Universidade Federal do Rio de Janeiro

COPPE/Programa de Engenharia de Sistemas e Computação

Disciplina: Inteligência Artificial

Profª. Inês de Castro Dutra

UMA INTRODUÇÃO AOS SISTEMAS MULTI-AGENTES

Regiane Felipe de Oliveira

Junho / 2003

Índice

Capítulo 1

Introdução

Nos últimos tempos, tem se observado cada vez mais uma mudança de enfoque dos sistemas de computação: o foco, anteriormente em sistemas centralizados e individuais, vem sendo transferido para sistemas dinâmicos e distribuídos. Em conseqüência disso, surgem a cada momento novas oportunidades e desafios tecnológicos, como a necessidade de se desenvolver sistemas que funcionem efetivamente em meio às rápidas mudanças de circunstâncias e ao crescimento elevado do volume de informação.

Tais desafios implicam na necessidade de uma melhoria nos modelos tradicionais de computação e da criação de novos paradigmas. Neste contexto, destaca-se o novo paradigma de agentes, que visa a possibilidade de uso de componentes que respondam dinamicamente a tais variações de circunstancias, conferindo maior autonomia ao software [1].

Sistemas baseados em agentes têm sido uma das mais importantes áreas em desenvolvimento na última década, tendo correlação com diversas sub-áreas da Tecnologia de Informação, incluindo a Inteligência Artificial Distribuída, ramo mais específico da Inteligência Artificial que compreende os Sistemas Multi-Agentes, principal enfoque deste trabalho.

O trabalho tem por objetivo introduzir alguns fundamentos dos Sistemas Multi-Agentes, e está subdividido da seguinte maneira: no capítulo 2 serão apresentados conceitos básicos para o entendimento dos Sistemas Multi-Agentes: a seção 2.1 aborda a Teoria de Agentes; a seção 2.2 descreve a Arquitetura dos Agentes. Na seção 2.3, a Linguagem de comunicação desses agentes é enfocada e algumas aplicações dos Sistemas Multi-Agentes são descritas na seção 2.4. O capítulo 3 apresenta algumas perspectivas do desenvolvimento de Sistemas Multi-Agentes nos próximos anos. Por fim, o presente estudo é concluído no capítulo 4, onde serão feitas algumas considerações finais.

Capítulo 2

Sistemas Multi-Agentes – Conceitos Básicos

A Inteligência artificial vem sendo largamente difundida e utilizada ao longo dos últimos anos. O esforço em projetar sistemas que "agem" racionalmente tem ocasionado diversas modificações na computação. Além disso, o grande crescimento da utilização de programas em rede, onde muitas vezes a informação se encontra distribuída através de seus nós, deu um grande impulso à computação distribuída, fomentando pesquisas na área de Inteligência Artificial Distribuída (IAD).

Uma das motivações para o crescimento dessa área reside no fato de que situações em que entidades computacionais possuem condições para resolver um determinado problema sem auxílio de outras entidades vêm se tornando cada vez mais raras. A necessidade de cooperação entre tais entidades se faz sentir. Os problemas são cada vez mais de natureza descentralizada, e, conseqüentemente, arquiteturas distribuídas vêm se mostrando mais úteis em sua resolução.

Nesse sentido, é introduzido o *agente*, entidade computacional autônoma, dotada de capacidades específicas e encapsuladas, como base para um novo tipo de sistema, que visa solucionar mais eficientemente tais problemas. A estratégia é agrupar os agentes que possuam parte do conhecimento do domínio de um determinado problema e fazer com que esses agentes interajam entre si, com o objetivo de complementarem suas habilidades [2].

Na IAD, destacam-se então duas classes de sistemas com múltiplos agentes:

- **Sistema de Resolução Distribuída de Problemas**, nos quais os agentes são projetados visando a cooperação mútua, para atingirem um dado objetivo, e assumindo-se que todos serão conhecidos *a priori*.
- **Sistemas Abertos**, onde cada agente é projetado independentemente da existência de outros, não necessariamente para um objetivo em comum, podendo fazer parte do sistema (ou não mais) conforme o necessário, isto é, dinamicamente.

É na classe de sistemas Abertos que se encontram os **Sistemas Multi-Agentes (SMA)**. Os SMA podem ser definidos como um conjunto de agentes autônomos que interagem num ambiente compartilhado, para resolver problemas que estão além de suas capacidades individuais [3], [4]. Assim, o comportamento global do sistema é derivado da interação entre os agentes que fazem parte do sistema [2].

A pesquisa em SMA é de natureza multidisciplinar: a Psicologia, a Biologia, a Sociologia, a Ciência Cognitiva, entre outras, deram sua parcela de contribuição para a área, alem da Ciência da Computação. Em particular, a Sociologia marca a transição dos sistemas de Inteligência Artificial tradicionais (monolíticos) para os sistemas multi-agentes, baseando-se numa corrente filosófica que sustenta de que toda atividade humana é uma prática social, só ocorrendo quando existe interação entre os indivíduos [4].

Outra motivação para os SMA é o fato de existirem metáforas que possibilitam analogias entre projetos de sistemas computacionais e o mundo real, tornando mais natural a abordagem dos multi-agentes. Desse modo, sistemas podem ser vistos como ambientes (em analogia com ambientes físicos), onde os agentes atuam, e podem ser agrupados de forma a se comportarem como comunidades ou sociedades de agentes, tal como no mundo real. Os agentes podem ainda ser programados como um conjunto de sub-agentes, que por sua vez formam grupos, sociedades, etc. A essa propriedade dá-se o nome de **Princípio da Recursividade em SMA** [4].

Existem, segundo o site [5], dois grandes grupos de SMA: SMA Reativos e SMA Cognitivos, que podem se descritos como segue:

- **Sistemas reativos**: "Os sistemas de agentes reativos são constituídos por um grande número de agentes. Estes são bastante simples e fortemente acoplados, apresentam uma granularidade fina, não possuem inteligência ou representação de seu ambiente e interagem utilizando um comportamento do tipo estímulo/resposta. Um comportamento inteligente emerge a partir das interações entre estes agentes e seu ambiente. Ou seja, os agentes não são inteligentes individualmente, mas o comportamento global é".

