



FACULDADE DE INFORMÁTICA
PUCRS – Brazil
<http://www.inf.pucrs.br>

Arquiteturas de Agentes

Murilo Juchem e Ricardo Melo Bastos

TECHNICAL REPORT SERIES

Number 013
April, 2001

Contact:

juchem@inf.pucrs.br

<http://www.inf.pucrs.br/~juchem>

bastos@inf.pucrs.br

<http://www.inf.pucrs.br/~bastos>

Murilo Juchem is a M.Sc. student of PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul – Brazil. He develops research in Multi-agent systems design techniques at the information system group of the PPGCC. He receives a federal graduate research grant from CNPq (Brazil) to support his research.

Ricardo Melo Bastos is a professor in the FACIN-PUCRS since 1986 and Ph.D. in computer science since 1998 (PPGC-UFRGS). He is a researcher in software engineering and multi-agent systems at the information system group of the PPGCC.

Copyright © Faculdade de Informática – PUCRS
Published by the Campus Global – FACIN – PUCRS
Av. Ipiranga, 6681
90619-900 Porto Alegre – RS – Brazil

Arquiteturas de Agentes

Relatório Técnico 013/2001

Murilo Juchem*
Ricardo Melo Bastos†

1 Introdução

Este trabalho tem como objetivo apresentar a revisão bibliográfica sobre arquiteturas de agentes, sendo resultado da adaptação dos trabalhos realizados na disciplina Trabalho Individual I, constante no Programa de Pós-Graduação em Ciência da Computação (PPGCC), da PUCRS, orientada pelo Prof. Dr. Ricardo Melo Bastos, tendo como foco a sub-área de Sistemas Multiagentes (MAS), dentro da área de Sistemas de Informação e com contribuições na linha de Inteligência Artificial (IA).

Sistemas Multiagentes é o nome dado à sub-área da Inteligência Artificial Distribuída que estuda o comportamento de um conjunto de agentes autônomos objetivando a solução de um problema que está além de suas capacidades individuais [JEN96]. Segundo [WOO95], um agente é um sistema computacional que está situado em algum ambiente, e que é capaz de ações autônomas neste ambiente objetivando alcançar seus objetivos de projeto.

O trabalho aqui proposto objetiva realizar uma investigação sobre algumas arquiteturas de agentes existentes para sistemas empresariais. Para tanto, realizaremos um estudo sobre os definições e conceitos referentes ao paradigma de multiagentes sendo estes examinados ao nível do indivíduo, ou seja, do agente.

Posteriormente, serão examinados alguns dos modelos de arquitetura de agentes já concebidas por pesquisadores da área com o objetivo de identificar suas propriedades e

*juchem@inf.pucrs.br

†bastos@inf.pucrs.br

características, através do estudo dos sistemas desenvolvidos em particular no domínio das aplicações que buscam solucionar problemas em organizações empresariais.

2 Agentes

2.1 Inteligência Artificial Distribuída (IAD)

[BON88] define Inteligência Artificial Distribuída como a sub-área da Inteligência Artificial que estuda a concorrência em computações de Inteligência Artificial. [OLI96] observa que esta definição pode levar à uma errônea compreensão do conceito de IAD como uma simples união entre IA e Sistemas Distribuídos, pois IAD é um conceito mais amplo.

Uma conceitualização de IAD mais compatível com a idéia de agente é dada por [JEN96] que diz que o objeto de investigação da IAD são os modelos de conhecimento, e as técnicas de comunicação e raciocínio necessárias para que agentes computacionais convivam em sociedades compostas de computadores e pessoas. Jennings ainda divide a IAD em duas áreas de pesquisa principais:

- *Resolução Distribuída de Problemas (DPS)* - que divide a solução de um problema em particular entre um número de módulos que cooperam compartilhado conhecimento sobre o problema e sobre as soluções envolvidas.
- *Sistemas Multiagente (MAS)* - estuda o comportamento de um conjunto de agentes autônomos (possivelmente preexistentes) cujo objetivo comum é a solução de um dado problema.

Segundo [OLI96], nos MAS a atenção do projetista não está necessariamente voltada para um problema específico. Esta abordagem consiste na coordenação do comportamento inteligente de um conjunto de agentes autônomos, que podem ter sido criados antes do surgimento de um problema em particular. Demazeau Apud [OLI96] define a abordagem MAS como "o estudo do comportamento de um agente autônomo, em um mundo multiagente".

