ou comparam-nas com informações previamente armazenadas, gerando uma resposta do tipo “sim” ou “não” ou um “grau de reconhecimento” que expressa o resultado da análise ou comparação. Um reconhedor também pode enviar mensagens de ativação em resposta ao reconhecimento. Supressores e censores são reconhedores que enviam exclusivamente mensagens de inibição. *Demons* são reconhedores que enviam mensagens em resposta a certas mensagens e não a outras, de modo que a presença ou ausência de resposta é que constitui a sua reação.

Frames são compostos por terminais ou *slots*. Cada terminal constitui um agente individual que recebe uma mensagem de ativação do agente *frame* e possivelmente de outros agentes, enviando em resposta mensagens de ativação para agentes memorizadores. Também seria possível reunir o *frame* e seus terminais em um único agente, mas optamos por deixar assim nesta versão para manter a uniformidade.

O comportamento dos agentes de coordenação consiste, portanto, em enviar determinados tipos de mensagens em resposta ao recebimento de outras mensagens, sendo que apenas reconhedores realizam algum processamento, mesmo assim muito simples. Os tipos das mensagens podem ser definidos pelo tipo do agente, ou podem ter que ser definidos, assim como seu conteúdo, quando for o caso. Mensagens administrativas são mensagens que trazem este tipo de indicação, assim como indicam os destinatários das respectivas mensagens. Isto permite identificar uma espécie de relação de subordinação entre agentes, onde os agentes que enviam mensagens administrativas para outros atuam como “administradores” destes. A partir disto, podemos organizar os agentes em camadas administrativas sucessivas, que podem então ser comparadas com a noção de *B-brain*, *C-brain* etc. na Sociedade da Mente.

O comportamento das agências atômicas depende de cada aplicação em particular.

Agências atômicas podem receber e enviar qualquer tipo de mensagem para qualquer agente do sistema, e podem realizar qualquer tipo de processamento sobre os dados contidos nas mensagens que recebem e dados que estejam eventualmente armazenados e que constituem parte de seu estado interno de modo geral. Além disso, mensagens podem ser recebidas em qualquer ordem e não precisam ser processadas logo que são recebidas, podendo ter seu conteúdo armazenado para processamento posterior. Por este motivo, é conveniente dividir o funcionamento de uma agência atômica em três partes:

Pré-processamento: realizado toda vez que o agente recebe uma mensagem. O pré-processamento inclui armazenar o conteúdo da mensagem quando for o caso e decidir se o agente vai ser ativado, inibido ou nenhum dos dois, assim como decidir qual o tipo de processamento que o agente vai realizar, se houver mais de um;

Processamento: corresponde à função propriamente dita do agente, segundo a qual as entradas são processadas gerando saídas. Se houver mais de um tipo de processamento disponível, não é necessário que todos eles utilizem todas as entradas;

Pós-processamento: corresponde ao envio das saídas sob a forma de mensagens para outros agentes.

A mesma divisão pode ser feita para os agentes de coordenação, mas é mais relevante no caso das agências atômicas, para cada uma das quais pré-processamento, processamento e pós-processamento podem ser diferentes. Esta divisão separa as atividades de comunicação do processamento propriamente dito. Em particular, as três atividades podem ser realizadas de forma concorrente por um agente.

Os aspectos essenciais da “primeira versão” da arquitetura são portanto os seguintes:

- cada agente de coordenação corresponde a um agente do sistema de IAD. Com exceção dos reconhedores, estes agentes não realizam nenhum processamento: eles apenas respondem a determinadas mensagens enviando outras mensagens ou modificando suas informações internas a respeito das mensagens que devem enviar;
- todos os agentes do sistema se comunicam através de troca de mensagens. Estas mensagens contém dados e não somente sinais de ativação ou inibição. O significado dos estados mentais parciais corresponde aproximadamente ao significado das mensagens.

A figura 4.1 mostra uma parte de um sistema fictício baseado na primeira versão da arquitetura. Os agentes A, B e C são agências atômicas que processam dados, possivelmente enviando seus dados de saída para outros agentes que não aparecem na figura. As *k-lines* D e E enviam mensagens específicas para A e B e para C e D, respectivamente.

Estas mensagens podem tanto conter dados de entrada como dados de controle que indicam o que A ou B devem fazer. Quando a *k-line* D é ativada, ela envia suas mensagens para A e B; analogamente, quando a *k-line* E é ativada, ela envia suas mensagens para B e C. Os agentes A, B e C podem receber mensagens adicionais dos agentes F, G e H. Os agentes I e J ativam as *k-lines* D e E nos instantes apropriados. As mensagens que D e E enviam para A, B e C quando ativadas podem ter sido definidas através de mensagens administrativas dos agentes K e L.

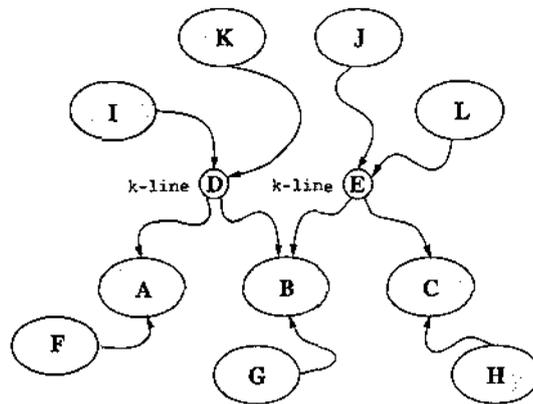


Figura 4.1: Um sistema fictício.

Nesta versão podemos porém identificar três inadequações para uma arquitetura de IAD:

- os agentes de coordenação, que são agentes muito simples, existem potencialmente em grande número;
- excessivo *overhead* de comunicação, uma vez que estes mesmos agentes de coordenação praticamente só recebem e enviam mensagens;
- agentes de coordenação, especialmente *k-lines* e suas variantes, são tipicamente criados e modificados dinamicamente, mas não é usual ter sistemas de IAD nos quais isto é feito com um grande número de agentes.

A segunda versão, descrita a seguir, é o resultado da tentativa de contornar estes problemas.

4.3.2 Agrupando agentes de coordenação

A solução que imaginamos para contornar os problemas da versão anterior foi agrupar agentes de coordenação do mesmo tipo em um agente do sistema de IAD; os agentes de

coordenação passam a ser então *objetos* que residem em agentes especiais e que implementam suas funcionalidades. Assim, em lugar de diversos agentes correspondentes a *k-lines*, tem-se um único agente que contém uma grande quantidade de *k-lines*, ou um pequeno número de agentes deste tipo; um em cada máquina, por exemplo. A mesma coisa para reconhecedores, *polynemes*, *pronomes*, *paranomes* etc. *Frames* e seus terminais passam a residir em um mesmo agente. Além destes, tem-se os agentes que correspondem às agências atômicas, que continuam sendo como na primeira versão.

Um tipo adicional de agente que pensamos em acrescentar nesta segunda versão são os chamados “agentes repositórios.” Estes agentes atuam como depósitos de dados que podem ser consultados ou modificados pelas agências atômicas. Podem ser considerados, portanto, como módulos de memória compartilhada distribuída, de modo que agências atômicas podem se comunicar tanto através de troca de mensagens como através dos dados contidos nos agentes repositórios.

Os tipos de mensagens continuam sendo essencialmente os mesmos da primeira versão, com a diferença de que as mensagens administrativas passam a incluir requisições para criação e destruição de objetos de coordenação e também a exigir a identificação do objeto a que se referem. Se na primeira versão bastava enviar uma mensagem de ativação para uma *k-line*, por exemplo, agora é necessário enviar uma mensagem de ativação para o agente que contém a *k-line*, contendo uma indicação de a qual das *k-lines* contidas no agente a mensagem se refere. Criar e destruir objetos de coordenação passa a ser simples, porque não envolve (em geral) a criação de um novo agente ou destruição de um agente do sistema.

A figura 4.2 mostra o subsistema equivalente ao mostrado na figura 4.1.

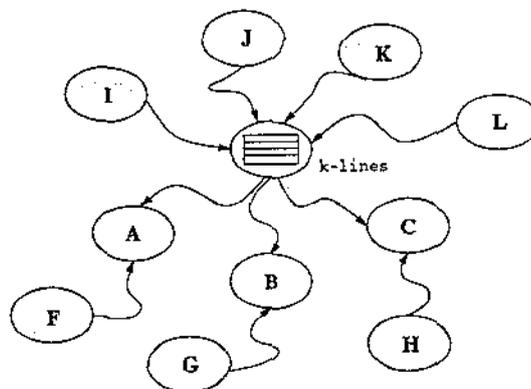


Figura 4.2: O sistema anterior na segunda versão da arquitetura.

Apesar do número de agentes ser menor, ainda existe um certo *overhead* de comunicação, uma vez que os agentes que contém objetos de coordenação praticamente só re-

cebem mensagens e enviam outras em resposta. Isto sugeriu uma modificação no esquema de comunicação entre os agentes, resultando em uma terceira versão que descrevemos a seguir.

4.3.3 Modificando o esquema de comunicação

Nesta versão, cada agente foi dividido em duas partes: o agente propriamente dito e uma área de memória compartilhada distribuída. No caso dos agentes que contém objetos de coordenação, esta área contém os próprios objetos; no caso das agências atômicas, esta área contém dados que podem ser manipulados tanto pelo agente como por outros agentes, sendo que além disso o agente também pode possuir dados internos, que residem no próprio agente. Esta modificação é mostrada na figura 4.3.

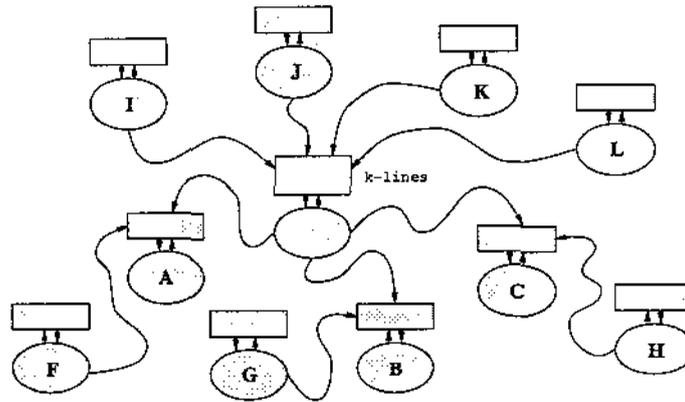


Figura 4.3: O sistema anterior na terceira versão da arquitetura.

Agora, em lugar de enviar mensagens, os agentes modificam dados diretamente nas áreas de memória compartilhada distribuída. Por exemplo, se o agente J quer ativar a *k-line* D, ele modifica o seu estado para ativo na área de memória compartilhada do agente de *k-lines*, que verifica continuamente o estado das *k-lines* e modifica dados adequadamente nas áreas de memória compartilhada de outros agentes. Assim, na terceira versão da arquitetura a comunicação passa a ser feita exclusivamente através de memória compartilhada distribuída. Isto torna os estados mentais parciais muito mais fáceis de serem caracterizados, pois eles correspondem a conteúdos destas áreas.

Observamos no entanto que cada uma das áreas de memória compartilhada distribuída possui uma estrutura própria, que deve ser conhecida pelos agentes que a acessam. Por exemplo, a área associada a um agente de *k-lines* contém uma tabela de *k-lines* que indica, para cada uma delas, seu estado e as modificações que devem ser feitas nas áreas

de memória compartilhada de outros agentes quando a *k-line* for ativada. De modo geral, os objetos de coordenação consistem em coleções de referências a outros objetos. Com base neste fato, decidimos reunir todos os módulos de memória compartilhada distribuída em um só, cada um deles passando a ser, com sua estrutura particular, uma seção (horizontal) em um *blackboard*, conforme descrevemos a seguir.

4.3.4 A versão final da arquitetura

O *blackboard* é constituído por seis seções que contém, respectivamente: objetos de dados específicos da aplicação, reconhecedores e suas variantes, *k-lines* e suas variantes, *pronomes*, *frames* e *paranomes* (figura 4.4). Todas as seções podem ser divididas em subseções conforme contenham diferentes variantes de seus respectivos objetos. Isto se aplica especialmente às seções de dados e de *frames*, que em geral vão conter diferentes tipos de dados e de *frames*.

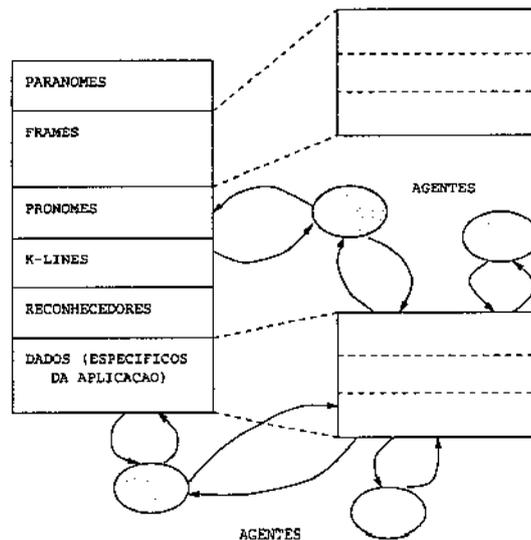


Figura 4.4: A versão final da arquitetura.

Os objetos do *blackboard* são essencialmente coleções de apontadores para outros objetos. Cada objeto, com exceção dos reconhecedores, possui um *estado* cujo significado é dependente da aplicação. O funcionamento dos objetos de coordenação consiste em causar a modificação dos estados de outros objetos a partir do seu próprio estado ou ainda dos estados de outros objetos. O funcionamento de cada um dos tipos de objetos, supondo que existam apenas os estados ativo ou inativo, é descrito a seguir:

Reconhecedores: o tipo mais simples de reconhecedor é mostrado na figura 4.5, e consiste em duas coleções de apontadores para outros objetos; se o número de objetos apontados pela primeira coleção que está no estado ativo for maior que um certo limite, os objetos apontados pela segunda coleção são ativados. Outra possibilidade é mostrada na figura 4.6: neste caso, tem-se três coleções de apontadores; se o número de objetos apontados pela primeira coleção que satisfazem algum tipo de comparação com os objetos apontados pela segunda coleção for maior que um determinado limite, os objetos apontados pela terceira coleção são ativados. Outras possibilidades podem ser imaginadas, uma vez que tanto o número de estados possíveis como seu significado e também as comparações que podem ser feitas são dependentes da aplicação particular a ser desenvolvida. Uma variante dos reconhecedores são os supressores, que inibem objetos em vez de ativá-los. Além disso, pode-se ter também combinações de ambos, ou seja, reconhecedores que ativam alguns objetos e inibem outros;

K-lines: uma *k-line* consiste em uma coleção de apontadores para outros objetos no *blackboard* (figura 4.7). Quando a *k-line* é ativada, os objetos apontados também o são. Uma segunda possibilidade é a *k-line* causar modificações no conteúdo dos objetos apontados;

Pronomes: um *pronome* é um tipo de *k-line* que pode ser usado como uma “variável.” Um *pronome* consiste em uma coleção de apontadores para outros objetos, que se tornam acessíveis através do mesmo. Além disso, um *pronome* pode ter associado a ele um reconhecedor que deve ser testado quando ocorre uma atribuição, de modo que o *pronome* só pode apontar para objetos que satisfazem determinadas características;

Frames: um *frame* consiste basicamente em uma coleção de *pronomes* chamados terminais, e possivelmente algum mecanismo adicional que define como ativar estes terminais; em um *script*, por exemplo, que é um tipo de *frame*, os terminais devem ser ativados sequencialmente. Um *frame* pode simplesmente relacionar objetos; neste caso, quando o *frame* é ativado, todos os seus terminais são ativados;

Paranomes: um *paranome* é semelhante a uma *k-line*, mas conecta objetos “maiores,” como *frames*, uns aos outros, de modo que quando um objeto é ativado, os outros também o são.

O *blackboard* também pode ser dividido verticalmente (em painéis), mas decidimos não incluir uma tal divisão na arquitetura por não ser genérica o suficiente. Em outras palavras, divisões verticais podem ser estabelecidas para aplicações particulares.

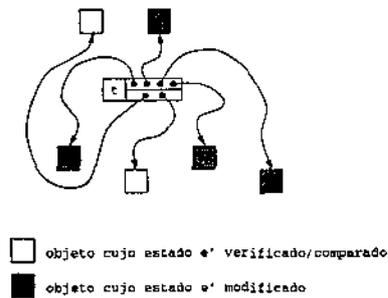


Figura 4.5: Um tipo de reconhecedor.

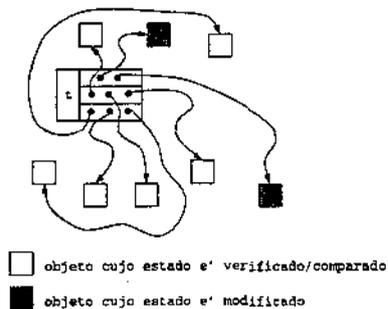


Figura 4.6: Outro tipo de reconhecedor.

Os agentes aos quais na versão anterior eram associados os módulos de memória compartilhada distribuída contendo objetos de coordenação podem continuar sendo agentes, ou podem ser implementados como *threads* do próprio *blackboard*, que idealmente consiste em uma memória compartilhada distribuída *multithreaded*. Neste caso, o *blackboard* não é mais apenas um meio de comunicação, ele também passa a ser uma estrutura de dados “ativa,” com uma funcionalidade própria.

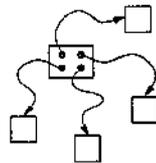


Figura 4.7: Uma k-line.

Um sistema baseado nesta arquitetura funciona da seguinte forma: o *blackboard* contém

todos os dados; um certo número de agentes (que podem ser *threads* do *blackboard*, como vimos anteriormente) inspeciona continuamente o estado dos objetos do *blackboard*, atualizando o estado de outros objetos adequadamente; e os agentes correspondentes às agências atômicas processam os dados específicos da aplicação, possivelmente utilizando também objetos de coordenação, como *pronomes*, para acessá-los. Assim, na versão final da arquitetura, temos que cada agente de coordenação corresponde a um objeto no *blackboard*, e estados mentais (parciais) correspondem a configurações (parciais) do *blackboard*.