- **Sistemas Cognitivos:** "Os sistemas de agentes cognitivos são geralmente constituídos de um pequeno número de agentes, onde cada agente possui uma quantidade considerável de recursos. Estes agentes são inteligentes, contêm uma representação parcial e explícita de seu ambiente, têm uma capacidade local de decisão e podem negociar uma informação ou um serviço. Eles são em geral dotados de conhecimentos, competências, intenções e planos, o que lhes permite coordenar suas ações visando à resolução de um problema" .

Em sistemas cognitivos, cada agente trata unicamente o subproblema que requer especificamente sua competência. A decomposição do problema pode não ser conhecida *a priori*, porém os agentes são capazes de distribuir os subproblemas entre si baseando-se em suas próprias competências. Além disso, eles podem desenvolver atividades cooperativas e coordenadas para resolver um problema. Assim, um SMA pode ser definido em função da autonomia de cada agente e dos meios que os mesmos dispõem para gerar suas interações.

Algumas características são tidas como grandes vantagens do uso de SMA sobre o uso de sistemas monolíticos [3]:

- **Rapidez**: o aproveitamento de processamento paralelo acrescenta melhoria significativa ao desempenho do sistema.
- **Redução de Comunicação**: somente soluções parciais de alto nível são transmitidas, ao invés da transmissão de dados básicos a um sistema centralizado.
- **Flexibilidade**: a utilização de agentes com diferentes habilidades permite que a cooperação entre eles se faça de forma dinâmica.
- **Confiabilidade**: se algum agente falha em sua operação, outro agente pode assumir o seu lugar.

Além disso, a autonomia conferida aos agentes é de grande valia para situações em que o sistema implementado é dinâmico, tornando impossível a previsão de todos os casos que devem ser tratados: como os agentes buscam alcançar seus próprios objetivos, eles podem perfeitamente ser projetados visando a adaptação a situações imprevistas.

Outros argumentos são utilizados pela comunidade de IAD para salientar vantagens do uso de SMA. Pesquisadores do assunto afirmam que [6]:

- Os SMA diferem dos outros modelos conhecidos pelo fato de permitirem soluções razoáveis, em termos de custos e de tempo envolvido, para a resolução de problemas particularmente complexos, ou cuja solução exigisse múltiplas intervenções (por exemplo, um sistema de tomada de decisão em grupo, em que uma decisão final é resultado de um processo baseado em decisões parciais de múltiplos atores, muitas vezes com objetivos divergentes).

- Os agentes permitem eliminar as diferenças entre os diferentes tipos de redes e sistemas de informação existentes e, conseqüentemente, eliminar, ou atenuar, as fronteiras entre todas essas redes e respectivos sistemas, do ponto de vista da aplicação. Ou seja, os agentes permitem - ao encapsularem a complexidade e especificidades inerentes a cada rede e sistema computacional - a concepção e construção de aplicações altamente distribuídas e heterogêneas.

Na tentativa de um melhor entendimento e desenvolvimento de SMA, os estudos na área podem ser classificados em três sub-áreas, que serão, de forma sucinta, descritas a seguir.

2.1 - Teoria de Agentes

O conceito de "agente" pode ser visto como um paradigma, isto é, uma visão de solução de um problema. Por esse motivo, e também pelo fato de esse paradigma envolver diversas áreas de aplicação, os pesquisadores não chegaram a um consenso acerca de uma definição única. Algumas são mais divulgadas e aceitas, como por exemplo, a de Wooldridge e Jennings (citado por [6]) e Franklin (citado por [4]).

Wooldridge propõe um modelo genérico para agente, como apresentado na Figura 2.1, e define um agente segundo características que ele chama de "visão fraca" e "visão forte".

A visão fraca engloba os atributos essenciais, isto é, um conjunto mínimo de características que, em geral, um agente deve possuir, tais como:

- *Autonomia*: agente exerce controle sobre suas próprias ações e estado interno e agem sem a intervenção direta das pessoas. Isto significa que devem possuir a capacidade de resolução, baseado no conhecimento, gerar possíveis cursos de ação, e que de acordo com as metas, possam tomar decisões e executar ações;
- *Sociabilidade*: habilidade de comunicação com outros agentes e também com pessoas, através de algum tipo de linguagem;
- *Reatividade*: agentes percebem seu ambiente (que pode ser o mundo real, um usuário, através da interface gráfica, um grupo de outros agentes, Internet, ou a combinação destes) e agem em relação às mudanças que nele ocorrem, de forma seletiva.Tais agentes são designados reativos e são em geral simples e fáceis de desenvolver. Baseiam-se sobre três componentes principais: percepção, ação e comunicação;
- *Proatividade*: agentes não simplesmente agem em resposta a seu ambiente, como também são capazes de exibir comportamento orientado a objetivo por tomar iniciativas;
- *Persistência*: Os agentes mantêm consistentemente o seu estado interno ao longo da sua existência.