É possível perceber que estas duas áreas diferem na forma de construção da solução do problema. A Resolução Distribuída de Problemas adota uma visão top-down dividindo

o problema em partes que corresponderão a módulos computacionais, sendo que "o processo de coordenação das ações é definido em tempo de projeto" [OLI96]. Os Sistemas Multiagente são compostos por entidades computacionais, denominadas agentes, com capacidades e objetivos individuais que, uma vez agrupados em sociedade, trabalhem juntos visando atingir o objetivo do sistema, sendo que "os agentes devem raciocinar a respeito das ações, mas também sobre o processo de coordenação em si" [OLI96].

2.2 Sistemas Multiagentes

Um sistema multiagente é formado por um conjunto de agentes que colaboram entre si para cumprir com a responsabilidade no sistema [AMA97].

Na visão de Bond e Gasser [BON88], um sistema multiagente pressupõe coordenação entre um conjunto existente de agentes autônomos e inteligentes. Fundamentalmente, está envolvida a busca por uma funcionalidade neste sistema que permita que estes agentes possam coordenar seus conhecimentos, objetivos, habilidades e planos individuais de uma forma conjunta, em favor da execução de uma ação ou da resolução de algum problema. Tanto para a ação, quanto para a resolução do problema, em sistemas multiagentes faz-se necessária a cooperação entre os agentes.

Jennings [JEN96] mostra que as características dos Sistemas Multiagente têm vantagens significativas sobre um solucionador de problemas monolítico, dentre elas:

- maior rapidez na resolução de problemas através do aproveitamento do paralelismo;
- diminuição da comunicação por transmitir somente soluções parciais em alto nível para outros agentes ao invés de dados brutos para um lugar central;
- mais flexibilidade por ter agentes de diferentes habilidades dinamicamente agrupados para resolver problemas; e
- aumento da segurança por permitir agentes assumirem responsabilidades de agentes que falham.

2.3 Agentes

O conceito de agente (vide figura 1) é de vital importância para a compreensão do que seja um Sistema Multiagente. Vejamos algumas das definições dadas para este termo.

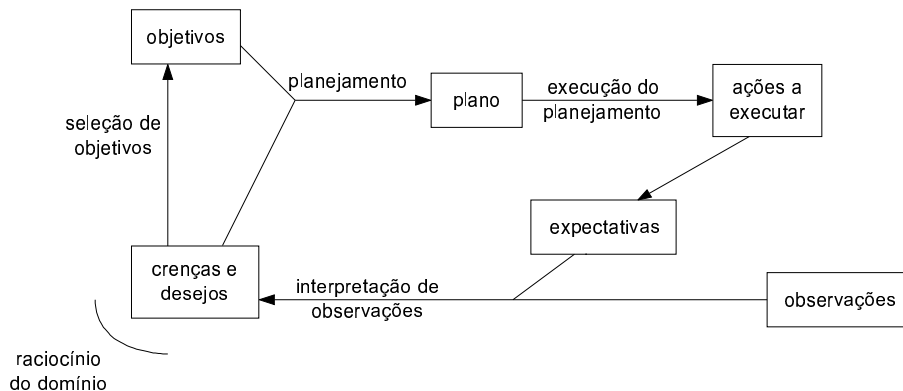


Figura 1: Um modelo de agente

Corrêa [COR94] considera que um agente pode ser definido como uma entidade que funciona contínua e autonomamente em um ambiente no qual existem outros processos e agentes.

Para Amandi [AMA97], um agente é uma entidade computacional com um comportamento autônomo que lhe permite tomar suas decisões para agir levando em consideração as mudanças acontecidas no ambiente em que atua e o seu desejo de alcançar seus objetivos.

[SHO93] afirma que um agente é uma entidade à qual se atribuem estados, denominados de estados mentais. Os estados mentais usuais são: crenças, decisões, capacidades, objetivos, intenções, compromissos e expectativas, conceitos análogos ou similares aos humanos. Com base nesta perspectiva, o que faz qualquer componente de hardware ou software ser considerado um agente é precisamente o fato dele poder ser analisado e controlado em termos destes estados mentais.

[D'AM95] considera que, do ponto de vista prático, são agentes: (a) robôs que atuam em um ambiente, interagindo com outros robôs ou com humanos em língua natural e utilizando sensores para captar informações do meio, tais como controle de motor e restrições de tempo; (b) um sistema que interage com outros sistemas ou com o ser humano.

Smith Apud [JEN96b] e outros definem um agente como "uma entidade de software persistente dedicada a um propósito específico". Selker Apud [JEN96b] toma um agente como "um programa de computador que simula um relacionamento humano fazendo algo que outra pessoa poderia fazer para você". Janca Apud [JEN96b] define que um agente é "uma entidade de software para qual tarefas podem ser delegadas".

