

Agentes e Sistemas Multi-agentes

Agentes Inteligentes

João Nuno Cardoso Gonçalves de Abreu

Universidade do Minho, Braga, Portugal
a84802@alunos.uminho.pt

Resumo O relatório apresentado têm como objectivo a concepção de um estado de arte sobre *Agentes e Sistemas Multi-agentes*, realizado no contexto da unidade curricular *Agentes Inteligentes*. Serão abordados domínios concretos sobre este tema, requisitos e funcionalidades, metodologia de software, análise de aplicações e plataformas disponíveis, como também a da sua aplicabilidade num domínio escolhido pelo autor do relatório.

Keywords: Agentes Inteligentes · Sistemas Multi-Agentes Inteligentes.

1 Introdução

Há mais de vinte anos que pesquisas na área de agentes e sistemas multi-agentes fornecem resultados destacáveis do ponto de vista teórico e prático. No entanto, depois de tantos anos, não há consenso geral sobre algumas noções básicas como "O que é um agente?", "O que é um sistema multi agente?", qual a terminologia correta a ser usada, etc. Este relatório tem o objectivo de servir como uma introdução "suave" ao campo de agentes e sistemas multi agentes. O principal objectivo é familiarizar os leitores com a terminologia e as definições básicas, bem como a sua aplicação a domínios concretos e a aplicabilidade desta tecnologia para um dado domínio.[1]

2 Requisitos e funcionalidades

2.1 Definições

Agente 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. [2]

No dicionário da Língua Portuguesa é apresentada uma definição de agente na forma: "*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.



Figura 1. Agente em interação com o meio.

No entanto, na comunidade científica, existe um acordo sobre uma das facetas dessas entidades: a que se prende com o conceito de autonomia. Por exemplo, *Wooldridge* 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.

Sistemas Multi-Agentes (SMA) Um Sistema Multi-Agente é um sistema computacional em que dois ou mais agentes interagem ou trabalham em conjunto de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objectivos. A investigação científica e a implementação prática de Sistemas Multi-Agente está focada na construção de *standards*, princípios e modelos que permitam a criação de pequenas e grandes sociedades de agentes semi-autónomos, capazes de interagir convenientemente de forma a atingirem os seus objectivos.[3]

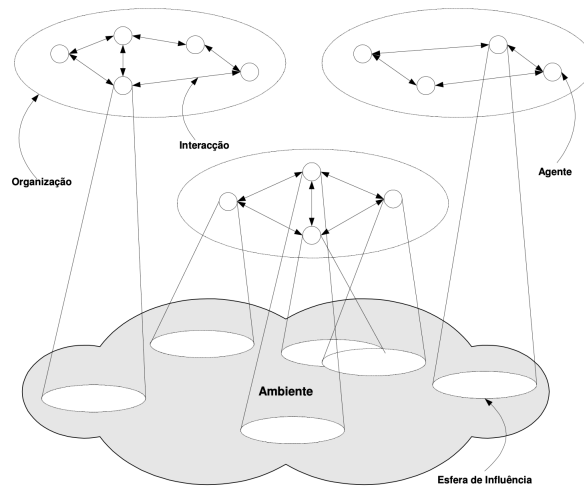


Figura 2. Estrutura de um Sistema Multi-Agente.

2.2 Atributos

Em particular, um agente ou sistema multi-agente inteligente 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:[2]

- **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 entre tanto lhes tenham sido colocados.

Um agente ou Sistema Multi-agente 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 emocionalismo. Por isso, os atributos mínimos para um agente ou sistema multi-agente forte são:

- **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 adota 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 constituinte de agentes e Sistemas Multiagente.