4.4 Aplicações

A arquitetura que apresentamos possui as seguintes características: ela inclui um certo número de componentes (agentes) encapsulados e independentes que são as agências atômicas, e todas estas operam sobre uma estrutura do tipo *blackboard* que constitui o meio de comunicação e controle, e que implementa os estados mentais parciais da Sociedade da Mente e os meios de coordenação dos mesmos. Os agentes do sistema (agências atômicas) observam estados mentais parciais e os utilizam, ou os criam, ou criam/modificam os próprios objetos no *blackboard*. Desta forma, a criação dinâmica de objetos de coordenação é uma característica essencial do funcionamento do sistema, embora a criação dinâmica de agências atômicas não o seja. Podemos especular que esta criação e modificação de objetos no *blackboard* constitua o principal mecanismo de desenvolvimento, aprendizado ou evolução de um sistema baseado nesta arquitetura. Algumas das possíveis aplicações que descreveremos ilustram isto.

A arquitetura pode ser estendida sem perder estas características, acrescentando novos mecanismos de coordenação no *blackboard*, que podem ser mecanismos adicionais da Sociedade da Mente—por exemplo, diferentes tipos de *frames*—, ou variantes destes, ou mesmo mecanismos completamente novos, mas com a mesma filosofia.

Como vantagens desta arquitetura, podemos dizer que ela reúne características do modelo de *blackboard* clássico—ou seja, onde o *blackboard* é ao mesmo tempo o meio de comunicação e a “área de trabalho” na qual soluções ou resultados são progressivamente construídos através do esforço cooperativo dos agentes do sistema—com características do modelo da Sociedade da Mente, em especial o fato de que os mecanismos de coordenação atuam de modo a integrar, relacionar e estabelecer relações entre fragmentos de conhecimento. Conforme dissemos no segundo capítulo, esta é justamente uma característica da Sociedade da Mente que tenta contornar uma deficiência comum na Inteligência Artificial simbólica. Uma vez que aspectos conexionistas não fazem parte desta arquitetura, não faz sentido falar nas características da Sociedade da Mente que tentam contornar deficiências dos modelos conexionistas; no entanto, o fato de a arquitetura não prever nada de conexionismo não significa que não seja possível montar uma aplicação “híbrida” usando esta

arquitetura e dando às agências atômicas, por exemplo, um caráter conexcionista.

Existe uma característica adicional do modelo da Sociedade da Mente, que vem a ser o fato de que diferentes agentes podem operar com diferentes representações, estratégias, algoritmos etc., e é esta característica que subjaz ao comportamento inteligente. Esta característica também está associada tanto ao modelo de *blackboard* como ao paradigma de IAD; no entanto, consideramos que isto merece alguns comentários. Nem toda decomposição, ou particionamento, através da qual um sistema pode ser constituído por agentes que operam sinergisticamente se adequa à forma pela qual a cooperação entre agentes dá origem à inteligência no modelo da Sociedade da Mente. Brooks [3] estabelece uma diferença entre uma decomposição *funcional* e uma decomposição *comportamental*. Uma decomposição funcional decompõe uma tarefa nas funções que a compõem, enquanto uma decomposição comportamental decompõe uma tarefa em comportamentos completos, mesmo que tenham componentes redundantes. Um exemplo de decomposição funcional em Robótica seria: percepção, modelamento, planejamento, execução, controle motor; enquanto que um exemplo de decomposição comportamental seria: raciocínio sobre objetos, construção de mapas, explorar, perambular, evitar obstáculos etc. [3]. Consideramos que uma decomposição comportamental provavelmente é a mais adequada para sistemas baseados na arquitetura que descrevemos, uma vez que as agências da Sociedade da Mente constituem comportamentos e não funções. Uma vantagem que Brooks aponta para este tipo de decomposição é que permite o desenvolvimento incremental: quanto mais comportamentos (agências), maior a competência do agente inteligente, o que ao mesmo tempo facilita a construção e a depuração de tais agentes.

A seguir descreveremos algumas possibilidades de aplicações que ilustram os aspectos citados anteriormente.

4.4.1 Um agente de senso comum

Uma das áreas de pesquisa mais comentadas em IA atualmente é a que trata da representação e manipulação do chamado “senso comum.” O pioneiro nesta área tem sido Douglas Lenat, cujo projeto CYC [8] vem se dedicando à construção de uma base de conhecimentos abrangente de senso comum desde 1984.

O conhecimento de senso comum corresponde, informalmente, àquilo que “todo mundo sabe”; é o conhecimento que uma pessoa adquire à medida que vive e toma contato com o mundo. Podemos falar também em raciocínio de senso comum, que são os métodos através dos quais fazemos inferências e predições que nem sempre são válidas mas que servem para as situações do dia-a-dia. O chamado senso comum é a combinação do conhecimento de senso comum com os métodos de raciocínio de senso comum, sendo que não é sempre claro o que é uma coisa e o que é outra. Na literatura, muitas vezes encontramos os termos “naïve” e “folk” para designar o conhecimento de senso comum relativo a uma

determinada área; por exemplo, “Naïve Physics,” “Folk Psychology” etc. Em português poderíamos dizer “Física Intuitiva” etc.

Acredita-se atualmente que é o senso comum que permite às pessoas exibirem um comportamento “inteligente,” deparando-se com situações cotidianas e agindo, tomando decisões etc. de forma adequada, se não infalível, dentro de limites de tempo razoáveis. Incorporar senso comum em agentes não serviria somente para simular o comportamento inteligente, no entanto, mas permitiria também um tratamento computacional mais adequado de diversos problemas, entre os quais podemos mencionar: compreensão de linguagem natural, buscas orientadas por associações de significados e outros.

Consideramos que um agente de senso comum deve ser construído de modo a não apenas conter uma grande base de conhecimentos e ser capaz de consultá-la para obter informações, mas também de modo a poder incorporar novos conhecimentos a esta base.

Um agente de senso comum seria um bom exemplo de aplicação para a arquitetura que desenvolvemos, pelos seguintes motivos: primeiro, o conhecimento de senso comum é caracterizado precisamente pelo inter-relacionamento entre uma grande quantidade de fragmentos de conhecimento, e é exatamente isto que os mecanismos de coordenação da Sociedade da Mente permitem fazer; segundo, o raciocínio de senso comum se adequa bem ao esquema de agências e decomposição comportamental, sendo que, na verdade, foi precisamente este tipo de raciocínio não-especializado do dia-a-dia que inspirou o modelo da Sociedade da Mente.

A base de conhecimentos de senso comum corresponderia ao conteúdo do *blackboard*. É importante lembrar que a arquitetura prevê que o *blackboard* seja implementado utilizando memória compartilhada distribuída e, neste caso, seria conveniente utilizar recursos mais sofisticados ainda, como replicação, permanência (cópia em disco) etc.

A maneira de representar o conhecimento usando os recursos da arquitetura seria definida a partir de uma ontologia, como por exemplo a ontologia geral proposta em [23] e que inclui: categorias e taxonomias; medidas, unidades e comparações entre elas; objetos compostos, suas partes e relações entre elas; tempo, espaço e mudança; eventos e processos; objetos físicos; substâncias; objetos mentais, crenças, desejos, intenções etc. A cada um destes itens corresponderiam certos tipos de objetos no *blackboard* e algumas agências atômicas responsáveis pela sua manipulação.

Por enquanto, o que descrevemos corresponde ao “núcleo” do agente, ou seja, representações de conhecimento e mecanismos de raciocínio de senso comum. As agências atômicas, ou agentes internos, que descrevemos até agora não são diretamente acessíveis, e sim através de outros agentes, que constituem a interface do agente de senso comum com o ambiente externo. Estes podem ser divididos em dois grupos: um grupo formado pelos agentes que correspondem a ferramentas de inserção de conhecimento, e outro grupo formado por agentes que respondem a diversos tipos de consulta.

As ferramentas de inserção do conhecimento seriam agentes através dos quais seria possível “transmitir” ao agente fragmentos de conhecimento. Cada um destes agentes poderia ser responsável por um ou mais aspectos particulares da ontologia escolhida. Por exemplo, um agente poderia ser usado para inserir conhecimento sobre coisas e atributos, outro para inserir fragmentos de conhecimento sobre medidas, outro para inserir conhecimento sobre categorias e assim por diante.

Suponhamos, por exemplo, que vamos inserir um fragmento de conhecimento sobre o objeto “laranja.” Para isto, usamos um agente que permita inserir o nome de um objeto e seus atributos. Objetos, conceitos, atributos etc. podem ser representados através de palavras em linguagem natural ou em alguma representação interna que possa ser manipulada pelos agentes internos e pelos agentes de interface e traduzida por estes em outras representações adequadas. Vamos supor aqui que os objetos de dados são cadeias de caracteres correspondentes a nomes legíveis. Assim, o fragmento de conhecimento relativo ao conceito de “laranja” poderia ser informado ao agente da seguinte forma:

LARANJA

cor: cor de laranja
forma: redonda
tamanho: aproximadamente 8cm de diametro
textura: rugosa
material: vegetal
peso: aproximadamente 200g
outros: comestivel, sabor de laranja etc.

A inserção destes fragmentos de conhecimento utilizando os agentes adequados constitui apenas o princípio do processo de incorporação do novo conhecimento ao agente. As fases seguintes deste processo envolvem a análise do novo fragmento de conhecimento e a confrontação dos resultados desta análise com todo o conhecimento que o agente já possui, de modo a estabelecer as relações apropriadas entre eles, com a participação dos agentes internos que descrevemos anteriormente. Neste processo, uma representação inicial, provisória, construída pelo agente-ferramenta a partir dos dados inseridos pelo usuário, é substituída por uma representação adequada para ser incorporada ao *blackboard*, e que inclui conexões com outros objetos ali já existentes. Como subprodutos do processo, também podem ocorrer modificações de representações existentes no *blackboard*. Isto pode ocorrer, por exemplo, quando a adição de um elemento em uma *acumulação* leva o agente a abstrair atributos destes elementos (*uniframing*) formando um *conceito* que engloba todos os elementos da acumulação. Neste caso, pode ser necessário consultar o usuário quanto à adequação ou não das modificações que o agente sugere fazer.

No exemplo anterior, uma representação de LARANJA e de seus atributos poderia fazer com que um reconhecedor ativasse o conceito de FRUTA. O agente “perceberia” isto

e perguntaria ao usuário se uma LARANJA é mesmo uma FRUTA. No caso, o usuário responderia que sim, e o agente construiria as ligações apropriadas para que a ativação do conceito de FRUTA ativasse fracamente o conceito de LARANJA como uma de suas instâncias. Por outro lado, se o conceito de GOLFINHO estivesse sendo inserido, o agente poderia perguntar se um GOLFINHO é um PEIXE, ao que o usuário deve responder que não, causando a construção das ligações apropriadas para que o fato de o GOLFINHO ter muitas características de PEIXE não ativasse erroneamente esta como sua categoria.

Como podemos observar, a representação de categorias está intimamente relacionada com a representação de coisas e seus atributos. Este inter-relacionamento, como observamos anteriormente, é típico do conhecimento de senso comum, e constitui um aspecto essencial deste. Por isso, além de agências atômicas que manipulam representações específicas, faz-se necessário ter também agências atômicas que utilizam representações diferentes.

De modo geral, o processo de incorporação do conhecimento envolve a criação de objetos de diversos tipos no *blackboard* e o estabelecimento de relações entre estes e os anteriormente existentes através de reconhecedores, *k-lines* etc. Podemos pensar neste processo como um tipo de “aprendizado” para este agente. Entretanto, os processos e a organização do conhecimento que descrevemos não têm necessariamente nada a ver com o modelamento de processos cognitivos reais; eles são meramente mecanismos de construção e manipulação de uma base de conhecimentos computacional, em uma abordagem muito semelhante à do projeto CYC, com a diferença de que identificamos o conjunto formado pela base de conhecimentos e pelos métodos de raciocínio que a manipulam com um agente baseado na Sociedade da Mente.

Este agente é um caso onde uma divisão vertical do *blackboard* pode ser conveniente. Esta divisão seria em duas partes: uma base de conhecimentos propriamente dita, comparável a uma memória de longo prazo, e uma “área de trabalho,” que pode ser vista como uma memória de curto prazo na qual o novo conhecimento é inicialmente colocado para então ser incorporado àquela, sendo que neste processo a representação inicial é transformada em uma representação mais adequada e conectada a outros fragmentos de conhecimento.

4.4.2 Um agente de busca bibliográfica

O agente de busca bibliográfica é um agente que gerencia uma coleção de publicações em algum formato eletrônico e pode ser consultado por um usuário para obter uma lista de publicações sobre um ou mais assuntos. Um agente deste tipo poderia simplesmente comparar os assuntos fornecidos pelo usuário para a busca com uma lista interna de assuntos e retornar as publicações que se referissem a estes, mas isto seria uma mera busca léxica. Seria preferível que o agente fosse capaz de associar assuntos semelhantes;

assim, por exemplo, se o usuário pedisse publicações relacionadas ao assunto “sociedades de agentes,” o agente saberia que este assunto é relacionado a “sistemas multi-agentes” e retornaria publicações sobre isto, entre outras.

Este agente também se caracteriza por fragmentos de dados que devem ser relacionados. No caso, estes dados correspondem a dados bibliográficos e assuntos. A associação de assuntos é um processo de reconhecimento; daí a adequação a esta arquitetura.

Um dos agentes internos processa textos à medida que estes são incorporados à coleção de publicações. Este processamento resulta em uma série de objetos de dados, que são: títulos, autores e palavras-chave extraídas do título, do *abstract* e das próprias palavras-chave eventualmente fornecidas no texto ou obtidas de outras partes do texto. Além disso, para cada texto é construído um ou mais *frames*. Os tipos dos *frames* poderiam ser:

```
<LIVRO> ou <PUBLICACAO>
  <TITULO>
  <AUTORES> ou <EDITORES>
  <ASSUNTO>
  <CAPITULOS> ou <ARTIGOS>
```

```
<CAPITULO> ou <ARTIGO>
  <TITULO>
  <AUTORES>
  <ASSUNTO>
```

Cada terminal dos *frames* aponta para uma *k-line* que por sua vez aponta para os objetos de dados ou *frames* pertinentes. Assim, por exemplo, um livro intitulado T cujos editores são E_1 e E_2 , cujo assunto é descrito pelas palavras-chave p_1, p_2, p_3, p_4 e p_5 , e que é formado por dois capítulos intitulados C_1 e C_2 , cujos autores são A_1 e A_2 e cujos assuntos são descritos por p_1, p_2 e p_3 , e p_1, p_4 e p_5 , respectivamente, é representado conforme mostrado na figura 4.8.

Cada *frame* deve conter também as referências bibliográficas completas do texto a que se refere em uma forma que possa ser apresentada ao usuário ao final da busca; alternativamente, o *frame* pode conter uma referência ao arquivo, de modo que o texto possa ser fornecido ao usuário diretamente.

Quando um *frame* é criado, também podem ser criados *paranomes* que conectam o *frame* a outros *frames* que tenham assuntos suficientes em comum. Reconhecedores conectam objetos de dados a *frames* de modo que um *frame* seja ativado quando um número suficiente destes objetos de dados seja ativado, e também conectam objetos de dados a outros objetos de dados.

Quando o usuário quisesse fazer uma busca, ele forneceria uma lista de palavras-chave e nomes de autores que seriam então comparadas por um agente com objetos de dados,

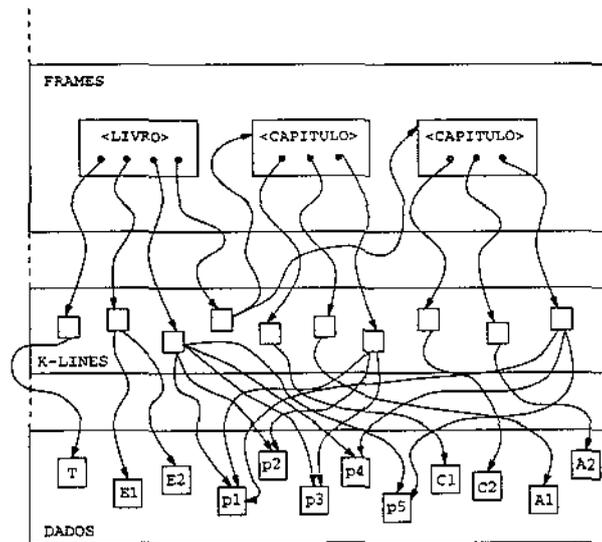


Figura 4.8: Representação de um livro.

ativando os que forem semelhantes (iguais, sinônimos etc.). Isto por sua vez faria com que reconhecedores ativassem outros objetos, até que algum *frame* fosse ativado. Os objetos de dados seriam conectados internamente de acordo com seu significado; por exemplo, o objeto que representa o assunto “sistema multi-agentes” estaria conectado ao objeto que representa o assunto “sociedades de agentes” através de um reconhecedor.

Estas conexões poderiam ser estabelecidas de várias maneiras. Uma possibilidade seria exigir que a entrada de dados fosse feita por um ser humano que estabelecesse conexões entre os novos dados e os dados já existentes no sistema através de uma ferramenta de visualização adequada, mas é uma solução inconveniente porque a quantidade de dados tende a se tornar grande e portanto difícil de visualizar. Outra possibilidade é que o agente trabalhe em estreita colaboração com um agente de senso comum ou uma versão restrita deste que o auxilia a organizar o conhecimento, indicando quais assuntos são relacionados, sinônimos, relações todo-parte etc. Este seria um exemplo de como *dois* agentes baseados na Sociedade da Mente podem ser *integrados* (figura 4.9).

