



**Universidade do Minho**

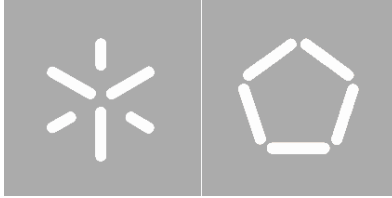
Departamento de Informática

## **Agentes Inteligentes**

Paulo Novais

Cesar Analide





**Universidade do Minho**

Departamento de Informática

## **Agentes Inteligentes**

Paulo Novais

Cesar Analide

Texto Pedagógico

Grupo de Inteligência Artificial

Centro de Ciências e Tecnologias da Computação



## **Resumo**

Neste documento são apresentados os conceitos fundamentais na área dos Agentes Inteligentes e dos Sistemas Multiagente, sendo expostas as principais arquitecturas para a concepção de agentes, estabelecendo um paralelo entre as noções de agente e de objecto. São abordadas, igualmente, as principais áreas de aplicação das tecnologias baseadas em agentes, assim como os elementos que desencadeiam a necessidade de agrupamento de agentes em estruturas sociais. São, ainda, expostos conceitos associados ao processo de comunicação entre agentes e apresentadas ferramentas para o desenvolvimento de agentes.



## Tabela de Conteúdos

A Inteligência Artificial.....	11
A Inteligência Artificial Distribuída .....	12
Estrutura do Documento.....	14
Agentes Inteligentes.....	15
Agentes.....	15
Agentes Inteligentes.....	16
Agentes versus Objectos .....	18
Agentes versus Sistemas Periciais.....	18
Agentes como Sistemas Intencionais .....	19
Tipos de Agentes de Software .....	19
Arquitecturas para Agentes de Software .....	20
Arquitectura Deliberativa.....	21
Arquitectura Reactiva.....	22
Arquitectura Híbrida .....	22
Arquitectura BDI.....	23
Sistemas Multiagente .....	25
Plataformas de Comunicação.....	26
Linguagens de Comunicação entre Agentes de Software .....	28

Ontologias.....	33
Sistemas Baseados em Quadros Negros.....	33
Áreas de Aplicação.....	34
Conclusões .....	35
Ferramentas para a Construção de Agentes .....	37
JESS.....	37
SICStus-Prolog .....	38
CORBA.....	38
JINI.....	39
FIPA-OS .....	40
JADE.....	40
ZEUS .....	41
Outras Ferramentas .....	41
Conclusões .....	42
Conclusões .....	43
Exercícios.....	45
Referências .....	47
Anexo.....	51



## Lista de Figuras

Figura 1: Agente em interacção com o meio.....	16
Figura 2: Tipologia de um agente .....	20
Figura 3: Arquitectura Deliberativa .....	21
Figura 4: Arquitectura Reactiva .....	22
Figura 5: Arquitectura Híbrida .....	23
Figura 6: Arquitectura BDI.....	24
Figura 7: A Linguagem KQML como linguagem por camadas.....	30
Figura 8: Tabela sumária de algumas performatives reservadas, para um emissor E e um receptor R .....	31
Figura 9: Tabela com alguns parâmetros reservados e os seus significados .....	31
Figura 10: Arquétipo de um Sistema Baseado em Quadros Negros .....	34



## Abreviaturas

AMR	Agent Message Router
BDI	Beliefs Desires e Intentions
CORBA	Common Object Request Broker
DCOM/OLE	Distributed Component Object Model/Object Linking Embedding
FIPA ACL	Foundation for Intelligent Physical Agents - Agent Communication Language
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
IDL	Interface Definition Language
JATLite	Java Agent Template Lite
JESS	Java Expert System Shell
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
RCDP	Resolução Cooperativa e Distribuída de Problemas
RPC	Remote Call Procedure
SAT	Speech Act Theory
SMA	Sistemas Multiagente
SP	Sistemas Periciais
XML	Extended Markup Language



## **A Inteligência Artificial**

A actividade humana está directamente ligada à inteligência, sendo esta requerida ao nível das mais diversas tarefas, nomeadamente na compreensão das linguagens, na capacidade de expressão e na aprendizagem [Ganascia, 1993].

No Dicionário da Língua Portuguesa, da Porto Editora [Costa & Melo, 1997], é apresentada a seguinte definição:

Definição 1: Inteligência: [...] faculdade de compreender, um talento, o raciocínio, a habilidade [...]

É difícil definir inteligência pois está coberta por uma áurea de mistério, que lhe advém, um pouco, de uma componente que emana da mente, do talento e da habilidade. A questão é mais séria e complexa quando se estudam as formas para a sua implementação em processos computacionais ou entidades inteligentes, descrevendo a atitude mental e desenvolvendo o talento e a habilidade. Os investigadores e cientistas têm aplicado os seus esforços no desenvolvimento de sistemas de computação com o objectivo de executarem tarefas similares às que um humano realiza, e tais sistemas possuem um certo grau de inteligência, i.e., uma Inteligência Artificial (IA) [Coelho, 1995].

A IA é entendida como sendo, simultaneamente, uma ciência e uma engenharia que faz (ou desenvolve) máquinas inteligentes, especialmente robôs. Isto leva a uma redefinição da inteligência como sendo a componente computacional da habilidade de realizar tarefas [Charniak & McDermott, 1985].

O relacionamento da IA com o desenvolvimento de sistemas computacionais para a execução de tarefas, habitualmente realizadas por humanos, não é mais do que uma simulação do comportamento de um humano na resolução do problema [Russel & Norvig, 1995]. Este relacionamento reflecte-se no desenvolvimento de técnicas que permitem conceber e programar máquinas que procuram mimetizar e estender as capacidades cognitivas do ser humano [McCarthy & Hayes, 1969].

Definição 2: A Inteligência Artificial é o ramo da Ciência da Computação que tem a ver com a automatização de comportamentos inteligentes, próprios dos seres humanos [Luger & Stubblefield, 1998].

Definição 3: Conhecimento: [...] forma do entendimento que emana do acto de conhecer [...] [Costa & Melo, 1997].

O conhecimento foi reconhecido como o principal factor de sucesso dos sistemas inteligentes, nomeadamente a partir da década de 70. Mais recentemente, há novas leituras para a questão do conhecimento, como por exemplo a gestão do conhecimento e o conceito de sociedade do conhecimento. Pode-se dizer que o conhecimento beneficia o desempenho dos sistemas e favorece a tomada de decisão. Mais conhecimento permite tomar melhores decisões (automaticamente ou não), bem balanceadas com menor risco e a menor custo. O conhecimento pode ser obtido através de peritos ou automaticamente a partir de dados, em algumas situações usando técnicas de descoberta de conhecimento.

Para que um sistema actue é necessário que possua informação. A posse de informação é determinada, de alguma forma, por mecanismos de armazenamento, o que pressupõe a sua representação interna. Em IA, tanto o conhecimento como a forma de o representar são um campo aberto à investigação [Coelho, 1995].

## **A Inteligência Artificial Distribuída**

Nos anos 70, a investigação em IA tinha feito significativos progressos em diferentes áreas do saber, em particular através do emprego de métodos de trabalho que faziam recurso à utilização de conhecimento de uma forma intensiva. Por outro lado, o sucesso conseguido com diferentes tipos de sistemas, tais como os sistemas de produção, em que o conhecimento é codificado e tratado a partir de estruturas de dados de baixa complexidade, altamente manipuláveis, conduziu a tentativas de construção de sistemas modelares que exploram a metáfora dos especialistas cooperantes. Um bom exemplo disto são os sistemas ACTORS [Agha, 1986] que capturaram, então, o cerne dos interesses. Lançando as pontes entre este tipo de sistemas e o trabalho então em curso sobre as redes de computadores, criou-se o enquadramento necessário para o influxo da tecnologia de redes, levando os sistemas distribuídos baseados em conhecimento a afirmarem-se como um ramo autónomo da pesquisa em IA [Santos, 1999] [Machado, 2001].

Embora constituindo um novo paradigma para a resolução de problemas, por força das raízes históricas em que era, então, desenvolvido o trabalho em IA, os primeiros investigadores em Inteligência Artificial Distribuída (IAD) adoptaram uma postura em tudo análoga à da IA, i.e., dado um problema, como poderiam construir sistemas, preferencialmente distribuídos, para a sua resolução? A ênfase deste esforço era colocada no problema e no modo de administrar os diferentes intervenientes para que trabalhassem em equipa. Procurava-se, assim, utilizar formas de organização social previamente conhecidas e de planeamento de tarefas em tempo real, de modo a potenciar e aumentar a produtividade do trabalho em grupo.

A IAD tornou-se numa área de investigação da maior importância, juntamente com outras áreas tradicionais da IA, a Representação de Conhecimento e a Aprendizagem. Hoje, a IAD integra conceitos e ideias de áreas

tão diversas como a Sociologia, a Economia e a Filosofia, para além da própria IA. Este domínio tão alargado e de natureza multidisciplinar, torna a sua definição difícil. A IAD deu inicialmente origem a duas áreas de investigação [Durfée & Rosenshien, 1994]: os Sistemas Multiagente (SMA) e a Resolução Cooperativa e Distribuída de Problemas (RCDP). Na RCDP, a funcionalidade é decomposta e distribuída por módulos, podendo o controlo ser centralizado. Num SMA, para além da distribuição e da descentralização da execução por vários módulos ou entidades, aqui designados por agentes, o controlo é distribuído por esses agentes. Estes, além de seleccionarem as suas próprias acções, socializam-se e cooperam.

Em particular, podem-se destacar as seguintes sub-áreas na IAD:

- A Inteligência Artificial Paralela;
- A Resolução Cooperativa e Distribuída de Problemas; e
- Os Agentes e os Sistemas Multiagente.

Definição 4: A Inteligência Artificial Distribuída é o estudo, a construção e a aplicação de Sistemas Multiagente à resolução de problemas, i.e., sistemas nos quais vários agentes inteligentes interagem perseguindo um conjunto de objectivos ou realizando um conjunto de tarefas [Weiss, 1999].

A IAD deverá caminhar no sentido de desenvolver mecanismos e métodos de resolução de problemas que permitam às entidades aí envolvidas (os agentes) melhorarem os processos de interacção entre si e com terceiros, i.e., há que potenciar e compreender os processos de comunicação entre entidades inteligentes. Os computadores actuais já não configuram sistemas isolados; estão, antes, interligados em rede e com os utilizadores. As plataformas de computação e sistemas de informação actuais são heterogéneos, abertos e distribuídos. Perante esta crescente complexidade, as tecnologias de IAD surgem como forma de gerir este elevado número de particularidades. Pode-se, então, reflectir sobre as razões que nos levam a optar pela vertente IAD/SMA [Weiss, 1999]:

- dispersão – a problemática da distribuição de tarefas subjacente a muitas aplicações, tem uma componente semântica, funcional, temporal e espacial;
- complexidade – os problemas são demasiados complexos para se pensar na sua resolução através de um processo centralizado, tanto por questões tecnológicas como por questões algorítmicas.

Como foi referido anteriormente, os computadores eram máquinas que, embora se limitassem a executar sequências de operações de forma mais ou menos eficiente, não tinham vontade própria. No entanto, a necessidade de lidar com ambientes altamente dinâmicos e onde o nível de abstracção é incomensuravelmente superior ao da instrução-máquina, levou ao aparecimento do conceito de agente inteligente, autónomo e flexível, i.e., uma entidade que procura resolver problemas através de processos inteligentes, com capacidade de decisão própria e capacidade de aprendizagem.

Definição 5: Um agente é um sistema computacional (de software, hardware ou misto) que, estando situado num determinado ambiente, actua sobre este através de acções autónomas tendo em vista atingir um dado objectivo [Weiss, 1999].

### **Estrutura do Documento**

Após este primeiro capítulo de introdução à temática da Inteligência Artificial, no capítulo sobre Agentes Inteligentes faz-se uma abordagem à computação baseada em agentes e aos Sistemas Multiagente, apresentando-se as características, propriedades e arquitecturas em que assentam estas entidades.

No capítulo de Ferramentas para a Construção de Agentes são apresentadas ferramentas habitualmente utilizadas para o desenvolvimento de agentes sendo abordados, ainda, sistemas integrados de agentes.

No capítulo final são apresentadas as principais ideias com que se conclui a discussão sobre agentes inteligentes.

Neste documento pode-se encontrar, ainda, uma série de exercícios e, em anexo, um conjunto de diapositivos utilizados como suporte à discussão do tema deste documento.



## **Agentes Inteligentes**

Um agente assume-se como uma figura metafórica utilizada em inúmeras áreas do conhecimento humano, que vão desde a Psicologia, a Economia e a Sociologia, até às Ciências da Computação. Nestas últimas, o conceito de agente foi introduzido pela mão da comunidade da Inteligência Artificial (IA), em especial por intermédio do sistema Actor [Hewitt, 1977]. Foram apresentadas, pela primeira vez, entidades designadas como objectos auto-contidos, interactivos e concorrentes, que comunicavam por troca de mensagens e que demonstravam possuir um certo tipo de comportamento. Designaram-se esses objectos por actores.