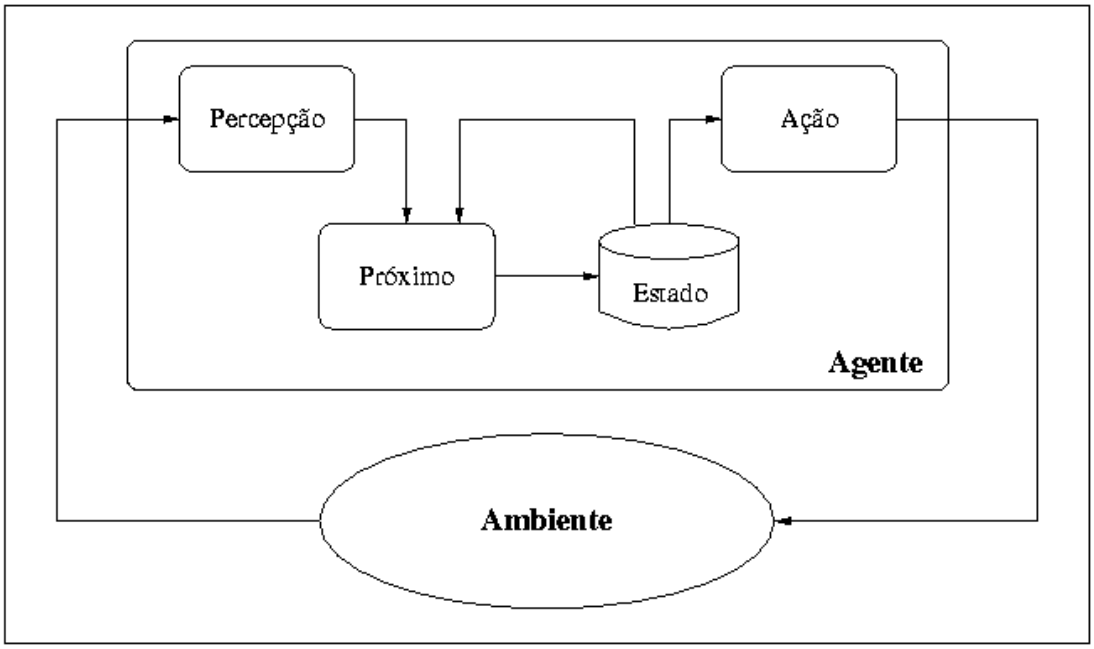

Figura 2.1 – Modelo Genérico de Agente

Outros atributos complementam o conjunto supracitado, designando a noção "forte" de agente. Esses atributos são também chamados de antropomórficos, e têm relação com características de seres humanos, no que diz respeito ao uso de noções mentais e emocionais. Alguns desses atributos são:

- *Mobilidade*: capacidade de migrar de uma plataforma para outra.Caso contrário, diz-se estático ou estacionário. (Esta característica é essencial para os agentes vistos segundo a perspectiva da comunidade de Sistemas Distribuídos).
- *Aprendizagem:* A capacidade de aprendizagem está intrinsecamente associada com a capacidade de manipulação e geração de conhecimento. Os agentes com capacidade de aprendizagem (por exemplo, assistentes pessoais, clientes de correio eletrônico inteligentes) vão, à medida da sua utilização, reconhecendo padrões de comportamentos, padrões de preferências, etc., e atualizando a sua base de conhecimentos.
- *Veracidade:* os agentes não devem comunicar, de forma deliberada, informações falsas;
- *Benevolência*: os agentes não têm metas contraditórias, e procuram realizar as tarefas que lhes foram solicitadas;

- *Racionalidade*: um agente atua para alcançar as suas metas, na medida que suas crenças, conhecimentos e capacidade de raciocínio dêem permissão ou condição;
- *Intencionalidade*: Capacidade de representação explicita dos objetivos de um agente. Estes agentes, ditos intencionais ou cognitivos, apresentam quatro componentes. Para além da percepção, ação e comunicação, apresentam ainda, capacidade de raciocínio sobre uma base de conhecimento;
- *Características Mentais*: A atribuição de noções mentais - conhecimento, crenças, intenções ou desejos - aos agentes foi um passo importante na comunidade de IAD. De entre as teorias envolvidas destacam-se os trabalho de Rao e Georgeff (citado por [6]) baseado em agentes BDI (agentes com crenças, desejos e intenções) e de Wooldrige e Jennings com agentes baseados em atitudes e pró-atitudes (citado por por [6]).

Outras tentativas de definir o conceito de agente foram surgindo mais recentemente, como por exemplo, a visão de pesquisadores da IBM, que caracterizam gentes em termos de três pontos principais: agenciamento, inteligência e mobilidade. Tal definição é dada da seguinte maneira:

"*Agenciamento é o grau de autonomia e autoridade investido no agente, e pode ser medido, pelo menos qualitativamente, pela natureza da interação entre o agente e outras entidades no sistema. No mínimo o agente deve ser executado assincronamente. O grau de agenciamento é realçado se um agente representa um usuário de alguma forma (perfil)... Um agente mais desenvolvido pode interagir com ... dados, aplicações... serviços... (ou) outros agentes. Inteligência é o grau do comportamento de raciocínio e aprendizado: a capacidade do agente de aceitar a declaração de objetivos do usuário e cumprir a tarefa delegada a ele. ...Além disso, na escala de inteligência estão sistemas que aprendem e se adaptam ao seu ambiente ... . Mobilidade é o grau no qual agentes viajam através da rede...*"(citado por [3])

Segundo o trabalho [3], agentes podem ser classificados de acordo com: 1) capacidade de resolução de problemas; 2) aprendizagem, cooperação e autonomia e 3) Outros agentes.

Quanto à capacidade de resolver problemas, tem-se:

a) **Agentes Reativos**: reagem a mudanças de seu ambiente ou mensagens provenientes de outros agentes; não são capazes de raciocinar sobre as suas intenções.Suas ações se realizam como resultado de regras que se disparam ou da execução de planos. Exemplo: sistemas especialistas de primeira geração, compostos de uma base de conhecimento que contém conjuntos de regras, fatos e uma máquina de inferências;
b) **Agentes Intencionais**: São capazes de raciocinar sobre suas intenções e conhecimentos, criar planos de ações, e executá-los. Exemplo: sistemas geradores de planos;
c) **Agentes Sociais**: possuem a capacidade dos agentes intencionais, e os modelos explícitos de outros agentes.

Quanto à aprendizagem, autonomia e cooperação tem-se:

a) **Agentes Colaboradores**: enfatizam sua autonomia e cooperação com outros agentes para realizar as suas tarefas. Podem ser usados para resolver problemas muito complexos e que, geralmente, são por natureza de solução distribuída.
b) **Agentes de Interface:** colocam ênfase na sua autonomia e capacidade de aprendizagem para realizar as suas tarefas. Essencialmente, os agentes de interface assistem e dão apoio ao usuário para aprender o uso de uma aplicação.

Outros tipos de classificação de agentes:

a) **Agentes Móveis**: Os agentes móveis são programas capazes de viajar através de redes de computadores, de interagir com Hosts, pedir informação em nome de seu usuário e voltar ao seu lugar de origem, onde recebeu as tarefas especificadas pelo seu usuário.
b) **Agentes de informação**: realizam a tarefa de administrar, manipular ou coletar informação proveniente de várias fontes distribuídas.
c) **Agentes Híbridos**: agentes híbridos são os que possuem duas ou mais combinações das capacidades dos tipos anteriormente mencionados.

2.2 Arquitetura de Agentes

Entende-se por arquitetura de agentes a maneira de se construir um sistema de forma a satisfazer as propriedades especificadas pela Teoria de Agentes, definindo que estruturas computacionais serão utilizadas [3].

Segundo Maes et al. (citado por [3]), a arquitetura de agentes está dividida em três abordagens: Abordagem Clássica, Abordagem Alternativa e Abordagem Híbrida.