O funcionamento interno do processo de recuperação de textos a partir de assuntos e autores ocorre através dos reconhecedores que conectam objetos de dados a outros objetos de dados relacionados, como sinônimos, ou o nome de um autor a suas principais áreas de pesquisa etc.—e objetos de dados a *frames*. Os objetos de dados a serem reconhecidos para ativar um *frame* incluem aqueles para os quais seus terminais apontam. Os reconhecedores devem atribuir pesos diferentes às suas entradas de acordo com sua relevância. O objetivo, novamente, é partir da ativação de um conjunto de objetos de dados que, ou foram

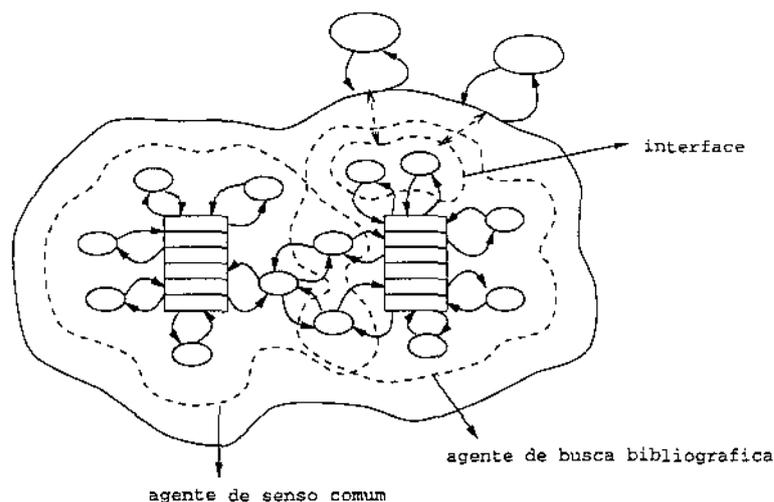


Figura 4.9: Agentes de busca bibliográfica e de senso comum integrados.

fornecidos pelo usuário como entrada, ou foram inferidos destes—com o auxílio do agente de senso comum, por exemplo—e chegar à ativação de um ou mais *frames*, cujos títulos e autores e demais referências podem ser então apresentados ao usuário como resultado da busca.

4.4.3 Um agente que conta uma história

A possibilidade de organizar, estruturar e conectar informações através dos mecanismos próprios da arquitetura do agente permite pensar em utilizar um agente desse tipo para navegar em uma história ou em um ambiente estruturado semelhante. Uma história consiste em uma coleção de informações cuja estrutura mais marcante é a narrativa, ou seja, a seqüencialidade dos eventos que a compõem, complementada por outros tipos de estruturas e conexões entre as informações, como simultaneidade, relações entre eventos e personagens, entre personagens e outros personagens, eventos e lugares etc. Estas ligações entre os dados estabelecem possíveis caminhos para exploração. Um agente baseado na Sociedade da Mente poderia orientar um usuário na exploração destes caminhos, exibindo as informações conforme fossem sendo encontradas e sugerindo alternativas de exploração de acordo com o contexto. Para isto, as informações subjacentes à história são organizadas como objetos de dados no *blackboard* conectados de forma tal que a ativação de certos dados causa a ativação de dados relacionados. O estado ativo desses objetos é interpretado como o “foco de atenção” do agente, correspondendo ao contexto cujas informações o usuário pode explorar no momento. À medida que este escolhe alternativas para explorar,

o foco de atenção do agente vai se modificando de modo a acompanhar a exploração do usuário. O resultado seria um tipo de navegação semelhante à de um hipertexto, mas com a diferença de que os *links* não são codificados, mas emergem a partir dos assuntos que são evocados durante o próprio processo. Desta forma, o agente “conta” uma história para o usuário de maneira semelhante a uma pessoa contando uma história para outra, onde esta solicita informações adicionais conforme o que a primeira vai lhe dizendo.

A mesma idéia pode ser utilizada para tratar outros tipos de coleções estruturadas de informações além de histórias. Um exemplo seria um museu virtual no qual o usuário pudesse explorar informações relativas a artistas e suas obras. Os períodos históricos a que os artistas pertenceram e a cronologia de suas obras corresponderiam à estrutura narrativa da história. Temas, técnicas, estilos, influências, escolas etc. proveriam conexões adicionais entre os dados.

O agente que conta uma história foi o escolhido para implementação a fim de realizar uma experiência com a arquitetura que desenvolvemos; por isso, ele é melhor descrito, juntamente com aspectos da implementação, no próximo capítulo.

4.4.4 Sociedades de agentes baseados na Sociedade da Mente

Até aqui, estivemos descrevendo agentes isolados baseados na Sociedade da Mente, e isto porque a arquitetura que projetamos é a arquitetura de *um* agente inteligente. Mas podemos pensar também em sistemas (ou ambientes) multi-agentes compostos por agentes baseados na Sociedade da Mente. Assim, em lugar de termos um único agente que orienta o usuário na exploração de um museu virtual, podemos ter vários agentes que enfatizam diferentes conexões entre as informações, como se possuíssem “estilos” ou até mesmo “personalidades” diferentes. Se tivéssemos diversos agentes que contam uma história em lugar de um único, as relações que cada um deles vê entre os personagens, por exemplo, poderiam ser diferentes, e mesmo a seqüencialidade da história, da mesma maneira que pessoas diferentes narram uma série de acontecimentos de formas naturalmente diferentes.

4.5 Comparação com a arquitetura M

Outra arquitetura que também utiliza idéias da Sociedade da Mente é a arquitetura do sistema M, ou arquitetura M [21]. Outra característica que esta arquitetura apresenta em comum com a que apresentamos é que um único agente (ou “assistente,” na terminologia de M) é constituído por uma coleção de agentes, ou seja, ambos constituem sistemas multi-agentes.

O objetivo do sistema M é dar assistência a um usuário que trabalha junto com outros em uma sala virtual de conferências ou *Virtual Meeting Room* (VMR). O sistema atua

tentando reconhecer, classificar, armazenar, recuperar e apresentar informações relacionadas com a interação humano-computador no ambiente VMR, no qual cada usuário possui um assistente deste tipo. O ambiente é composto de objetos como documentos eletrônicos, imagens, marcadores, quadros etc. Desta forma, um assistente M tem um objetivo bem definido, que é auxiliar o usuário de um ambiente de trabalho cooperativo, o VMR, a organizar suas coisas. O assistente M relaciona objetos uns com os outros através de diferentes tipos de raciocínio: espacial, estrutural (relativo às partes que compõem um objeto), funcional (relativo a como o objeto funciona), temporal, causal (relativo a eventos e mudanças de estado), baseado em explicações e baseado em casos. Os cinco primeiros correspondem a agentes que trabalham com quatro estruturas: um sistema de *scripts*, um sistema baseado em regras, uma rede semântica e um conjunto de *blackboards*. A rede semântica serve para representar atributos de objetos; objetos são então representados por *k-lines/polynemes* que conectam e ativam os atributos adequados nesta rede semântica. Cada *blackboard* contém *trans-frames* e *pronomes* que representam ações e atributos de objetos relacionados a estas ações. O sistema baseado em regras ao mesmo tempo estabelece fatos a respeito do ambiente e dirige a atuação dos agentes. O sistema de *scripts* contém *scripts* que permitem aos agentes avaliar as teorias representadas pelos vários *blackboards*. Cada *blackboard* contém uma “teoria” emergente formada por classificações de objetos e ações, e através dos *scripts* eles são ordenados em um *ranking*.

Desta forma, o agente (ou assistente) M é construído tendo em vista um objetivo, e para isto utiliza diversas técnicas, entre elas algumas inspiradas nos mecanismos da Sociedade da Mente. Estas técnicas teriam sido escolhidas por se adequarem ao objetivo do agente. Assim, a representação dos objetos do VMR como *k-lines/polynemes* conectando atributos representados em uma rede semântica estaria relacionada ao fato de que a representação de atributos dos objetos do VMR é relevante para o tipo de tarefa que o agente realiza.

A arquitetura que apresentamos é uma arquitetura genérica, ou seja, não está relacionada a um objetivo ou problema específico. Assim, enquanto M combina idéias da Sociedade da Mente com outras coisas para atingir seu objetivo, para nós o objetivo é, ao contrário, incorporar idéias da Sociedade da Mente. Esta abordagem deixa muita coisa a ser definida caso se pretenda pensar em aplicações específicas, mas isto constitui, de certo modo, uma flexibilidade esperada. Nossa intenção é que a arquitetura apresentada seja geral o suficiente para permitir pensar em aplicações bastante diversas, como tentamos mostrar na seção anterior.

Finalmente, em ambos os casos existe uma “coesão” que faz com que o sistema M seja chamado de um “assistente” e o nosso seja chamado de agente. Isto é bastante condizente com o modelo da Sociedade da Mente, no qual a “sociedade da mente” é uma *sociedade* de componentes que formam *uma* mente e não várias.

4.6 Conclusão

Neste capítulo descrevemos o desenvolvimento de uma arquitetura para agentes inteligentes baseada na Sociedade da Mente [13]. A arquitetura de um agente pode ser dividida em uma parte de funcionalidade e uma parte de interação com outros agentes. No caso, como estamos pensando em um agente isolado, a arquitetura se refere à parte da funcionalidade do agente. Nosso objetivo era desenvolver uma arquitetura de um agente que refletisse aspectos do modelo da Sociedade da Mente e para isto decidimos projetar a arquitetura de um agente que é, ele mesmo, um sistema multi-agentes composto por agentes menores. A arquitetura foi descrita a partir de uma perspectiva de seu desenvolvimento ou evolução, ou seja, descrevemos como aspectos do modelo da Sociedade da Mente foram progressivamente transformados em componentes ou propriedades de uma arquitetura de Inteligência Artificial Distribuída. Em seguida, descrevemos algumas possibilidades de aplicações e mencionamos o fato de que podemos projetar um sistema multi-agentes composto por agentes construídos de acordo com esta arquitetura. Finalmente, fizemos uma comparação entre a arquitetura de um agente baseada na Sociedade da Mente e a arquitetura *M* de agentes integrados [21], que é a arquitetura de um assistente computacional composto por agentes menores e que utiliza alguns conceitos da Sociedade da Mente combinados com outros como redes semânticas, raciocínio baseado em casos etc.

O próximo capítulo trata da implementação de um agente que conta uma história, conforme descrito na seção 4.4.3. Nele descreveremos com mais detalhes os princípios de funcionamento do agente, que se baseiam nos mecanismos da Sociedade da Mente incorporados na arquitetura, entre os quais se destacam a possibilidade de conectar fragmentos de informação e o *loop* de memorização, usado para trazer à tona assuntos relacionados dentro da história que o agente conta. Descreveremos também a implementação realizada e discutiremos extensões e outros aspectos.

Capítulo 5

A implementação de um agente

“L’informatique autorise aujourd’hui des recherches beaucoup plus élaborées qui n’ont pas encore été explorées. La lecture interactive sur terminal à écran renouvelle en particulier les données du problème. Les hypertextes (matrices de textes potentiels) écrits pour la lecture interactive se présentent comme des systèmes de parcours multiples. En 1987, le lecteur ne peut y intervenir qu’en choisissant des embranchements au sein du récit. Mais des systèmes capables de dialoguer avec le lecteur, et mettant à sa disposition de multiples outils pour travailler sur une proposition initiale ou une collection de données, sont concevables grâce aux progrès de l’intelligence artificielle et des systèmes experts. La distinction entre auteur et lecteur pourrait ainsi s’effacer pour certains genres littéraires. Dans nombre d’activités économiques et sociales, cette distinction subira sans doute le même sort au profit d’un réseau anonyme de production. Des milieux informationnels et des bases de ressources seront les conditions de possibilité des parcours de lecture et de la génération de textes. Les équipes chargées de les construire et de les aménager assumeront-elles la responsabilité de cette écriture au deuxième degré, l’écriture des possibles ?”

Pierre Lévy, la Machine univers: création, cognition et culture informatique, p. 14 [10]

5.1 Introdução

No capítulo anterior, descrevemos o desenvolvimento de uma arquitetura para agentes inteligentes baseada na Sociedade da Mente e apresentamos alguns exemplos de agentes que poderiam ser construídos usando a mesma. Entre estes está um agente que “conta uma história” orientando o usuário na exploração de uma vasta coleção de informações relacionadas à história. Neste capítulo, descrevemos a implementação de um agente deste tipo.

O agente que conta uma história é baseado em uma idealização de um diálogo no qual uma pessoa conta uma história a outra de forma interativa. Neste diálogo, o indivíduo que conta a história essencialmente fornece informações solicitadas pelo outro, que faz perguntas de modo a selecionar as informações que o primeiro lhe apresenta. Assim, em um diálogo típico, o narrador começa falando de algum evento que tenha acontecido, e o outro faz perguntas do tipo: “e o que aconteceu então?,” “mas por que isto aconteceu?,” “quem é X” (onde X é um personagem) etc. É claro que uma conversa normal na qual podemos dizer que uma história é narrada de modo algum se resume a este tipo de diálogo, que constitui somente uma abstração das partes do diálogo que estão estritamente relacionadas a contar a história; ademais, podemos dizer que este tipo de diálogo nunca ocorre isolado na vida real, mas sempre combinado com ou inserido em outros tipos de diálogo. O que nos interessa aqui no entanto são apenas as características deste diálogo idealizado no que se refere à *interação* entre os dois indivíduos, o que conta a história e o seu interlocutor, e o *tipo de informação* que este diálogo transporta.

O que podemos observar que ocorre em um tal diálogo idealizado e restrito é que, de modo geral, a pessoa que ouve quer saber da pessoa que conta o que aconteceu antes, ou o que aconteceu depois, ou quer informações adicionais sobre os personagens envolvidos, ou sobre algum objeto relevante, ou ainda sobre os lugares que o outro menciona. Desta forma, o *tipo de informação* que o diálogo transporta compreende: a narração de um acontecimento, a descrição de um personagem, a descrição de um objeto, a descrição de um lugar e descrições de relacionamentos entre quaisquer desses itens. A *interação* se dá da seguinte forma: o narrador apresenta informações sobre um acontecimento, personagem, objeto ou lugar; o interlocutor solicita informações adicionais a respeito de algo que tenha sido mencionado; o narrador apresenta a informação solicitada, e assim por diante.

Para projetar o agente que conta uma história para um usuário humano, decidimos transpor o padrão descrito anteriormente para outro *modo de interação*. Enquanto em um diálogo entre duas pessoas o modo de interação usual é o verbal, no caso do agente optamos por um modo de interação que envolve texto, imagens, *buttons* e menus. Assim, o agente apresenta informações através de texto e imagens e o usuário informa o agente sobre o que ele quer saber sobre a história através de *palavras-chave* que podem ser apresentadas sob a forma de *buttons*, menus, listas ou outros recursos de interface semelhantes a estes. As palavras-chave entre as quais o usuário pode escolher dependem das informações anteriormente apresentadas pelo agente, de modo que existe uma restrição relativa às informações que o usuário pode solicitar.

Por exemplo, suponhamos que o agente tenha acabado de descrever, através de texto e figuras, uma cena na qual os personagens A, B e C estão planejando uma trama. Decidimos explorar informações sobre o personagem A. O agente apresenta então algumas informações biográficas básicas sobre A e pode apresentar, digamos, informações adicio-

nais detalhando a participação deste personagem em particular na trama. Depois podemos escolher explorar as informações relativas ao personagem B e, como acabamos de explorar aquelas relativas a A, o agente pode mostrar, além das informações básicas sobre B, informações adicionais sobre o relacionamento entre A e B e sobre a participação de B na trama.

Uma história apresenta as seguintes características que a tornam apropriada para uma abordagem deste tipo:

- existe uma *seqüencialidade*, ou seja, um encadeamento de eventos. Esta seqüencialidade em geral repousa sobre uma ordem cronológica subjacente, mas também pode ser estabelecida em função de relações “menos objetivas” de causa-e-conseqüência, por exemplo. (Relações “menos objetivas” trazem a possibilidade de ter agentes com “pontos de vista” diferentes.)
- os eventos que compõem a história são protagonizados por personagens que além disso podem ter outros relacionamentos entre si.
- a história se passa em lugares que usualmente também são relacionados entre si geograficamente. Em particular, um acontecimento pode se passar em mais de um lugar ao mesmo tempo: por exemplo, algo pode ocorrer na “escada” que fica na “casa” que fica na “floresta” etc.
- objetos concretos ou abstratos também podem desempenhar papéis importantes nos acontecimentos.
- finalmente, todos estes itens também são relacionados uns com os outros: cada acontecimento envolve certos personagens, certos objetos, e se passa em certos lugares; personagens podem ser relacionados a determinados lugares, por exemplo, os lugares onde vivem; objetos podem pertencer a personagens ou podem estar localizados em lugares.

Assim, podemos ver que uma história é composta por uma coleção de informações relacionadas entre si. O agente orienta o usuário na exploração destas informações, onde um assunto evoca outros assuntos relacionados dentro da estrutura da história. Nós utilizamos características do modelo da Sociedade da Mente transpostas para a arquitetura do agente de modo a espelhar estes requisitos para seu funcionamento. Deste modo, utilizamos objetos de dados para conter as unidades de informação que o agente manipula, *k-lines* para agrupar unidades relacionadas ao mesmo assunto, reconhecedores para estabelecer conexões entre diferentes assuntos e entre partes de um mesmo assunto e *frames* para representar a estrutura da história. O processo pelo qual reconhecedores evocam estados mentais parciais a partir de outros estados mentais parciais é usado para evocar

informações com potencial para serem exploradas, e o mecanismo do *loop* de rememoração é usado para trazer assuntos completos para o “foco de atenção” do agente a partir das informações evocadas no processo anterior.

Um agente deste tipo é mais adequado para apresentar uma história que seja complexa e multi-ramificada. Em uma história muito simples, tanto a quantidade de informações como os possíveis caminhos para exploração seriam inerentemente poucos, de modo que esta abordagem poderia não trazer nenhuma vantagem. No entanto, na realidade isto depende fortemente de *como* a informação é organizada, ou seja, da “escrita dos possíveis” da epígrafe que abre este capítulo. Exemplos de histórias interessantes seriam: a saga completa de *O Senhor dos Anéis* de Tolkien, a História da Civilização Ocidental ou parte dela, a História da Arte etc.