O avanço tecnológico, em matéria de computação e de comunicação, permitiu o aparecimento da Inteligência Artificial Distribuída, passível de se subdividir, em termos da resolução de problemas, em Sistemas de Processamento Distribuído e nos Sistemas Multiagente [Bond, 1990]. As aplicações do tipo distribuído são apelativas quando aplicadas na obtenção de soluções para problemas que contemplam múltiplos intervenientes e encapsulam a complexidade e as especificidades próprias dos sistemas físicos que lhes deram origem. Os intervenientes têm uma base de conhecimento individual ou partilhada, com capacidade de processamento e de aquisição de conhecimento, assim como de formas de raciocínio, e são por isso catalogados como inteligentes, i.e., partindo de uma definição muito genérica de agente, pode-se afirmar que estes são entidades em que os humanos delegam tarefas a serem executadas, relacionadas com a sua capacidade de cálculo, de memorização, ou de perseverança, apenas para nomear umas entre tantas.

### **Agentes**

Definir o conceito de agente tem sido tarefa de difícil concretização, devido ao facto de um agente ser usado para caracterizar muitas e diversas espécies de entidades. Resulta, daí, que as múltiplas definições existentes na literatura se prendam com inúmeros e distintos contextos de utilização e de desenvolvimento de aplicações.

No dicionário da Língua Portuguesa [Costa & Melo, 1997] é apresentada uma definição de agente na forma:

Definição 6: Agente: [...] algo que age, capaz de produzir um efeito [...].

Uma outra definição passa por considerar um agente como algo que obtém informação e conhecimento do seu universo de discurso através de sensores e actua nesse ambiente através de actuadores [Russel & Norvig, 1995] (Figura 1). Pode-se dizer que os agentes são componentes persistentes e activos que percebem, raciocinam, actuam e comunicam [Huhns & Singh, 1998]. Pode-se perceber que os agentes são entidades que habitam em ambientes complexos, sentem esse ambiente e actuam de modo autónomo, procurando executar um conjunto de tarefas para as quais receberam procuração [Maes, 1990], [Novais et al., 2000].



**Figura 1: Agente em interacção com o meio**

Na comunidade científica existe, no entanto, um certo acordo sobre uma das facetas dessas entidades: a que se prende com o conceito de autonomia. Por exemplo, Wooldridge [Wooldridge, 1999] afirma que um agente corporiza um sistema computacional capaz de uma acção flexível e autónoma, desenvolvido num determinado meio ou sobre um dado universo de discurso, o que se apresenta como uma definição mais abrangente (entende-se por flexível um agente que é reactivo, pro-activo e com capacidade de socialização).

## **Agentes Inteligentes**

Wooldridge e Jennings [Wooldridge & Jennings, 1995] apresentam aquilo a que nos dias de hoje se pode definir como o fio condutor que levou a uma das primeiras formulações e quantificações, sob um ponto de vista formal, da noção de agente ou Sistema Multiagente, em termos de propriedades do foro antropopático [Analide & Neves, 2002], e adjectivados como sendo de fracas ou fortes.

Em particular, um agente ou Sistema Multiagente é considerado fraco quando não apresenta um conjunto mínimo de atributos, a partir dos quais se possam definir e quantificar um certo conjunto de atitudes como, por exemplo:

**Autonomia:** os agentes operam sem a intervenção de outros agentes, e têm controlo sobre as suas acções e o seu estado de conhecimento interno,

**Reactividade:** os agentes têm percepção do que ocorre no seu universo de discurso e respondem adequada e atempadamente a mudanças ocorridas nesse ambiente.

**Pro-actividade:** os agentes são capazes de tomar a iniciativa, conduzindo as suas próprias acções segundo um comportamento que é dirigido por objectivos.

**Sociabilidade:** os agentes interactivam com outros agentes, comunicando com estes, competindo ou cooperando na resolução de problemas que entretanto lhes tenham sido colocados.

A noção de agente fraco tem a sua génese em áreas como a Computação Distribuída e a IA, onde se entende o agente como um paradigma para a automação dos procedimentos de cooperação, situações que se desenvolvem quando se olha para a resolução de problemas em ambientes distribuídos.

Por outro lado, um agente ou Sistema Multiagente é considerado forte, quando a(s) entidade(s) com que nos deparámos é(são) eminentemente cognitiva(s), passível(eis) de desenvolver a sua própria consciência, de se apresentar como tendo um conjunto de mais-valias como a perceptibilidade, a sentimentalidade e o emocionismo [Wooldridge & Jennings, 1995]. Por conseguinte, um conjunto mínimo de atributos tem, de ser considerado, e em que se destacam [Ferber, 1999] [Nwana, 1996] [Russel & Norvig, 1995]:

**Mobilidade:** um agente diz-se móvel quando se movimenta através da rede executando as tarefas de que foi incumbido e cumprindo objectivos.

**Intencionalidade:** a intencionalidade é a capacidade que o agente apresenta para a definição de objectivos assim como de estratégias para os atingir. Esta atitude é analisada em moldes semelhantes aos apresentados na qualificação de mobilidade.

**Aprendizagem:** a aprendizagem é a capacidade que o agente apresenta para adquirir conhecimento. A actualização da base de conhecimento é feita através da assimilação de padrões de comportamento ou de preferências manifestadas pelo próprio agente.

**Competência:** um agente é competente quando conduz com sucesso e eficiência as tarefas que é incumbido de realizar. A competência está normalmente relacionada com a confiança depositada no agente, por terceiros.

**Veracidade:** um agente não fornece, propositadamente, informação falsa.

**Racionalidade:** um agente racional não aceita realizar tarefas que lhe pareçam impossíveis de executar, contraditórias com os seus princípios (e.g., ética), ou quando não são compensados em termos do risco, custo e esforço envolvido.

**Benevolência:** um agente benevolente adopta como seus, os objectivos de terceiros, desde que estes não entrem em conflito com os que perfilha. Os agentes benevolentes são levados a realizar todas as tarefas que lhe sejam imputadas.

**Emotividade:** certas características próprias do ser humano têm vindo a migrar e a constituir-se como parte integrante de agentes e Sistemas Multiagente [Analide & Neves, 2002]. Os primeiros passos nesta área foram dados, entre outros, por Rao e Georgeff [Rao & Georgeff, 1995], que introduziram as arquitecturas do tipo BDI (i.e., Beliefs, Desires, Intentions).

### **Agentes versus Objectos**

Sempre que se fala em agentes e em objectos existe, não raras vezes, uma falta de rigor nos conceitos e uma falta de clareza nas definições. Importa realçar que objectos são entidades que encapsulam um dado estado de conhecimento, são capazes de realizar operações sobre esse estado, e comunicam entre si por troca de mensagens [Wooldridge, 1999] [Jennings, 2001].

Uma das principais diferenças está relacionada com o grau de autonomia. Os objectos têm controlo sobre o seu próprio estado de conhecimento, mas não têm controlo sobre o seu funcionamento. Os agentes controlam o seu estado e o seu comportamento, são autónomos e, ao contrário dos objectos, podem recusar pedidos de assistência.

Uma outra distinção situa-se ao nível da comunicação. Nos objectos a definição dos mecanismos de comunicação dá-se através da indicação dos métodos que podem ser invocados, o que faz com que estes procedimentos sejam intrínsecos dos objectos em que se desencadeiam. Nos agentes, o conceito de mensagem e da linguagem a utilizar é definido ao nível do colectivo ou comunidade de agentes, e não individualmente.

Em termos de comportamento, os agentes são flexíveis, cultivam a autonomia, percebem o seu ambiente e agem de acordo com os seus objectivos.

Em termos gerais, a programação orientada ao objecto reporta-se a um paradigma de programação, enquanto que a computação baseada em agentes consubstancia um paradigma de computação.

### **Agentes versus Sistemas Periciais**

A tecnologia dos Sistemas Periciais (SP) corporiza uma das mais importantes tecnologias desenvolvidas na área científica da IA nos anos 80 [Hayes-Roth et al., 1993]. Um SP tem como principal objectivo o apoio à resolução de problemas e à tomada de decisão num determinado domínio de conhecimento, actuando essencialmente como um consultor, ou seja, os SP:

- lidam com uma representação do ambiente ou universo de discurso, não o manipulando directamente, nem sentindo no imediato o resultado das suas acções;

- destinam-se a assistir os peritos na resolução de problemas numa determinada área do conhecimento, enquanto que os agentes resolvem problemas;
- aplicam-se a tarefas de alto nível, enquanto que os agentes se dedicam a tarefas comuns;
- não são pro-activos nem autónomos, respondendo de modo passivo e executando sempre a mesma acção.

### **Agentes como Sistemas Intencionais**

Às definições de agente apresentadas anteriormente são usualmente acrescentadas propriedades geralmente atribuídas a humanos. Um agente pode, assim, ser caracterizado como um sistema intencional, o qual se socorre dessas propriedades para guiar as suas atitudes no processo de resolução de problemas, e que vão desde as crenças, as intenções, os desejos, a formas de compromisso [Shoham, 1993].

### **Tipos de Agentes de Software**

Quanto à mobilidade têm-se dois tipos de agente, ou seja:

- Agentes Móveis, que se movimentam entre diferentes equipamentos ou redes (e.g., na Internet);
- Agentes Estacionários, adstritos a um equipamento, do qual não podem migrar.

Quanto às tarefas que realizam têm-se, igualmente, dois tipos de agente, ou seja:

- Agentes Pessoais, agentes que realizam tarefas de apoio ao utilizador (podendo mesmo substituí-lo) ao organizar e disponibilizar, de acordo com os requisitos e preferências deste, informação recebida de diferentes fontes de conhecimento;
- Agentes de Serviço, agentes que se encarregam de tarefas de natureza repetitiva, não estando, em princípio, na ribalta.

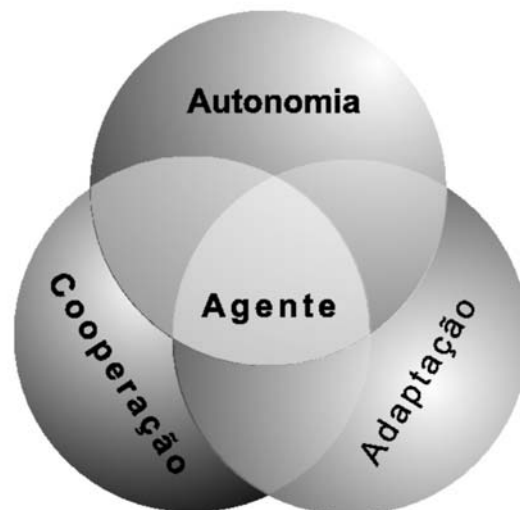
Quanto ao modo de selecção, há que considerar diferentes tipos de agente, como:

- Agentes Reactivos, que agem de acordo com a percepção que têm do mundo ou universo de discurso, através da invocação de um conjunto de regras de percepção/acção;
- Agentes com Memória, que baseiam as suas tomadas de decisão em memórias do passado;
- Agentes orientados por Objectivos, que usam, para uma tomada de decisão, para além do conhecimento que têm acerca do mundo ou universo de discurso, informação particular acerca dos problemas em equação (a qual é obtida mediante a invocação de procedimentos, de análise e/ou descoberta de conhecimento);
- Agentes baseados na Utilidade, que se socorrem, quando há que decidir entre diferentes objectivos a cumprir, ou então quando se está face a uma situação de (potencial) conflito, de medidas de satisfação pessoal;

- Agentes Adaptativos, que alteram o seu modo de interactuar com o seu mundo ou universo de discurso ao longo do tempo.

Uma outra possibilidade de classificação vem de Nwana [Nwana, 1996], baseando-se na presunção de que um agente deve subscrever um conjunto mínimo e ideal de atributos, tais como a autonomia, a cooperação e a aprendizagem/adaptação (Figura 2). Esta visão vai de encontro à de Wooldridge e Jennings [Wooldridge & Jennings, 1995], e contempla um conjunto de entidades, tais como:

- Agentes Colaborativos;
- Agentes de Interface;
- Agentes de Aprendizagem;
- Agentes Inteligentes.



**Figura 2: Tipologia de um agente**

Em resumo, um agente deve, em princípio, apresentar:

- A capacidade de aquisição de conhecimento;
- Um comportamento orientado por objectivos;
- Uma “rara” facilidade em adquirir novas aptidões.

### **Arquitecturas para Agentes de Software**

Quando alguém se debruça sobre o tema arquitecturas para agentes de software está, implícita ou explicitamente, a reportar-se a entidades com capacidades deliberativas, reactivas e/ou pro-activas, i.e., uma arquitectura para agentes de software trata tais entidades, sob um ponto de vista conceptual, como sendo formadas por uma amálgama de termos do foro predicativo, consolidados através de programas lógicos ou teorias da percepção, acção e formas de raciocínio. Neste contexto, tem-se que a percepção alimenta tais

programas em lógica ou teorias (um alimento que se traduz em teoremas a demonstrar), os quais por sua vez condicionam a acção (e.g., o que é que se irá perceber a seguir).

