

**Tecnologia para Sistemas Inteligentes**  
**Apontamentos para as aulas sobre**

**Sistemas de Agentes: Introdução, conceitos básicos e**  
**arquitecturas**

**Luís Miguel Botelho**

**Departamento de Ciências e Tecnologias da Informação**  
**Instituto Superior de Ciências do Trabalho e da Empresa**

**Abril de 2012**

# **Tecnologias para Sistemas Inteligentes**

## **Apontamentos para as aulas**

### **Índice**

<b>1</b>	<b>ENTERTENIMENTO: ORGANIZAÇÃO DE NOITES</b>	<b>2</b>
<b>2</b>	<b>COMÉRCIO ELECTRÓNICO MEDEADO POR AGENTES</b>	<b>3</b>
<b>3</b>	<b>ARQUITECTURAS DE SISTEMAS MULTI-AGENTE</b>	<b>4</b>
3.1	CONTROLO E COMUNICAÇÃO CENTRALIZADOS	4
3.2	ARQUITECTURA “BLACKBOARD”	5
3.3	ARQUITECTURA DE NEGOCIAÇÃO	6
3.4	ARQUITECTURA DE SUBSUNÇÃO	7
3.5	ARQUITECTURA INTERNA DOS AGENTES	8
<b>4</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>9</b>

# Sistemas de Agentes: Introdução, conceitos básicos e arquiteturas

Os Sistemas Baseados em Conhecimento (SBC) tradicionais são sistemas monolíticos no sentido em que são concebidos para realizarem um conjunto de tarefas previamente especificadas sem a cooperação de outros sistemas. A partilha de informação ou de conhecimento entre SBCs ocorre, com pouca frequência e sempre através das possibilidades de comunicação específicas do sistema operativo em que funcionam (e.g., ficheiros, filas de mensagens, bases de dados partilhadas). Mais frequentemente, o utilizador efectua uma consulta num SBC e usa o resultado dessa consulta noutra consulta noutra SBC.

Nos últimos anos, assistiu-se ao advento de um novo paradigma computacional (não só na Inteligência Artificial, mas em toda a computação), a que se chama genericamente computação distribuída. Em Inteligência Artificial (IA), a computação distribuída aparece sob duas designações: IAD (Inteligência Artificial Distribuída) e Sistemas Multi-Agente. Para muitos autores, estas duas expressões identificam a mesma realidade. Para outros, existem diferenças entre elas. Nestes apontamentos fala-se apenas de Sistemas Multi-Agente, sem a preocupação de identificar uma eventual distinção em relação à IAD.

Um sistema multi-agente é um sistema constituído por um conjunto de subsistemas especializados em determinadas tarefas que unem esforços para enfrentar problemas para os quais nenhum dos subsistemas isolados teria capacidade. A cada um desses subsistemas especializados chama-se agente. Existem agentes de hardware (“robots”) e agentes de software (“softbots”), mas este texto trata apenas de agentes de software.

Em termos genéricos, um agente de software pode ser visto como um programa muito especializado (opinião não pacífica) com uma certa autonomia, isto é, com permanência no tempo, com a capacidade de decidir o que fazer, quando fazer, e como fazer. Muitas pessoas aceitam que um agente é apenas um programa especializado numa tarefa muito específica e bem definida. Num caso extremo, um agente pode ser uma regra de produção pois trata-se de um pequeno “programa” que efectua uma ou várias acções, em determinadas condições. No entanto, num caso típico, um agente é mais do que uma regra de produção: pode ser um conjunto de regras de produções, pode ser um procedimento especializado, e pode até ser um sistema distribuído formado por vários componentes localizados num único ou em vários computadores.

Na visão tradicional dos Sistemas Baseados em Conhecimento (SBCs), o problema principal situa-se na criação e manutenção de uma base de conhecimentos abrangente, e na utilização de um motor de inferência geral e eficiente. Na visão Multi-Agente, o problema principal aparece na comunicação e na coordenação dos vários agentes que integram o sistema global.

Na abordagem tradicional, os sistemas podem ser melhorados e expandidos, dilatando a base de conhecimentos e, eventualmente, criando novas capacidades no motor de inferência. Este processo é problemático, na medida em que se trata de alterar programas complexos e com grande dimensão. Em muitos casos, uma pequena alteração conceptual pode envolver modificações difíceis de concretizar num programa grande e complexo.