Segundo [WOO99], não há definição universalmente aceita do termo agente, mas há

um consenso geral de que autonomia é a idéia central da noção de agência.

Para [RUS95], autônomo vem a ser algo como "não submetido a controle imediato de um humano", como no termo "veículo autônomo". Na visão de Moffat [MOF95], o significado de autonomia para um agente é a capacidade para atingir seus próprios objetivos.

[WOO99] explica que parte da dificuldade em se definir o termo agente surge do fato de que para diferentes domínios de aplicação os atributos associados ao conceito de agência assumem diferentes níveis de importância. Assim, da mesma forma que para alguns domínios de aplicação a capacidade de aprendizado a partir das experiências é de suma importância, para outras, aprender pode constituir uma capacidade não só pouco importante, mas indesejável.

Todavia, apesar de todas as divergências sobre a definição do que seja um agente, alguma forma de definição é necessária, caso contrário há risco de que o termo perca o significado dentro do escopo no qual está sendo abordado. A definição apresentada aqui, e que será utilizada neste trabalho, é uma adaptação feita por [WOO99] da apresentada por [WOO95]:

“um agente é um sistema computacional que está situado em algum ambiente, e que é capaz de ações autônomas neste ambiente visando atingir seus objetivos propostos”.

[WOO95] visualiza um agente como sendo uma entidade com capacidade de resolução de problemas encapsulada. Inserido nesta visão, define o agente como tendo as seguintes propriedades:

- *autonomia* - executam a maior parte de suas ações sem interferência direta de agentes humanos ou de outros agentes computacionais, possuindo controle total sobre suas ações e estado interno;
- *habilidade social* - por impossibilidade de resolução de certos problemas ou por outro tipo de conveniência, interagem com outros agentes (humanos ou computacionais), para completarem a resolução de seus problemas, ou ainda para auxiliarem outros agentes. Disto surge a necessidade de que os agentes tenha capacidade para comunicar seus requisitos aos outros e um mecanismo decisório interno que defina quando e quais interações são apropriadas;

- *reatividade* - percebem e reagem à alterações no ambientes em que estiverem inseridos.
- *pró-atividade* - agentes, do tipo deliberativo, além de atuar em resposta às alterações ocorridas em seu ambiente, apresentam um comportamento orientado a objetivos, tomando iniciativas quando julgarem apropriado.

Autonomia

Entende-se autonomia como a capacidade do agente executar as suas atividades sem a necessidade de intervenção humana. Para ter autonomia, o agente deve ter um certo grau de inteligência, capacitando-o a sobreviver em um ambiente dinâmico e por vezes não benigno [COR94].

Segundo [RUS95], para definir um agente racional ideal é necessário considerar seu "conhecimento embutido". No caso das ações do agente basearem-se completamente no conhecimento embutido, ao ponto que este não considere suas percepções, podemos dizer que o agente tem falta de autonomia. O comportamento de um agente pode ser baseado tanto em sua própria experiência¹, quanto em seu conhecimento embutido usado na construção do agente para um ambiente em particular. Então, assim como a evolução provê os animais de conhecimentos embutidos que lhes permite aprender por si próprios, seria lógico fornecer a um agente artificial algum conhecimento inicial e capacidade de aprendizado.

Assim, podemos concluir que o grau de autonomia de um agente está diretamente relacionada ao nível de independência que um agente tem em relação àquilo que lhe é nato (ou, "instintivo").

Perspectivas de Análise

Jennings [JEN96] propõe um framework que fornece uma estrutura para analisar e classificar a maior parte das atividades de pesquisa em MAS, do qual podemos citar duas perspectivas:

- *perspectiva do agente* - enfoca elementos que caracterizam o agente envolvido em um MAS. São eles: categorias de agente, estrutura e manutenção do conhecimento,

¹"Um sistema é autônomo ao nível que seu comportamento é determinado por sua própria experiência" [RUS95]

habilidades de raciocínio, habilidades de adaptação e aprendizado, e arquiteturas de agente.

- *perspectiva de grupo* - reúne aspectos de grupo, tais como: organização, coordenação, cooperação, negociação, comportamento coerente, planejamento, comunicação e interação.

Perspectiva do Agente

Uma classificação de MAS quanto sua heterogeneidade é apresentada por Jennings [JEN96]:

- *baixa heterogeneidade* - os agentes são idênticos ou diferem apenas pelos recursos disponíveis a eles.
- *média heterogeneidade* - os agentes diferem também pelos métodos de resolução de problemas.
- *alta heterogeneidade* - os agentes somente compartilham uma mesma linguagem de interação, sendo que suas outras características podem ser completamente diferentes.