2.3 Tipos de Agentes

Os agentes podem ser agrupados em 4 classes com base no seu grau de inteligência e capacidade em: [4]

- **Simple Reflex Agents** - são agentes que não fazem referência ao histórico e agem apenas com base na percepção atual. A história da percepção é a história de tudo que um agente percebeu até hoje. A função do agente é baseada na regra de ação de condição. Uma regra de ação de condição é uma regra que mapeia um estado, ou seja, uma condição para uma ação. Se a condição for verdadeira, a ação será executada, caso contrário, não. Esta função do agente só é bem-sucedida quando o ambiente é totalmente observável. Para agentes reflexos simples operando em ambientes parcialmente observáveis, loops infinitos são frequentemente inevitáveis. Pode ser possível escapar de loops infinitos se o agente puder realizar ações aleatoriamente. Os problemas com agentes reflexos simples são: inteligência limitada; nenhum conhecimento das partes não perceptivas do estado; geralmente muito grande para gerar e armazenar; caso ocorra alguma alteração no ambiente, o conjunto de regras precisa de ser atualizado.

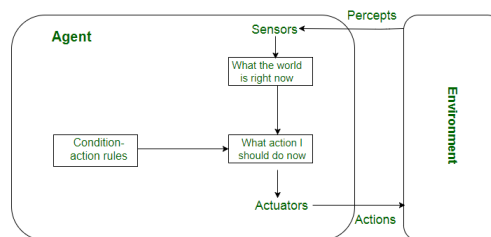


Figura 3. Simple Reflex Agents.

- **Model-based reflex agents** - estes agentes funcionam encontrando uma regra cuja condição corresponda à situação atual. Um agente baseado em modelos pode lidar com ambientes parcialmente observáveis usando um modelo sobre o mundo. O agente deve acompanhar o estado interno que é ajustado por cada percepção e que depende do histórico de percepção. O estado atual é armazenado dentro do agente que mantém algum tipo de estrutura que descreve a parte do mundo que não pode ser vista. Atualizar o estado requer informações sobre: como o mundo evolui de forma independente do agente, e como as ações do agente afetam o mundo.

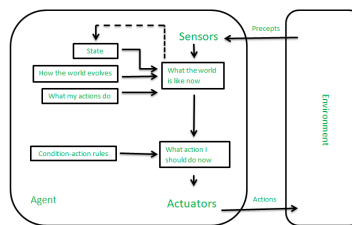


Figura 4. Model-based reflex agents.

- **Goal-based agents** - estes agentes tomam decisões com base em quão longe estão do seu objetivo (existe descrição das situações desejáveis). Todas as suas ações têm como objetivo reduzir a distância da meta. Isso permite ao agente escolher entre múltiplas possibilidades, seleccionando aquela que atinge um estado objetivo. O conhecimento que suporta as suas decisões é representado de forma explícita e pode ser modificado, o que torna estes agentes mais flexíveis.

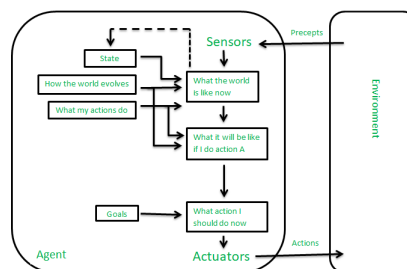


Figura 5. Goal-based agents.

- **Utility-based agents** - os agentes que são desenvolvidos tendo os seus usos finais como blocos de construção são chamados de agentes baseados

em utilidade. Quando há várias alternativas possíveis, para decidir qual é a melhor, são usados agentes baseados em utilidade. Eles escolhem ações com base numa preferência para cada estado. Às vezes, atingir o objetivo desejado não é suficiente. Podemos procurar uma viagem mais rápida, segura e barata para chegar a um destino. A felicidade do agente deve ser levada em consideração. O utilitário descreve o quão "feliz" o agente está. Por causa da incerteza do mundo, um agente utilitário escolhe a ação que maximiza a utilidade esperada. Uma função de utilidade mapeia um estado num número real que descreve o grau de felicidade associado.

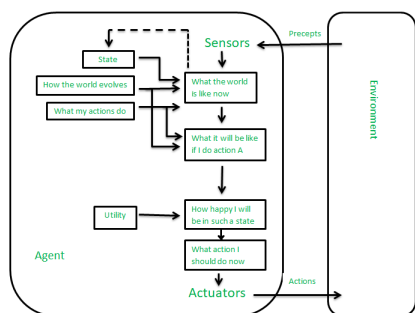


Figura 6. Utility-based agents.