A arquitectura de um agente é dada pelo plano das suas unidades funcionais e pela forma de organização e interacção [Maes, 1990]. Por outro lado, várias são as questões que se levantam quando se aborda esta problemática, sendo de salientar:

- De que modo é que um agente deve seleccionar a acção ou acções a serem executadas?
- De que forma é que deve ser abordado e equacionado o problema da representação de conhecimento, e que mecanismos ou tipos de raciocínio se irão utilizar?
- Como é que irá um agente comunicar e colaborar com terceiros?
- Deverá um agente ter capacidades de auto-aprendizagem?

É neste contexto que se deve entender a contribuição da IA para a definição de uma taxionomia para uma seriação dos ambientes computacionais mais adequados a agentes ou SMA e em que pontuam as arquitecturas deliberativas, reactivas e híbridas [Wooldridge & Jennings, 1994].

### Arquitectura Deliberativa

Um agente deliberativo é utilizado para resolver problemas em que há precedência na acção, socorrendo-se, normalmente, de uma representação simbólica e/ou predicativa do mundo ou universo de discurso, cujo estado de conhecimento pode ser modificado por alguma forma de raciocínio lógico-matemático, i.e., a selecção de qual a acção a desencadear, num determinado instante, é baseada em formas de raciocínio, quais manipulações puramente sintácticas, sobre essas representações (Figura 3).



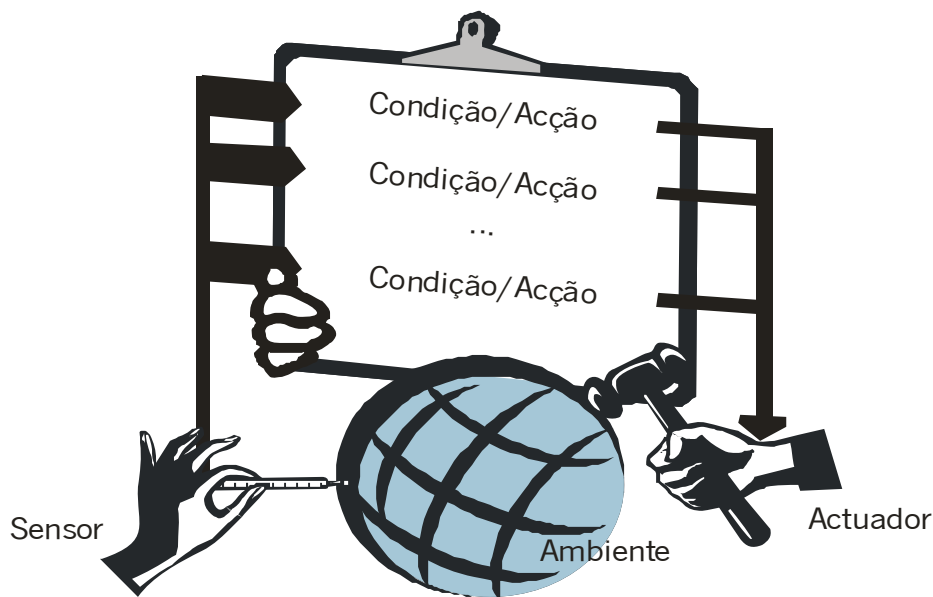
**Figura 3: Arquitectura Deliberativa**

A ideia de agente deliberativo baseado em puro raciocínio lógico-matemático é extremamente atraente. No entanto, dois importantes problemas têm-se vindo a colocar:

- Um problema de transposição e/ou representação: como traduzir o mundo real em termos de um programa em lógica ou teoria?
- Um problema de raciocínio: como levar os agentes a raciocinar?

### Arquitectura Reactiva

Um agente reactivo não possui qualquer representação simbólica e/ou predicativa do mundo ou universo de discurso, nem requer formas de raciocínio complexas, i.e., um agente comporta-se como um autómato inserido no meio ambiente que o rodeia. Procura lidar com a percepção que tem do mundo através da recepção de itens de informação do tipo atómico, que lhe são passadas através de sensores. Não existindo representações do tipo simbólico nem mecanismos ou formas de raciocínio, o comportamento inteligente do sistema pode ser visto como o resultado da invocação de regras do tipo condição/acção, quando indexadas a certas formas de comportamento (Figura 4).



**Figura 4: Arquitectura Reactiva**

Um exemplo paradigmático desta situação passa pela Arquitectura de Subsunção [Brooks, 1986], em que se tem um sistema organizado em camadas, em que cada camada responde por certo tipo de comportamento do sistema, i.e., as camadas do sistema têm capacidade própria de percepção/acção, podendo ser invocadas de forma concorrente [Russel & Norvig, 1995].

### Arquitectura Híbrida

Os agentes reactivos são de certa forma redutores, o que dificulta a implementação de estratégias direccionadas para atingir um certo objectivo. Por outro lado, os agentes deliberativos ao apresentarem



formas de raciocínio e/ou de representação de conhecimento porventura complexas, tornam-se, não raras vezes, incapazes de reagir de uma forma rápida a estímulos vindos do exterior.

Não será de estranhar, por conseguinte, que a combinação das virtualidades das arquitecturas baseadas em agentes deliberativos e reactivos dê corpo à construção de arquitecturas que apresentem ou subscrevam certas formas de hibridismo. A ideia principal passa por categorizar as funcionalidades do agente em camadas dispostas hierarquicamente, onde à camada reactiva é atribuída alguma forma de prioridade sobre a deliberativa, de tal modo que se aproveite uma das suas características mais importantes, que é o de dar uma resposta rápida a eventos detectados no ambiente ou universo de discurso (Figura 5).



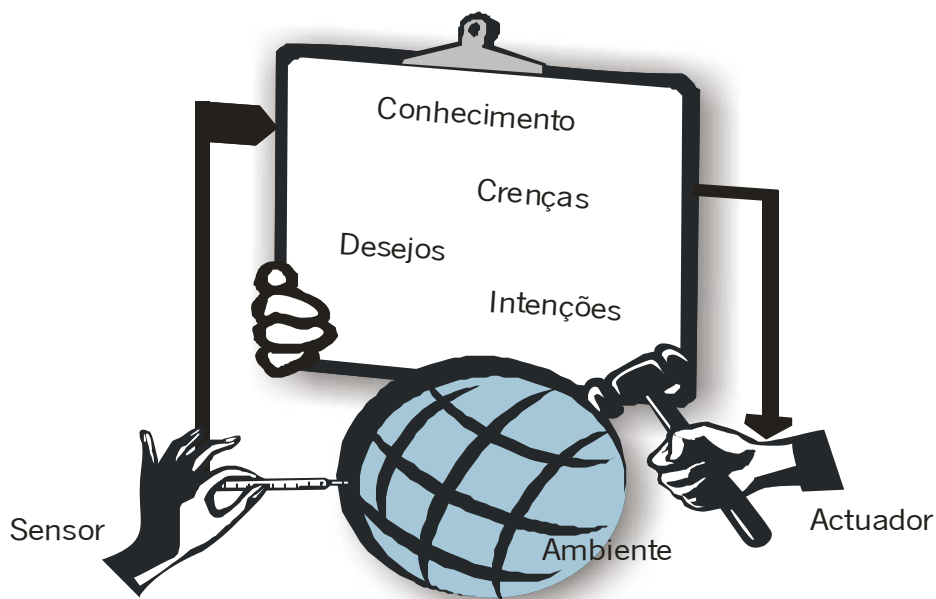
**Figura 5: Arquitetura Híbrida**

Esta arquitectura apresenta significativos ganhos em termos de adaptabilidade, robustez e desempenho, quando comparada com as que estiveram na sua génese, sendo frequentemente usada em sistemas robóticos.

### **Arquitetura BDI**

Uma arquitectura para agentes de software que suporte a concretização das Crenças, Desejos e Intenções (uma arquitectura BDI – Beliefs, Desires, Intentions) de um agente, é claramente uma arquitectura deliberativa [Wooldridge, 2000].

Nesta abordagem à problemática subjacente à construção de arquitecturas para agentes de software, tem-se como objectivo obter aquela que melhor se adequa à descrição do estado interno de conhecimento de um dado agente com base nos seus estados mentais, estados estes de que o agente fará uso para determinar o seu curso de acção [Rao & Georgeff, 1995] (Figura 6).



**Figura 6: Arquitectura BDI**

#### Crenças

- a aceitação de que uma afirmação é verdadeira ou então de que algo existe (e.g., a crença de que o clima pode ser modificado pelo homem de uma forma benéfica para a sociedade);
- alguma coisa que é aceite como verdadeira ou real;
- uma opinião ou convicção firmemente cimentada (e.g., qualquer um deve estar preparado para lutar pelos seus ideais). Ora esta situação surge como que a contrapor-se àquela em que, por exemplo, uma dada população acredita que as condições em que actualmente se processa o tráfego rodoviário são as adequadas.

#### Desejos

- um forte sentimento de se querer ter algo ou se desejar que alguma coisa aconteça (e.g., a Joana não pretende tornar-se esposa e mãe);
- appetite.

Os desejos de um agente referem-se ao estado de conhecimento ou satisfação que o agente deseja atingir, não sendo necessário conhecer a forma como o pretende fazer. O desejo é sinónimo de motivação.

#### Intenções

- ter um plano de acção, objectivo ou intenção (e.g., a empresa X pretende despedir um certo número Y de colaboradores);
- designar ou destinar alguém ou alguma coisa como forma para se atingir um dado objectivo ou fim (e.g., o peru tem como fim destinar-se à alimentação do ser humano);

- obtenção de construções do tipo lógico-matemático a partir do momento em que se fixa a mente num certo objecto, mundo ou universo de discurso.

As intenções referem-se a um conjunto de acções ou tarefas que o agente seleccionou, e que o podem ajudar na concretização dos seus objectivos. Um conjunto de intenções é o resultado de uma deliberação.

A título exemplificativo, apresentar-se-á uma construção BDI simplificada, tendo em linha de conta que o agente é um dos alunos matriculados na disciplina de Sistemas Multiagente, do 4º ano da Licenciatura em Engenharia de Sistemas e Informática, do Conselho de Cursos de Engenharia, da Universidade do Minho, a qual é dada na forma [Novais, 2003]:

- Crenças: o conjunto  $\{ c_1, c_2, c_3 \}$  em que  $c_1$  = “acabar o curso é importante”,  $c_2$  = “quem acaba o curso com boa nota tem melhores hipóteses de emprego”,  $c_3$  = “a disciplina de Sistemas Multiagente é estruturante”;
- Desejos: o conjunto  $\{ d_1, d_2, d_3 \}$  em que  $d_1$  = “passar à disciplina de Sistemas Multiagente”,  $d_2$  = “acabar o curso”,  $d_3$  = “acabar o curso com uma boa classificação”;
- Intenções: o conjunto  $\{ i_1 \}$  em que  $i_1$  = “concluir a disciplina de Sistemas Multiagente com uma boa classificação”.

## Sistemas Multiagente

O estudo dos Sistemas Multiagente (SMA) teve a sua génese na área da Inteligência Artificial Distribuída (IAD) volvidos são 25 (vinte e cinco) anos. Os SMA têm-se vindo a afirmar-se como uma das áreas do conhecimento em que se realiza investigação de qualidade, surgindo a todo o momento novos produtos, quer seja nos sectores comercial, industrial ou de serviços, ou mesmo em termos de investigação e desenvolvimento. Por outro lado, o crescimento exponencial da Internet e a sua propensão para ligar entre si pessoas e máquinas, está a fazer com que se justifique esta nova era, a dos SMA, como consequência directa de tal grau de interconectividade. Contudo, construir SMA requiere muito mais do que a compreensão de protocolos de comunicação e paradigmas computacionais associados à programação distribuída. É preciso dominar tecnologias que têm a ver com as linguagens de comunicação entre agentes, arquitecturas de agentes, resolução de problemas em ambientes distribuídos, concepção de protocolos, assim como ser perito em teoria das organizações [Novais, 2003].

Minsk, em *The Society of Mind* [Minsky, 1986], lançou os SMA como a base para a concretização da inteligência em sistemas computacionais, assumindo, contudo, que cada entidade constituinte pode ser o mais simples possível e que, da sua interacção, podem emergir novas formas de inteligência.

De [Durfee & Rosenchien, 1994] [O'Hare & Jennings, 1996] podemos retirar que um SMA pode ser definido na forma:

Definição 7: Um Sistema Multiagente compreende um conjunto de entidades que cooperam de forma a solucionar um dado problema; o que, normalmente, estará para lá das suas capacidades individuais.

A importância dos SMA assenta na especificidade de cada agente e nas suas capacidades de relacionamento. A interacção entre agentes num SMA obriga, assim, à análise, definição, especificação e implementação de um certo conjunto base de funcionalidades, aqui concretizadas em termos das camadas:

- A plataforma de comunicação;
- A linguagem de comunicação;
- A ontologia.

Em resumo, será através da activação de protocolos de comunicação que se efectuará a troca e a descodificação de mensagens. Por outro lado, são também necessários protocolos de interacção que tornem possível o diálogo, i.e., a troca estruturada de informação através de mensagens.

A problemática do processo da interacção exige que se atenda à forma de coordenação entre agentes. Os dois modos de coordenação que se irão equacionar dão pelo nome de cooperação e competição. Na cooperação, os agentes integram o seu conhecimento e outras mais valias para alcançar um objectivo comum, o que em termos individuais não conseguiriam. Quando se passa para a competição, constata-se que os agentes têm interesses próprios, por vezes antagónicos; neste caso a negociação irá fornecer os mecanismos e as formas de coordenação.