A Abordagem Clássica (ou Deliberativa) é tratada como um tipo particular de sistema conhecido como inteligência Artificial Simbólica. Essa arquitetura é definida como uma representação explícita dos agentes, através de um modelo simbólico do mundo.

As mais importantes arquiteturas dessa natureza são baseadas num modelo fundamentado em três principais atitudes mentais, que são as crenças, desejos e intenções, e são conhecidas por **Arquitetura BDI** (do inglês *beliefs, desires* e *intentions*). Um exemplo de arquitetura desenvolvida segundo a abordagem clássica é o IRMA (*Intelligent Resource-boundered Machine Architecture*) (BRATMAN, citado por [4]).

De forma esquemática, Wooldridge (citado por [6]) propõe uma arquitetura BDI genérica, como apresentado na figura 2.2.

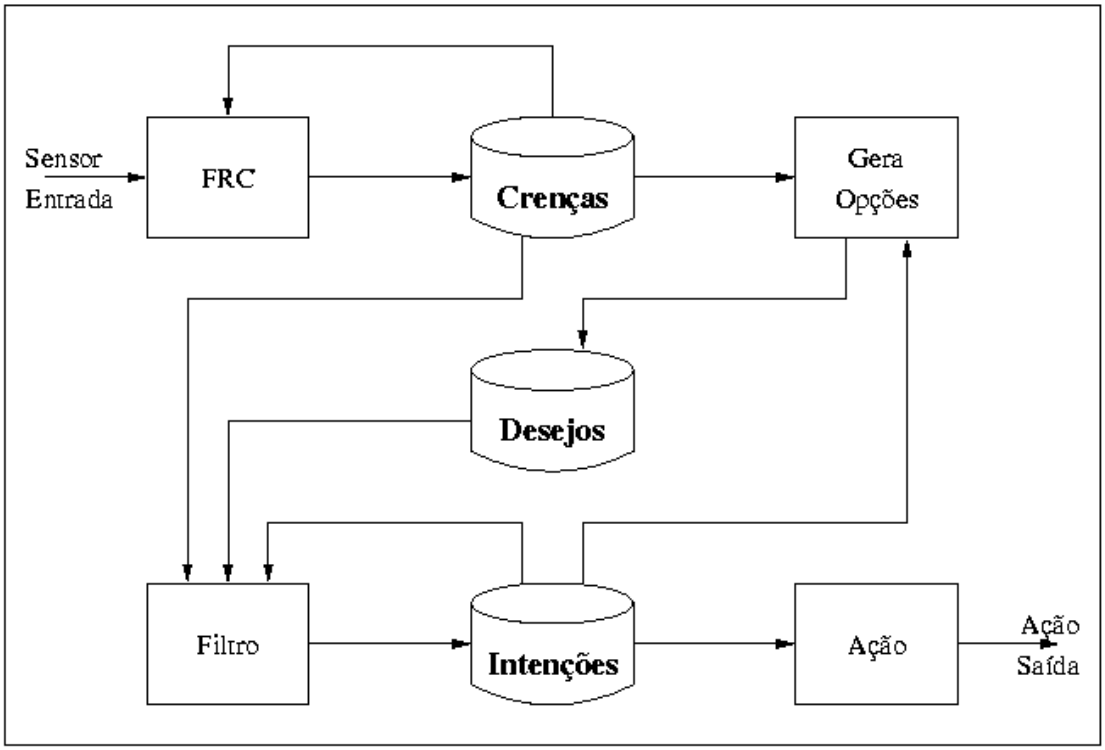

Figura 2.2 – Arquitetura BDI genérica

Resumidamente, esta arquitetura de agente está estruturada da seguinte forma: as *crenças* representam aquilo que o agente sabe sobre o estado do ambiente e dos agentes naquele ambiente (inclusive sobre si mesmo). Os *desejos* representam estados do mundo que o agente quer atingir (isto é, quer se tornem verdadeiros). O*bjetivos* são um subconjunto dos desejos que são todos compatíveis entre si. As *intenções* representam seqüências de ações específicas que um agente se compromete a executar para atingir um determinado objetivo.

A *função de revisão de crenças* (representada na figura por *FRC*) recebe a informação referente a alterações no ambiente e, consultando as crenças anteriores do agente, atualiza estas crenças para que elas reflitam o novo estado do ambiente. Com esta nova representação do estado do ambiente, é possível que novas opções, referentes a estados a serem atingidos, fiquem disponíveis. A função *gera opções* verifica quais as novas alternativas para possíveis ações (consultando as intenções com as quais o agente já esteja comprometido), e então uma escolha dentre as novas opções com as quais o agente se comprometerá é realizada atualizando então os desejos do agente. Definidos o conhecimento e a motivação do agente, é necessário decidir que curso específico de

ações será utilizado para alcançar os objetivos atuais do agente. Para isto, é preciso considerar os demais cursos de ações com os quais o agente já tenha se comprometido, para evitar ações incoerentes. Finalmente, a função *filtro* atualiza o conjunto de intenções do agente, com base nas crenças e desejos já atualizados e nas intenções já existentes. Dado um conjunto de intenções, a escolha de qual ação específica será realizada no ambiente pelo agente a cada momento, pela função *ação*, é relativamente simples.

A Abordagem Alternativa é uma arquitetura reativa que não inclui nenhuma espécie de modelo simbólico central de mundo e não usa raciocínio simbólico complexo. Ao contrário, propõe que o comportamento inteligente pode ser gerado sem representações explícitas ou raciocínio abstrato, mas como uma propriedade emergente de certos sistemas complexos [3].

A Arquitetura Híbrida propõe um subsistema deliberativo que planeja e toma decisões da maneira proposta pela Inteligência Artificial simbólica e um reativo capaz de reagir a eventos que ocorrem no ambiente sem ocupar-se de raciocínios complexos. Segundo pesquisadores, arquiteturas híbridas têm algumas vantagens sobre as anteriormente expostas, mas não é claro como gerenciariam diferentes níveis de comportamento abstrato.

A figura 2.3 ilustra uma arquitetura genérica de SMA.