Quanto a um agente envolvido em um MAS, Jennings também apresenta alguns assuntos a serem abordados: categorias de agente, estruturas e manutenção de conhecimento, habilidades de raciocínio, habilidades de aprendizado, e arquitetura de agente (que será examinado no item 4 - Arquiteturas de Agente).

Categorias de agente

Segundo Jennings [JEN96], para que possa agir de maneira autônoma, agentes podem ter várias habilidades: percepção e interpretação de mensagens, raciocínio baseado em crenças, tomada de decisão, planejamento, e habilidade para executar planos incluindo passagem de mensagens. Jennings categoriza os agentes quanto ao nível de capacidade de resolução de problemas:

- *Reativos* - reagem a alterações no ambiente ou a mensagens de outros agentes. Não têm capacidade de raciocínio sobre suas intenções, reagindo tão somente sobre regras e planos estereotipados. Suas ações podem ser: atualizar a base de fatos e enviar mensagens para outros agentes ou para o ambiente.

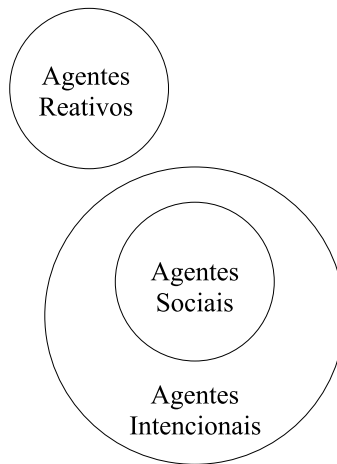


Figura 2: Categorias de agentes

- *Intencionais*² - tem a habilidade de raciocínio sobre suas intenções e crenças, e criar e executar planos de ações. São considerados como sistemas de planejamento selecionando objetivos - de acordo com suas motivações - e raciocinar sobre estes - detecção e resolução de conflitos e coincidências de objetivos, selecionar e criar planos (agendamento de ações), detecção de conflitos entre planos (alocação de recursos), e, se necessário, executar e revisar planos.
- *Sociais* - agentes intencionais são considerados sociais quando possui modelos de outros agentes, sobre os quais raciocina para tomar decisões e criar planos.

Estruturas e Manutenção do Conhecimento

As estruturas de conhecimento utilizadas na construção de agentes podem ser de diferentes tipos: crenças (ou fatos), objetivos (ou intenções), preferências, motivações, desejos, etc. É usual que a aquisição de novas crenças por parte de um agente seja resultado de mensagens enviadas por outros agentes ou pelo ambiente no qual está inserido.

Inserido em um MAS, um agente possui somente uma visão parcial da situação e do problema a ser resolvido. A informação processada pelo agente é muitas vezes parcial, incerta ou até mesmo errônea. Assim sendo, o agente necessita de mecanismos de manutenção da consistência de sua base de conhecimento ou para revisar suas crenças quando adquire algum conhecimento que contraditório.

²Intencional também é apresentado com outros termos, tais como: Racional, Cognitivo e Deliberativo.

Huhns e Bridgeland Apud [JEN96] define níveis de consistência das crenças entre um grupo de agentes computacionais (inconsistência, consistência local, consistência local-e-compartilhada, consistência global) e propõe um algoritmo de manutenção da verdade que garante consistência local para cada agente e consistência global para dados compartilhados entre agentes.

Habilidades de Raciocínio

Agentes precisam raciocinar sobre diversos aspectos da realidade: (i) tratam com obrigações, permissões e proibições (lógica deôntica); (ii) tempo, que é um fator que a maior parte dos sistemas têm que levar em consideração (lógica temporal); (iii) um agente pode precisar explorar várias hipóteses antes de tomar uma decisão (raciocínio hipotético).

Moore Apud [JEN96] enfatiza a relação entre conhecimento e ação. Para ele, conhecimento é necessário para desempenhar ações, e a aquisição de novos conhecimentos são o resultado destas ações.

Além de raciocinar sobre suas próprias crenças, desejos e intenções, agentes têm que raciocinar sobre o conhecimento e o comportamento de outros agentes. Halpern Apud [JEN96] afirma que a maioria dos modelos de conhecimento e crenças são baseados nos modelos de possíveis mundos, nos quais um agente é dito conhecedor de um fato e se este fato é verdadeiro em todos mundos que ele pensa são possíveis. Alguns dos problemas encontrados são: (i) onisciência lógica - agentes podem derivar todas fórmulas válidas e suas conseqüências lógicas; e (ii) conhecimento mútuo - não só todos sabem que um fato é verdadeiro, mas eles conhecem todos que sabem, e assim por diante.