Na abordagem multi-agente, os sistemas podem ser melhorados pelo acréscimo de mais agentes com novas capacidades, ou pela substituição de alguns agentes por outros. Em geral, numa arquitectura multi-agente, essas alterações não se repercutem tão profundamente nos programas que constituem o resto do sistema, como no caso dos sistemas monolíticos.

Os sistemas multi-agente e as sociedades de agentes podem ser e têm sido usados em diversos domínios de aplicação, incluindo o entretenimento; o comércio electrónico medeado por agentes; a procura, filtragem e composição de informação na Internet; a gestão de agendas pessoais, entre outros. As sub-secções que se seguem descrevem dois desses domínios de aplicação.

# **1 Entertemento: Organização de Noites**

O Projecto Europeu Agentcities.RTD [Willmott et al 2001] desenvolveu uma rede de cidades de agentes que prestam serviços no domínio do entretenimento urbano nocturno. A aplicação testada tem diversos tipos de agentes, incluindo os agentes organizadores de noite, os agentes intermediários de informação (“*information brokers*”), os agentes representantes de serviços (e.g., representantes de restaurantes e de bares), os agentes de informação sobre domínios específicos (e.g., agentes de informação de restaurantes, e agentes de informação de cinemas), e os agentes de assistência pessoal. Além destes, existem agentes que estruturam a sociedade mas que não estão directamente associados a um dado domínio de aplicação. Entre estes, destacam-se os agentes de directoria que prestam serviços de páginas amarelas, e os agentes de ontologias que prestam informação sobre as ontologias do domínio. Os agentes de páginas amarelas são agentes em que os outros agentes podem registar os seus serviços, e podem procurar prestadores de serviços de que necessitam.

Do ponto de vista do consumidor individual, a interacção típica processa-se da seguinte forma. O utilizador possui um ou mais agentes pessoais através dos quais pode interagir com a aplicação desenvolvida. Supondo que o utilizador pretende organizar uma noite numa dada cidade, apresenta o seu pedido ao seu Agente Pessoal. Este, contacta com o Organizador de Noites a quem passa o pedido e as preferências do utilizador.

O Organizador de Noites [Dale e Ceccaroni 2002] consulta os agentes de directoria (DF – “*Directory Facilitator*”, [FIPA 2002-23]) para encontrar os agentes intermediários, os agentes representantes de serviços, e os agentes de informação necessários para organizar a noite de acordo com os interesses do utilizador. Uma vez conhecidas as identificações dos agentes necessários para organizar a noite, o Organizador de Noite delega sub-tarefas nos outros agentes. Por exemplo, pode consultar um agente intermediário de informação [Botelho et al 2003] com o fim de descobrir um restaurante de comida italiana situado a curta distância do hotel do utilizador. Tendo este restaurante, pode consultar o mesmo ou outro intermediário de informação para encontrar um bar de música sul americana até ao qual o utilizador possa chegar dando um passeio nocturno agradável em numa zona segura da cidade. Tendo o restaurante e o bar, o Organizador de Noite contacta o agente representante do restaurante para efectuar a reserva de uma mesa para seis pessoas na área de fumadores, e consulta o agente de informação geográfica para criar um percurso pedestre desde o restaurante até ao bar.

Neste tipo de cenário, enquanto que os agentes organizadores de noites se especializam no seu trabalho sem terem que se preocupar com a interacção com cada utilizador, os agentes pessoais especializam-se na interacção com o seu utilizador, podendo mesmo aprender algumas das suas preferências de modo a não ter que o importunar com todos os pormenores da interacção com os outros agentes.

O agente Organizador de Noites também não é capaz de resolver todos os sub-problemas da organização de uma noite. Para encontrar agentes capazes de o ajudar consulta agentes de directoria, e para resolver alguns dos sub-problemas conta com a ajuda de agentes intermediários.

Os agentes intermediários de informação têm também que contar com agentes de directoria e com agentes de ontologias para encontrar agentes capazes de fornecer a informação necessária para os pedidos que lhe são apresentados.

Um sistema como o descrito, devido à autonomia dos agentes que o compõem e à flexibilidade da sua organização inter-agente, é muito mais robusto do que os sistemas monolíticos mais tradicionais pois será capaz de enfrentar muito mais problemas mesmo na presença de imprevistos. Se o agente pessoal não encontra o agente Organizador de Noites, pode contentar-se com os agentes intermediários de informação. Dessa forma, parte das necessidades do utilizador são satisfeitas; o problema, embora não possa ser totalmente resolvido, é parcialmente solucionado.