Na próxima seção detalharemos os mecanismos mencionados no parágrafo anterior, descrevendo a organização interna do agente em termos dos objetos do *blackboard* e dos seus agentes internos. Na seção 5.3 descreveremos como este agente foi efetivamente implementado. Finalmente, discutiremos os resultados obtidos e algumas extensões que poderiam ser feitas.

5.2 A organização interna do agente

Como mencionamos na introdução, utilizamos as estruturas da arquitetura do agente para organizar as informações relativas à história e coordenar seus agentes internos de modo a obter o comportamento desejado para o agente como um todo. Podemos então descrever o funcionamento do agente que conta uma história em duas partes: a organização e o funcionamento dos objetos do *blackboard* e o funcionamento dos agentes internos. Começaremos pelos objetos do *blackboard*, uma vez que estes objetos são usados por todos os agentes internos e a diferenciação entre estes não é relevante para a descrição daqueles.

5.2.1 A organização das informações

Todas as informações relativas à história são organizadas em *itens de dados* relativamente pequenos, como textos curtos e figuras. Assim, por exemplo, cada cena da história é descrita por um pequeno número de textos e algumas figuras que representam partes da cena, ou diferentes perspectivas, de perto, de longe etc. Um dos textos pode descrever a abertura da cena, situando-a, dizendo onde se passa, quem está presente etc.; outro pode ser a descrição de um incidente; os textos seguintes podem descrever as reações dos diversos personagens presentes a este incidente, e um texto final pode sugerir que a cena tem uma continuidade, conseqüências etc., ou seja, que existem cenas posteriores.

As imagens associadas podem representar, por exemplo, o instante anterior ao incidente, o momento do incidente e as reações dos personagens após o incidente. Analogamente, personagens, lugares e objetos são descritos por coleções semelhantes de textos e imagens. Cenas, personagens, lugares e objetos constituem “assuntos” da história. De modo geral, cada assunto é representado por uma coleção de itens de dados, que são *objetos de dados* colocados na seção de dados do *blackboard*, sendo que esta pode ser subdividida de acordo com os tipos de dados utilizados (texto, imagens etc.).

Os itens de dados que descrevem uma cena, personagem, lugar ou objeto podem ser divididos em duas coleções: uma constituída de informações “essenciais” ou básicas e a outra constituída de informações incidentais ou contextuais. Por sua vez, cada conjunto de informações, seja essencial ou contextual, pode ser constituído de um ou mais itens de dados. Podemos ter, por exemplo, uma informação contextual representada por um texto e por uma figura associados. Quando uma informação consiste em mais de um item, estes são agrupados através de uma *k-line*. Itens de dados podem ser associados a mais de um assunto; por exemplo, uma figura que mostra um personagem diante do lugar onde mora pode ser associada tanto ao personagem quanto ao lugar.

Consideremos o assunto “o personagem A.” Alguns itens de dados podem apresentar sua descrição, informações biográficas etc. Estes itens de dados correspondem a informações básicas relativas ao personagem A e são conectados por uma *k-line*. Além desses, no entanto, existem outros itens de dados que não exatamente representam o personagem A, mas acrescentam informações relativas a ele e que estão relacionadas a determinados contextos. Um exemplo de informação deste tipo é a que detalha a participação deste personagem específico na trama descrita em uma determinada cena. Esta informação não pode ser considerada como relativa à cena, pois ela fala da participação de um personagem particular. Logo, ela é relativa ao personagem A. Da mesma forma que esta, podem existir muitas outras informações do mesmo tipo, ou seja, que relacionam o personagem A a alguma outra coisa, mas que são principalmente relativas a este personagem. Além disso, estas informações podem ser tanto itens de dados sozinhos como coleções de itens de dados, da mesma forma que a descrição básica do personagem, e neste caso estes itens de dados são conectados por *k-lines*. No entanto, apenas uma *k-line*—aquela que conecta itens que descrevem o personagem—pode ser considerada como realmente representando o personagem.

Além dos itens de dados, existem outros objetos de dados que são as chamadas *palavras-chave*. As palavras-chave são utilizadas na interação com o usuário e correspondem aos “assuntos” que o agente reconhece e sobre os quais ele pode exibir informações. Exemplos de palavras-chave são: “Cena 1,” “Cena 2,” “A Cena do Desaparecimento de Bilbo Baggins,” “Gandalf” (nome de um personagem), “O Castelo” (nome de um lugar). Palavras-chave podem incluir também ícones e pequenas figuras; elas consistem essencial-

mente em representações abreviadas que evocam os assuntos apropriados para o usuário, e através da seleção das mesmas estes assuntos são efetivamente evocados na “mente” do agente.

A estrutura da história é representada através de *frames*. *Frames* “narrativos” representam a estrutura seqüencial da história, e seus terminais são conectados às *k-lines* correspondentes a cenas. Outros *frames* descrevem relacionamentos entre personagens, relações geográficas entre lugares etc. Na seção 5.3 mostraremos os tipos de *frames* que utilizamos em uma implementação particular; histórias diferentes podem exigir *frames* diferentes, exceto *frames* narrativos, que não obstante podem ter diferentes complexidades (como níveis de aninhamento, múltiplas ramificações etc.).

5.2.2 As conexões entre as informações

O inter-relacionamento entre os objetos no *blackboard* é feito através de reconhecedores, que funcionam baseados no significado dos *estados* desses objetos. Em princípio, consideramos que existem três níveis de ativação: inativo, ativo e “hiperativo,” que podem ser vistos como correspondendo aos valores numéricos 0, 1 e 2, respectivamente. O significado destes estados está relacionado a uma noção intuitiva de “atenção.” De acordo com esta noção, podemos pensar nos objetos de dados ativos como sendo objetos de dados relativos a assuntos que estão no “foco de atenção” do agente, onde os demais objetos de coordenação controlam e/ou delineiam este foco de atenção. Entretanto, o fato de um item de dados estar no foco de atenção do agente não significa necessariamente que o(s) assunto(s) ao(s) qual(is) ele está associado se encontra(m) no foco de atenção. Dizemos que um assunto está no foco de atenção do agente somente se a palavra-chave associada estiver ativa.

O agente funciona essencialmente através da evolução de seu foco de atenção, que compreende itens de dados passíveis ou não de serem apresentados, dependendo da existência ou não de um número de itens ativos suficiente para trazer o assunto correspondente para o foco de atenção, segundo um processo que descreveremos mais adiante. O estado hiperativo está relacionado com a seleção de um assunto para ser efetivamente apresentado. O significado do estado hiperativo é o de um “estreitamento” temporário do foco de atenção do agente ou, em outras palavras, a ascensão momentânea de um determinado assunto a um nível de atenção consideravelmente superior ao dos demais assuntos que também se encontram no foco de atenção. Nos parágrafos seguintes, mostraremos como ocorre a evolução do foco de atenção do agente e como assuntos são momentaneamente trazidos acima do nível de atenção ativo através do funcionamento dos reconhecedores.

Os reconhecedores estabelecem ligações entre objetos, como dissemos anteriormente. De modo geral, quando determinados objetos estão ativos (ou hiperativos), isto faz com que reconhecedores modifiquem os estados de outros objetos. A diferença entre *frames* e

reconhecedores no que se refere a estabelecer relacionamentos entre dados está no próprio propósito dos dois tipos de mecanismos: reconhecedores modificam estados de objetos a partir dos estados de outros objetos, enquanto *frames* representam estruturas. Logo, são reconhecedores que ativam dados relativos a personagens que têm algum relacionamento entre si, por exemplo, mas é um *frame* que representa este relacionamento específico. Se o personagem A é filho do personagem B, digamos, um reconhecedor pode evocar um a partir do outro, mas é um *frame* que indica que um é pai e outro é filho.

5.2.3 O mecanismo de evocação de informações relacionadas

O foco de atenção do agente evolui basicamente de duas maneiras que se complementam: primeiro, assuntos evocam “impressões parciais” de outros assuntos relacionados, isto é, itens de dados pertencentes a tais assuntos; segundo, estas “impressões parciais” causam a emergência de assuntos no foco de atenção do agente. Para explicar estes dois processos vamos partir da situação na qual existem objetos de dados ativos, isto é, no foco de atenção do agente, e objetos de dados hiperativos, ou seja, também no foco de atenção do agente, mas temporariamente ocupando o centro deste foco de atenção.

Os objetos de dados hiperativos são os objetos de dados mostrados ao usuário e, portanto, são o assunto ou contexto do diálogo entre o agente e o usuário. Uma coleção de reconhecedores reage aos objetos de dados hiperativos fazendo com que outros objetos de dados se tornem ativos e também possivelmente que outros se tornem inativos. Estes reconhecedores representam relações entre assuntos *diferentes*. Em um exemplo mencionado anteriormente, o agente apresenta uma cena na qual os personagens A, B e C estão envolvidos no planejamento de uma trama. Se a cena se passa na casa de um deles, os dados correspondentes a esta cena evocam dados relativos a A, B, C e a casa. Ou seja, a partir dos dados hiperativos que descrevem a cena, reconhecedores ativam dados relativos a A, B, C e a casa, e possivelmente desativam outros dados, modificando assim o foco de atenção do agente. Desta forma, o que caracteriza o funcionamento desta primeira coleção de reconhecedores é que eles trazem novos dados para o foco de atenção, eventualmente deslocam outros dados para o foco de atenção, e mantêm outros inalterados. Neste processo, assuntos evocam outros assuntos, pelo menos parcialmente.

5.2.4 O mecanismo de evocação de assuntos

Uma segunda coleção de reconhecedores compreende aqueles que conectam itens de dados, *k-lines* e palavras-chave. Estes objetos são organizados em *loops*, conforme mostrado na figura 5.1. Um *loop* desses funciona da seguinte maneira: um reconhecedor detecta se há itens de dados relativos a um certo assunto ativos em número suficiente e, se houver, ativa a palavra-chave correspondente. Por sua vez, outro reconhecedor detecta a ativação

da palavra-chave e ativa a *k-line* que conecta os itens de dados essenciais relativos ao assunto. Através deste processo, assuntos que foram parcialmente ativados por estarem relacionados ao assunto corrente são trazidos para o foco de atenção na medida em que aquelas ativações parciais são suficientes para isto, e as palavras-chave correspondentes também são ativadas.

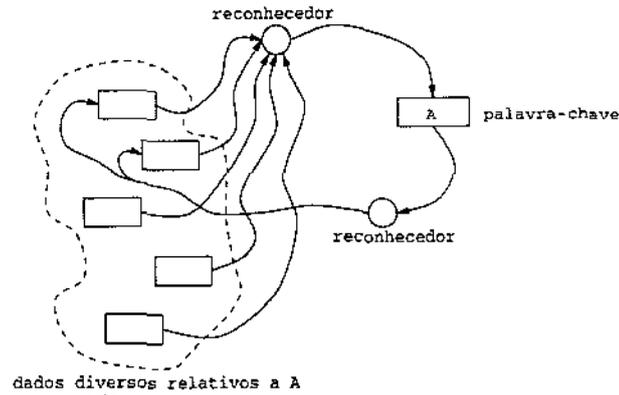


Figura 5.1: Reconhedores conectando itens de dados, *k-lines* e palavras-chave.

Ao final do processo, tem-se uma coleção de itens de dados e palavras-chave ativas. As palavras-chave ativas são então mostradas ao usuário como opções de exploração correspondentes a assuntos sobre os quais o agente “gostaria de falar,” por assim dizer.

5.2.5 O processo de evolução do foco de atenção

O processo completo, envolvendo as duas coleções de reconhedores descritas anteriormente e ainda uma terceira que iremos descrever agora, pode ser sumarizado como segue: inicialmente, existe uma coleção de itens de dados temporariamente hiperativos, que correspondem ao assunto que está sendo apresentado pelo agente. Reconhedores reagem a estes dados hiperativos ativando itens de dados relacionados. Estes itens de dados ativos atuam como impressões parciais que constituem a entrada (no sentido de *input*) para o *loop* descrito anteriormente. Reconhedores ativam palavras-chave a partir destes itens de dados. Outros reconhedores ativam *k-lines* e/ou itens de dados adicionais a partir das palavras-chave. A esta altura do processo consideramos que os dados inicialmente hiperativos não se encontram mais neste estado, uma vez que este é, por definição, um estado temporário. As palavras-chave ativas são apresentadas ao usuário, que deve escolher somente uma para ser explorada. Ao fazer isto, ele causa a hiperativação da palavra-chave escolhida, e a terceira coleção de reconhedores causa a hiperativação dos itens de dados relacionados à palavra-chave.

Neste momento, haverá uma grande quantidade de itens de dados ativos, correspondentes a todas as palavras-chave que foram oferecidas ao usuário como alternativas de exploração. O mecanismo de hiperativação é o que permite indicar, entre todos estes itens de dados, os que devem ser apresentados. Quando uma palavra-chave é selecionada, ela é hiperativada, como dissemos anteriormente. A terceira coleção de reconhedores pode responder ao estado hiperativo de uma palavra-chave de duas maneiras: hiperativando os objetos aos quais está conectado de forma *incondicional* ou então hiperativando apenas objetos que já estejam *ativos*, isto é, hiperativando objetos de forma *condicional*. A hiperativação incondicional é utilizada para informações básicas ou essenciais, enquanto a hiperativação condicional é usada para informações incidentais ou contextuais. Deste modo, uma informação que não é essencial mas sim contextual só vai ser mostrada se já tiver sido previamente ativada, ou seja, se já estiver no foco de atenção do agente, e portanto estiver no contexto adequado.

Podemos observar que uma palavra-chave desempenha um papel semelhante ao de um *polyneme* no *loop* que acabamos de descrever mas, diferentemente de um *polyneme*, ela não é conectada diretamente a *k-lines* e itens de dados, mas isto é feito através de reconhedores, sendo que alguns destes reconhedores hiperativam objetos incondicionalmente, baseados apenas no estado da palavra-chave, enquanto outros hiperativam objetos condicionalmente, tomando também o estado do próprio objeto como entrada.

Retomando o exemplo dos personagens envolvidos na trama, o que poderia acontecer seria que a hiperativação dos dados relativos à cena na qual eles planejam a trama faria com que reconhedores reagissem ativando os itens de dados que descrevem as participações de cada um dos personagens na trama, por exemplo, o personagem A. Portanto, temos um item relativo a A ativo. Suponhamos que, por A ser citado diretamente na cena, isto faz com que a descrição de A também seja ativada. Assumindo também que a ativação destes dois itens de dados seja suficiente para que o reconhedor do *loop* ative a palavra-chave correspondente ao personagem A, então o outro reconhedor reage à ativação da palavra-chave ativando a *k-line* que por sua vez ativa os itens de dados que contém a descrição básica do personagem A, e que podem vir a ser os mesmos que já estavam ativos, ou não.

Pode acontecer que poucos itens de dados sejam ativados sobre um determinado assunto em um determinado momento, e estes itens sejam insuficientes para fazer o reconhedor ativar a palavra-chave correspondente, ou seja, insuficientes para trazer o assunto para o foco de atenção; em um momento seguinte, pode acontecer de mais alguns itens relativos ao mesmo assunto serem ativados sem que outros sejam desativados, e o número de itens pode então ultrapassar o limiar de reconhecimento, levando à ativação da palavra-chave. Assim, apesar de cada um dos casos por si só poder ser insuficiente para evocar o assunto, eles causaram uma acumulação de dados ativos suficientes para evocá-lo.

O *loop* que descrevemos anteriormente constitui um *loop* de rememoração ou *reminding*, no qual um conjunto de “estímulos”—itens de dados ativados por terem alguma relação com itens de dados mostrados no momento—evoca um “simulus,” ou seja, uma “impressão mais completa” a partir daquela “impressão parcial” de um assunto. Neste caso, os reconhecedores que ativam itens de dados relacionados ao assunto corrente desempenham o papel dos reconhecedores ligados a agências sensoriais.

Finalmente, existem também no *blackboard* os *pronomes*, que estão diretamente relacionados ao funcionamento dos agentes internos e por isso serão melhor discutidos na próxima seção. Por enquanto, é suficiente notar que os *pronomes* estabelecem “contextos” ou limites para o foco de atenção de cada um dos agentes internos, determinando quais os itens de dados e quais as palavras-chave que cada um dos agentes internos é capaz de “ver.” Isto encerra a descrição do funcionamento dos objetos do *blackboard*, cujo aspecto mais importante é o *loop* que conecta itens de dados e palavras-chave.

5.2.6 Os agentes internos

O funcionamento dos agentes internos se baseia em três coisas: primeiro, no uso que estes fazem do foco de atenção; segundo, na maneira pela qual os *pronomes* definem o foco de atenção dos agentes internos; terceiro, no uso que os agentes internos fazem dos *frames*.

Cada um dos agentes internos representa um dos ângulos através dos quais o agente como um todo vê a história. Assim, um agente interno trata da seqüência de acontecimentos, outro trata das relações entre personagens, e outro trata das relações entre lugares. Poderia haver ainda outro agente que tratasse dos objetos que aparecem na história. É importante notar a diferença que existe entre os agentes *internos* e entre diferentes agentes baseados na Sociedade da Mente contando a mesma história: enquanto estes são agentes completos que podem ver a mesma história de maneiras diferentes—com uma seqüencialidade diferente, com relações diferentes entre personagens, objetos e lugares—, aqueles são *partes* de um agente destes, vêem apenas *parte* da história, e juntos formam *um* agente que conta uma história.