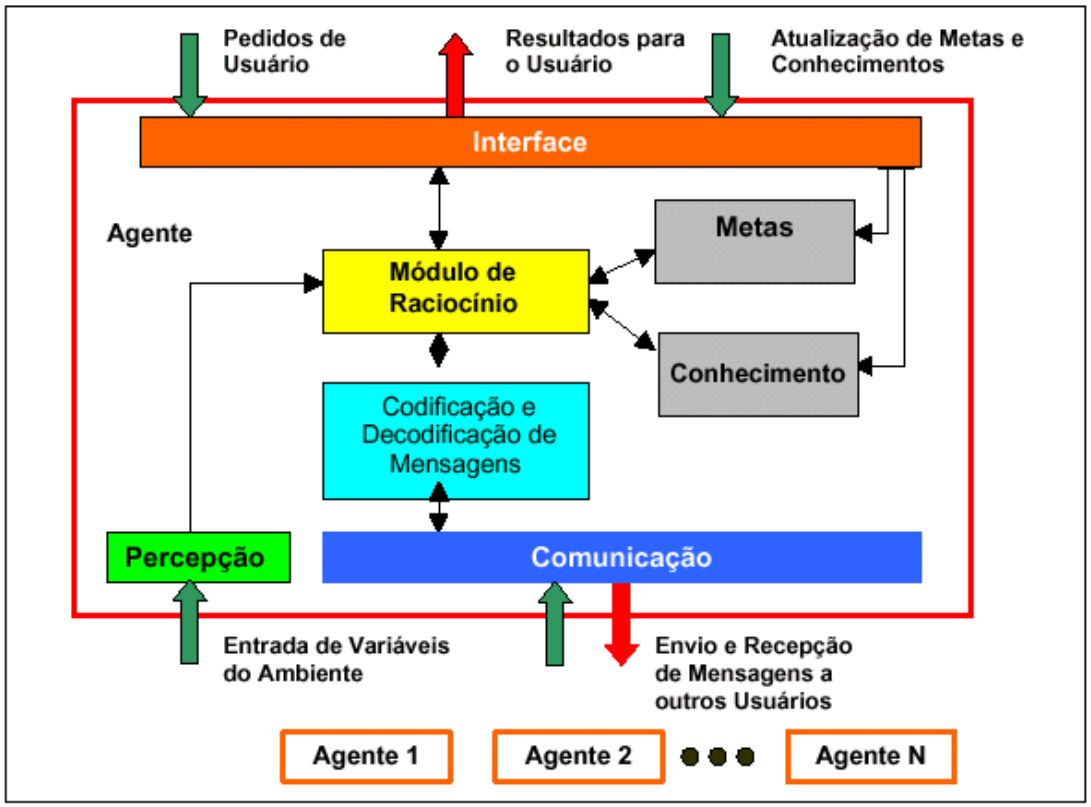

Figura 2.3 – Arquitetura genérica de SMA

Os componentes dessa arquitetura serão descritos a seguir:

- *Interface* com o usuário: encarrega-se de receber informações do usuário, enviá-los ao módulo de raciocínio e apresentar os resultados ao usuário.
- *Módulo de raciocíni*o: encarrega-se, com base no conhecimento e nas metas, de avaliar as diferentes alternativas de solução além de negociar e de selecionar a melhor opção. É capaz de negociar com outros agentes. Pode, com base nas mensagens que recebe dos outros agentes, ou da percepção das variáveis do ambiente, atualizar sua base de conhecimento.
- *Meta*s: correspondem aos estados finais na busca de soluções de agentes;
- *Base de conhecimento*: refere-se às informações que o agente dispõe do ambiente;
- *Módulo de codificação e decodificação de mensagen*s: encarrega-se de codificar as mensagens do agente no formato de alguma linguagem de comunicação de

agentes. É capaz de receber mensagens de outros agentes, decodificá-las, e enviá-las ao módulo de raciocínio;

- *Módulo de percepção*: refere-se aos meios com os quais o agente monitora as variáveis do ambiente que o cerca;
- *Módulo de comunicação*: encarrega-se de enviar e receber mensagens de outros agentes mediante protocolos de transporte [3].

2.3 Linguagens de Agentes

Entende-se por linguagem de agentes a definição de uma linguagem na qual se possa programar sistemas que satisfaçam os conceitos propostos pela Teoria de Agentes, de modo a reduzir inconsistências e variações de notações particulares, definindo uma comunicação global entre desenvolvedores.

Inicialmente, os pesquisadores implementaram sistemas de agentes com linguagens de programação da Inteligência Artificial Clássica, como por exemplo, Lisp e Prolog. Depois de alguns anos, os percebeu-se que linguagens específicas (orientadas a agentes) podem facilitar o desenvolvimento e manutenção desses sistemas. Passou-se então a utilizar linguagens de comunicação entre agentes

As linguagens concorrentes, orientadas a objetos, são consideradas precursoras das linguagens de agentes. O fato de essas linguagens unirem os aspectos de concorrência -distribuição, tolerância a falhas, capacidade de recuperação de informações, etc.- com os aspectos de linguagens orientadas a objetos - modularidade, extensibilidade, flexibilidade, etc. – as tornam apropriadas para o desenvolvimento e a manutenção de sistemas distribuídos. Além disso, a propriedade que o agente possui de mostrar um comportamento autônomo inteligente faz com que o conceito de agente estenda o conceito de objeto ativo.

Algumas linguagens concorrentes, orientadas a objeto, têm sido usadas para desenvolver SMA. Entretanto, seu principal limite é a ausência de primitivas de nível superior para programação multi-agentes, tornando difícil seu uso para o desenvolvimento de aplicativos.