O agente Organizador de Noites pode não ser capaz de encontrar um dos agentes intermediários de informação, mas encontrará outro que o substitua. Mesmo que este seja pior, o problema não fica sem resposta.

## **2 Comércio Electrónico Medeado por Agentes**

Vários sistemas de agentes têm sido criados no domínio do comércio electrónico mediado por agentes [Sandholm e Wang 2002][Collins et al. 1999] [Tsvetovaty et al. 1997]. Os agentes pertencentes a aplicações de comércio electrónico poderão representar vendedores e compradores de bens e serviços, poderão representar casas de leilão, poderão suportar serviços de anúncio de intenções de compra e venda de bens e serviços, poderão suportar a negociação desinteressada entre partes com mútua desconfiança, poderão ser intermediários capazes de encontrar um vendedor que satisfaça as necessidades de um comprador e vice-versa.

Além destes agentes com funções directamente ligadas ao mercado, poderão existir outros agentes que prestam serviços úteis em domínios ligados ao comércio electrónico mas que não têm um interesse directo nas transações. Entre estes agentes, destacam-se os agentes de reputação que mantêm um registo da reputação de intervenientes na aplicação; os agentes de formação de coligações que são capazes de formar equipas de agentes capazes de satisfazer as necessidades de um dado cliente (introduzidas na aplicação através de um agente de representação pessoal ou de um agente de representação de uma empresa); e agentes capazes de manter registo dos compromissos incorridos por cada agente perante os outros com quem troca mensagens.

Os agentes de formação de coligações podem ser solicitados por agentes representantes de pessoas ou de empresas que necessitem de serviços complexos, por exemplo, a organização de uma reunião com vários intervenientes nacionais e estrangeiros com necessidade de tradução simultânea, com necessidade de marcação de hotel e de restaurante e possivelmente com a necessidade de organizar um passeio turístico na cidade numa das noites da reunião. A tecnologia e modelos usados na criação de coligações no domínio do comércio electrónico pode ser usada em muitos outros domínios.

Os agentes de reputação recebem informações de outros agentes sobre a qualidade do serviço prestado por terceiros agentes e podem ser consultados por qualquer agente como forma de filtrar possíveis parceiros de negócio. A capacidade de manter informação sobre a reputação de outros agentes pode ser usada e é útil noutros domínios, por exemplo no domínio da saúde.

Os agentes que mantêm registo dos compromissos podem consultar os serviços de transporte e encaminhamento de mensagens das plataformas de agentes para determinar que compromissos são criados pelos agentes pelo simples facto de trocarem mensagens uns com os outros. Há mensagens individuais, ou conjuntos de mensagens enviadas e recebidas que funcionam como contratos, isto é, que criam compromissos por parte de quem as envia ou recebe. A possibilidade de determinar os compromissos de cada agente perante os outros devido às suas interacções é um serviço útil em grande número de domínios de aplicação.

O vendedor e o comprador, que são os clientes finais deste tipo de aplicação, poderão participar neste tipo de sistema dinâmico através de simples agentes representantes. Toda a infraestrutura social e diversos mecanismos de negócio evoluem livre e autónomamente como as sociedades humanas, pela criação, remoção e alteração de serviços e de infraestruturas.

### **3 Arquitecturas de sistemas multi-agente**

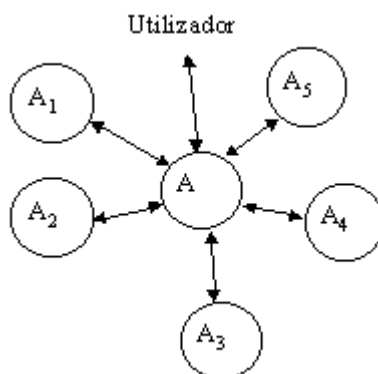
Os aspectos mais importantes a considerar em relação a sistemas multi-agente são a comunicação entre os agentes que integram o sistema, a coordenação do seu trabalho de modo que haja a possibilidade de enfrentar problemas que nenhum dos agentes do sistema seria capaz de resolver se estivesse isolado, e a estrutura social dos vários agentes no sistema.