- **Learning Agent** - um agente de aprendizagem em IA é o tipo de agente que pode aprender com as suas experiências anteriores ou que possui capacidades de aprendizagem. Ele passa a atuar com conhecimentos básicos, podendo então agir e adaptar-se automaticamente através da aprendizagem.

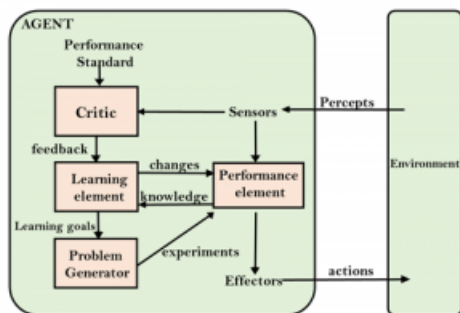


Figura 7. Learning Agent.

2.4 Comunicação

Uma grande vantagem do uso de SMAs é a possibilidade de coordenar agentes colocando-os a trabalhar em conjunto de forma a atingir um objectivo em comum. Para isso precisamos de **comunicação**.

Interoperabilidade - capacidade de um sistema comunicar de forma transparente com outro sistema.

Os padrões de linguagem de comunicação facilitam a criação de software interoperável, dissociando a implementação da interface. Desde que os programas estejam comprometidos com os detalhes dos padrões, não importa como eles são implementados. Hoje, existem padrões para uma ampla variedade de domínios. Por exemplo, programas de e-mail de diferentes fornecedores conseguem interoperar por meio do uso de padrões de correio como *SMTP*.

A linguagem de comunicação do agente (como qualquer outro padrão) deve garantir consistência (as mesmas palavras ou expressões devem ter o mesmo significado para todos os programas) e compatibilidade (todos os programas devem usar as mesmas palavras e expressões para “dizer” as mesmas coisas).[1]

SMA fechado - Arquitetura de desenho estático, com componentes e funcionalidades pré-definidos. As propriedades do sistema são conhecidas antecipadamente:

- Linguagem comum;
- Cada agente pode ser desenvolvido como um *expert*;
- Os agentes são normalmente cooperativos;
- Múltiplas pessoas podem trabalhar em prol do desenvolvimento do sistema, ao mesmo tempo.

Vantagens	Desvantagens
Carga e perícia distribuídas	Custos de manutenção podem ser elevado
Simplicidade e previsibilidade	Pode ser menos tolerantes a falha
	Difícil a interoperabilidade com outros sistemas

SMA aberto - o sistema não tem um desenho ou arquitectura prévios, apenas agentes no seu seio. Os agentes não têm, necessariamente, consciência da existência dos outros. É exigido um mecanismo para identificar, localizar e procurar outros agentes. Os agentes podem ser não-cooperativos, maliciosos ou não-confiáveis.

Vantagens	Desvantagens
agentes e/ou grupos de agentes são concebidos separadamente (modularidade)	comportamento do sistema não é previsível
flexíveis e tolerantes a falhas	protocolos, linguagem, ontologias podem varia entre agentes
desenvolvimento incremental do sistema	podem surgir comportamentos maliciosos, difíceis de evitar
manutenção facilitada	
sociedade aberta e dinâmica	

2.5 Coordenação

Um SMA pode ter forma de comunicar mas se não houver uma coordenação entre agentes não conseguiremos alcançar nenhum objectivo.

Coordenação - Ato de trabalhar em grupo de forma harmoniosa.

Podemos ter 2 tipos de coordenação, que serão apresentados abaixo:

- **Cooperativa** - 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.

Independentemente de serem colaboradores ou competidores, os agentes que formam um Sistema Multi-Agente irão interagir uns com os outros. Esta interacção não é indesejada mas sim intrínseco ao próprio conceito de agente que pressupõe a sociabilidade como forma de um agente atingir os seus próprios objectivos. As interacções pressupõem que os agentes conheçam os outros agentes presentes no ambiente ou, pelo menos, que estejam a par da sua existência, ou seja, que os agentes sejam projectados como membros de uma sociedade multi-agente. No entanto, estas interacções têm de ser convenientemente coordenadas pois caso contrário a sociedade de agentes pode degenerar numa sociedade completamente descoordenada e caótica.