Isto é feito através de *pronomes* e *frames*. Cada agente tem associado a ele dois *pronomes* que delimitam o seu foco de atenção. Um dos *pronomes* se refere às palavras-chave e o outro se refere aos itens de dados. O agente só “vê” os objetos que estes *pronomes* apontam. Os focos de atenção dos agentes internos podem se sobrepor ou não. No momento em que o agente como um todo oferece palavras-chave para o usuário escolher, na verdade cada um dos agentes internos oferece as suas, e é conveniente que isto seja representado também na interface com o usuário, principalmente quando agentes internos diferentes podem sugerir a mesma palavra-chave. No caso em que dois agentes internos apresentam a mesma palavra-chave, normalmente os dados que cada um deles apresenta caso seja escolhido incluem tanto alguns itens comuns como alguns itens diferentes, porque

os agentes internos representam perspectivas diferentes, de modo que um vai dar mais ênfase aos acontecimentos, outro vai dar mais ênfase aos personagens etc.

Os *frames* mostram ao agente a estrutura subjacente da história conforme descrevemos anteriormente. Os *frames* não *relacionam* os dados, quem faz isso são os reconhecedores; eles mostram a estrutura destas relações. Os *frames* devem ser usados pelos agentes internos quando estes querem usar estas estruturas. Um exemplo de uso dos *frames* seria usá-los para apresentar as palavras-chave de modo a informar o usuário de suas relações com o assunto corrente. Cada agente tem que saber a maneira certa de usar os *frames* que lhe dizem respeito, lembrando que, por definição, grande parte do significado de um *frame* vem justamente da forma como ele é usado. Para dar um exemplo mais concreto: o agente narrativo pode usar o *frame* narrativo para mostrar um “mapa” da história, de modo que o usuário possa visualizar que parte da história ele já conhece, quais as suas ramificações etc.

Os agentes internos são coordenados pelos objetos do *blackboard*. A palavra-chave que o usuário escolhe determina qual agente interno terá precedência para apresentar informações, mas estas informações irão evocar novos assuntos que se encontram também nos focos de atenção dos outros agentes. Desta forma, existe uma cooperação implícita entre os agentes internos que é decorrente do compartilhamento do foco de atenção global; isto quer dizer que, do ponto de vista dos reconhecedores e dos demais objetos do *blackboard* (exceto *pronomes*, evidentemente, pois estes são que as determinam), não há fronteiras entre os focos de atenção dos diferentes agentes internos. Ao mesmo tempo existe uma competição também implícita, uma vez que a evolução do foco de atenção combinada com a escolha do usuário a cada vez faz com que um agente interno possa atuar.

5.3 Implementação

Nesta seção descrevemos como foi feita uma implementação de um agente que conta uma história. A história escolhida foi uma história simples e conhecida, mas que permitiu observar as características do agente. A implementação foi feita em C++ usando programação orientada a objetos e testada em sistemas Unix (Linux e SunOS). O agente consiste em um único programa e os agentes internos correspondem a diferentes objetos. Começaremos descrevendo a organização do *blackboard* e depois os agentes internos.

5.3.1 Os objetos do *blackboard*

Os objetos do *blackboard* são instâncias de classes que formam uma hierarquia (figura 5.2). O topo da hierarquia corresponde a uma classe denominada *IdentifiedObject*. Esta

classe possui apenas um membro de dados do tipo cadeia de caracteres que corresponde ao identificador ou nome dado ao objeto, e dois métodos que lêem e escrevem este identificador. As demais classes desta hierarquia são derivadas desta, o que significa que todos os objetos do *blackboard* possuem um identificador, isto é, um nome. O propósito disto é para ser usado para depuração e visualização do conteúdo do *blackboard*. A classe *ObjectWithState* acrescenta ao objeto um *estado* ou *nível* de ativação e os métodos associados. Os únicos objetos do *blackboard* que são instâncias de uma classe que não é derivada de *ObjectWithState* mas sim diretamente de *IdentifiedObject* são os reconhecedores, pois estes não possuem um estado ou nível de ativação. Os demais objetos do *blackboard* são instâncias de classes derivadas de *ObjectWithState* e portanto possuem um nível de ativação.

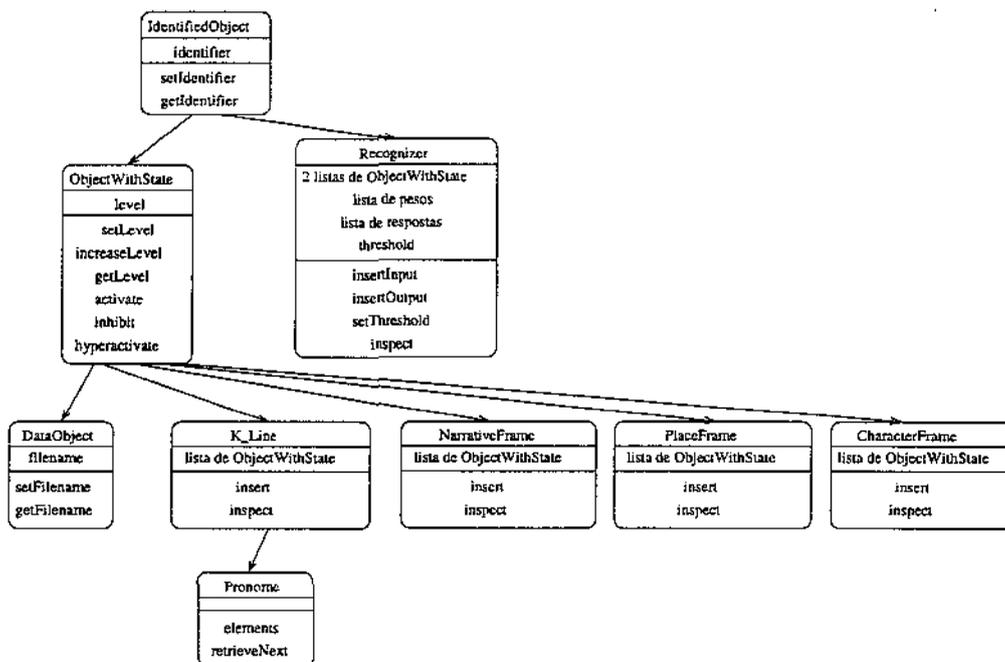


Figura 5.2: A hierarquia de classes correspondentes aos objetos do *blackboard*.

Os objetos de dados são instâncias da classe *DataObject*. A informação que os objetos deste tipo efetivamente contém é apenas o nome do arquivo onde se encontra o texto ou imagem correspondente. Desta forma, não é necessário manter todas as informações no *blackboard*, elas são lidas dos arquivos quando for necessário. Os métodos desta classe permitem ler e escrever o nome do arquivo. O nome do arquivo é escrito apenas na construção do agente ao iniciar a execução. Posteriormente, o nome do arquivo vai ser lido pelos agentes internos a fim de exibir as informações para o usuário. Como esta classe

é derivada de `ObjectWithState`, estes objetos também possuem um estado; este estado vai ser verificado e alterado pelos outros objetos, em especial pelos reconhecedores, que descrevemos a seguir.

Cada reconhecedor contém duas listas de apontadores para objetos do tipo `ObjectWithState`: uma delas corresponde aos objetos cujo estado vai ser verificado, e a outra corresponde aos objetos cujo estado poderá ser modificado pelo reconhecedor. À primeira lista está associada uma lista de “pesos,” que são valores pelos quais os valores numéricos correspondentes aos estados devem ser multiplicados, para então serem somados e comparados com o “limiar” associado ao reconhecedor. À segunda lista por sua vez está associada uma lista de “respostas” que determina como modificar o estado de cada objeto caso o limiar do reconhecedor seja ultrapassado. Os métodos da classe `Recognizer` permitem construir as duas listas, com pesos e respostas, respectivamente, e determinar o limiar do reconhecedor. Estes métodos devem ser usados durante a inicialização do agente. O último método permite “inspecionar” um reconhecedor, ou seja, percorrer os objetos apontados pela primeira lista multiplicando o valor correspondente ao estado de cada um pelo respectivo peso, comparar com o limiar do reconhecedor e, conforme o caso, modificar o estado de cada um dos objetos apontados pela segunda lista de acordo com a respectiva resposta.

Uma *k-line* é uma lista de apontadores para objetos do tipo `ObjectWithState`. Quanto aos métodos, um é usado para construir a lista, e outro é usado para verificar o estado da *k-line* e ativar os objetos da lista quando esta estiver ativa (lembrando que a classe `K-Line` é derivada de `ObjectWithState`). A classe `Pronome` é derivada da classe `K-Line` porque seu conteúdo é o mesmo, acrescido de dois métodos que permitem percorrer a lista de objetos. Isto é necessário porque, neste agente, *pronomes* são usados como listas de objetos a serem percorridas pelos agentes internos.

Finalmente, temos três classes de *frames*: `NarrativeFrame`, `PlaceFrame` e `CharacterFrame`, que representam, respectivamente, a estrutura seqüencial da história, lugares adjacentes e personagens relacionados. Cada um dos *frames* consiste em uma lista de apontadores para outros objetos, que neste caso devem ser *k-lines* correspondentes às descrições básicas de cenas, lugares e personagens. O uso que os agentes internos poderiam fazer dos *frames* não foi, na realidade, explorado nesta implementação, na qual a ênfase ficou com o processo de evocar assuntos. Os métodos de cada uma destas classes têm a mesma funcionalidade dos métodos da classe `K-Line`, embora esta tenha uma estrutura interna diferente; por isso as classes de *frames* não são derivadas dela.

5.3.2 O blackboard

O próprio *blackboard* também é um objeto, que contém uma coleção de objetos das diversas classes que descrevemos anteriormente. A classe `Blackboard` define uma tabela para

cada tipo de objeto e dois pronomes de contexto para cada um dos agentes internos, um delimitando as palavras-chave e outro delimitando os itens de dados que serão visíveis pelo agente correspondente.

Os métodos da classe *Blackboard* são basicamente de dois tipos: métodos de inspeção e métodos de *reset*. Os métodos de inspeção servem para verificar o estado de cada coleção de objetos no *blackboard* e modificar o estado de outros objetos de forma apropriada. Estes métodos por sua vez utilizam os métodos definidos nas classes correspondentes a cada tipo de objeto. Não existem métodos deste tipo para os objetos de dados, pois estes não são objetos de coordenação e portanto não modificam os estados de outros objetos, apenas têm o seu próprio estado verificado e/ou modificado por estes. Os métodos de *reset* são usados para implementar o caráter transitório da hiperativação: por meio destes métodos, objetos hiperativos são colocados no estado inativo.

Essencialmente, estes é o conteúdo do *blackboard* que é usado durante a execução do agente. Além disso, no entanto, o objeto *blackboard* possui recursos para sua própria construção a partir de um arquivo de configuração, ou seja, um analisador léxico, que é um outro objeto, e um analisador sintático, que consiste em uma série de métodos privados. Desta forma, podemos fazer com que o mesmo agente seja capaz de contar histórias diferentes apenas trocando o arquivo de configuração. O arquivo de configuração possui uma sintaxe própria e algumas exigências que devem ser seguidas para que o *blackboard* possa ser adequadamente construído. Em particular, existem alguns identificadores que são obrigatórios: o *frame* narrativo principal tem que se chamar *main*, os *pronomes* de contexto narrativos devem se chamar *ncontext* e *nkey*, os *pronomes* de contexto relativos a lugares devem se chamar *pcontext* e *pkey*, e da mesma forma os *pronomes* de contexto relativos a personagens devem se chamar *ccontext* e *ckey*. O formato do arquivo de configuração é mostrado no excerto a seguir.

```
/* Sleeping Beauty (Disney/Ladybird Books, 1991) */
```

```
data sc1txt      "Scene 1"          "sleeping-beauty-1.txt"
data sc1pic      "Scene 1 pic"      "sleeping-beauty-1.gif"
...

data kintxt      "King"             "king.txt"
data kinpic      "King"             "king.gif"
data quetxt      "Queen"           "queen.txt"
data quepic      "Queen"           "queen.gif"
...

data castxt      "Castle"           "castle.txt"
```

```

data caspic "Castle"          "castle.gif"
...

k_line sce1 "Scene 1"  sc1txt sc1pic end
...
k_line kin  "King"      kintxt kinpic end
k_line que  "Queen"     quetxt quepic end
...
k_line cas  "Castle"    castxt caspic end

keyword kw1  "Scene 1"
...
keyword kw31 "King"
keyword kw32 "Queen"
...
keyword kw41 "Castle"

recognizer r1a  "Scene 1 (act)"      kw1:1 ; 1 ; sce1:active end
recognizer r1h  "Scene 1 (hyp)"      kw1:1 ; 2 ; sce1:hyperactive end
...
recognizer r31a "King (act)"         kw31:1 ; 1 ; kin:active end
recognizer r31h "King (hyp)"         kw31:1 ; 2 ; kin:hyperactive end
recognizer r32a "Queen (act)"        kw32:1 ; 1 ; que:active end
recognizer r32h "Queen (hyp)"        kw32:1 ; 2 ; que:hyperactive end
...
recognizer r41a "Castle (act)"       kw41:1 ; 1 ; cas:active end
recognizer r41h "Castle (hyp)"       kw41:1 ; 2 ; cas:hyperactive end
...

recognizer lnk1 "-" sce1:1 ; 2 ; kintxt:active quetxt:active aurtxt:active
      castxt:active end
...
recognizer lnk31 "-" kin:1 ; 2 ; sc1txt:active sc4txt:inactive sc4atxt:active
      sc7txt:inactive sc7atxt:active
      castxt:active end
recognizer lnk32 "-" que:1 ; 2 ; sc1txt:active sc4txt:inactive sc4atxt:active
      sc7txt:inactive sc7atxt:active
      castxt:active end

```

```

...
recognizer lnk41 "-" cas:1 ; 2 ; sc1txt:active sc18txt:active sc26txt:inactive
      sc26atxt:active kintxt:active quetxt:active end
...

recognizer lnk45 "-" sc1txt:1 sc1pic:1 ; 1 ; kw1:active end
...
recognizer lnk75 "-" kintxt:1 kinpic:1 ; 1 ; kw31:active end
recognizer lnk76 "-" quetxt:1 quepic:1 ; 1 ; kw32:active end
...
recognizer lnk85 "-" castxt:1 caspic:1 ; 1 ; kw41:active end
...

recognizer lnk89 "-" sce1:1 ; 2 ; sc2txt:active sc2atxt:inactive end
...

n_frame main "Sleeping Beauty" sce1 sce2 sce3 sce4 sce5 sce6 sce7 sce8 sce9
      sce10 sce11 sce12 sce13 sce14 sce15 sce16 sce17 sce18 sce19 sce20
      sce21 sce22 sce23 sce24 sce25 sce26 sce27 sce28 sce29 sce30 end
n_pronome ncontext "(narrative context)" sc1txt sc2txt sc2atxt sc3txt sc3atxt
      sc4txt sc4atxt sc5txt sc5atxt sc6txt sc6atxt sc7txt sc7atxt sc8txt
      sc8atxt sc9txt sc10txt sc11txt sc11atxt sc12txt sc12atxt sc12btxt
      sc13txt sc13atxt sc14txt sc15txt sc16txt sc16atxt sc17txt sc18txt
      sc19txt sc20txt sc20atxt sc21txt sc21atxt sc22txt sc22atxt sc23txt
      sc23atxt sc24txt sc24atxt sc25txt sc26txt sc26atxt sc27txt sc27atxt
      sc28txt sc28atxt sc29txt sc29atxt sc30txt end
n_pronome nkey "(narrative keywords)" kw1 kw2 kw3 kw4 kw5 kw6 kw7 kw8 kw9
      kw10 kw11 kw12 kw13 kw14 kw15 kw16 kw17 kw18 kw19 kw20 kw21 kw22
      kw23 kw24 kw25 kw26 kw27 kw28 kw29 kw30 end

c_frame cfr1 "parents" kin que aur end
c_frame cfr2 "companions" flo fau mer end
...

c_pronome ccontext "(character context)" kintxt quetxt aurtxt flotxt fautxt
      mertxt maltxt pritxt hubtxt anitxt end
c_pronome ckey "(character keywords)" kw31 kw32 kw33 kw34 kw35 kw36 kw37
      kw38 kw39 kw40 end

```

```

p_frame pfr1 "near" cas for end
p_frame pfr2 "in" cot for end
...

p_pronome pcontext "(place context)" castxt fortxt cottxt duntxt end
p_pronome pkey "(place keywords)" kw41 kw42 kw43 kw44 end

```

5.3.3 Os agentes internos

Os agentes internos também são objetos, baseados em quatro classes (figura 5.3): a classe *Agent* e as três classes derivadas, *NarrativeAgent*, *PlaceAgent* e *CharacterAgent*, que correspondem aos três agentes internos. A classe *Agent* contém um apontador através do qual o agente tem acesso ao *blackboard*. A classe *NarrativeAgent* possui um apontador para o *frame* narrativo principal, que deve ser identificado no arquivo de configuração como *main* para que possa ser reconhecido; possui também um método que dá início ao funcionamento do agente, hiperativando um conjunto de dados *default* de modo a iniciar o processo através do qual outros assuntos são trazidos ao foco de atenção dos agentes. Além disso os três agentes possuem quatro métodos: um primeiro método que inicializa as relações do agente com o *blackboard*, um segundo que mostra dados para o usuário, um terceiro que mostra palavras-chave e finalmente outro método que seleciona uma palavra-chave.

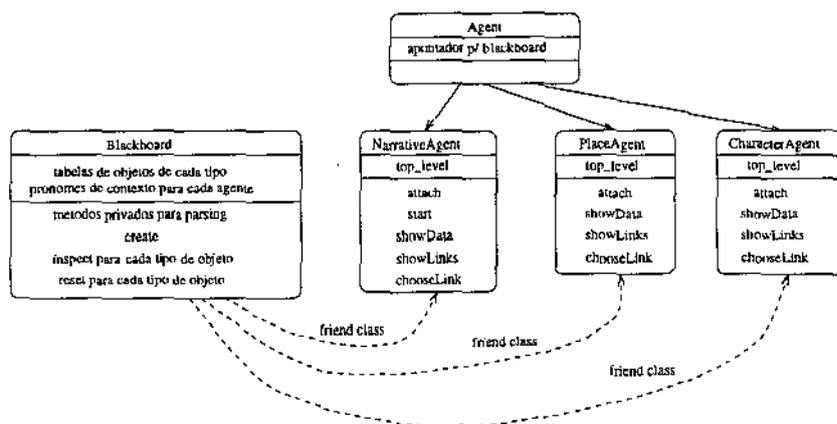


Figura 5.3: A hierarquia de classes correspondentes aos agentes.