Alguns exemplos de linguagens de agentes são relacionados a seguir:

- **Agent0** - Linguagem baseada em um modelo de agente, no qual os componentes principais são o estado mental e um conjunto de ações (ações privadas e comunicativas), e um interpretador para definir e programar agentes. Foi definida após a introdução de um novo paradigma de programação, chamado *programação orientada a agente*, como uma especialização de programação orientada a objetos. Uma extensão de Agent0 introduz alguns operadores para planejar ações e alcançar objetivos.
- **MetateM** - Linguagem concorrente orientada a agente, baseada na lógica, que oferece uma especificação temporal lógica de um agente e permite a execução de suas regras a fim de implementar o comportamento do agente. Agentes se comunicam através de um mecanismo de transmissão e a extensão de mensagens transmitidas é restrita através da possibilidade de organizar agentes em grupos.
- **APRIL** e **MAIL** - linguagens definidas pelo projeto IMAGINE da ESPRIT, definidas e integradas em um dos primeiros ambientes para desenvolver aplicações multi-agentes. APRIL oferece meios de programação de nível inferior que permitem multitarefa, comunicação, união de padrões e capacidades de processamento simbólico. MAIL oferece meios de programação de nível superior para cooperação entre agentes. Em particular, MAIL é baseada em um conjunto de primitivas permitindo a um agente propor um objetivo a outros agentes, a fim de refinar um plano compartilhado com outros agentes e/ou aceitar ou rejeitar uma tarefa requerida por outro agente.
- **Telescript** – Linguagem considerada a primeira linguagem comercial de agente. Agentes Telescript são processos de software capazes de se mover de um lugar para outro, e de se comunicar uns com outros baseados em diversos padrões. Telescript oferece um ambiente composto de: i) a linguagem Telescript para a definição de agentes; ii) um instrumento, o interpretador Telescript; iii) um conjunto de protocolos para suportar o transporte de agentes entre plataformas; iv) um conjunto de ferramentas para desenvolver aplicações.

- **KQML** (Knowledge Query and Manipulation Language) – Linguagem que representa parte de um grande esforço de um grupo chamado ARPA Knowledge Sharing Effort, que impulsionou o desenvolvimento de técnicas e metodologias para construir bases de conhecimento em larga escala para que possa ser compartilhado e reutilizável. KQML é uma forma de utilizar e manipular mensagens para suportar compartilhamento de conhecimento entre agentes. A linguagem possui um conjunto de mensagens que viabilizam a comunicação entre os agentes. KQML foi construída com o objetivo de independer de qualquer tipo de padronização, e por este motivo não se preocupa em saber o que efetivamente está escrito no conteúdo da mensagem, e sim, em identificar o inicio e fim da mensagem. Além de ser uma poderosa linguagem, é também uma padronização no que diz respeito ao compartilhamento de conhecimentos e informações [7].

Vale ressaltar que esses foram apenas alguns exemplos, e que além dessas, existem linguagens destinadas exclusivamente à comunicação de agentes.

2.4 . Aplicações de SMA

Sistemas Multi-Agentes vêm sendo utilizados em diversas áreas. De um modo geral, SMA podem (ou devem) ser aplicados a problemas com as seguintes características: (Engenharia de Sistemas Multi-Agentes):

- O problema não pode ser descrito a priori, por conseqüência de perturbações em tempo real no ambiente e/ou processos de negócio de natureza dinâmica;
- Interações complexas, necessidade de negociação, compartilhamento de informação e coordenação;
- A distribuição dos dados do domínio e das capacidades de resolução de responsabilidades é intrínseca ao problema;

- Necessidade de manter a autonomia de subpartes, sem a perda da estrutura organizacional.

Algumas áreas que vêm utilizando uma abordagem de SMA no desenvolvimento de seus sistemas são descritas a seguir [4]:

- **Controle de Trafego Aéreo:** É uma aplicação reconhecidamente complexa. A aplicação de sistemas multi-agentes mais comentada é um sistema para controle de tráfego aéreo que está sendo testado para uso no aeroporto de Sydney na Austrália.
- **Indústria:** Modelagem de uma linha de produção através de agentes cooperantes.
- **Gerência de Negócios:** Agentes podem ser empregados em sistemas de *workflow* (i.e., sistemas que visam garantir que as tarefas necessárias serão feitas pelas pessoas certas nas horas certas).
- **Interação Humano-Computador:** Esta área utiliza agentes *credíveis - que são* agentes capazes de provocar a crença no fato de que eles s ao seres reais, como personagens de desenhos animados, característica que pode ser obtida fazendo os agentes exibirem algo parecido com personalidades - e agentes improvisacionais para melhorar a qualidade das interfaces de software.
- **Ambientes de Aprendizagem:** Modelagem dos alunos em tutores inteligentes.
- **Entretenimento:** Agentes podem ser utilizados para aumentar o realismo de personagens de jogos e sistemas para a industria de entretenimento em geral.
- **Aplicações Distribuídas:** Para domínios naturalmente distribuídos, a metáfora de *agente* é bastante adequada; telecomunicações, transportes e sistemas para a área de saúde são alguns exemplos.
- **Aplicações para a Internet:** Esta é uma área muito comum para aplicações multi-agentes. Existem varias classes de sistemas que se enquadram neste item, como sistemas para gerência de informação (em particular agentes conhecido como *personal digital assistants* ou assistentes digitais pessoais, que auxiliam na gerência da sobrecarga de informação recebida pelas pessoas via Internet), sistemas de comércio eletrônico e sistemas para busca de informações na Internet.

- **Simulação Social:** O objetivo destas simulações multi-agentes é auxiliar cientistas sociais em seus estudos. Para isto é preciso representar aspectos cognitivos e sociais de comunidades de pessoas, vindo ao encontro dos objetivos dos sistemas multi-agentes. Este tipo de simulação pode auxiliar bastante na compreensão de questões importantes para as ciências sociais.

Outras referências para trabalhos nestas e em outras áreas de aplicação podem ser encontradas em Wooldridge (citado por [4]).

Capítulo 3

Perspectivas do Desenvolvimento de SMA

É comum em qualquer domínio da alta tecnologia que os sistemas disponíveis no mercado sejam sustentados - em termos de concepções e atualizações - pela pesquisa acadêmica. Assim, baseadas no grau de interesse do universo acadêmico na área de SMA, algumas perspectivas podem ser traçadas, a partir do quadro existente nos dias de hoje, no que diz respeito ao futuro do desenvolvimento de sistemas multi-agentes.

A AgentLink (http://www.agentlink.org) , uma rede de excelência para computação baseada em agentes, apresenta em [1], quatro fases distintas no desenvolvimento de SMA, indicativas da tendência majoritária de aplicações baseadas em agentes ao longo do tempo. Os marcos que distinguem as fases são definidos ao longo de cinco dimensões:

- O grau de participação de agentes que compartilham conhecimento sobre um domínio comum e objetivos comuns;
- O grau de participação de agentes que são projetados por uma mesma equipe ou por equipes diferentes;
- A natureza das linguagens de comunicação e protocolos de interação utilizados por agentes que participam de um SMA, que são classificados num intervalo que vai de linguagens *ad hoc*, passa por linguagens fixamente padronizadas e chega às linguagens emergentes;
- A escalabilidade, isto é, quantos agentes participando ativamente são suportados pelo sistema, quantos usuários e a complexidade do sistema como um todo, e;
- A metodologia de projeto (se existir) usada pelos projetistas do sistema.