3 Metodologia de software

As principais vantagens do SMA são a **robustez** e a **escalabilidade**. Robustez refere-se à capacidade de, se o controlo e as responsabilidades forem suficientemente partilhados entre os agentes dentro do SMA, o sistema pode tolerar falhas de um ou mais agentes. A escalabilidade do SMA origina-se de sua modularidade. Deve ser mais fácil adicionar novos agentes a um SMA do que adicionar novos recursos a um sistema monolítico. [1]

No entanto, SMAs trazem um preço consigo. Existem também muitas desvantagens em comparação com os sistemas de software tradicionais como: a **limitação na previsibilidade, compreensibilidade e controlo; confiabilidade restrita** para fins computacionais.[5]

Embora um SMA seja confiável no sentido *SPOF* (single point of failure) de que não há um ponto único ou central de falha, porque o processamento é distribuído entre muitos elementos, ele só é confiável num sentido coletivo. O sistema inteiro não falhará se um componente do sistema falhar. Isso significa que o agente individual não importa. Um único agente certamente não é um provedor de serviços confiável (já que ele é um elemento autónomo, de qualquer maneira, que tem o direito de determinar a própria reacção e respostas), e não é adequado para fins computacionais. O facto de ser um sistema inclinado para a facilidade de se obter erros e visto que são sistemas relativamente dinâmicos, a estrutura e organização escolhidas para os sistema pode vir a ser pouco adequada no futura caso sejam introduzidos novos agentes.[5]

4 Análise de aplicações/projectos existentes e plataformas disponíveis

Atualmente, um grande número de plataformas de agentes e linguagens para o desenvolvimento de SMAs podem ser encontradas na literatura. Esta secção apresenta uma ampla variedade de abordagens (comerciais e académicas) encontradas no domínio do agente.

- **JADE** - é uma plataforma de agente *open-source* eficiente e estável, em conformidade com o *FIPA*, desenvolvida pela *TILAB*. O JADE é amplamente utilizado tanto em pesquisas como em projetos comerciais e tem uma comunidade de utilizadores e programadores muito activa.[7]
- **IMPACT** - o principal objectivo deste projecto é desenvolver ambos teoremas como implementações de software que facilitem a criação, *deployment*, interacção e aspectos colaborativos de agentes de *software* num ambiente heterogéneo e distribuído. Esta ferramenta também fornece um conjunto de servers (yellow pages, registos, tipos e interface) que facilitam a interoperabilidade dos agentes numa aplicação. Também fornece um *Agent Delopment Environment* para criação, testes e deployment de agentes.[8]
- **MadKit** - é uma plataforma multi-agente escalável escrita em Java e construída sobre o modelo de organização AGR (Agent/Group/Role): agentes estão situados em grupos e funções. MadKit permite alta heterogeneidade nas arquitecturas de agentes e nas linguagens de comunicação. A comunicação nesta ferramenta é feita num mecanismo *peer to peer*, e permite aos programadores um rápido desenvolvimento de aplicações distribuídas usando princípios de multi-agentes.[9]
- **INGENIAS** - esta ferramenta tenta ligar diretamente à implementação a análise e *design*. Quando comparada com outros ambientes de desenvolvimento, esta ferramenta possui funcionalidades diferentes, como a geração

de código e mudanças feitas. Também tem instalações de *debugging* e apresenta capacidade receptiva (podemos integrar novas funções com uso de *plugins*).[10]

5 Aplicabilidade a domínio - Indústria

A abordagem SMA é adequada para apoiar os requisitos atuais de sistemas de controlo moderno em domínios industriais, fornecendo flexibilidade, robustez, escalabilidade, adaptabilidade, configurabilidade e produtividade. SMA está a ser aplicado com sucesso numa ampla gama de domínios, ou seja, comércio electrónico, gráficos (por exemplo, computador jogos e filmes), transporte, logística, robótica, fabricação, telecomunicações e energia. Como exemplos, pode-se citar a aplicação de soluções de sistemas multi agentes na fábrica de motores *Daimler Chrysler* em Stuttgart [11], Tankers International que opera um dos maiores reservatórios de petróleo do mundo [12], *Air Liquide America* para otimizar a distribuição de gases médicos e industriais [13] e navios da Marinha dos EUA para controlar sistemas de aquecimento, ventilação e ar condicionado (AVAC) [14]. Uma análise profunda das aplicações industriais de SMA pode ser encontrada em [15,16].