O agente inteiro foi implementado como um programa que possui quatro objetos globais: o *blackboard* e os três agentes internos. Existe apenas um objeto de cada tipo.

O funcionamento do agente inicia com a criação do *blackboard* a partir do arquivo de configuração, seguida da inicialização dos agentes internos e da chamada ao método que seleciona um conjunto de dados (default) para ser apresentado ao usuário. A partir de então, é executado um *loop* no qual os estados dos objetos no *blackboard* são atualizados de acordo com os estados de outros objetos, os dados relativos ao assunto escolhido são mostrados, os estados dos objetos no *blackboard* são reinicializados—isto é, os objetos hiperativos são desativados—, os estados dos objetos são novamente atualizados, as palavras-chave ativas são mostradas ao usuário e este escolhe uma, que é então hiperativada. O estado dos objetos do *blackboard* é novamente atualizado, os dados hiperativos são mostrados e assim por diante.

5.4 Resultados e extensões

O agente que foi efetivamente implementado conta uma história muito simples, *Sleeping Beauty*, utiliza apenas textos, possui uma interface extremamente simplificada e não é distribuído. Por outro lado, tudo isto pode ser modificado sem interferir significativamente no funcionamento dos objetos no *blackboard*, ou seja, nos mecanismos baseados na Sociedade da Mente, que constituem a característica essencial do agente.

Embora a história escolhida tenha sido simples, foi possível observar como ocorrem os processos que descrevemos na seção 5.2 e, principalmente, como isto seria escalável para uma história muito mais complexa. Quanto mais elaborada a história, mais interessante seria o comportamento do agente, contanto que este seja adequadamente configurado.

Mostramos a seguir uma seqüência das telas que o agente exhibe. Cada uma mostra as informações exibidas e as palavras-chave que cada um dos agentes internos oferece como assuntos para exploração.

Narrative agent shows: <<Scene 1>>

```
-----
>> Long, long ago, in a faraway kingdom, a beautiful
>> baby girl was born to the king and queen. They named
>> the new princess Aurora, and they gave a magnificent
>> feast to celebrate her birth.
-----
```

Narrative agent suggests:

1. Scene 1
2. Scene 2

Place agent suggests:

1. Castle

Character agent suggests:

1. King
2. Queen
3. Aurora/Briar Rose

Agent (n/p/c): n

Link number: 2

Narrative agent shows: <<Scene 2>>

>> The three good fairies of the kingdom were invited,
>> and each one had a special gift for the princess.
>>
>> "I give you the gift of beauty," said Flora.
>>
>> "I give you the gift of song," said Fauna.

Narrative agent suggests:

1. Scene 1
2. Scene 3

Place agent suggests:

1. Castle

Character agent suggests:

1. King
2. Queen
3. Aurora/Briar Rose
4. Flora
5. Fauna

Agent (n/p/c): n

Link number: 2

Narrative agent shows: <<Scene 3>>

>> But just as the third fairy, Merryweather, was about to
>> speak, a flash of lightning filled the room.

>>

>> The wicked fairy Maleficent appeared. She was furious

>> that she had not been invited to the feast.

Narrative agent suggests:

1. Scene 1
2. Scene 2
3. Scene 4

Place agent suggests:

1. Castle

Character agent suggests:

1. King
2. Queen
3. Aurora/Briar Rose
4. Flora
5. Fauna
6. Merryweather
7. Maleficent

Agent (n/p/c): c

Link number: 7

Character agent shows: <<Maleficent>>

Narrative agent suggests:

1. Scene 1
2. Scene 2
3. Scene 3
4. Scene 4
5. Scene 18
6. Scene 22
7. Scene 25

Place agent suggests:

1. Castle
2. Maleficent's dungeon

Character agent suggests:

1. King

2. Queen
3. Aurora/Briar Rose
4. Flora
5. Fauna
6. Merryweather

Agent (n/p/c): p

Link number: 2

Place agent shows: <<Maleficent's dungeon>>

Narrative agent suggests:

1. Scene 1
2. Scene 2
3. Scene 3
4. Scene 4
5. Scene 18
6. Scene 22
7. Scene 25

Place agent suggests:

1. Castle

Character agent suggests:

1. King
2. Queen
3. Aurora/Briar Rose
4. Flora
5. Fauna
6. Merryweather
7. Maleficent

Agent (n/p/c): n

Link number: 4

Narrative agent shows: <<Scene 4>>

>> "You will regret your mistake," she thundered at the
>> king and queen, "for I, too, have a gift for the
>> princess! Before her sixteenth birthday, she will

```
>> prick her finger on the spindle of a spinning wheel,  
>> and she will DIE!"  
>>  
>> Then, with her evil laughter echoing through the castle,  
>> Maleficent vanished.
```

Narrative agent suggests:

1. Scene 1
2. Scene 2
3. Scene 3
4. Scene 5
5. Scene 18
6. Scene 22
7. Scene 25

Place agent suggests:

1. Castle

Character agent suggests:

1. King
2. Queen
3. Aurora/Briar Rose
4. Flora
5. Fauna
6. Merryweather
7. Maleficent

Quanto à interface, o ideal seria ter uma interface gráfica. Os agentes internos podem ser caracterizados visualmente em uma tal interface. Além disso, outros tipos de dados poderiam ser utilizados além de texto e imagens, como por exemplo sons, incluindo música e voz, e animações. É importante notar que uma reestruturação da interface poderia sugerir agentes internos diferentes.

O agente que conta uma história se parece, em certo sentido, com um *browser* de hipertexto. As palavras-chave podem ser comparadas com os *links* de um hipertexto, e os dados mostrados para o usuário podem ser considerados como uma página de hipertexto. Mas, se por um lado o papel das palavras-chave é equivalente ao papel dos *links* de um hipertexto, por outro lado existe uma diferença importante que é o fato de que *links* são em geral programados e incluídos nas próprias páginas, enquanto neste caso os possíveis "caminhos" de exploração *emergem* gradativamente a partir das conexões internas entre objetos no *blackboard*. As páginas de um hipertexto também são previamente codificadas,

enquanto a “página” de dados que o agente que conta uma história mostra para o usuário a cada iteração é dinamicamente construída de acordo com o contexto, que por sua vez depende do que ocorreu no “passado” do agente. Podemos dizer, por um lado, que seria possível projetar um sistema de hipertexto sofisticado que tivesse a mesma funcionalidade deste agente, sem ter que recorrer ao modelo da Sociedade da Mente; por outro lado, este agente parece-nos uma maneira interessante de fazer isto.

Na seção 5.2, mencionamos duas extensões que poderiam ser implementadas. Uma delas é a que permite simular uma espécie de “esquecimento” gradual. Nesta extensão, não temos mais apenas três estados ou níveis de ativação mas vários: o nível inativo continua correspondendo a um valor numérico de zero; o nível ativo passa a corresponder a um valor de 100 (digamos) e o nível hiperativo passa a corresponder a 200, sendo que a diferença crucial é que agora todos os valores (inteiros) entre zero e 100 passam a ser válidos, e o estado hiperativo representa uma espécie de “excitação” temporária marcadamente distinta dos demais estados.

Nesta extensão, os conceitos de desativação, ativação e hiperativação continuam exatamente os mesmos: um objeto que é desativado passa para o estado inativo, ou nível de atividade zero; um objeto que é ativado passa para o estado ativo, ou nível de atividade 100; e um objeto que é hiperativado passa para o estado hiperativo, ou nível de atividade 200. Como na versão básica, durante o *reset* os objetos hiperativos são desativados. A diferença é que os objetos ativos não são mais simplesmente deixados ativos, mas têm seu nível de atividade reduzido a cada *reset*. Este pode ser reduzido de uma ou mais unidades, contanto que não decresça abaixo de zero. Poderíamos até mesmo associar a cada objeto um “delta” de decréscimo, que corresponderia ao número de unidades de atividade que lhe são subtraídas a cada *reset*, ou seja, a cada ciclo de atividade do agente. Desta forma, podemos fazer com que objetos tenham tendências diferentes a serem esquecidos. No caso extremo, um objeto cujo “delta” fosse igual a 100 ficaria ativo apenas durante a iteração corrente: se a palavra-chave que o inclui fosse escolhida, ele seria mostrado; caso contrário, seria esquecido. Desta forma podemos ter dados que são mostrados somente em determinadas circunstâncias muito particulares, ou seja, somente quando um assunto é explorado a partir de outro.

Outra extensão envolve apenas o uso de um tipo de dado adicional: um “indicador” ou “contador” que impede que certos dados sejam mostrados mais de uma vez (ou mais de duas, três etc.). O reconhecedor que hiperativa um objeto (um dado ou uma *k-line*) que queremos que seja mostrado uma única vez possui uma entrada e uma saída adicionais, ambas conectadas ao indicador. Assim o reconhecedor passa a ter duas entradas: a palavra-chave, com peso 1, e o indicador, com peso -1, e continua com um limiar de 2. A resposta do reconhecedor é hiperativar o dado ou *k-line* e ao mesmo tempo ativar o indicador. Desta forma, o reconhecedor só responde quando a palavra-chave estiver ativa

e o indicador inativo. Depois que o indicador é tornado ativo, junto com a primeira hiperativação do objeto, o limiar de reconhecimento nunca mais é satisfeito.

Esta extensão parte da versão básica, ou seja, com apenas três níveis de atividade, mas pode ser combinada com a extensão anterior. Reunindo as duas extensões, teremos um resultado interessante, pois podemos considerar como se depois de um certo tempo o agente “esquecesse” que já havia apresentado certos dados e pudesse então repeti-los. Podemos interpretar isto de forma um pouco diferente também: podemos pensar que o agente não esqueceu de nada, mas supôs que o seu interlocutor pode ter esquecido de algo que ele mencionou há um certo tempo. Se utilizarmos “deltas” diferentes para cada objeto, podemos ter indicadores com “delta” igual a zero fazendo com que dados sejam apresentados uma única vez e depois nunca mais, e outros indicadores com “delta” não-nulo resultando no efeito anteriormente descrito.

Como dissemos na introdução, a possibilidade de estabelecer ligações “subjéctivas” entre dados permite conceber um *ambiente* de exploração de uma história composto por mais de um agente deste tipo em vez de um só. Esta possibilidade se torna interessante a partir do momento em que seja possível atribuir a cada agente um “estilo,” “personalidade” etc. de modo a permitir que o usuário tenha uma expectativa razoável—isto é, um modelo mental—do que pode obter de cada agente. Quanto mais complexa a história, mais esta abordagem se torna adequada. Se a história em questão for a História da Civilização Ocidental, por exemplo, podemos ter um agente que “vê” a seqüencialidade dos fatos exclusiva ou predominantemente a partir de sua ordem cronológica, um segundo agente que o faz a partir dos determinantes *políticos* dos fatos, um terceiro agente que estabelece as relações entre os fatos a partir de seus determinantes *econômicos*, e assim por diante.

Outra possibilidade, mencionada no capítulo 4, seria adaptar a mesma idéia do agente que conta uma história para um agente que permite ao usuário explorar alguma outra coleção estruturada de informações, como um museu virtual.

Finalmente, podemos tentar pensar no significado dos fenômenos que ocorrem internamente ao agente que conta uma história. Como dissemos anteriormente, podemos nos referir tanto a itens de dados que se encontram no foco de atenção do agente, como a assuntos que ali se encontram, sendo que nem sempre o assunto correspondente a um item que está no foco de atenção também vai estar no foco de atenção do agente. Isto ocorre porque existe um limiar—um número mínimo de itens de dados ativos—a partir do qual um assunto é trazido para o foco de atenção. Assim, podemos considerar *como se* os assuntos no foco de atenção do agente ocupassem sua atenção “consciente,” enquanto itens de dados que estão no foco de atenção sem que seus assuntos o estejam se encontram no “subconsciente” do agente e, havendo estímulos adicionais—ou seja, mais itens de dados associados ao mesmo assunto ativos—, podem ultrapassar o limiar da “consciência.” Não

queremos dizer com isto que este é o significado ou a natureza da consciência humana, mas apenas que podemos pensar em um significado deste tipo no caso particular deste agente extremamente simplificado.

5.5 Conclusão

Neste capítulo, descrevemos a implementação de um exemplo de um agente inteligente usando a arquitetura baseada na Sociedade da Mente que foi desenvolvida neste trabalho. Começamos apresentando a idéia de um agente que conta uma história segundo a idealização de um diálogo no qual uma pessoa conta uma história para outra à medida que esta lhe solicita informações adicionais. Este diálogo paradigmático é transposto para um modo de interação no qual o agente apresenta informações na forma de texto e imagens para o usuário e este solicita informações através da escolha de um entre diversos itens que o agente lhe oferece sob a forma de *buttons* ou menus. Em seguida, discutimos as características que uma história deve ter para que este tipo de diálogo possa se desenrolar. Na seção seguinte, mostramos como as informações sobre a história são organizadas como objetos no *blackboard* e como estes implementam as relações entre os diversos assuntos que a história envolve. Descrevemos os processos por meio dos quais diferentes assuntos são evocados e trazidos para o “foco de atenção” do agente, tornando-se passíveis de serem selecionados para exploração pelo usuário. Apresentamos então os agentes internos que compõem o agente, cada um dos quais sendo capaz de “ver” uma parte do foco de atenção. Na seção 5.3 descrevemos a implementação em C++ em um sistema Unix de um agente que conta uma história simples. Na seção 5.4 discutimos os resultados e *insights* obtidos, algumas variações, como a possibilidade de uso de outros tipos de dados (*multimedia*), e duas extensões: uma que simula o “esquecimento” gradual de contextos e outra que tem por fim evitar que o agente apresente informações repetidas. Finalmente, mencionamos a possibilidade de se ter um ambiente no qual não somente um, mas diversos agentes contam uma mesma história para o usuário, mas com “estilos” diferentes; e fizemos alguns comentários sobre uma possível interpretação do fenômeno interno ao agente de acordo com o qual um assunto é trazido para o foco de atenção; e também a possibilidade de usar a mesma idéia para a exploração de outras coleções estruturadas de informações inter-relacionadas.

Capítulo 6

Conclusão

Este capítulo encerra este trabalho apresentando suas conclusões. A primeira parte consiste em uma discussão que cobre todos os aspectos do trabalho, e que podemos sumarizar da seguinte forma: as razões que nos levaram a escolher a Sociedade da Mente como um modelo teórico para servir de fundamento para uma arquitetura para agentes inteligentes e a IAD como o paradigma computacional para sua realização, a adequação dos conceitos da Sociedade da Mente ao paradigma de IAD, o processo de desenvolvimento da arquitetura, as características da arquitetura obtida e a implementação que foi feita de um exemplo de aplicação. Na segunda parte, descrevemos algumas direções futuras nas quais podem ser feitas novas explorações. Finalmente, apresentamos as conclusões finais.

6.1 Discussão

A Inteligência Artificial é um empreendimento que tem um caráter duplo. De um lado, ela é uma parte da ciência cognitiva; de outro, ela é uma parte da ciência da computação ou mesmo da engenharia. Assim, por um lado ela constitui uma tentativa de explicar e/ou modelar computacionalmente a inteligência e a mente, e por outro lado ela busca desenvolver aplicações. Estas aplicações não são quaisquer, porém; elas apresentam características específicas, baseando-se nos possíveis mecanismos que também estão por trás da inteligência humana, em lugar de simplesmente copiar comportamentos de forma *ad hoc* [24]. Schank menciona um programa de reconhecimento de palavras que compara o sinal sonoro de entrada com padrões previamente armazenados em uma base de dados. Este programa não seria uma aplicação de IA. Já um outro programa que se baseasse, por exemplo, no modelamento de diversos níveis de compreensão, passando pelo reconhecimento de fonemas até o reconhecimento de palavras constituiria uma aplicação de IA, e uma das conseqüências de se usar esta abordagem seria a escalabilidade, ficando o primeiro programa restrito ao reconhecimento das palavras que estão na sua base de

dados.

A Sociedade da Mente é uma coleção de idéias que perfazem um modelo coerente da mente e da inteligência e que apresenta as seguintes características importantes para o contexto deste trabalho: trata-se de um modelo computacional da mente, onde por modelo computacional entendemos um modelo descrito em termos dos conceitos mais básicos de representação e processo, e descreve estruturas e mecanismos bem definidos. Estas características tornam a Sociedade da Mente atraente como um modelo para uma arquitetura para agentes inteligentes.

A Sociedade da Mente é controvertida no que se refere à validade como uma teoria de mente. Alguns críticos (por exemplo [9]) objetam que Minsky não leva em consideração as neurociências e os resultados experimentais que tem sido obtidos nestas áreas. Reeke observa que o modelo que Minsky propõe é, em vários aspectos, inadequado se forem levadas em conta certas evidências experimentais obtidas a respeito do funcionamento do cérebro. Ele aponta, por exemplo, que o modelo da Sociedade da Mente parece ignorar a grande variabilidade em termos de mecanismos de conexões entre os neurônios encontrada no cérebro. Reeke também sugere que a Sociedade da Mente é um modelo da mente considerada como um computador paralelo com memória distribuída, e que os diversos mecanismos que este modelo propõe são mais próximos de mecanismos e conceitos computacionais do que de mecanismos que sejam plausíveis de serem encontrados no cérebro. A partir de críticas deste tipo, podemos pensar da seguinte maneira: a Sociedade da Mente talvez não seja um modelo adequado ou realista do funcionamento da mente humana, mas sim um modelo de uma inteligência possível, e portanto pode ser vista como um modelo conceitual adequado para projetar um certo tipo de mente artificial, a qual aqui chamamos de “agente inteligente.” Um agente inteligente não tem que possuir necessariamente uma inteligência humana, mas apenas alguns aspectos do que possa ser considerado como uma inteligência possível. Projetar um agente inteligente não constitui uma tentativa de modelar a mente humana, pelo menos não necessariamente.