As quatro fases são descritas a seguir, e ilustradas pela figura 3.1.

3.1 . Fase 1 – Desenvolvimento Atual (2000-2003)

Os SMA existentes atualmente são tipicamente projetados por uma única equipe para um determinado ambiente de uma corporação, onde os agentes desenvolvidos compartilham um alto nível de objetivos comuns em um único domínio. Tais sistemas podem ser caracterizados como "*Closed Systems*".

As linguagens de comunicação e protocolos de interação são tipicamente protocolos *in-house*[1] e são definidos de acordo com a prioridade que a equipe de projetistas atribui a cada interação entre os agentes.

Os sistemas são, geralmente, escaláveis apenas quando sob condições controladas, ou de simulação. As abordagens do projeto tendem a ser *ad-hoc*, inspirada no paradigma de agentes, o invés da utilização de metodologias específicas. Um exemplo de sistema desenvolvido nessa fase é o sistema de gerenciamento de redes.

É esperado que, devido às circunstâncias atuais, ainda venha a existir uma demanda substancial por SMA do tipo *Closed*, pelo fato de esses sistemas serem mais seguros em relação aos sistemas abertos (*Open Systems*). Assim, mesmo com a evolução e total mudança da natureza dos SMA, a importância dos *Closed Systems* não deve ser subestimada.

3.2 . Fase 2 – Desenvolvimento a Curto Prazo (2004-2005)

Nesta próxima fase do desenvolvimento de SMA, os sistemas tenderão a ser desenvolvidos para ambientes corporativos mistos, fazendo com que os agentes envolvidos tenham poucos objetivos em comum, embora as intenções entre eles ainda se refiram a um domínio comum. No entanto, a despeito de tal diversidade, todos os agentes serão projetados pela mesma equipe durante todo o projeto e estes compartilharão do mesmo conhecimento sobre o domínio.

Linguagens de comunicação padronizadas serão utilizadas entre os agentes, porém os protocolos de interação permanecerão não-padronizados. Tais sistemas estarão aptos a manipular um grande número de agentes em ambientes pré-determinados, e serão usadas para o desenvolvimento desses sistemas metodologias do tipo *top-down*, ou *middle-out*, que suportem aplicações baseadas em arquiteturas orientadas a serviço.

[1] Protocolos desenvolvidos especificamente para tal propósito, isto é, protocolos particulares.

Um exemplo de sistemas dessa fase seria um sistema que permitisse coordenação automática de cronogramas entre diversos departamentos de uma mesma empresa, ou sistemas de operações centralizadas em uma rede.

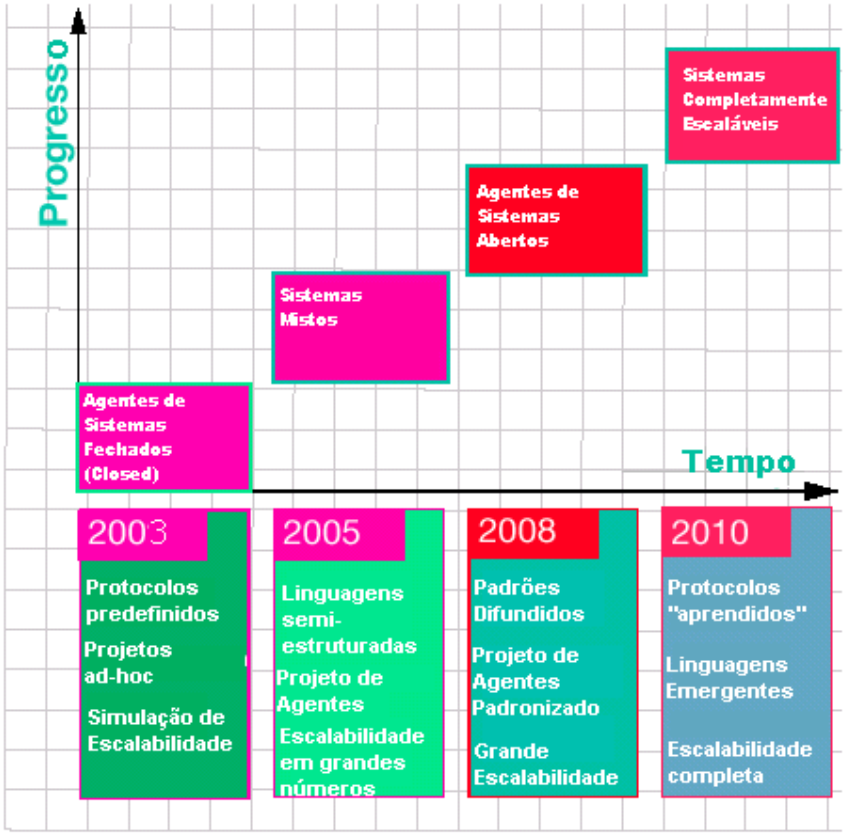

Figura 3.1 – Fases de Desenvolvimento dos SMA ao longo do tempo

3.3 . Fase 3 - Desenvolvimento a Médio Prazo (2006-2008)

Na terceira fase, os SMA permitirão a participação de agentes heterogêneos, desenhados por diferentes equipes de projetistas. Cada agente estará apto a participar de tais sistemas, provendo comportamentos observáveis de acordo com requisitos e padrões estabelecidos. No entanto, tais sistemas – do tipo *Open Systems* – serão utilizados tipicamente em domínios particulares de aplicações.

As linguagens e protocolos utilizados nestes sistemas serão compatíveis entre si e padronizados, podendo até ser importados de bibliotecas de protocolos alternativos, de domínio público.

Os sistemas terão uma grande escalabilidade, suportando um grande número de agentes, embora tipicamente somente agentes de um mesmo domínio. A terceira fase compreenderá o desenvolvimento de agentes denominados "pontes" (*bridge agents*), que estarão aptos a traduzir mensagens de diferentes domínios.

Nesta fase, o desenvolvimento de SMA utilizará metodologias de projetos padronizadas e específicas para agentes e tipos de sistemas de agentes.