### **Plataformas de Comunicação**

A plataforma de comunicação é o meio físico para a transmissão da informação entre os agentes. Sendo os agentes entidades heterogéneas, o protocolo de comunicação assume, desse modo, um papel fundamental para a interpretação dos dados, conhecimento e programas a transmitir. A Internet é, por definição, um veículo importante para a transmissão da informação de e para entidades que configuram os SMA. É uma rede pública, acessível a todos, fácil de usar, tem baixos custos e fornece serviços populares tais como o WWW, FTP, Email e Telnet. No entanto, a Internet e os seus serviços não resolvem, por si, toda uma miríade de problemas. Outras plataformas de comunicação para entidades distribuídas, heterogéneas e inteligentes, são necessárias.

Este serviço (a comunicação entre agentes) segue uma classificação dada por Ferber [Ferber, 1999], e que é traduzida na forma:

A ligação entre o remetente e o endereço pode ser do tipo ponto a ponto, em que o agente remetente envia uma mensagem ao destinatário, identificando-o, ou difusa, em que os potenciais destinatários de uma mensagem constituem o colectivo do sistema.

A natureza do meio ou canal de comunicação pode ser do tipo por reencaminhamento directo, em que o agente envia a mensagem ao canal de comunicação e este, por sua vez, encarrega-se de a fazer chegar

(directamente) ao respectivo endereço, ou então por reencaminhamento por propagação de nível, em que uma mensagem é difundida pelo ambiente e a sua intensidade vai decrescendo à medida que a distância entre as partes aumenta, ou mesmo por reencaminhamento por anúncio público, situação em que a mensagem é colocada num espaço comum que é visível para todas as entidades do sistema.

Intenção da comunicação pode ser do tipo intencional, nos casos em que o processo que determina a comunicação é encetado voluntariamente pelo agente, ou não intencional, quando este processo se dá de uma forma que é independente da vontade do emissor (e.g., linguagem corporal).

### **A Plataforma CORBA**

A plataforma CORBA (Common Object Request Broker) permite a interoperabilidade de objectos entre redes em ambientes heterogéneos [Vinoski, 1997]. As aplicações distribuídas são vistas como objectos fechados com funcionalidades visíveis para o exterior. Os objectos são declarados para o sistema através de uma interface que integra um alto nível de abstracção (e.g., através da linguagem declarativa IDL – Interface Definition Language). Na arquitectura CORBA tudo passa por se padronizar a interface de serviços ou funcionalidades do sistema, para dar-se a conhecer e proceder-se a implementações independentes. A comunicação entre aplicações é realizada pelo envio, para a arquitectura CORBA, de uma descrição na linguagem IDL, do serviço pretendido. É da responsabilidade da arquitectura CORBA encontrar a aplicação que satisfaz os requisitos desse pedido, realizar o pedido e retornar o resultado à aplicação cliente. Os objectos comunicam através da Invocação Remota de Procedimentos (RPC – Remote Call Procedure).

### **A Plataforma DCOM/OLE**

A plataforma DCOM/OLE (Distributed Component Object Model/Object Linking Embedding) da Microsoft privilegia a interoperabilidade entre objectos desenvolvidos segundo diferentes linguagens de programação, tendo em vista os sistemas operativos Windows.

### **A Plataforma JATLITE**

No que concerne à plataforma JATLite (Java Agent Template Lite) está-se face a uma plataforma desenvolvida na Universidade de Stanford, para facilitar a comunicação entre agentes heterogéneos e distribuídos, com recurso à linguagem de programação Java. A comunicação é realizada através da troca de mensagens, especialmente quando a troca é feita através da linguagem KQML (Knowledge Query and Manipulation Language). A plataforma JATLite assenta num componente denominado AMR (Agent Message Router). Qualquer agente do sistema deve registar-se no AMR, usando um nome e uma palavra-chave. O AMR gere o processo de encaminhamento de mensagens do emissor e entrega-as ao receptor, mesmo quando existem falhas de comunicação (e.g., rede interrompida, agente inactivo). O AMR mantém, por outro lado, uma actualização dos endereços dos agentes do sistema.

### **Outras Plataformas de Comunicação**

O JATLite é suportado por um motor em linguagem Java, mas outras plataformas suportadas por outros motores de linguagens interpretadas tais como Telescript, TCL ou Obliq estão adequadas à construção de

aplicações baseadas em código móvel. Mesmo linguagens como o Postscript e o SQL podem ser tomadas em consideração, e comportar-se como tal.

### **Linguagens de Comunicação entre Agentes de Software**

Uma linguagem de comunicação configura a forma pela qual é comunicado o conteúdo de uma mensagem [Labrou & Finin, 1997]. Uma diferença essencial entre os sistemas orientados ao objecto e os sistemas orientados ao agente está precisamente na linguagem que é utilizada. Geralmente, nos sistemas orientados ao objecto, o significado duma mensagem pode variar de objecto para objecto. Nos sistemas orientados ao agente, a linguagem usada deve ser comum e o conteúdo semântico deve ser independente do agente. Por outro lado, os agentes de software para interagir entre si necessitam de certos recursos, tais como:

- Uma linguagem comum;
- Uma forma de interpretar o conhecimento trocado;
- A capacidade de trocar esse conhecimento.

Aqui, tal como nas linguagens de programação convencional, há que atender:

- Às linguagens do tipo procedimental;
- Às linguagens do tipo declarativo.

As linguagens referidas em primeiro lugar permitem a troca de programas, procedimentos ou guiões, que são executados directa e eficientemente. As linguagens do tipo declarativo permitem a troca de definições e regras, as quais são processadas de forma indirecta. O maior problema que está subjacente à problemática das linguagens procedimentais ocorre quando são recebidas diferentes solicitações, em simultâneo, e é necessário estabelecer prioridades. Nas linguagens declarativas, essas operações são realizadas de uma forma natural.

Por outro lado, muitas são as linguagens que têm sido apresentadas de forma a resolver o problema da comunicação entre agentes, sendo de mencionar:

- KIF (Knowledge Interchange Format);
- KQML (Knowledge Query Manipulation Language);
- FIPA ACL (Foundation for Intelligent Physical Agents - Agent Communication Language);
- Ontolingua;
- RDF (Resource Description Format);
- XML (Extended Markup Language);
- BRML (Basic Rule Markup Language);
- DAML (DARPA Agent Markup Language).

## Actos de Discurso

Os agentes são entidades com propriedades antropomórficas, como já foi referido anteriormente. Desta forma, a comunicação entre seres racionais é um modelo de referência para a comunicação entre outros tipos de entidades, como por exemplo, os catalogados como agentes.

Uma análise restrita da comunicação humana é dada pela teoria designada por SAT (Speech Act Theory), que reduz a linguagem humana aos elementos indicativos de acção (e.g. pedidos, sugestões). Inerente à geração de frases, a teoria afirma um valor semântico subjacente. Por exemplo, quando se pede algo, não se está só a gerar uma frase mas a formalizar o pedido. A SAT analisa o discurso humano segundo três vectores [Weiss, 1999]:

- locução (a articulação do emissor);
- elocução (o significado pretendido para a articulação dada);
- perlocução (a acção resultante da articulação no receptor).

A SAT usa o termo “performative” para indicar o tipo especial de elocução que implica acção. Classificando estes elementos verbais, é possível reduzir a riqueza e a ambiguidade semântica que está subjacente ao acto de discurso. Obviamente, a mensagem recebida pode ser ambígua e de difícil resolução, no entanto, o agente deve ser capaz de a identificar correctamente.

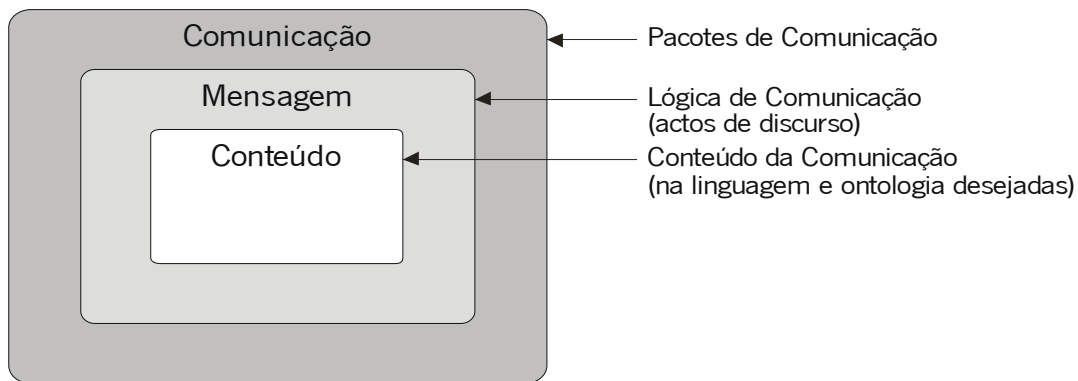
## A Linguagem KQML

A KQML assume-se como uma linguagem de comunicação entre agentes que se desenvolve segundo três camadas, dadas na forma (Figura 7):

- o Conteúdo, o qual passa não só pela explicitação da informação a ser transmitida, mas também pela indicação da linguagem em que esta é representada. O agente receptor interpreta a mensagem recebida;
- a Mensagem, a qual não só é concebida com base no conteúdo, mas também transporta uma intenção. A estes itens que dão corpo à mensagem, é ainda possível complementá-la com informação como a identidade dos agentes ou a ontologia adoptada (para a passagem de informação entre agentes);
- a Comunicação, a qual passa pela consideração de um conjunto de parâmetros de baixo nível (e.g., identidade do emissor/receptor ou identificador único associado à mensagem), sem os quais não é possível efectuar a transmissão de informação entre agentes.

Aqui o elemento aglutinador (em termos da linguagem KQML) é a lista, cujo primeiro elemento denota uma intenção, sendo a sua parte restante uma sequência constituída por um leque de parâmetros na forma de pares atributo/valor. Por ontologia entende-se a especificação (explícita) duma conceptualização, i.e., a visão simplificada e abstracta do mundo ou universo de discurso, com a representação formal de objectos, conceitos, áreas de interesse ou relações [Gruber, 1993]. Denota o vocabulário a utilizar em termos de

representação de informação, num domínio específico e ao qual se associa um conjunto de definições e axiomas que restringem o significado dos termos. Garante não só a consistência, mas também concorre para que haja compatibilidade entre a informação trocada e aquela que é propriedade do agente:



**Figura 7: A Linguagem KQML como linguagem por camadas**

Um agente pode ainda tomar um conjunto de atitudes, tais como: interrogar, afirmar, acreditar, requerer, atingir, subscrever ou oferecer.

Um exemplo de uma mensagem poderia ser dada na forma:

```
(tell : sender agent1
      : receiver agent2
      : ontology sales
      : language prolog
      : content "sell(productx,100)"
      : label label1)
```

Esta linguagem apresenta como características principais:

- as mensagens são independentes dos conteúdos que transportam;
- as mensagens socorrem-se de atitudes para gerir o conteúdo das mensagens (e.g., inserir, requerer, interrogar, ...);
- as primitivas da linguagem definem as acções ou operações admissíveis;
- um grupo de agentes que fale KQML pode ser enriquecido com recurso a agentes denominados facilitadores, os quais disponibilizam serviços que vão desde o registo de dados em bases de dados, ao encaminhamento de mensagens, à corretagem.

Cada mensagem KQML é designada por performative (termo que deriva da SAT) e segue a sintaxe do Knowledge Interchange Format (KIF) [Finin e tal, 1994]. Obviamente, dado o incontável universo das aplicações da linguagem, o número de performatives é questionável. Para alguns domínios poderá ser excessivo, enquanto para outros será insuficiente. A garantia de conformidade de uma aplicação com o padrão surge apenas quando as performatives reservadas estão de acordo com as especificações. O



funcionamento de cada agente deve assentar sobre a gestão de uma base de conhecimento. No entanto, os preceitos a que deve obedecer um repositório de conhecimento são desnecessários, surgindo assim o conceito de Base de Conhecimento Virtual (e.g., um agente pode actuar sobre uma base de dados vulgar, sendo essa a sua Base de Conhecimento Virtual). Dado que a comunicação com um dado agente se prende com o conceito abstracto de manipulação do seu repositório de conhecimento, a linguagem é utilizada para comunicação fim-a-fim, fazendo uso de intermediários e até entre os mais diferentes tipos de software e hardware (Figura 8) [Finin e tal, 1994].

A tabela da Figura 9 apresenta um sumário dos parâmetros reservados e os seus significados.