A idéia de projetar uma arquitetura para agentes inteligentes baseada na Sociedade da Mente foi concebida pensando em aplicações da IA. Em particular, uma abordagem que nos pareceu atraente foi a chamada IA → AI (Inteligência Artificial → Ampliação da Inteligência, no original AI → IA: *Artificial Intelligence* → *Intelligence Augmentation*) [7]. Esta abordagem sugere essencialmente que a ênfase da pesquisa em IA deve se deslocar para o uso dos computadores para ampliar ou aumentar a inteligência em lugar de imitá-la. A idéia de ampliação da inteligência (*Intelligence Augmentation*) em si não é nova, podendo ter suas origens traçadas quase que até as origens da própria computação, onde já nos anos 60 Engelbart utilizava o termo *augmentation* para sugerir que a maior utilidade dos computadores estaria nesta direção. É esta a idéia que está por trás de muitas pesquisas em interfaces e interação humano-computador assim como em outras áreas, en-

tre as quais se destaca o uso dos computadores na educação. A abordagem de Fischer vai um pouco mais além ao sugerir especificamente que as técnicas de IA podem ser utilizadas para a ampliação da inteligência. Embora Papert já tenha indicado que a filosofia LOGO teria suas origens na IA, no caso ele sugeriu que, uma vez que a IA exige uma exploração sistemática das atividades relacionadas a inteligência e em particular das atividades de resolução de problemas, esta característica poderia ser proveitosamente transferida para o terreno da educação sob a forma de uma espécie de introspecção sistemática com o propósito de melhor compreender para melhor dominar os processos de resolução de problemas [19]. Fischer, por outro lado, sugere que a “inteligência” do computador pode ter uma relação de *complementaridade* com a inteligência humana de modo a ampliá-la. A ênfase recai então na resolução colaborativa de problemas envolvendo agentes humanos e computacionais e na divisão de trabalho entre estes, visando esta complementaridade.

Nesse contexto, estivemos elaborando idéias para os chamados *domínios multi-agentes*, que seriam essencialmente um paradigma de interação humano-computador baseado em agentes substituindo aplicações. A partir disto surgiu a idéia de projetar a arquitetura de um agente, usando a Sociedade da Mente como modelo de modo a ter um agente razoavelmente sofisticado, para não dizer inteligente, desde que usemos esta palavra não no sentido restrito de inteligência humana mas no sentido mais amplo de um agente cujas habilidades são capazes de se combinar sinergisticamente e produtivamente com as habilidades humanas.

Para este tipo de objetivo, torna-se permissível utilizar o que Dennett chama de “rodas cognitivas” [5], ou seja, mecanismos que obtêm um determinado resultado mesmo que não o façam de forma idêntica aos mecanismos encontrados na natureza. Podemos utilizar “rodas cognitivas” a fim de obter programas ou agentes “inteligentes,” embora não como modelos da inteligência real. A analogia é com a idéia da roda como mecanismo de locomoção: rodas não são encontradas na natureza, e no entanto elas são muito mais fáceis de projetar e em geral mais eficazes do que seria o caso se tivéssemos que construir “pernas” para carros e outros veículos. Outra comparação que encontramos frequentemente é entre pássaros e aviões, onde estes representam um meio de voar sem imitar os mecanismos naturais. Em suma, utilizar mecanismos diferentes dos naturais pode ser até mesmo preferível devido à facilidade e outros fatores. Isto se aplica também no caso dos agentes para ampliação da inteligência.

Desta forma, o fato de a Sociedade da Mente ser ou não válida como uma teoria da mente não é relevante para sua validade como modelo para uma arquitetura de agentes inteligentes. A objeção de Reeke de que ela é um modelo excessivamente computacional [9] deixa de ser uma objeção para se tornar uma vantagem, da mesma forma que rodas e asas de aviões são mais fáceis de construir e controlar do que pernas e asas móveis como as dos pássaros. Ao lado desse direcionamento computacional, a estrutura da Sociedade

da Mente e a meta de refletir uma inteligência “genérica” em lugar de algum tipo de inteligência especializada constituem os principais aspectos que tornam a Sociedade da Mente atraente para este propósito. Outros aspectos importantes que podemos mencionar são que a Sociedade da Mente é um modelo descentralizado no qual os componentes individuais não precisam possuir inteligência; e também, que a Sociedade da Mente incorpora realmente pelo menos algumas características normalmente associadas à inteligência, como a “intencionalidade” na acepção de Searle: para ele, intencionalidade é a propriedade que um pensamento ou idéia tem de se referir a alguma coisa, de ser “sobre” alguma coisa (*aboutness*) [2, 9, 25]. Na Sociedade da Mente, a intencionalidade corresponderia ao fato de que todo estado mental parcial causa, por meio da ação dos reconhecedores e demais mecanismos de coordenação, o estabelecimento de outros estados mentais parciais a ele relacionados. Desta forma, o nome ou idéia de uma “caneta” evoca todas as características e o aspecto do objeto “caneta,” assim como imagens de canetas mais ou menos específicas, as possibilidades de uso de uma ou mais canetas, tanto práticas como artísticas e outras, e assim por diante.

Estes aspectos, em particular o fato de a Sociedade da Mente ser um modelo de inteligência genérica, são relevantes para o propósito deste trabalho porque este foi o de adequar a estrutura básica da Sociedade da Mente a uma arquitetura genérica para agentes inteligentes, em contraste com uma arquitetura voltada para alguma aplicação específica. Esta adequação se concentra nos mecanismos de coordenação apresentados na Sociedade da Mente, sendo que o aspecto mais importante é que os mecanismos básicos não são específicos para certos domínios da mente, mas sim válidos para todos, embora com variações para domínios particulares; por exemplo, o conceito de *frames* está presente em todos os domínios, embora existam tipos diferentes de *frames*, como os *frames* de linguagem, os *frames* de imagens etc.

Decidimos que um agente cuja arquitetura se baseia na Sociedade da Mente deveria ser ele próprio um sistema multi-agentes, ou seja, um sistema de Inteligência Artificial Distribuída (IAD). Este nos pareceu ser um paradigma especialmente apropriado por ser basicamente compatível com a Sociedade da Mente.

A Inteligência Artificial Distribuída pode ser descrita como tendo diversas facetas, entre as quais destacamos as seguintes:

- uma, puramente voltada para aplicações, como controle industrial, controle de tráfego etc. Aplica muitos conceitos da Inteligência Artificial—regras, *fuzzy logic* etc.—mas está basicamente voltada para resolver problemas, combinando para isto as várias técnicas que a IA oferece;
- outra, influenciada por teorias diversas das ciências humanas, como: teoria das organizações, economia, lingüística, sociologia etc. O sistema multi-agentes é visto como

um modelo de uma organização, mercado, sociedade, grupo etc. Visa tanto utilizar noções das ciências citadas no contexto da IAD como também fazer modelamentos e simulações para obter resultados para estas ciências;

- uma terceira visão considera a IAD como um modelo ou paradigma computacional no qual o sistema é composto por componentes descentralizados e o comportamento do sistema como um todo é o resultado do comportamento dos componentes e da interação entre eles. Esta vertente se alinha com certos tópicos de computação emergente, e tem muito em comum com o paradigma mais geral de processamento ou computação distribuída, embora com ênfases distintas.

Entre estas três visões da IAD, que não são de modo algum mutuamente exclusivas, a que nos interessa aqui é a última.

Em uma perspectiva mais ampla, podemos dizer que o modelo computacional que a IAD propõe parece ser um modelo interessante para refletir as propriedades da inteligência, no sentido de que ele consiste basicamente em múltiplos processos interagindo, cada um deles sendo autônomo mas também influenciando-se constantemente uns aos outros. Um modelo deste tipo poderia ser capaz de captar a complexidade dos processos e interações que subjazem à inteligência. Um ponto de vista semelhante pode ser encontrado em [28], no qual a importância dos poderes causais envolvidos nessa constante interação é enfatizado. Em outras palavras, a mente poderia ser comparada a uma sinfonia; para entender uma sinfonia, é importante ter conhecimento sobre os músicos e os instrumentos musicais, mas também é necessário estudar a estrutura da sinfonia, teoria musical, harmonia e tudo mais que faz com que um conjunto de sons seja mais do que isso, tornando-se uma sinfonia.

A partir da escolha da Sociedade da Mente como modelo conceitual e da IAD como paradigma computacional para a arquitetura de um agente, este trabalho também pode ser visto como uma tentativa de conciliar estes dois modelos. A IAD especifica que o sistema deve ser descentralizado e composto por componentes que operam concorrentemente, interagindo através de troca de mensagens ou de uma estrutura compartilhada, como é o caso do *blackboard*. Mas resta dizer quem são os agentes, como eles atuam, e sobre o quê. A definição destes aspectos é baseada na Sociedade da Mente, particularmente no que se refere a interação entre os agentes, feita através dos estados mentais parciais e dos mecanismos que os coordenam.

Dois tipos de agentes foram selecionados para serem transformados em elementos da arquitetura do agente: agências atômicas e agentes de coordenação. Chamamos de agências atômicas aos agentes de um nível de descrição relativamente alto, isto é, abstrato, da Sociedade da Mente. Isto significa que estes agentes são agências de um nível mais baixo

de descrição, podendo ser decompostas em agentes mais simples. Ao decidir transformá-los em agentes da arquitetura de IAD, no entanto, eles deixam de poder ser decompostos, e por isso os chamamos de agências atômicas. Agências atômicas correspondem então a agentes da arquitetura de IAD.

Quanto aos agentes de coordenação, que implementam os mecanismos de coordenação da Sociedade da Mente, eles correspondem ao principal aspecto da Sociedade da Mente que decidimos utilizar na arquitetura. Estes deveriam então ser de uma forma ou de outra transformados em elementos da arquitetura de forma a preservar sua funcionalidade. Assim, o desenvolvimento da arquitetura pode ser visto como a sucessiva transformação destes mecanismos/agentes em elementos mais adequados da arquitetura.

Também é importante mencionar que a Sociedade da Mente é um modelo de natureza conexionista, baseado em processamento paralelo, enquanto que o paradigma da IAD é o de processamento distribuído, simbólico, procedural. Este foi outro aspecto que teve que ser endereçado no desenvolvimento da arquitetura.

A primeira idéia foi transformar tanto agências atômicas como agentes de coordenação em agentes da arquitetura de IAD. Inicialmente, pensamos em dar às agências atômicas uma estrutura interna de natureza conexionista, de modo que os agentes de coordenação se referissem a esta estrutura interna. Isto manteria a funcionalidade dos agentes de coordenação exatamente como é descrita na Sociedade da Mente. A esta interpretação “literal” se seguiu a idéia de pensar nas agências atômicas como agentes que processam dados, com um certo número de entradas e um certo número de saídas, sendo algumas destas entradas e saídas simplesmente sinais de ativação e de inibição, ou seja, sem carregar informação. Cada uma das entradas e saídas seria como se fosse “conectada” a outro agente específico. Estas “conexões” poderiam ser mudadas através de mensagens específicas cujo conteúdo indicaria quais as conexões a alterar. Desta forma conciliamos a natureza conexionista da Sociedade da Mente com a natureza simbólica e procedural da IAD. Juntando-se a isso a idéia de que cada agente de coordenação seria um agente do sistema de IAD, só que com determinados padrões de entradas e saídas—por exemplo, uma *k-line* responde a entradas de ativação enviando sinais de ativação para outros agentes quaisquer—, tem-se a primeira versão da arquitetura. A idéia seria então refinar a noção de entradas e saídas, estabelecendo exatamente quais os seus tipos—de ativação, de inibição, de informação, de administração (isto é, que define ou altera conexões) etc.— e conseqüentemente quais os tipos de mensagens que seriam trocadas entre os agentes, definindo um protocolo de comunicação.

Disto surgiu a identificação do problema desta primeira versão, que era o número excessivo de agentes de coordenação, que ainda por cima teriam que ser criados e destruídos dinamicamente, mas o mais problemático era o fato de terem uma granularidade muito baixa, ou seja, eram meramente reativos, com uma funcionalidade muito restrita. Daí

a segunda versão, que reuniu agentes de coordenação dentro de um único agente do sistema de IAD, transformando-os em entradas em uma tabela. No entanto, encontramos o problema do *overhead* de comunicação, porque no caso destes agentes a sua principal funcionalidade consiste em receber e enviar mensagens, e nesse sentido reuni-los em um único agente não adianta nada para reduzir a quantidade de mensagens e contornar o fato de que a troca de mensagens acabaria ocupando uma porção considerável do tempo.

O problema do *overhead* de comunicação foi atacado tornando o estado dos agentes de coordenação acessível aos agentes que quisessem alterá-lo, assincronamente, ou seja, como se esse estado residisse em um módulo de memória compartilhada. Finalmente, para resolver o problema das referências, colocamos todos os estados juntos em um *blackboard*. E já que a funcionalidade dos agentes de coordenação é bem definida, definimos que *threads* do próprio *blackboard* inspecionam os estados e alteram outros estados de acordo. Nesse estágio o que eram agentes de coordenação foram transformados em objetos ativos (na verdade reativos) em um *blackboard*. Ativos no sentido de que eles não somente são alterados por outros, mas também causam alterações em outros.

A arquitetura resultante é baseada no modelo de *blackboard* mas, em vez de este conter dados e/ou metas, este contém objetos ativos que implementam os estados mentais parciais e agentes de coordenação da Sociedade da Mente, sendo que cada tipo de objeto é caracterizado por uma funcionalidade própria. Os agentes do sistema propriamente ditos, ou seja, as agências atômicas, são em princípio heterogêneos, de modo que cada um executa funções diferentes. Como mencionado anteriormente, um sistema desse tipo pode ser implementado em diversos níveis, refletindo os múltiplos níveis de descrição na Sociedade da Mente; neste caso, as agências só são atômicas no nível mais elementar de todos, enquanto nos demais elas são sistemas multi-agentes construídos de acordo com a arquitetura baseada na Sociedade da Mente. Em termos de organização, em princípio os agentes são autônomos, não havendo relações de subordinação explícitas. No entanto, um agente pode “comandar” outro, definir o que outro vai fazer etc. através dos estados mentais parciais.

Outros dois aspectos que podem ser mencionados são a possibilidade de ocorrerem conflitos e a criação dinâmica de agentes. Conflitos poderiam ocorrer entre estados mentais parciais quando dois ou mais agentes ou objetos de coordenação tentassem evocar estados mentais parciais envolvendo alguns dos mesmos objetos. No entanto, o controle de concorrência deve impedir que mais de um agente ou objeto de coordenação possa efetivamente modificar o estado de um objeto ao mesmo tempo, de modo que o conflito não ocorre, e os objetos que foram modificados por mais de um agente ficam no estado em que o último destes agentes os colocou. Por outro lado, pode ser que conflitos possam ter efeitos interessantes, e neste caso pode ser conveniente implementar uma operação de modificação dos estados dos objetos que leve em conta o estado corrente, ou seja, as

modificações passadas.

Criação dinâmica de agentes pode se referir à criação de agentes propriamente ditos ou objetos de coordenação em tempo de execução. A arquitetura prevê a criação, modificação e destruição de objetos no *blackboard* como principal mecanismo de evolução e aprendizado do sistema, mas não prevê a criação dinâmica de agências atômicas que não tenham sido previamente programadas, embora estas possam ter sua execução iniciada condicionalmente durante o funcionamento do sistema como um todo e possam também ser replicadas. Mesmo assim, existe uma maneira pela qual isto pode ser pelo menos simulado, explorando as próprias características da arquitetura. Essencialmente, uma agência atômica corresponde a um procedimento ou programa, que também pode ser representado por um *script*, que é um tipo de *frame* que contém instruções. Deste modo, podemos ter um agente no qual todas ou algumas das agências atômicas são iguais, e são como “processadores virtuais” que executam *scripts*. Criar uma nova agência atômica consiste então em criar um *script* e replicar uma agência para executá-lo. Isto não quer dizer que esta seja a única forma de usar *scripts*; pelo contrário. Mas esta pode simular a criação dinâmica de agências atômicas e de comportamentos, inclusive porque a linguagem dos *scripts* pode ser bastante poderosa, desempenhando no caso o mesmo papel de uma linguagem de programação interpretada.

A aplicação que escolhemos para implementar foi plenamente satisfatória em mostrar as potencialidades dos mecanismos da Sociedade da Mente. Consideramos que ela ilustra uma de suas propriedades mais importantes, a de permitir conectar e relacionar fragmentos de informações; além disso utilizamos diretamente um dos conceitos propostos na Sociedade da Mente, o *loop* de rememoração.

Ela poderia ser criticada por ser *ad hoc*, mas consideramos que isto é justificável por se tratar de uma aplicação, que não pretende modelar uma mente real; qualquer aplicação teria que ser desenvolvida da mesma forma, em princípio.

O fato de termos usado uma história simples não nos impediu de observar as potencialidades do agente, ao contrário, foi o que nos permitiu pensar que a mesma organização das informações e dos objetos de coordenação mas com uma história muito mais complexa poderia levar a resultados muito interessantes.

6.2 Direções futuras

Consideramos que a arquitetura desenvolvida cumpre o objetivo deste trabalho, que foi de projetar uma arquitetura genérica baseada nos mecanismos de coordenação da Sociedade da Mente e na sua adaptação ao paradigma de IAD ou sistemas multi-agentes. No entanto, precisamente por pretender ser uma arquitetura genérica, e não partir de um problema ou da proposta de modelar alguma competência mental específica, ela acaba ficando

limitada a um nível muito genérico de definição dos mecanismos, dando uma impressão de indefinição. Por exemplo, os *frames* são definidos apenas conforme a sua estrutura básica, como uma coleção estruturada de terminais, mas sem definir qual é ou pode ser esta estruturação.