Um exemplo de sistema desenvolvido nessa fase seria o sistema de obtenção de recursos eletrônicos, permitindo a participação de qualquer provedor, e não somente de grupos fechados de usuários.

3.4 . Fase 4 – Desenvolvimento a Longo Prazo (2009 em diante)

A quarta fase dessa projeção prevê o desenvolvimento de SMA abertos (*Open Systems*) envolvendo múltiplos domínios de aplicação e envolvendo agentes heterogêneos desenvolvidos por diferentes equipes de projetistas. Os agentes que não possuírem a capacidade necessária para participar de tais sistemas (ditos *seeking agents*) poderão "aprender" em curso, isto é, adquirir o comportamento apropriado para que essa participação aconteça, sem a necessidade de ser modificado antes de sua entrada no sistema.

Embora existam linguagens de comunicação e protocolos de interação disponíveis para o uso, nesta fase os sistemas permitirão o surgimento e utilização de evolucionários meios de interação, ao invés de um protocolo imposto. Certamente, tais protocolos e linguagens evolucionários serão meros refinamentos de padrões previamente desenvolvidos, porém serão adaptados a seus contextos particulares de uso.

Nesta fase os sistemas serão completamente escaláveis, isto é, não serão restritos a limites arbitrários (agentes, usuários, complexidade, etc). Assim como na fase anterior, será notado o uso de rigorosas metodologias de projeto, específicas para agentes.

Os sistemas desenvolvidos nessa fase suportarão completamente qualquer ambiente computacional.

Capítulo 4

Considerações Finais

Muito ainda há a se estudar e se definir a respeito de Sistemas Multi-Agentes. Devido a uma ampla divulgação obtida pelos SMA no final da década de 90, existe uma vasta literatura disponível sobre o assunto, para aqueles interessados em aprofundar conhecimentos nessa área.

Tendo em mente os aspectos mencionados neste trabalho, algumas considerações devem ser feitas, no desenvolvimento de SMA:

- Deve-se considerar aspectos relacionados com o domínio do problema, tais como:
    a) Número de agentes necessários ao sistema;
    b) A relação temporal do sistema, isto é, se é, por exemplo, um sistema de tempo real;
    c) A possibilidade de novos objetivos serem acrescentados dinamicamente ao sistema;
    d) Custo da comunicação;
    e) Custo de falhas;
    f) Que tipo de envolvimento o usuário terá no sistema, etc.

- Deve-se levar em conta aspectos arquiteturais, tais como Robustez do sistema, protocolos de comunicação, etc, e aspectos ambientais, tais como tipo de ambiente, grau de conhecimento dos agentes sobre as mudanças ocorridas no ambiente, etc.
- Problemas inerentes ao desenvolvimento de SMA, como os descritos a seguir, devem ser levados em conta:
    a) **Formulação do problema** - como formular, descrever, apagar e nomear problemas e sintetizar resultados entre os grupos de agentes inteligentes?

b) **Comunicação e interação** - como alcançar a comunicação e interação entre agentes? Quais as linguagens e protocolos de comunicação serão usados em SMA? Quando e o quê comunicar?
c) **Tomada de decisões** - como assegurar que os agentes ajam corretamente na tomada de decisões ou na execução de ações, de tal modo a evitar interações prejudiciais devido às decisões particulares que podem afetar decisões globais?
d) **Planos, ações e conhecimento** - como permitir que agentes individuais representem e argumentem sobre os planos, ações e o conhecimento de outros agentes, com o propósito de se coordenarem entre si? Como argumentar sobre o estado dos seus processos coordenados (por exemplo: iniciação, execução, e finalização)?
e) **Perspectivas e intenções** - como reconhecer e reconciliar perspectivas opostas e intenções em conflito entre agentes que tentam coordenar suas ações?
f) **Projeto e desenvolvimento de sistemas distribuídos** - como projetar e desenvolver sistemas distribuídos práticos? Como projetar plataformas tecnológicas e metodologias de desenvolvimento para SMA?

Desse modo, o campo de SMA mostra-se propenso a novas explorações, seja em níveis mais ou menos profundos, oferecendo uma enorme gama de tema para trabalhos futuros, nos diversos aspectos envolvidos nessa área.

Capítulo 5

Referências Bibliográficas

[1] LUCK, Michael; MCBURNEY, Peter; PREIST, Chris, Agent Technology: Enabling Next Generation Computing - A Roadmap for Agent-Based Computing, version 1.0, Southampton, EUA, JAN 2003. Disponível em: www.agentlink.org/roadmap/roadmap.pdf

[2] JUCHEM, M.; BASTOS, R. M.*, Engenharia de Sistemas Multiagentes: Uma Investigação sobre o Estado da Arte*, Campus Global – FACIN – PUCRS, Porto Alegre – RS , abr 2001. Disponível em: www.inf.pucrs.br/tr/tr014.pdf

[3] TAVARES, J. M. de S. B.; GOTTGTROY, M. de P. B.; Comunicação entre Agentes Inteligentes, monografia de graduação, UFRN, Natal – RN, Jul 1999. Disponível em: www.dimap.ufrn.br/~sagri/RelatorioCurso.PDF

[4] BORDINI, R. H., VIEIRA, R. e MOREIRA, Á. F. Fundamentos de sistemas multiagentes. In Ferreira, C. E., ed., *Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001), Volume 2, XX Jornada de Atualização em Informática (JAI), Fortaleza-CE, Brasil, 30 de julho - 3 de agosto*. Sociedade Brasileira de Computação. capitulo 1, 3-41.

[5] Projeto AgP - Ambiente de Aprendizagem Baseado em Portfólios usando Arquitetura Multi-Agentes -http://www.ppgia.pucpr.br/projetos/agp/pesquisa.html.

[6] SILVA, A. ; DELGADO, J.; Agentes de Software: Conceitos e Tecnologias, Atas do 3º Encontro Nacional do Colégio de Engenharia Eletrotécnica. Ordem dos Engenheiros. Matosinhos – Portugal. Jun 1997

[7] PORTO, P. R. P.; Engenharia de Software Baseada em Agentes, *II Oficina de Inteligência Artificial – II OIA*, Pelotas - RS, Jun 1998. Disponível em: http://gpia.ucpel.tche.br/iioia/9ricardo.doc