Nas secções que se seguem apresentam-se quatro arquiteturas de sistemas multi-agente, dando ênfase especial aos aspectos da comunicação, da coordenação ou controlo das contribuições dos vários agentes para a resolução de problemas, e da topologia do sistema.

#### ***3.1 Controlo e Comunicação Centralizados***

A cooperação conseguida através de controlo centralizado não explora o paradigma multi-agente até às últimas consequências -- ainda é um tipo de arquitectura semelhante ao modelo computacional tradicional.

No controlo centralizado, tem-se um conjunto de agentes que são usados por um agente especial que tem a função de coordenar e integrar a acção dos vários agentes. Este modelo de controlo é semelhante ao que se passa num sistema com multiprocessamento em que um processo lança e sincroniza diversos sub-processos, recebe e integra os resultados das suas computações. Existem variantes a este paradigma centralizado puro, em que os vários agentes que são iniciados pelo agente coordenador comunicam entre si. A Figura 1 ilustra o que se passa na cooperação através de controlo centralizado.



**Figura 1 - Sistema Multi-Agente com controlo e comunicação centralizados**

Na Figura 1, cada círculo representa um agente, e cada arco representa um fluxo de informação e de controlo. Em geral, o agente controlador (i.e., agente A na figura) inicia um agente e “diz-lhe” o que espera que ele faça. O agente controlado efectua um conjunto de computações e envia ao agente controlador o resultado dessas computações. Daí que existam fluxos de informação do controlador para os outros agentes e destes para o controlador.

A comunicação com o exterior poderá estar a cargo do agente controlador, de alguns dos agentes especialistas, ou de ambos. Mas levando o paradigma Multi-Agente mais a sério, cada utilizador do sistema multi-agente deve ter um agente de comunicação específico. Consequentemente, a comunicação com o exterior estaria a cargo de agentes de utilizador.

Eventualmente, a integração de alguns dos resultados parciais produzidos pelos vários agentes especialistas estará a cargo de um ou mais agentes integradores especializados em diferentes formas de integrar resultados parciais. No entanto estas possíveis alterações afastam-se do conceito puro de arquitectura centralizada.

Na arquitectura centralizada, a topologia é fixa na altura do desenho do sistema. Não se prevê que possam surgir novos agentes nem se prevê que a interação entre eles mude ao longo do tempo.

No controlo centralizado de um sistema multi-agente, a autonomia dos agentes que constituem o todo fica muito reduzida, já que os agentes não podem tomar a iniciativa. É o agente controlador que se encarrega de chamar os agentes que quiser, quando quiser.

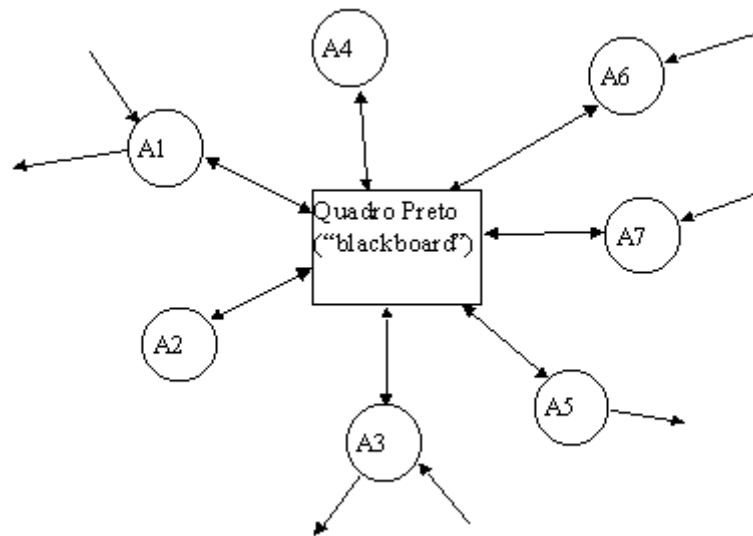
A principal vantagem desta abordagem baseada num agente controlador centralizado é a maior facilidade com que é possível impor uma sequência de acções desejada. Além disso, os agentes que constituem o sistema podem comunicar em linguagens diferentes, basta que o agente controlador e eventualmente os agentes integradores conheçam as linguagens faladas pelos agentes especialistas.