Nome	Significado
advertise	E é particularmente apto ao processamento de uma performative
ask-if	E quer saber se uma frase está na BCV de R
delete	E quer remover da BCV de R uma frase base
deny	a performative embebida não se aplica a E (a partir de agora)
insert	E requer a R que um conteúdo seja inserido na sua BCV
pipe	E deseja que R faça o encaminhamento de todas as mensagens para outro agente
register	E pode entregar performatives a um dado agente
tell	a frase está na BCV de E
untell	a frase não está na BCV de E

**Figura 8: Tabela sumária de algumas performatives reservadas, para um emissor E e um receptor R**

Parâmetro	Significado
:content	a informação acerca da qual a performative expressa uma atitude
:force	se o emissor algumas vez negar o significado da performative
:in-reply-to	a etiqueta esperada numa resposta
:language	a identificação da linguagem usada no parâmetro :content
:ontology	a identificação da ontologia usada no parâmetro :content
:receiver	o receptor da performative
:reply-with	se o emissor espera uma resposta e, se tal, qual a etiqueta da resposta
:sender	o emissor da performative

**Figura 9: Tabela com alguns parâmetros reservados e os seus significados**

## A Linguagem KIF

A linguagem KIF denota um formalismo computacional para a troca de conhecimento entre programas. É uma linguagem do tipo declarativo e suporta a interpretação de expressões formuladas em termos da Lógica Predicativa de Primeira Ordem.

## A Linguagem ACL

A linguagem ACL da FIPA permite uma abordagem diferente ao problema da comunicação entre agentes, sendo composta por:

- o KQML como linguagem de comunicação;
- o KIF como formalismo;
- ontologias que definem um conjunto de classes, funções e alfabetos com aplicação a um determinado mundo ou universo de discurso, a que se associa uma axiomática que permite a interpretação das mensagens.

## A Linguagem XML

A XML (eXtended Markup Language) é, neste momento, a linguagem mais promissora para o armazenamento e a difusão de informação na Web. A linguagem HTML é sem dúvida um padrão para a escrita de páginas publicadas na Internet, mas sofre de limitações em matéria de armazenamento da informação. Pelo contrário, a XML oferece ao utilizador uma linguagem com um alto nível de flexibilidade que permite descrever virtualmente qualquer tipo de informação (e.g., entidades, objectos, agentes, estruturas de dados complexos). Em conjunto com style sheets ou DTDs, um documento ou ficheiro XML pode facilmente ser apresentado em formato HTML e visualizado num qualquer web browser. A informação pode ser pesquisada, extraída, integrada, ordenada, filtrada, manuseada ou simplesmente usada a partir de documentos em XML. A XML está integralmente orientada para a difusão rápida da informação na Web em quantidade e em qualidade (e.g., a mensagem referida na página em linguagem KQML pode ser reescrita em XML).

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<tell>
<sender> agent1 </sender>
<receiver> agent2 </receiver>
<ontology> sales </ontology>
<language> prolog </language>