A análise das aplicações industriais pesquisadas de soluções baseadas em agentes permite extrair as seguintes conclusões:

- Adopção relativamente pequena de agentes na indústria, sendo o implementado aplicativos limitados em termos de funcionalidade.
- As soluções abordam principalmente o controle de alto nível ou software puro sistemas (por exemplo, o comércio electrónico).
- Pouco entusiasmo por parte dos fornecedores de tecnologia e da indústria empresas.

As razões para esta fraca adopção na indústria já foram amplamente discutidas em a literatura de vários autores, nomeadamente [15,17]. Resumidamente, os principais *road-blockers* são o investimento inicial necessário, a necessidade de adoptar o pensamento distribuído, a interoperabilidade em sistemas heterogéneos distribuídos, a padronização ausente, o restrições em tempo real e a falta de maturidade da tecnologia.

6 Conclusão

A IA oferece um conjunto de vantagens para melhorar o desempenho de sistemas complexos automáticos, e os sistemas multi agentes, como um paradigma derivado de IA, é adequado para atender aos requisitos atuais impostos às empresas. Apesar de já ser adoptado em diversos domínios, a tecnologia multi agente ainda tem um longo e difícil caminho a ser percorrido para uma aceitação mais ampla desses conceitos de IA, seja em que domínio for.

Referências

1. Agents and Multi-Agent Systems: A Short Introduction for Power Engineers, by Dr.Mevludin Glavic, May 2006
2. Sebenta de Agentes Inteligentes, Paulo Novais e César Analide (2006)
3. Sistemas Multi-Agente, Luis Paulo Reis, paginas.fe.up.pt/~lpreis/Tese/Capitulo3.PDF
4. Agents in Artificial Intelligence, Sahil Bansall, 2019, www.geeksforgeeks.org/agents-artificial-intelligence/
5. CAS-Wiki, Multi-Agent System - CasGroup, wiki.cas-group.net/index.php?title=Multi-Agent_systemAdvantages_andDisadvantages
6. A Review on Multi-Agent platforms and Environmental Decision Support Systems Simulation tools, Thania Rendón-Sallard and Miquel Sànchez-Marrè, upcommons.upc.edu/bitstream/handle/2117/87344/R06-4.pdf
7. JADE, jade.tilab.com
8. IMPACT, www.cs.umd.edu/projects/impact
9. MadKit, www.madkit.org
10. INGENIAS, ingenias.sourceforge.net
11. Schild, K. and Bussmann, S.: Self-Organization in Manufacturing Operations, Communications of the ACM, 50(12), 74–79 (2007).
12. Himoff, J., P. Skobelev, P., Wooldridge, M.: Magenta Technology: Multi-agent Systems for Industrial Logistics. In Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems, 60–66 (2005).
13. Harper, C., Davis, L.: Evolutionary Computation at American Air Liquide. In Evolutionary Computation in Practice, Studies in Computational Intelligence, 88, Springer Berlin/Heidelberg, 313–317 (2008).
14. Maturana, F., Staron, R., Hall, K., Tichy, P., Slechta, P., Marík, V.: An Intelligent Agent Validation Architecture for Distributed Manufacturing Organizations. In Camarinha-Matos, L. (ed.), Emerging Solutions for Future Manufacturing Systems, Springer, 81–90 (2004).
15. Leitão, P.: Agent-based Distributed Manufacturing Control: A State-of-the-art Survey. Engineering Applications of Artificial Intelligence, 22(7), 979–991 (2009).
16. Monostori, L., Váncza, J., Kumara, S.: Agent-Based Systems for Manufacturing. Annals of the CIRP, 55/2, 697–720 (2006).
17. Marik, V., McFarlane, D.: Industrial Adoption of Agent-Based Technologies. IEEE Intelligent Systems, 20(1), 27–35 (2005).