A desvantagem desta abordagem é a dificuldade associada ao acréscimo de novos agentes com novas capacidades. Se fosse necessário acrescentar novas capacidades ao sistema, ter-se-ia que criar mais agentes e modificar o agente controlador para que pudesse usar os novos agentes. Eventualmente, teria mesmo de se modificar alguns dos agentes já existentes.

### 3.2 *Arquitectura “Blackboard”*

No caso extremo de controlo totalmente descentralizado (não se pode falar propriamente de controlo), a cooperação emerge do comportamento dos diversos agentes e não existe nenhum agente encarregado de a impor. Uma das possibilidades mais divulgadas consiste em utilizar uma zona de memória acessível a todos os agentes do sistema. Tradicionalmente, esta memória partilhada chama-se quadro preto (“blackboard”).

Num sistema de quadro preto (Figura 2), existem agentes especializados em obter informação do exterior e de a escrever no quadro preto (sujeita a eventuais transformações). Os outros agentes especializados em processar informação examinam permanentemente o quadro preto e testam as condições em que estão preparados para actuar. Sempre que, no quadro preto, surgem condições reconhecidas por um agente, esse agente efectua as acções adequadas. A informação produzida por cada agente é escrita no quadro preto, resultando na eventual activação de outros agentes. Nesta arquitectura, podem existir agentes especialistas em integrar resultados parciais. É também necessário dispor de agentes encarregados de apresentar ao utilizador os resultados produzidos pelo sistema. Cada um desses agentes apresentadores deve ser especializado no modo de apresentação preferido pelo utilizador.



**Figura 2 - Sistema Multi-Agente com controlo descentralizado**

Na Figura 2, os agentes A<sub>1</sub>, A<sub>3</sub>, A<sub>6</sub> e A<sub>7</sub> obtêm informação do exterior que escrevem no quadro preto; A<sub>2</sub> e A<sub>4</sub> processam informação existente no quadro preto e produzem acções e nova informação que é escrita no quadro preto; A<sub>1</sub>, A<sub>3</sub> e A<sub>5</sub> apresentam os resultados produzidos ao seu utilizador específico.

Cada agente que lê o quadro preto tem que conhecer a linguagem usada por alguns dos outros agentes que produzem informação para o quadro preto.

A principal vantagem de um sistema Multi-Agente com controlo descentralizado é a facilidade com que se pode expandir o sistema. Para tal, pode bastar o acréscimo de agentes capazes de efectuar novas tarefas.

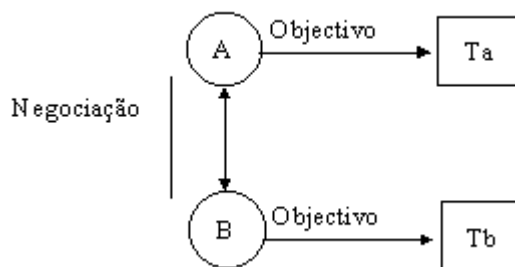
A arquitectura de quadro preto, os sistemas não tem uma topologia fixa. A cada momento podem surgir ou desaparecer agentes e as relações entre eles evoluem no tempo.

Num sistema multi-agente em que os vários agentes existem em computadores separados, o quadro preto tem que ser implementado através de um agente com a função específica de manter a informação produzida pelos outros agentes. Nestas circunstâncias poderá existir uma grande sobrecarga de comunicação entre o agente quadro preto e os outros agentes. Possivelmente, cada agente tem que se encarregar de enviar uma mensagem ao agente quadro preto perguntando se determinada condição se verifica. Quando o agente quadro preto recebe uma destas perguntas tem que verificar e responder ao agente interessado. Pelas razões apresentadas, a arquitectura quadro preto pode não ser uma boa solução quando os agentes residem em vários computadores, ou quando por qualquer outra razão, não podem partilhar zonas de memória comuns.

### 3.3 *Arquitectura de Negociação*

Em sistemas em que a cooperação se consegue através da negociação entre os vários agentes, existe uma forma de controlo totalmente descentralizado. Este tipo de controlo baseia-se na seguinte ideia. O agente A tem o objectivo de que a tarefa  $T_a$  seja executada, o agente B tem o objectivo de que a tarefa  $T_b$  seja executada, nenhum dos agentes consegue satisfazer o seu objectivo isoladamente, mas A e B podem cooperar para executar ambas as tarefas. Neste tipo de situações, a negociação consiste numa troca de mensagens entre A e B, em que A se propõe ajudar B a satisfazer o seu objectivo se, em troca, B ajudar A a satisfazer o seu objectivo (Figura 3). Se essa troca de mensagens permitir chegar a um acordo, os dois agentes passam a cooperar para satisfazer os objectivos de ambos.