Uma abordagem alternativa que contornaria este problema seria delimitar alguma área de competência, ou algum comportamento, e projetar uma arquitetura voltada para o seu modelamento. Isto poderia permitir uma melhor apreciação tanto da Sociedade da Mente como da sua combinação com sistemas multi-agentes, no sentido de ser mais concreta e mais detalhada do que a que apresentamos. Entretanto, é necessário lembrar que esta seria uma abordagem um tanto quanto parcial e restrita a um determinado domínio da mente, o que não é o caso da arquitetura apresentada.

Entre os aspectos ou competências específicas da Sociedade da Mente que poderiam ser explorados podemos mencionar: visão computacional (sistemas de *frames* de imagens); processamento de linguagem natural (sistemas de *frames* de linguagem); planejamento (sistemas de *scripts*, definição de uma linguagem de *scripts*, mecanismos de aprendizado associados, aprendizado de novas habilidades procedurais); compreensão de histórias (sistemas de *frames* que modelam situações estereotipadas).

Uma crítica ao modelo da Sociedade da Mente diz respeito à indefinição das agências que compõem a mente. Em poucos lugares na Sociedade da Mente podemos encontrar uma proposta clara de quem ou mesmo como são as agências da mente, ao passo que os agentes de coordenação são relativamente bem especificados. Há o exemplo da agência construtora, das agências que avaliam quantidades (sobre o assunto da conservação), mas de modo algum uma proposta abrangente. Ficam em aberto questões como: quem são as agências que operam com os *frames* de linguagem, quem são as agências que utilizam os *trans-frames* de modo a tirar conclusões de seus encadeamentos, por exemplo, e outras.

Na Sociedade da Mente, tanto operações como representações são feitas em termos de agentes, o que é um mérito porque ambas têm que ser ultimamente definidas em termos de componentes neurais. No entanto, à medida que se adota níveis mais altos de descrição, os diferentes poderes causais de representações e processos tornam-se distintos e justificam a descrição em termos de duas classes de entidades diferentes, quais sejam, representações e processos. Assim, se optamos por abordar a Sociedade da Mente a partir de um nível alto de descrição, definir as estruturas envolvidas em um determinado comportamento não define automaticamente quais as funções envolvidas. A Sociedade da Mente, como dissemos anteriormente, não explicita quais são os processos relevantes na mente, embora especifique as estruturas: *k-lines*, *frames* etc.

Desta forma, um caminho promissor para exploração seria definir as operações que manipulam as estruturas apresentadas. Modelar uma mente inteira seria uma tarefa gigantesca, mas também aqui vale a sugestão de explorar competências ou comportamentos

isolados. Isto deve ser feito levando em conta que a própria noção de competência isolada é artificial: competências não são isoladas, embora exista um grau considerável de autonomia, como demonstram as neurociências. Na própria Sociedade da Mente as agências são compostas por agentes conectados entre si mas também se conectam com outras agências, ou seja, com agentes das outras agências. Uma competência isolada seria uma construção virtual, mas a ciência de modo geral não pode tentar abarcar tudo de uma vez sob o risco de não conseguir explicar nada. Assim, focalizar em uma competência isolada deve servir não somente para entender as características desta competência em particular, mas também para mais adiante poder entender como ela se relaciona com outras áreas de competência.

Para sumarizar, uma sugestão para pesquisas futuras é a seguinte: explorar alguma competência isolada; definir quais as representações e quais as operações relevantes para a mesma; definir os objetos de dados e refinar os objetos de coordenação de modo a refletir adequadamente as representações identificadas e definir as agências atômicas de modo a refletir as operações. A arquitetura desenvolvida pode ser apropriada para servir de base para isto, porque ela provê os objetos de coordenação básicos e ao mesmo tempo é aberta o suficiente para acomodar variações destes. Ela é apropriada para diversas etapas do processo de exploração: em um primeiro momento, pode-se modelar as agências em um nível alto de abstração, implementando-as como procedimentos ou funções; em um momento seguinte, estas agências podem ser abertas e terem sua própria funcionalidade explorada em termos dos agentes que as constituem, e assim por diante até chegar ao nível conexionista. Desta forma, uma teoria de alguma competência específica, baseada na Sociedade da Mente, pode ser sucessivamente validada através da implementação computacional em níveis cada vez mais detalhados.

O reconhecimento das diferentes competências da mente também leva à seguinte observação: como dissemos anteriormente, uma característica importante da Sociedade da Mente é que os mecanismos de coordenação básicos são os mesmos para todos os domínios da mente, embora existam variantes específicas associadas a cada um destes, como o domínio do processamento da linguagem, o domínio visual-espacial etc.; ao mesmo tempo, a Sociedade da Mente afirma que a inteligência emerge da existência na mente de agências que utilizam representações e processos diferentes. Então parece que existe um *interplay* entre a maior heterogeneidade possível construída em cima de um certo número de elementos básicos que são os mesmos ao longo dos domínios, e que em última análise permitem que estes se comuniquem.

Escolhemos os tipos básicos de mecanismos de coordenação para fins de definição de uma arquitetura genérica, mas uma das direções de pesquisa futuras seria o refinamento desses mecanismos, ou seja, definir melhor todos os tipos e variedades de *k-lines*, *pronomes*, *frames* e reconhecedores. Esta seria uma proposta para uma continuidade deste trabalho.

Na Sociedade da Mente, Minsky discute vários tipos de mecanismos de coordenação, dos quais abstraímos as funcionalidades que incorporamos à presente arquitetura. Seria necessário então fazer o caminho contrário, ou seja, refinar, detalhar o funcionamento de cada um deles. Uma questão que fica em aberto é se seria possível fazer isso sem pensar em uma aplicação, seja deliberadamente ou não—em outras palavras, abordar este problema sem delimitar uma área seria o mesmo que tentar achar uma panacéia para resolver o problema da representação do conhecimento, só que partindo dos princípios da Sociedade da Mente! Uma possível solução já é sugerida neste enunciado, que seria delimitar uma área. Seria possível então, por exemplo, refinar e definir todo um sistema de *frames* de linguagem, tendo em vista empregá-los em um agente de processamento de linguagem natural.

O agente que descrevemos no capítulo 5 é um exemplo do que pode ser feito no sentido de usar a arquitetura como base para o desenvolvimento de aplicações. O significado teórico dos constructos da Sociedade da Mente serve como uma indicação sobre como eles podem ser usados no contexto de uma aplicação. No caso do agente que conta uma história, exploramos a possibilidade de conectar itens de dados relacionados uns aos outros, assim como o conceito do *loop* de rememoração, descrito no capítulo 2. Estas mesmas características também são fundamentais para o agente de busca bibliográfica que descrevemos brevemente no final do capítulo 4.

O desenvolvimento de aplicações usando esta arquitetura traz em geral a necessidade de uma engenharia do conhecimento a fim de organizar as informações de acordo com as estruturas propostas. A organização do *blackboard* do agente que conta uma história é um exemplo de como as informações referentes a uma história foram organizadas. No caso, isto foi feito de forma *ad hoc*, pensando exclusivamente na aplicação. A organização escolhida poderia, no entanto, ser usada para uma história bem mais complexa, isto é, com mais dados e conexões entre estes. Uma possibilidade interessante seria aprofundar o aspecto da engenharia de conhecimento para aplicações desse tipo, que inclui não apenas agentes que contam histórias mas, como anteriormente mencionado, também agentes que orientam o usuário na exploração de conjuntos de informações inter-relacionadas e/ou estruturadas, como um museu virtual e outros.

Outra idéia seria desenvolver um agente de senso comum como o que descrevemos muito brevemente no capítulo 4. Senso comum é um assunto do qual muito se tem falado hoje em dia, mas o único projeto de grande porte de que se tem notícia é o CYC [8], cujos próprios autores apontam que seria conveniente que houvessem outros grupos de pesquisa explorando outras alternativas. Talvez isto se deva justamente ao fato de que um projeto deste porte é necessariamente muito grande, envolvendo esforços conjuntos, e integrados, de muitas pessoas. A existência de um fundamento comum—no caso, uma arquitetura—poderia ser um ponto de partida para um projeto desta envergadura. A arquitetura

certamente teria que ser adaptada e seria necessária uma implementação sofisticada e robusta, utilizando pelo menos um sistema de suporte distribuído, mas é um ponto de partida concreto. Imaginamos que um tal projeto pudesse começar pela definição de uma ontologia geral dividida em um certo número de “domínios” que seriam subdivididos até que cada um dos domínios tivesse operações de manipulação de informações bem definidas associadas a ele e assim pudesse ser colocado sob a responsabilidade de um ou mais agentes capazes de realizar estas operações. Como a decomposição das operações deve ser comportamental e não funcional, diferentes domínios podem ser projetados quase independentemente, cada um possuindo representações de dados e ligações entre dados próprias. A partir disso, viria a integração dos domínios em domínios maiores, sendo que o inter-relacionamento final de todos os domínios constitui uma característica crucial do senso comum.

Finalmente, uma outra possibilidade dentro da linha de aplicações seria o desenvolvimento de um sistema de processamento de linguagem natural baseado na Sociedade da Mente. Minsky propõe diversas representações relacionadas com a linguagem e seu uso, como os *frames* de linguagem. A arquitetura que desenvolvemos pode ser usada como base para a construção de um sistema desse tipo mas, como observamos anteriormente, seria necessário definir quais os agentes para um tal sistema, uma vez que a Sociedade da Mente não especifica quais seriam eles. Do ponto de vista do desenvolvimento de aplicações, no entanto, estes poderiam ser definidos a partir dos requisitos de funcionalidade desejados.

6.3 Conclusão

Podemos avaliar a arquitetura desenvolvida a partir de, pelo menos, duas perspectivas: podemos determinar quais as suas vantagens em comparação com outras arquiteturas de IAD, e podemos estabelecer qual a vantagem de ter partido de um modelo conceitual em vez de projetar uma arquitetura para resolver um problema específico.

Embora não seja talvez o mais comum, não é pouco usual que arquiteturas de sistemas multi-agentes ou de sistemas de IAD de modo geral sejam projetadas a partir de modelos conceituais, em particular de modelos de organizações ou sociedades humanas, assim como de padrões de interação. Como exemplos, podemos mencionar a arquitetura de Rede de Contratos descrita no capítulo 3, que se baseia em um modelo de interação baseado em negociação e contratação de serviços, e abordagens baseadas na estrutura de mercado [34]. As vantagens de tais arquiteturas devem ser buscadas na identificação de aspectos dos modelos conceituais utilizados que se aplicam ao contexto computacional. No caso da arquitetura baseada na Sociedade da Mente, os aspectos que podemos identificar são os seguintes: a totalidade dos objetos do *blackboard* constitui uma estrutura que é ao mesmo tempo compartilhada entre os agentes do sistema, fortemente interconectada e altamente

organizada; os agentes do sistema são autônomos e independentes, podendo cada um deles utilizar suas próprias estratégias e estruturas internas; e os agentes do sistema interagem através dos estados mentais parciais realizados no *blackboard* com o auxílio dos objetos de coordenação que nele residem ao lado dos objetos de dados propriamente ditos.

Estes três aspectos, juntos, determinam uma característica do sistema que é a de acomodar um relativo isolamento entre os agentes e ao mesmo tempo permitir que estes influenciem uns aos outros a um nível bastante profundo. O compartilhamento de objetos tanto passivos (dados) como reativos (objetos de coordenação) no *blackboard* é o que permite isto. Por outro lado, a organização do *blackboard* e o funcionamento dos diferentes tipos de objetos de coordenação impõe uma super-estrutura a esse compartilhamento de informações. O resultado é fundamentalmente diferente daquele obtido em um sistema no qual os agentes são independentes e interagem através de mensagens de solicitação, disponibilização e contratação de serviços, por exemplo, porque as informações que podem ser compartilhadas são essencialmente distintas. Ao mesmo tempo, também é diferente do que ocorre na maior parte dos sistemas baseados no modelo de *blackboard*, pois este em geral contém dados (incluindo dados de controle, como metas) que os agentes utilizam e modificam mas que não afetam uns aos outros como é o caso dos objetos de coordenação na arquitetura que desenvolvemos. Nesta, portanto, o *blackboard* e os seus objetos transcendem o conceito tradicional de *blackboard* e passam a constituir um mecanismo poderoso de coordenação.

A vantagem da arquitetura que desenvolvemos em comparação com outras arquiteturas de IAD pode ser, então, expressa da seguinte forma: ela oferece vantagens na medida em que seja possível identificar elementos de uma aplicação em potencial que se ajustem bem ao tipo de mecanismo de coordenação que ela possui. Em outras palavras, a arquitetura propõe os mecanismos, e se estes captam, ou se adequam, àquilo que é necessário na aplicação, então trata-se de uma arquitetura apropriada.

A vantagem de se projetar uma arquitetura a partir de um modelo teórico em lugar de fazê-lo com o intuito de solucionar um problema está no fato de que os mecanismos que a arquitetura oferece se baseiam em um modelo teórico, de modo que a própria teoria associada provê indicações quanto ao seu significado e potencialidades, e serve como uma orientação sobre como eles podem ser utilizados.

Bibliografia

- [1] Avron Barr and Edward A. Feigenbaum. Representation of knowledge. In *The Handbook of Artificial Intelligence*, volume 1. Addison-Wesley, 1981.
- [2] Nick Bourbaki. Turing, Searle, & thought. *AI Expert*, pages 52–59, July 1990.
- [3] Rodney A. Brooks. Artificial intelligence through building robots. A.I. Memo 899, MIT, 1988.
- [4] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.
- [5] Daniel C. Dennett. Cognitive wheels: The frame problem in AI. In Margaret A. Boden, editor, *The Philosophy of Artificial Intelligence*, pages 147–170. Oxford University Press, 1990.
- [6] Robert Englemore and Tony Morgan. *Blackboard Systems*. Addison-Wesley, 1988.
- [7] Gerhard Fischer. Rethinking and reinventing artificial intelligence from the perspective of human-centered computational artifacts. In *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence*, number 991 in Lecture Notes on Artificial Intelligence, pages 1–11. Springer, 1995.
- [8] R. V. Guha and Douglas B. Lenat. Enabling agents to work together. *Communications of the ACM*, 37(7):127–142, July 1994.
- [9] George N. Reeke Jr. Marvin Minsky, The Society of Mind. *Artificial Intelligence*, 48:341–348, 1991.
- [10] Pierre Lévy. *la Machine univers: création, cognition et culture informatique*. Éditions La Découverte, 1987.
- [11] Marvin Minsky. A framework for representing knowledge. In John Haugeland, editor, *Mind Design*, pages 95–128. MIT Press, 1975.

- [12] Marvin Minsky. K-lines: A theory of memory. *Cognitive Science*, 4:117–133, 1980.
- [13] Marvin Minsky. *The Society of Mind*. Simon & Schuster, 1986.
- [14] Marvin Minsky. Preface: Connectionist models and their prospects. In D. L. Waltz and J. Feldman, editors, *Connectionist Models and Their Implications: Readings from Cognitive Science*, pages vii–xvi. Ablex, 1988.
- [15] Marvin Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, Summer:35–51, 1991.
- [16] H. Penny Nii. Blackboard systems: Blackboard application systems, blackboard systems from a knowledge engineering perspective (part two). *AI Magazine*, pages 82–106, August 1986.
- [17] H. Penny Nii. Blackboard systems. In Avron Barr, Paul R. Cohen, and Edward A. Feigenbaum, editors, *The Handbook of Artificial Intelligence*, volume IV, pages 1–82. Addison-Wesley, 1989.
- [18] Eugénio Oliveira and Fernando Augusto Mouta. A distributed AI architecture enabling multi-agent cooperation. In *Proceedings of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Edinburgh, Scotland, 1993.
- [19] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Harvester Wheatsheaf, 1980.
- [20] Doug Riecken. A conversation with Marvin Minsky about agents. *Communications of the ACM*, 37(7):23–29, July 1994.
- [21] Doug Riecken. M: An architecture of integrated agents. *Communications of the ACM*, 37(7):107–116, July 1994.
- [22] Ana Paula Rocha and Eugénio Oliveira. Studying conflicts in cooperative multi agent systems. In *submitted to First Australian Workshop on Distributed Artificial Intelligence*, Canberra, Australia, 1995.
- [23] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*, capítulo 8. Prentice-Hall, 1995.
- [24] Roger C. Schank. Where's the AI? *AI Magazine*, 12(4):138–49, 1991.
- [25] John R. Searle. Is the brain's mind a computer program? *Scientific American*, 262(1):20–31, January 1990.

- [26] Jaime S. Sichman, Yves Demazeau, and Olivier Boissier. When can knowledge-based systems be called agents? In *IX Simpósio Brasileiro de Inteligência Artificial*, pages 172–185, 1992.
- [27] Jaime Simão Sichman and Yves Demazeau. Exploiting social reasoning to enhance adaptation in open multi-agent systems. In *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence*, number 991 in Lecture Notes on Artificial Intelligence, pages 253–263. Springer, 1995.
- [28] Aaron Sloman. The emperor’s real mind: review of Roger Penrose’s *The Emperor’s New Mind: Concerning Computers, Mind and the Laws of Physics*. *Artificial Intelligence*, 56:355–396, 1992.
- [29] Reid G. Smith. The Contract Net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.
- [30] Reid G. Smith and Randall Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1):61–70, January 1981.
- [31] Michael Stumm and Songnian Zhou. Algorithms implementing distributed shared memory. *IEEE Computer*, pages 54–64, May 1990.
- [32] Ming-Chit Tam, Jonathan M. Smith, and David J. Farber. A taxonomy-based comparison of several distributed shared memory systems. *Operating Systems Review*, pages 40–67, May 1990.
- [33] Andrew S. Tanenbaum. *Modern Operating Systems*, capítulo 10. Prentice-Hall, 1992.
- [34] Michael P. Wellman. Market-oriented programming (abstract). In *Proceedings of the 12th Brazilian Symposium on Artificial Intelligence*, number 991 in Lecture Notes on Artificial Intelligence, pages 26–27. Springer, 1995.