<content> sell(productx,100) </content>
<label> label1</label>
</tell>
```

## **Ontologias**

A troca de informação entre agentes exige mais que um formato comum e uma linguagem de comunicação comum. Estas entidades, especialistas na resolução de um determinado problema, autónomas e heterogéneas, têm de concordar em um modelo conceptual da realidade, que permita capturar o conhecimento próprio dessa mesma realidade e que a permita compreender, i.e., acordar numa ontologia comum [Neches et al., 1991].

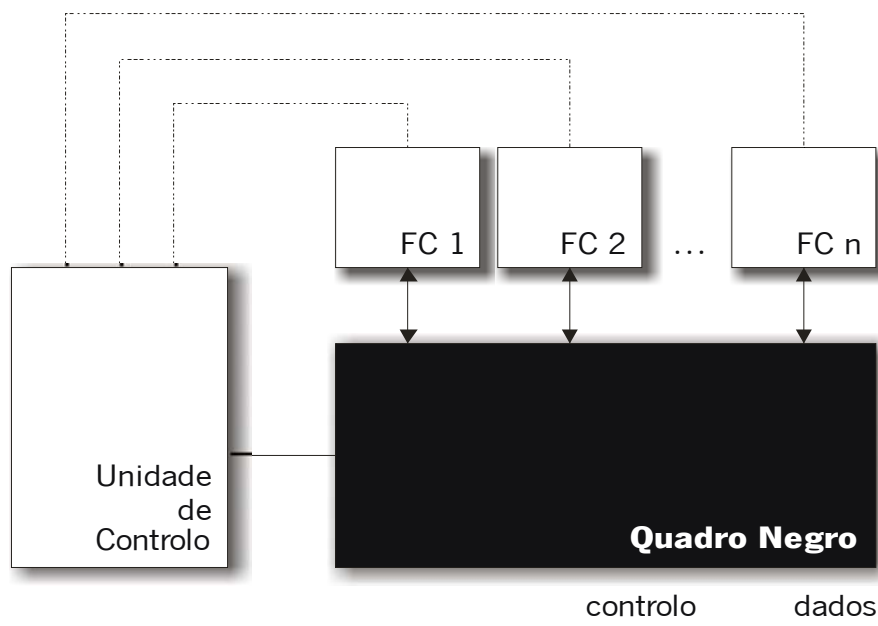
O termo ontologia teve a sua origem na disciplina de Filosofia, onde designa uma explicação sistemática da existência. Em [Gruber, 1993] definiu-se uma ontologia como sendo a explicação explícita de uma conceptualização. A ontologia fornece por outras palavras o vocabulário de representação para o domínio em questão, e um conjunto de definições e axiomas que restringem o significado dos termos nesse vocabulário, de forma a permitir uma interpretação consistente e única. A adesão a uma ontologia comum garante a consistência (a mesma expressão tem o mesmo significado em qualquer agente) e a compatibilidade (qualquer conceito é designado pela mesma expressão em qualquer agente) da informação presente no sistema [Gruber, 1993].

## **Sistemas Baseados em Quadros Negros**

Um Sistema Baseado em Quadros Negros configura uma espécie de meta-arquitectura, i.e., uma arquitectura a partir da qual é possível gerar novas arquitecturas. Os sistemas que correm nessas arquitecturas afirmam-se por apresentarem formas de organização, estruturação, e utilização do conhecimento deveras eficientes. Os Sistemas Baseados em Quadros Negros têm vindo a ser utilizados como ferramenta de prototipação rápida em arquitecturas para SMA, assim como no desenvolvimento desses sistemas [Neves et al., 1997] [Machado et al., 1997].

Estes sistemas são constituídos por três peças chave [Luger & Stubblefield, 1998] (Figura 10):

- As fontes de conhecimento FC 1, FC 2 , ..., FC n, ou agentes, que denotam os diversos especialistas ou actores que aí co-habitam (i.e., os intervenientes na resolução dos problemas);



**Figura 10: Arquétipo de um Sistema Baseado em Quadros Negros**

- A unidade de controlo que é o componente do sistema responsável pela monitorização das alterações ao estado do conhecimento do Quadro Negro, assim como das acções a executar pelos agentes e/ou SMA no processo de geração de solução(ões) para um dado problema.
- O Quadro Negro, que se assume como um repositório de informação e que se comporta como uma base de dados. Tem como função dar a conhecer o estado dos processos computacionais que aí decorrem, e actua como uma plataforma de comunicação entre as fontes de conhecimento ou agentes, assim como facilita a gestão global do sistema;

### Áreas de Aplicação

Têm-se vindo a dar a conhecer, a cada dia que passa, novos e potenciais campos para a utilização de tecnologia baseada em sistemas computacionais que recorrem a agentes e/ou SMA, para a resolução de problemas, sendo de mencionar [Wooldridge, 2000], [He et al., 2003]:

- Na área industrial:
  - Controlo de processos;
  - Produção;
  - Controlo/Simulação de sistemas de controlo de Tráfego Aéreo.
- No sector Comercial:
  - Comércio Electrónico;
  - Gestão de Empresas.
- No ensino (e.g., em sistemas de e-learning).

- No domínio da Medicina:  
Monitorização de pacientes;  
Cuidados de Saúde.
- Entretenimento:  
Jogos.

De notar que estas áreas acabadas de referir, têm alguns denominadores comuns, tais como:

- A distribuição geográfica, a temporalidade, um certo conteúdo semântico ou mesmo funcional;
- A complexidade.

## **Conclusões**

Neste capítulo faz-se uma abordagem à computação baseada em agentes e/ou SMA para a resolução de problemas. Estas entidades não são vistas como elementos computacionais simples e/ou com esquemas de interacção pré-definidos, mas como entidades que revelam um comportamento próprio quando inseridas num determinado ambiente, e que coordenam as suas próprias acções com base em conhecimento formal sobre a dinâmica dessas mesmas acções e as suas consequências, quer sobre o ambiente onde estão inseridas, quer sobre as outras entidades que fazem parte desse mesmo ambiente.



## Ferramentas para a Construção de Agentes

Neste capítulo faz-se uma apresentação de ferramentas para construção de agentes sendo abordados, ainda, sistemas integrados de agentes.

### JESS

A Java Expert System Shell (JESS) [Friedman-Hill, 2002] é uma ferramenta deveras adequada para o desenvolvimento de Sistemas Periciais. A JESS foi desenvolvida por Ernest Friedman-Hill no Sandia National Laboratories em 1995, durante um projecto interno de investigação. Originalmente destinada a ser uma implementação Java do CLIPS [Riley, 2002], possui agora um conjunto de características que a diferencia de qualquer produto similar.

Através do uso de uma sintaxe próxima da linguagem LISP, é possível fornecer um conjunto de regras que disparam quando certas condições são dadas como factos. A unificação do conhecimento factual com um conjunto de regras que compõe a BC do agente é feita através de um algoritmo conhecido como Rete (designação latina para rede) [Forgy, 1982], que torna a JESS muito mais rápido que uma simples cascata de instruções «se ... então ...».

O Rete funciona através da construção de nodos em que cada um destes representa um ou mais testes sobre o lado esquerdo de uma regra.

Através da implementação em Java e do uso de uma linguagem funcional interpretada, a JESS possibilita a criação de agentes inteligentes que são simultaneamente portáteis e de fácil adaptação. No entanto, o que é considerado como uma das suas maiores vantagens é também uma das suas maiores fraquezas: de forma a fornecer portabilidade é assumido um custo relativo de eficiência.

### **SICStus-Prolog**

O SICStus-Prolog [Almgren et al., 1993] é uma implementação da linguagem de programação em lógica PROLOG, desenvolvida pelo Swedish Institute of Computer Science (SICS). Partes do sistema foram desenvolvidas através do projecto Industrialização do SICStus Prolog em colaboração com a Ericsson Telecom AB, NobelTech Systems AB, Infologics AB e Televerket. A linguagem de programação PROLOG foi desenvolvida na Universidade de Marselha e faz uso de um subconjunto da lógica de primeira ordem (clausulado de Horn) como forma de produzir programas curtos e declarativos.

O SICStus Prolog fornece um conjunto de bibliotecas, ferramentas e um ambiente de execução para o desenvolvimento de programas. A resolução do programa dá-se através de um algoritmo de encadeamento inverso, o qual procura activar regras cujas pré-condições não estejam directamente garantidas [Russel & Norvig, 1995]. Este software possibilita a criação de agentes com mecanismos de raciocínio complexos suportados por lógica. A arquitectura baseada em quadros negros LINDA [Carreiro & Gelernter, 1989] [Almgren et al., 1993], também fornecida sob a forma de bibliotecas, permite uma fácil e rápida troca de informação, sob a forma de tuplos, entre agentes.

No SICStus Prolog, apesar dos programas em lógica serem independentes da plataforma, tal não acontece com o ambiente de execução, condicionando, assim, alguma da portabilidade. No entanto, versões recentes fornecem um módulo designado por JASPER, o qual disponibiliza uma ligação bidireccional com o Java. Contudo, a construção de applets, vista como uma das principais vantagens do Java, é inacessível.

### **CORBA**

A Common Object Request Broker Architecture (CORBA) [Harkey & Orfali, 1998] [Rosenberger, 1998] é considerada como o projecto de middleware mais importante e ambicioso na área dos Sistemas Distribuídos. Ao invés de ser patrocinado por uma única empresa (tal como acontece com o Distributed Component Object Model - DCOM – da Microsoft), a CORBA é apoiada por um consórcio de mais de 800 empresas, designado por Object Management Group (OMG).

A CORBA define um middleware que é capaz de agregar qualquer outra forma existente de middleware cliente/servidor, possibilitando aos sistemas existentes a integração no barramento de objectos e, simultaneamente, fornecendo mecanismos para que novos sistemas tomem partido das vantagens das novas capacidades. Os objectos (e até certo ponto, os agentes) são considerados a metáfora unificadora, sendo a interoperabilidade a principal preocupação.

Os objectos/agentes são descritos através de uma linguagem independente da plataforma ou da implementação, designada por Interface Definition Language (IDL). Logo, o interface e a implementação são claramente separados. A CORBA assenta sobre um protocolo que dá pelo nome de Internet Inter-Orb Protocol (IIOP) que possibilita a manipulação de objectos remotos. O componente central da arquitectura é o Object Request Broker (ORB), o qual possibilita que cada objecto interactue transparentemente com outros objectos através da sua interface. Por outro lado, a implementação de um objecto interactua com o ORB através de um



Object Adapter. Os processos de codificação e decodificação inter-plataforma das mensagens (designados, respectivamente, por marshalling e unmarshalling), os quais garantem interoperabilidade, são completamente transparentes.

Através do uso dos CORBAservices, as funcionalidades do ORB podem ser complementadas. O OMG fornece definições base (através de IDLs e especificações de serviço) para um grande número de serviços, entre os quais estão: o Naming Service (permite a localização de componentes através de um nome), o Event Service (através do registo num canal de eventos o interesse numa informação específica é anunciado, sendo esta acedida por métodos de push e pull ), o Transaction Service (fornece coordenação entre componentes através de um protocolo de actualização em duas fases (two-phase commit) [Samaras et al., 1993] [Connolly et al., 1999] e o Trader Service (possibilita aos objectos/agentes a publicitação das suas capacidades e a licitação de tarefas).

Apesar da CORBA ser um pouco mais lenta e bastante mais complexa que a sua concorrência directa, a IRM Java, dispõe de vantagens tais como: suporte de descoberta dinâmica e meta-dados, os benefícios de um standard aberto, independência da linguagem e um protocolo de troca de informação neutro.

## JINI

A tecnologia JINI [Kumaran, 2001], desenvolvida pela Sun Microsystems sob a alçada do Java, prende-se com a movimentação dos sistemas de centrados-em-disco para centrados-na-rede. Devido à proliferação do acesso a redes e, em particular, o acesso à Internet, um conjunto de aparelhos (não necessariamente computadores) necessitam de acesso rápido e fácil à infra-estrutura de comunicação subjacente.

A JINI é um conjunto de Application Programming Interfaces (APIs) e protocolos de rede que possibilitam a criação de sistemas distribuídos enquanto federações de serviços/objectos. Fazendo uso de RMI e de mecanismos de seriação de objectos fornecidos pelo Java, a JINI fornece elevada portabilidade e uma curva suave de aprendizagem para aqueles que se encontrem familiarizados com estas tecnologias.

Ao invés de olhar para a coordenação como um problema central, esta tecnologia agrupa conjuntos de objectos distribuídos remotos como “fornecedores de serviços” com iguais direitos. Esta estrutura federada é altamente flexível e a tolerância à falha é obtida através da inexistência de um ponto de falha único.

Tudo que se ligue à rede é considerado um “fornecedor de serviços”. Desde um simples telefone celular a um super-computador, de uma câmara digital a um indivíduo, tudo pode fornecer um serviço à comunidade. A JINI define uma infra-estrutura de execução ao nível de rede, disponibilizando mecanismos para adicionar, remover, localizar e aceder a serviços. Os “serviços de localização” são a componente central da arquitectura, registando-se a si mesmos em “serviços de localização”, caso estes já existam. O procedimento de “descoberta de serviços de localização” tem lugar através de pacotes de anúncio produzidos por mecanismos com funcionalidades JINI. De seguida, procedimentos de “junção” ou “localização” podem ocorrer, de forma a um objecto/agente se registar num “serviço de localização” específico ou de forma a

encontrar um “fornecedor de serviço”. Uma clara separação entre interface e implementação é conseguida através dos procedimentos JINI.

## **FIPA-OS**

O FIPA-OS [Poslad, 2000] é um conjunto de ferramentas open-source para o desenvolvimento de agentes compatíveis com o padrão da Foundation for Intelligent Physical Agents (FIPA). Os padrões definidos pela FIPA vão desde a arquitectura dos agentes até à comunicação e serviços específicos (e.g., leilões).

São definidos três tipos de componentes: obrigatórios (componentes que apontam para a implementação de interoperabilidade e portabilidade), com implementações comutáveis, e opcionais (e.g., a JESS fornece um mecanismo para construir agentes FIPA-OS que fazem uso das vantagens existentes em sistemas periciais).

Existem 4 componentes obrigatórios: Agent Shell, Task Manager, Conversation Manager e Message Transport Service (MTS). A Agent Shell providencia um ambiente simples para a implementação de agentes estendendo classes. O Task Manager fracciona uma tarefa composta em elementos individuais que são pedaços de código auto-contidos, com a capacidade de receber e enviar mensagens e sem preferência quanto ao local onde o código deve ser executado. O Conversation Manager permite que um protocolo específico (seja ele FIPA ou definido pelo utilizador) seja seguido durante uma troca de mensagens (possibilitando, por exemplo, o agrupamento de mensagens relacionadas). O MTS fornece uma das mais importantes características dos agentes inteligentes - a comunicação - possibilitando a troca de mensagens entre agentes.

## **JADE**

O Java Agent Development Framework (JADE) [Bellifemine et al., 1999] é um software desenvolvido em Java que permite a construção de aplicações baseadas em agentes. Surge graças aos esforços da CSELT S.p.A. e da Universidade de Parma sob uma licença de software de desenvolvimento aberto. O JADE é compatível com as especificações FIPA. Mais do que um conjunto de APIs, o JADE é visto como um middleware para agentes, fornecendo uma plataforma de agentes e uma arquitectura de desenvolvimento. Os programadores necessitam apenas de se concentrar nos mecanismos inteligentes e no funcionamento interno dos seus agentes, deixando o transporte de mensagens, a codificação/descodificação de estruturas de dados (marshalling/unmarshalling) e a gestão geral ao sistema subjacente. A versão 2.x do JADE é compatível com as especificações FIPA2000, implementando: o Agent Management System (AMS), o Directory Facilitator (DF) e o Agent Communication Channel (ACC). A passagem de mensagens segue a FIPA-Agent Communication Language (FIPA-ACL).

No JADE, cada Máquina Virtual Java (MVJ) fornece uma área para a execução concorrente de agentes. A comunicação é atingida através de três mecanismos: IRM entre diferentes MVJs, sinalização de eventos dentro de uma MVJ e IIOP entre diferentes plataformas de agentes.

As principais vantagens do JADE situam-se a nível da interoperabilidade, do uso de um standard bem fundamentado, da portabilidade do código e de um conceito neutro de agente.

## **ZEUS**

O ZEUS [Nwana & Wooldridge, 1997] [Nwana, 1996] é um conjunto de ferramentas para construção de agentes desenvolvido pelos laboratórios da British Telecom. Estas ferramentas foram já distinguidas com vários prémios nacionais e internacionais. O ZEUS foi implementado em Java procurando as mais-valias desta linguagem: independência da plataforma, execução concorrente (multi-threading), orientação à Internet e um conjunto de IPAs em constante evolução (as mesmas razões escolhidas por algumas das ferramentas anteriormente apresentadas).