Outra forma de negociar consiste no acerto de preços de prestação de serviços e não na permuta de objectivos. Nestes casos um agente negocia com outro agente a prestação de um dado serviço mediante um determinado pagamento.



**Figura 3 - Negociação com vista à cooperação para satisfação dos objectivos mútuos**

Os sistemas com arquitectura de negociação descritos não têm topologia fixa. Podem surgir novos agentes a todo o momento. Agentes existentes podem deixar de existir. As relações entre agentes são dinâmicas.

A negociação entre agentes envolve essencialmente dois tipos de dificuldades. Primeiro, para que o agente A saiba a quem deve propor uma cooperação, tem que ter um modelo de todos, ou pelo menos de parte dos agentes do sistema multi-agente. Esta necessidade pode conduzir a agentes com bases de conhecimentos muito extensas devido à representação (implícita ou explícita) de um grande número de modelos de outros agentes, para além do conhecimento sobre as tarefas do domínio. Se os agentes não possuírem modelos dos outros, as propostas de negociação têm que ser enviadas a todos os agentes, e esperar que os interessados estabeleçam o diálogo. Esta solução envolve uma comunicação massiva entre os agentes do sistema, o que levanta sérios problemas. O mais importante deles consiste em gerir as prioridades com que cada agente processa cada mensagem que lhe chega.

Uma outra solução consiste em criar agentes que prestam um serviço de *páginas amarelas*, isto é, agentes que mantêm uma descrição dos serviços prestados por outros agentes. Nestes casos, quando um agente necessita da prestação de um dado serviço, pede ao agente que presta o serviço de *páginas*



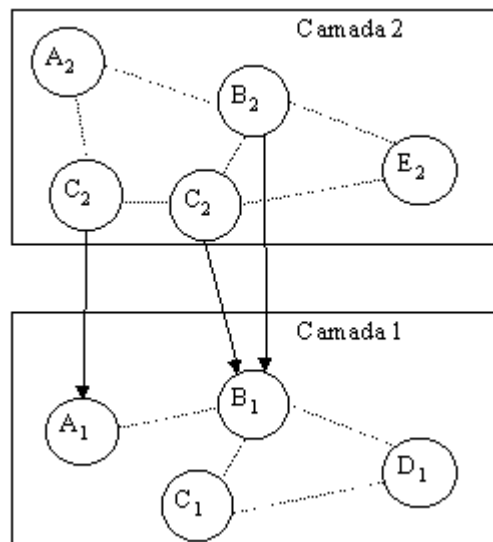
*amarelas* que lhe diga quais os agentes que executam a tarefa pretendida. Depois de obter a resposta, o agente entra em negociação directa com o outro agente.

O segundo problema tem que ver com a dificuldade prática de garantir que, após o agente B ter ajudado A a executar a tarefa  $T_a$ , o agente A ajuda o agente B a executar  $T_b$ . Para garantir que os contractos estabelecidos por negociação são cumpridos por ambas as partes, podem adoptar-se duas soluções (entre outras). Na mais simples, os agentes são programados para respeitarem todos os contractos com que se comprometem. Esta solução conduz à redução da autonomia dos agentes porque impede que eles possam adquirir outros objectivos mais prioritários do que cumprir um contracto estabelecido. A segunda solução consiste em impor um sistema normativo e punitivo, extra agente, que os obrigue a cumprir os seus contractos. Esta solução deixa de ser totalmente descentralizada, na medida em que o sistema normativo e punitivo pode ser visto como um agente especializado em controlar a acção dos outros agentes.

### 3.4 *Arquitectura de Subsunção*

Na arquitectura de subsunção (“subsumption architecture”), alguns agentes do sistema têm a possibilidade de exigir a alguns dos outros agentes que executem determinadas tarefas. Cada um dos agentes que recebem ordens de outros agentes tem dois estados internos possíveis. Num dos estados, não interrompe a sua actividade actual para atender as ordens recebidas; no outro estado, pode interromper a sua actividade actual para atender as ordens recebidas. Em geral, quando as ordens não são atendidas são ignoradas, embora se possam criar esquemas alternativos (e.g., cada agente pode ter uma fila de espera de ordens pendentes). Este tipo de controlo descentralizado chama-se “controlo por supressão e inibição” porque, dependendo do estado interno dos agentes do sistema, a produção de mensagens com ordens de processamento pode ser inibida, e as mensagens recebidas pelos agente receptores podem ser ignoradas (suprimidas).

Na sua versão original, a arquitectura de subsunção não era uma arquitectura baseada em agentes, mas sim uma arquitectura por camadas usada para organizar os diversos componentes de robôs. Cada camada era formada por uma rede de topologia fixa com diversos elementos. Cada elemento de cada camada poderia enviar ordens a determinados elementos da camada inferior (ligação ponto a ponto, Figura 4). Este tipo de arquitectura foi chamada arquitectura de subsunção porque um sistema com  $n$  camadas “herda” (i.e., inclui ou subsume) o comportamento do sistema formado pelas suas  $n-1$  camadas inferiores.

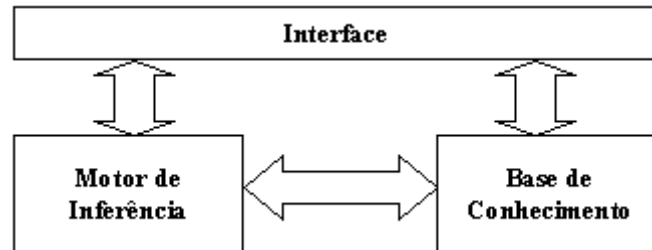


**Figura 4 - Arquitectura de Subsunção**

As vantagens desta arquitectura face à arquitectura baseada na negociação são essencialmente duas. Dado que cada camada tem uma arquitectura fixa, e que cada agente está ligado apenas a um número limitado de outros agentes, não surge a necessidade de manutenção de modelos internos dos outros agentes do sistema e diminui a necessidade de comunicação entre agentes.

### 3.5 *Arquitectura Interna dos Agentes*

Cada um dos agentes que compõem um sistema multi-agente pode ser mais ou menos inteligente e mais ou menos baseado em conhecimento. Na Figura 5 representa-se o caso extremo de agentes inteligentes baseados em conhecimento com uma arquitectura interna igual à dos sistemas baseados em conhecimento monolíticos. Noutro extremo da escala, os agentes de um sistema multi-agente podem também ser sistemas distribuídos formados por vários componentes. Em muitos casos, os componentes não são inteligentes ou são pouco inteligentes. A inteligência do sistema resulta da interacção e da cooperação entre todos os seus componentes.



**Figura 5 - Agente inteligente baseado em conhecimento**

## **4 Referências Bibliográficas**

- [Botelho et al 2003] Botelho, L.; Mendes, H.; Figueiredo, P.; e Marinheiro, R. 2003. "Send Fredo Off to Do this, Send Fredo Off to Do that". Proc. of the International Workshop on Cooperative Information Agents (CIA-2003).
- [Collins et al. 1999] Collins, J.; Sundareswara, R.; Tsvetovat, M.; Gini, M.; e Mobasher, B. 1999. "Search Strategy for Bid Selection in Multi-Agent Contracting". *Proceedings of IJCAI99 Workshop on Agent-mediated Electronic Commerce (AmEC-99)*
- [Dale e Ceccaroni 2002] Dale, J.; e Ceccaroni, L. 2002. "Pizza and a Movie: A Case Study in Advanced Web Services". *Proc. of the AAMAS2002 Workshop "Challenges in Open Agent Systems"*
- [FIPA 2002-23] Foundation for Intelligent Physical Agents. 2002. "FIPA Agent Management Specification". Report 00023. <http://www.fipa.org/specs/fipa00023/>
- [Sandholm e Wang 2002] Sandholm, T.; e Wang, X. 2002. "(Im)possibility of Safe Exchange Mechanism Design". *In Proc. of National Conference on Artificial Intelligence (AAAI2002)*
- [Tsvetovatyy et al. 1997] Tsvetovatyy, M., Gini, M., Mobasher, B. e Wieckowski, Z. 1997. "MAGMA: an agent-based virtual market for electronic commerce". *In Journal of Applied Artificial Intelligence* 11, pp. 501-524
- [Willmott et al 2001] Willmott, S.; Dale, J.; Burg, B.; Charlton, P; e O'Brien, P. 2001 "Agentcities: a worldwide open agent network". *Agentlink News*. 8:13-15