Cada agente é composto de 3 camadas: uma camada de definição do agente (responsável por enunciar e implementar as capacidades particulares de um agente), uma camada organizacional (definindo as relações estabelecidas entre conjuntos de agentes) e uma camada de coordenação (fornecendo técnicas de negociação e coordenação de forma a que um agente implemente uma das mais importantes características dos agentes inteligentes: comportamento social).

Para além de providenciar um enquadramento arquitectural para a construção de agentes de software, o ZEUS fornece um conjunto de ferramentas que ajudam na definição da informação manipulada, isto é, na definição de restrições, tarefas e ontologias que alimentarão um gerador de código. O conjunto de ferramentas é composto por: Ontology Editor, Task Description Editor, Organisation Editor, Agent Definition Editor, Coordination Editor, Fact/Variable Editor, Constraint Editor e Code Generator. O ZEUS inclui também um conjunto de bibliotecas e utilitários, tais como: uma biblioteca de grafos de coordenação, uma API para a execução de tarefas com avaliação de custo, comunicação KQML, algoritmos de planeamento e escalonamento, serviços de "páginas amarelas" e visualizadores.

A principal força do ZEUS situa-se na sua capacidade de construir, colocar em funcionamento e controlar agentes com pouco esforço de programação e portabilidade máxima. No entanto, a curva de aprendizagem é íngreme.

## **Outras Ferramentas**

Existem muitas iniciativas no campo das arquitecturas e ferramentas para o desenvolvimento de agentes. No entanto, as iniciativas comerciais centram-se sobretudo na tecnologia de agentes móveis. Destacam-se os projectos: Aglets (IBM Japan) [Labrou & Finin, 1997], Grasshoper (IKV++) [GmbH, 1998] e Voyager (ObjectSpace) [ObjectSpace, 1997].

No caso do Aglets a IBM deu por terminado o projecto, no entanto (talvez devido a alguma pressão pública e à popularidade do software) a iniciativa continua sob a forma open-source.

O Grasshopper foi o primeiro ambiente de agente móveis a ser compatível com a Mobile Agent System Interoperability Facility (MASIF) lançada pelo OMG. Através da utilização da CORBA, as tradicionais aplicações cliente/servidor e as infra-estruturas de agentes móveis são unificadas.

O Voyager é um projecto que, tal como o Grasshopper, tem por base a CORBA. É totalmente construído em Java e combina o poder da invocação remota de métodos com os agentes móveis autónomos, que podem movimentar-se de hospedeiro para hospedeiro executando o código do proprietário (particularmente importante em cenários de conectividade intermitente). É acompanhado por um conjunto de serviços distribuídos tais como: directoria, persistência e difusão selectiva por publicação/subscrição.

## **Conclusões**

Neste capítulo são apresentadas algumas ferramentas para a construção de agentes. As ferramentas para a construção de agentes devem fornecer não apenas mecanismos para a troca de mensagens entre os elementos constituintes de uma comunidade (facilitando assim o comportamento social – CORBA e JINI), mas também meios de implementação de acções inteligentes que resultem de processos, e.g., de avaliação de regras (JESS e SICStus-Prolog).

São ainda abordados sistemas integrados de agentes (FIPA-OS, JADE e ZEUS).

## Conclusões

O conceito ou a definição de agente está longe de ser consensual. No entanto, tal poderá assumir-se como uma vantagem e não como um senão, uma vez que deste modo é possível um elevado grau de abstracção na análise dos problemas. Existem autores que afirmam que o conceito de agente não existe enquanto meio inequívoco de classificação dos processos de desenvolvimento e análise de software, mas sim enquanto abstracção útil para a abordagem a problemas porventura complexos.

Apesar da indefinição quanto ao conceito, existe um conjunto de características consideradas elementares para a criação de agentes inteligentes: a autonomia, a pro-actividade, a reactividade e o comportamento social. A maneira como cada agente consegue manipular este conjunto de características determina a sua arquitectura básica, dando assim origem, por exemplo, a agentes CDI.

A distinção entre um agente e um objecto é também problemática. No entanto, um agente é desde logo um Paradigma Computacional enquanto os objectos voltam-se para os Paradigmas da Programação. O dinamismo perante o meio, a manipulação do conhecimento, os mecanismos de restrição de acesso ao mesmo e os conceitos de mensagem e linguagem são algumas das características diferenciadoras. Apesar de tudo, a utilização de objectos para a implementação de agentes é útil (ou até mesmo desejável).

Por imposição do meio (com distribuição geográfica ou temporal) ou por facilidade de modelação, é útil combinar diferentes agentes que, de forma competitiva ou cooperativa, procuram atingir um dado objectivo.

Surgem, assim, os SMA como o paralelo digital dos mecanismos de natureza social para a resolução de problemas próprios dos seres vivos. A existência de agentes altamente aptos para determinadas tarefas poderá ser menos útil do que se poderá pensar (assumindo uma actuação isolada). Isto é, as forças sociais que parecem ocupar um lugar de destaque na evolução dos seres vivos poderão ser benéficas ao nível das sociedades de agentes, potenciando um melhor desempenho. As áreas de aplicação deste tipo de sistemas

são variadas, como se viu, e vão desde o comércio electrónico até sistemas de controlo de tráfego aéreo, entre outras.

Um tema de particular importância, quando se lida com agregados de agentes, (sejam estes artificiais ou naturais) prende-se com a capacidade de comunicação entre as partes. A utilização de um mecanismo linguístico inteligível entre todos os intervenientes é essencial para que os processos cooperativos/concorrenciais se desenvolvam. Para tal, surgem linguagens como a KQML e a FIPA-ACL e, recentemente, a aplicação da XML.

## Exercícios

Os agentes autónomos e inteligentes podem ser caracterizados através das suas propriedades:

- De entre diversas propriedades destacam-se a pro-actividade, reactividade, adaptação ao meio e a racionalidade. Descreva estas propriedades, comparando-as.
- De entre diversas propriedades temos, a capacidade social, a mobilidade e a adaptação ao meio. Descreva estas propriedades.

O conceito de agente é bastante abstracto. No entanto algumas características podem ser enunciadas como fundamentais para a classificação de uma entidade como agente inteligente. Quais são, na sua opinião, essas características?

Considera um objecto (no contexto da Programação Orientada a Objectos) e um agente entidades equivalentes? Justifique?

Um dos principais problemas dos sistemas usando agentes tem a ver com a dificuldade que os seres humanos têm em acreditar nas capacidade dos agentes e atribuir-lhes alguma credibilidade, nomeadamente quando os agentes se destinam a substituir o ser humano.

Considere o caso de um Agente para representar um comprador por comércio electrónico. Que características deveria ter esse agente para que o utilizador pudesse confiar nele? Sugira as funcionalidades que disponibilizaria num agente que desenvolvesse para o efeito.

Considere que se transformam vários sistemas periciais desenvolvidos para o mesmo domínio ou para domínios similares (admitindo alguma complementaridade) em Agentes. Que dificuldades práticas poderiam dificultar a partilha de conhecimentos entre esses Agentes?

Considere uma aplicação na qual um agente necessita de informação oriunda de diversos sistemas computacionais, processando essa informação de modo a produzir resultados.

- Em que situações haverá uma maior eficiência se o agente for móvel, podendo deslocar-se para outros sistemas computacionais? Em que situações não será eficiente a mobilidade de tal agente?
- Que limitações adicionais se poderão pôr relativamente à mobilidade de tal agente?

Considere um Agente Inteligente para comércio electrónico, o qual interage com um utilizador e com outros agentes (por exemplo: vendedores). Indique que aplicações possíveis poderão ter os métodos de aprendizagem automática a um agente desse tipo?

Em grande parte dos sistemas multiagente, os agentes representam recursos (por exemplo: robot, empresa, etc.), tarefas (por exemplo: encomendas) e fornecimento de informação (por exemplo: agente informador de condições climáticas). Que outros tipos de agentes poderiam existir quanto ao tipo de função desempenhada (para além de recursos, tarefas e fornecimento de informação)? Dê exemplos.



## Referências

- [Agha, 1986] Agha, G. (1986). *Actors, a Model of Concurrent Computing in Distributed Systems*. MIT Press.
- [Almgren, et al., 1993] Almgren, J., Anderson, S., Carlsson, M., Floyd, L., Haridi, S., Frisk, .C, Nilsson, H., and Sundberg, J. (1993). *SICStus Prolog Library Manual*. Swedish Institute of Computer Science: Sweden.
- [Analide & Neves, 2002] Analide, C. e Neves, J. (2002). *Antropopatia em Entidades Virtuais*. Em Actas do Workshop de Teses e Dissertações em Inteligência Artificial (WTDIA02), Recife, Brasil.
- [Bellifemine et al., 1999] Bellifemine, F. Poggi, A. and Rimassa. G. (1999). JADE: a FIPA-compliant Agent Framework. In *Proceedings of the International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM99)*, pages 97 108, London, UK.
- [Bond, 1990] Bond, A. (1990). A Computational Model for Organizations of Cooperating Intelligent Agents. *SIGOIS Bulletin*, 11.
- [Brooks, 1986] Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14-23.
- [Carreiro & Gelernter, 1989] Carreiro, N. and Gelernter, D. (1989). Linda in Context. *Communications of the ACM*, 4(32).
- [Charniak & McDermott, 1985] Charniak, E. and McDermott, D. (1985). *Introduction to Artificial Intelligence*. Addison-Wesley Publishing Company, Inc.
- [Coelho, 1995] Coelho, H. (1995). *Inteligência Artificial em 25 Lições*. Fundação Calouste Gulbenkian.
- [Connolly et al., 1999] Connolly, T. Begg, C. and Strachan, A. (1999). *Database Systems: a Practical Approach to Design, Implementation, and Management*. Addison- Wesley, 2 edition.

- [Costa & Melo, 1997] Costa, J. e Melo, A. (1997). Dicionário da Língua Portuguesa. 7 a Edição, Porto Editora, Portugal.
- [Durfee & Rosenchien, 1994] Durfee, E. and Rosenchien, J. (1994). Distributed Problem Solving and Multiagent Systems: Comparisons and Examples. In Proceedings of the International Workshop on Distributed Artificial Intelligence, Seattle, USA.
- [Ferber, 1999] Ferber, J. (1999). Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley.
- [Finin e tal, 1994] Finin, T., Fritzson, R., McKay, D. and McEntire, R. (1994). KQML as an Agent Communication Language. Proceedings of the 3<sup>rd</sup> International Conference on Information and Knowledge Management (CIKM94), Maryland, ACM Press.
- [Forgy, 1982] Forgy, C.. Rete (1982). a Fast Algorithm for Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence, (19):17-37.
- [Friedman-Hill, 2002] Friedman-Hill, E. (2002). JESS: The Expert System Shell for the Java Platform. Distributed Computing Systems, Sandia National Laboratories, USA.
- [Ganascia, 1993] Ganascia, J. (1993). Inteligência Artificial. Instituto Piaget.
- [GmbH, 1998] IKV++ GmbH. Grasshoper. (1998). An Intelligent Mobile Agent Platform Written in 100Pure Java. Available at <http://www.ikv.de/products/grasshoper>.
- [Gruber, 1993] Gruber, T. (1993). A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.
- [Harkey & Orfali, 1998] Harkey, D. and Orfali, R. (1998). Client/Server Programming with Java and CORBA. John Wiley & Sons, New York, USA, 2 edition.
- [Hayes-Roth et al., 1993] Hayes-Roth, D., Waterman, A., and Lenat, D. (1993). Building Expert Systems. Addison-Wesley.
- [He et al., 2003] He, M., Jennings, N., and Leung, H. (2003). On Agent-Mediated Electronic Commerce. IEEE Trans on Knowledge and Data Engineering.
- [Hewitt, 1977] Hewitt, C. (1977). Viewing Control Structures as Patterns of Passing Messages. Artificial Intelligence, 8(3).
- [Huhns & Singh, 1998] Huhns, M. and Singh, M. (1998). Readings in Agents, chapter Agents and Multiagent Systems: Themes, Approaches and Challenges. M. Huhns and M. Singh (eds), Morgan Kaufmann Publishers.
- [IBM, 1997] IBM Tokyo Research Laboratory (1997). The Aglets Workbench: Programming Mobile Agents in Java.

- [Jennings et al., 2001] Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., and Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2).
- [Kumaran, 2001] Kumaran, L. (2001). *Jini Technology: An Overview*. Prentice Hall.
- [Labrou & Finin, 1997] Labrou, Y. and Finin, T. (1997). Semantics and Conversations for an Agent Communication Language. In Huns, M. and Singh, M., editors, *Readings in Agents*. Morgan Kaufmann Publishers.
- [Luger & Stubblefield, 1998] Luger, G. and Stubblefield, W. (1998). *Artificial Intelligence Structures and Strategies for Complex Problem Solving*. Addison-Wesley.
- [Machado, 2001] Machado, J. (2001). *Agentes Inteligentes como Objectos dum Sistema Distribuído de Realidade Virtual*. Tese de Doutoramento, Departamento de Informática, Universidade do Minho, Portugal.
- [Machado et al., 1997] Machado, J., Novais, P., Abelha, A., and Neves, J. (1997). Addressing Problem Solving on a Distributed Computational Architecture. In *Proceedings of the 3rd International Congress of Industrial Engineering*, Gramado, Brasil.
- [Maes, 1990] Maes, P. (1990). Situated Agents Can Have Goals. In Maes, P., editor, *Designing Autonomous Agents*. MIT Press.
- [McCarthy & Hayes, 1969] McCarthy, J. and Hayes, P. J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence*, B. Meltzer and D. Michie (eds), Edinburgh University Press, 4.
- [Minsky, 1986] Minsky, M. (1986). *The Society of Mind*. Simon & Schuster.
- [Neches et al., 1991] Neches, R., Fikes, R., Gruber, T., Patil, R., Senator, T., and Swartout, R. W. (1991). Enabling Technology for Knowledge Sharing. *AI Magazine*, 12(3).
- [Neves et al., 1997] Neves, J., Machado, J., Analide, C., Novais, P., and Abelha, A. (1997). Extended Logic Programming Applied to the Specification of Multi-Agent Systems and Their Computing Environments. In *Proceedings of the 1997 IEEE - International Conference on Intelligent Processing Systems*, China.
- [Novais et al., 2000] Novais, P., Brito, L., and Neves, J. (2000). Experience-Based Mediator Agents as the Basis of an Electronic Commerce System. In *Proceedings of the Workshop 2000 - Agent-Based Simulation*, Passau, Germany.
- [Novais, 2003] Novais P., 2003, “Teoria dos Processos de Pre-Negociação em Ambientes de Comércio Electrónico”, Tese Doutoramento, Departamento de Informática, Universidade do Minho, Braga, Portugal.
- [Nwana, 1996] Nwana, H. (1996). Software Agents: An Overview. *Knowledge Engineering Review*, 11(3):1 40.

- [Nwana & Wooldridge, 1997] Nwana, H. and Wooldridge, M. (1997). Software Agent technologies. *British Telecommunications Technology Journal*, 4(14):16-27.
- [Poslad, 2000] Poslad, S. Bourne, R. and Hayzelden, L. (2000). *Agent Technology for Communications Infrastructure*, chapter *Agent Technology for Communications Infrastructure: an Introduction*. John Wiley and Sons.
- [ObjectSpace, 1997] ObjectSpace (1997). *ObjectSpace Voyager Core Package Technical Overview*.
- [O'Hare & Jennings, 1996] O'Hare, G. and Jennings, N. (1996). *Foundations of Distributed Artificial Intelligence*. Wiley Sons, New York, USA.
- [Riley, 2002] Riley, G. (2002). *CLIPS - A Tool for Building Expert Systems*. available at <http://www.ghg.net/clips/CLIPS.html>.
- [Rao & George, 1995] Rao, A. and George, P. (1995). *BDI Agents: from Theory to Practice*. In *Proceedings of the First International Conference on Multi-Agent Systems - ICMAS95*, San Francisco, USA.
- [Rosenberger, 1998] Rosenberger, J. (1998). *CORBA in 14 Days*. SAMS Publishing, New York, USA.
- [Russel & Norvig, 1995] Russel, S. and Norvig, P. (1995). *Artificial Intelligence - A Modern Approach*. Prentice Hall.
- [Samaras et al., 1993] Samaras, G. Britton, K. Citron, A. and Mohan, C. (1993). *Two-Phase Commit Optimizations and Trade's in the Commercial Environment*. In *Proceedings of the 9th International Conference on Data Engineering*, pages 520-529.
- [Santos, 1999] Santos, M. (1999). *Sistemas de Classificação em Ambientes Distribuídos*. Tese de Doutorado, Departamento de Informática, Universidade do Minho, Portugal.
- [Shoham, 1993] Shoham, Y. (1993). *Agent-oriented programming*. *Artificial Intelligence*, 60(1):51-92.
- [Vinoski, 1997] Vinoski, S. (1997). *CORBA: Integrating Diverse Application within Distributed Heterogeneous Environments*. *IEEE Communications Magazine*, 14.
- [Weiss, 1999] Weiss, G. (1999). *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. MIT Press.
- [Wooldridge, 1999] Wooldridge, M. (1999). *Intelligent Agents*. In Weiss, G., editor, *Multiagent Systems - A modern Approach to Distributed Artificial Intelligence*. MIT Press.
- [Wooldridge, 2000] Wooldridge, M. (2000). *Reasoning about Rational Agents*. MIT Press.
- [Wooldridge & Jennings, 1994] Wooldridge, M. and Jennings, N. (1994). *Agent Theories, Architectures, and Languages: A Survey*. In Wooldridge, M. and Jennings, N., editors, *Proceedings of the ECAI-Workshop on Agent Theories, Architectures and Languages*, The Netherlands.
- [Wooldridge & Jennings, 1995] Wooldridge, M. and Jennings, N. (1995). *Intelligent agents: Theory and Practice*. *The Knowledge Engineering Review*, 10(2):115-152.

## **Anexo**

# Agentes Inteligentes

Paulo Novais  
Cesar Analide

Departamento de Informática  
Escola de Engenharia  
Universidade do Minho

Setembro 2006



## Índice

- Inteligência Artificial
- Inteligência Artificial Distribuída
- Agente Inteligente
  - Noção e definição
  - Propriedades
  - Arquitecturas
- Sistemas Multiagente
- Comunicação em SMA
- Sistema Baseados em Quadros Negros
- Áreas de Aplicação
- Coordenação em SMA
- Conclusões

## Inteligência Artificial

- Inteligência
  - “faculdade de compreender, um talento, o raciocínio, a habilidade”;
- Inteligência Artificial (IA)
  - “ramo da Ciência da Computação que tem a ver com a automatização de comportamentos inteligentes, próprios dos seres humanos”.

[Luger&Stubblefield, 1998]

## Inteligência Artificial Distribuída

- Inteligência Artificial Distribuída (IAD):
  - “Ramo da IA que estuda a resolução de problemas através de sistemas computacionais distribuídos”.

[Bond & Gasser, 1988]



---

---

---

---

---

---

---

---

## IAD – Agentes e Sistemas Multiagente

- “Estudo, construção e aplicação de sistemas em que diversas entidades computacionais (agentes ou Sistemas Multiagente (SMA)) interagem e perseguem um conjunto de objectivos e/ou realizando um conjunto de tarefas”.

[Weiss, 1999]

---

---

---

---

---

---

---

---

## Contexto

“As plataformas de computação e os sistemas de informação actuais são heterogêneos, abertos, e distribuídos”.

- Os agentes e os SMA surgem como uma nova Metodologia Computacional da engenharia de *software*;
- A ideia chave passa a ser “O agente inteligente que interage”.

---

---

---

---

---

---

---

---

## Agente

- “Algo que age, capaz de produzir um efeito”.



## Noção de Agente

- “Um agente é algo que obtém informação e conhecimento do seu ambiente através de sensores e actua nesse ambiente através de actuadores”;  
[Russell&Norvig, 1995]
- “Agentes como componentes persistentes e activos que percebem, raciocinam, actuam e comunicam”;  
[Huhns&Singh, 1997]
- “Agentes são entidades que habitam em ambientes complexos, sentem esse ambiente e actuam de modo autónomo, procurando executar um conjunto de tarefas para as quais receberam procuração”.  
[Maes, 1990]

## Definição de Agente

Generalizando:

“Um agente corporiza um sistema computacional capaz de revelar uma acção **autónoma** e **flexível**, desenvolvido num determinado universo de discurso. A flexibilidade do agente está relacionada com as suas capacidades de **reacção, iniciativa, aprendizagem e socialização**.”

[Wooldridge, 1999]



## Propriedades de Agentes (I)

Noção **fraca** de agente - Conjunto mínimo de propriedades/características; [Wooldridge & Jennings, 1995]

- **Autonomia:**  
os agentes operam sem intervenção de outros agentes e controlam as suas acções e o seu estado de conhecimento interno;
- **Reactividade:**  
os agentes percebem os eventos que ocorrem no seu universo de discurso e respondem adequada e atempadamente a mudanças ocorridas nesse ambiente;

---

---

---

---

---

---

---

---

## Propriedades de Agentes (II)

- **Pró-Actividade:**  
os agentes tomam iniciativa, conduzindo as suas próprias acções mediante um comportamento dirigido por objectivos;
- **Sociabilidade:**  
os agentes relacionam-se com outros agentes, comunicando, competindo ou cooperando na resolução de problemas que lhes sejam colocados.

---

---

---

---

---

---

---

---

## Noção forte

- Noção **forte** de agente: um agente é considerado forte, quando as entidades com que se depara são eminentemente cognitivas, passíveis de desenvolver a sua própria consciência, de se apresentar como tendo um conjunto de mais-valias como a perceptibilidade, a sentimentalidade e o emocionismo.

[Wooldridge & Jennings, 1995] [Ferber, 1999], [Nwana, 1996], [Russell & Norvig, 1995]

---

---

---

---

---

---

---

---

## Características (I)

- **Mobilidade:**  
capacidade de se movimentar através da rede formada pelos seus pares, executando as tarefas de que foi incumbido;
- **Intencionalidade:**  
capacidade que o agente apresenta para a definição de objectivos e das estratégias para os atingir;

---

---

---

---

---

---

---

---

## Características (II)

- **Aprendizagem:**  
capacidade que o agente ostenta de adquirir conhecimento. A actualização da base de conhecimento é feita através da assimilação de padrões de comportamento ou de preferências;
- **Competência:**  
um agente é competente quando conduz com sucesso e eficiência as tarefas de que é incumbido. A competência está normalmente relacionada com a confiança depositada no agente por terceiros;

---

---

---

---

---

---

---

---

## Características (III)

- **Veracidade:**  
fala-se da veracidade de um agente quando este não fornece, de forma intencional, informação falsa;
- **Racionalidade:**  
um agente racional não aceita realizar tarefas que lhe pareçam impossíveis de executar, contraditórias com os seus princípios ou quando não são compensados em termos do risco, custo ou esforço;

---

---

---

---

---

---

---

---

## Características (IV)

### ■ Benevolência:

um agente benevolente adopta como seus os objectivos de terceiros, desde que estes não entrem em conflito com os seus princípios de natureza ética e/ou deontológica, o que significa que não realizarão todas as tarefas que lhes sejam atribuídas;

### ■ Emotividade:

certas características próprias do ser humano têm vindo a migrar e a estabelecer-se como parte constituinte dos agentes.

---

---

---

---

---

---

---

---

## Agentes versus Objectos

- Agentes são **autónomos**: podem recusar pedidos, os objectos não;
- Os objectos controlam o seu estado e não o seu comportamento; os agentes controlam o seu **estado** e o seu **comportamento**;
- A comunicação nos objectos faz-se através da definição de métodos que são invocados; nos agentes, o conceito de **mensagem** e de **linguagem** é definido ao nível da comunidade de agentes em que a acção ocorre.

---

---

---

---

---

---

---

---

## Agentes versus Sistemas Periciais

Um Sistema Pericial tem como principal objectivo o apoio à resolução de problemas e à tomada de decisão num determinado domínio de conhecimento (consultor).

- lidam com uma **representação do universo** de discurso, não o manipulando directamente nem percepcionando, no imediato, o resultado das suas acções;
- destinam-se, essencialmente, a **assistir peritos** numa determinada área do conhecimento (os agentes resolvem problemas);
- aplicam-se, geralmente, a **tarefas de alto nível** (os agentes dedicam-se a tarefas comuns);
- **não têm capacidade** de tomar **iniciativa** nem têm **autonomia**, respondendo de modo passivo e realizando sempre a mesma acção.

---

---

---

---

---

---

---

---

## Arquitecturas Deliberativas

Seguem a abordagem clássica da IA, onde os agentes actuam com pouca autonomia e possuem modelos simbólicos ou predicativos explícitos dos seus ambientes (universo de discurso), cujo estado de conhecimento pode ser modificado por alguma forma de raciocínio lógico-matemático.



- Um problema de transposição e representação: como traduzir o mundo real em termos de um programa em lógica?
- Um problema de raciocínio: como levar os agentes a raciocinar?

## Arquitecturas Reactivas

Um agente reactivo não possui qualquer representação simbólica ou predicativa do universo de discurso, nem requer formas de raciocínio complexas, *i.e.*, tal agente comporta-se como um autómato inserido no meio ambiente que o rodeia. Procura lidar com a percepção que tem do mundo através da recepção de itens de informação do tipo atómico, que lhe são passados através de sensores.



## Arquitecturas Híbridas

Combinam as características das duas anteriores, a ideia principal passa por categorizar as funcionalidades do agente em camadas dispostas hierarquicamente, onde à camada reactiva é atribuída alguma forma de prioridade sobre a deliberativa, de tal modo que se aproveite uma das suas características mais importantes, que é a de dar resposta rápida a eventos detectados no ambiente.



## Arquitecturas CDI

Nesta abordagem à problemática que está subjacente à construção de arquitecturas para agentes, tem-se como objectivo obter aquela que melhor se adequa à descrição do estado interno de conhecimento de um agente, com base nos seus estados mentais, estados estes de que o agente fará uso para determinar o seu curso de acção:

[Rao & Georgeff, 1995]

- Intenções dos agentes guiam as Acções;
- Intenções baseiam-se em Crenças.



## Sistemas Multiagente

- Um SMA compreende um conjunto de entidades (agentes) que cooperam por forma a solucionar um dado problema, o que normalmente está para lá das suas capacidades individuais.

[Durfee & Rosenshien, 1994]

## Sistemas Multiagente

A interacção entre os agentes de um SMA obriga à análise, definição, especificação e implementação de um conjunto base de funcionalidades:

- Plataforma de comunicação:
  - Meio físico de transmissão da informação;
- Linguagem de comunicação:
  - Significado individual das mensagens;
- Ontologia:
  - Definição do modo de estruturação da comunicação;
- Arquitectura:
  - Determina a interligação entre os sistemas.

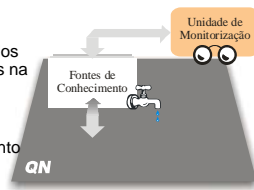
## Linguagens de Comunicação

- KQML<sup>[1]</sup> *Knowledge Query Manipulation Language*;
  - Linguagem externa para comunicações entre agentes;
  - Invólucro para formatar mensagens que determina o significado locutório da mensagem;
  - Não está preocupado com o conteúdo da mensagem;
- KIF<sup>[2]</sup> *Knowledge Interchange Format*;
  - Destina-se a representar o conhecimento sobre um dado domínio de discurso;
  - Forma de definir o conteúdo de mensagens KQML;
- FIPA/ACL<sup>[3]</sup> *Foundation for Intelligent Physical Agents – Agent Communication Language*;
  - Definição da estrutura das mensagens;
- Ontolingua<sup>[4]</sup>;
- XML<sup>[5]</sup> *Extended Markup Language*.

<sup>[1]</sup> <http://www.csee.umbc.edu/kqml>  
<sup>[2]</sup> <http://logic.stanford.edu/kil/kil.html>  
<sup>[3]</sup> <http://www.fipa.org>  
<sup>[4]</sup> <http://www.ksl.stanford.edu/software/ontolingua>  
<sup>[5]</sup> <http://www.xml.com>

## Sistemas Baseados em Quadros Negros

- Um SBQN configura uma espécie de meta-arquitectura (uma arquitectura a partir da qual é possível gerar novas arquitecturas).
- QN - repositório de conhecimento;
- FC - representam o conhecimento dos diversos especialistas intervenientes na resolução do problema;
- UM - é o componente do sistema responsável pelo controlo das alterações ao estado de conhecimento do QN, assim como das acções a executar.



## Áreas de Aplicação

- Industriais
  - Controlo de processos, produção;
- Comerciais
  - Comércio electrónico, gestão de empresas;
- Ensino
  - e-Learning;
- Medicina
  - Monitorização de doentes.

## Coordenação em SMA

- “Acto de trabalhar em grupo, de forma harmoniosa”;  
[Malone et al., 2001]
- Cooperativo – passa por um processo de tomada de decisão em que as partes envolvidas negociam, em termos de alcançarem um ou mais objectivos;
- Competitiva – passa por um processo de decisão em que as partes envolvidas competem tendo em conta um único objectivo.

---

---

---

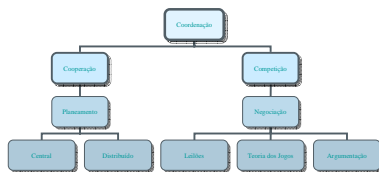
---

---

---

---

## Coordenação em SMA



---

---

---

---

---

---

---

## Ferramentas

- AgentBuilder
  - <http://www.agentbuilder.com/>
- Zeus
  - <http://more.btexact.com/projects/agents/zeus/index.htm>
- Jatlite
  - <http://java.stanford.edu/>
- Aglets
  - <http://www.tri.ibm.com/aglets/>
- Outros
  - <http://www.multiagent.com/Software/index.html>

---

---

---

---

---

---

---

## Conclusões

- Um agente corporiza um sistema computacional capaz de revelar uma acção **autónoma** e **flexível**, desenvolvido num determinado ambiente;
- Um SMA é um sistema computacional em que os **agentes interagem** ou trabalham em conjunto por forma a desempenhar um determinado conjunto de tarefas;
- Coordenar agentes implica colocá-los a trabalhar em conjunto de forma a **atingir um objectivo** comum;
- Coordenação em ambientes **cooperativos** ou **competitivos**.

## Bibliografia

- [Luger & Stubblefield, 1998] George F. Luger, William A. Stubblefield, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Addison Wesley Longman, Inc., 1998.
- [Bond & Gasser, 1988] Alan Bond, Les Gasser, "Readings in Distributed Artificial Intelligence", Morgan Kaufmann Publishers, San Mateo, EUA, 1988.
- [Weiß, 1999] Gerhard Weiß, "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", MIT Press, EUA, 1999.
- [Russell & Norvig, 1995] Stuart J. Russell, Peter Norvig, "Artificial Intelligence – A Modern Approach", Prentice Hall International Inc., EUA, 1995.
- [Huhns & Singh, 1998] Michael N. Huhns, Munindar P. Singh, "Agents and Multiagent Systems: Themes, Approaches and Challenges", Huhns, Singh (editors), Readings in Agents, pp. 1 23, Morgan Kaufmann Publishers, San Francisco, USA, 1998.
- [Maes, 1990] Pattie Maes, "Situated Agents Can Have Goals", Designing Autonomous Agents, Maes (editor), MIT Press, 1990.
- [Wooldridge, 1999] Michael J. Wooldridge, "Intelligent Agents", in Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence, Weiß (editor), chapter 1, pp. 27 77, MIT Press, Cambridge, EUA, 1999.

## Bibliografia

- [Wooldridge & Jennings, 1995] Michael J. Wooldridge, Nicholas R. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, 10 (2), pp. 115 152, 1995.
- [Ferber, 1999] Jacques Ferber, "Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence", Addison Wesley Longman, 1999.
- [Nwana, 1996] Hyacinth S. Nwana, "Software Agents: An Overview", Knowledge Engineering Review, 11 (3), pp. 1 40, 1996.
- [Rao & Georgeff, 1995] Anand S. Rao, Michael P. Georgeff, "BDI Agents: from Theory to Practice", Proceedings of the First International Conference on Multi Agent Systems – ICMAS'95, São Francisco, EUA, 1995.
- [Durfee & Rosenschein, 1994] Edmund H. Durfee, Jeffrey S. Rosenschein, "Distributed Problem Solving and Multi Agent Systems: Comparisons and Examples", Proceedings of the International Workshop on Distributed Artificial Intelligence, Seattle, 1994.
- [Malone et al., 2001] Olson, G. M., Malone, T. W., and Smith, J. B. (Eds.) "Coordination Theory and Collaboration Technology". Mahwah, NJ: Erlbaum, 2001.



## Recursos Electrónicos

- Multiagent Systems:
  - <http://www.multiagent.com/>
- Agent Link:
  - <http://www.agentlink.org/>



---

---

---

---

---

---

---