Shoham Apud [JEN96] investigou a relação entre conhecimento e crença. Contrariamente ao bem conhecido slogan "conhecimento é crença verdadeira", Shoham propõe que "crença deve ser visto como conhecimento que pode ser anulado".

Jennings [JEN96] afirma que para raciocinar (e alterar) crenças e ações de outros agentes, agentes inteligentes precisam raciocinar sobre (e alterar) o comportamento de outros agentes. Para isso, agentes devem raciocinar sobre planos de outros agentes para reconhecer suas ações potenciais.

Gmytrasiewicz, Durfee, Zlotkin e Rosenschein Apud [JEN96] têm estudado o impacto de mentiras no raciocínio. Isto vem a ser a transmissão de informações incompletas ou até mesmo erradas para outros agentes com o intento de obter alguma vantagem sobre outros para atingir algum objetivo.

Habilidades de Adaptação e Aprendizado

Em um ambiente dinâmico podem ocorrer imprevistos. Quando uma alteração no ambiente ocorre, o agente que nele está inserido pode necessitar adaptar-se à esta nova realidade, ou até mesmo à mudanças de comportamento de outros agentes aos quais ele se relaciona direta ou indiretamente.

Weiss Apud [JEN96] aborda a questão onde agentes aprendem coletivamente para coordenar suas ações na direção uma solução para um problema em comum. Para Jennings [JEN96], duas importantes restrições têm sido consideradas: restrição de incompatibilidade - na qual ações podem ser mutuamente excludentes; e (ii) restrição de informação local - na qual cada agente conhece uma parte de seu ambiente.

Sian Apud [JEN96] apresenta um MAS composto de sistemas especialistas com conhecimento diversificado, no qual um módulo de raciocínio indutivo permite a cada agente adquirir novos conhecimentos através da interação com outros sistemas especialistas.

3 Arquiteturas de Agente

Neste capítulo, serão apresentadas as classificações arquiteturais de um agente. Também serão apresentadas algumas das arquiteturas existentes, procurando destacar suas principais características.

3.1 Tipos de Arquiteturas de Agente

Para definir arquiteturas internas dos agente é necessário, primeiramente, saber qual o tipo de agente do qual se está tratando. Neste caso, um agente pode ser classificado como cognitivo ou reativo. [OLI96] define que, no caso de um agente cognitivo, temos uma íntima relação à idéia de agente racional, que tem como característica a capacidade de escolher as ações a executar, dentre as existentes em seu repertório, coerentemente com seus objetivos. Quando um agente sempre escolhe a ação mais coerente com seus objetivos, podemos dizer que possui "racionalidade ideal", o que é computacionalmente inviável, pois requer que o agente tenha um conhecimento total e correto sobre o problema e o ambiente, sendo que a escolha poderia requerer um infinito número de passos de inferência.

Um agente cognitivo é um agente racional que possui alguma representação explícita

de seu conhecimento e objetivos. Um agente reativo não necessariamente é um agente racional: seu comportamento pode ser definido através de um padrão estímulo-resposta (...) Um agente pode ser 'mais cognitivo' do que outro, conforme o grau de racionalidade explícita de seu comportamento. [OLI96]

Arquiteturas Cognitivas ou Deliberativas

[BAS98] define que os agentes cognitivos, que em seu trabalho são chamados agentes deliberativos, possuem uma representação simbólica do mundo, sendo que suas deliberações (também chamadas decisões) são feitas por meio de um processo baseado em raciocínio lógico. Este raciocínio trabalha sobre um conjunto de símbolos que, sendo fisicamente concebíveis, podem ser combinadas formando-se estruturas sobre o qual se pode operar.

Segundo [COR94], o agente passa por um processo de deliberação explícita para a escolha da ação, que é feito através de:

1. uma representação simbólica do mundo³;
2. um plano; ou
3. uma função de utilidade para a ação (vide item 3.2 - Agente Baseado em Utilidade).

[STE96] adota o princípio de que agentes são racionais:

(...) se um agente tem conhecimento de que uma de suas ações levarão a um de seus objetivos, este selecionará aquela ação.

Oliveira classifica as arquiteturas de agentes cognitivos como funcionais ou baseadas em estados mentais.

Arquiteturas Funcionais No caso da arquitetura funcional (vide figura 3), o agente é composto por módulos que representam cada uma das funcionalidades necessárias para sua operação. Demazeau Apud [OLI96] define que um agente cognitivo deve possuir conhecimento, um conjunto de objetivos, e capacidades de percepção, comunicação, decisão e raciocínio.

Outra arquitetura é apresentada por [STE96], sendo composta por três partes principais:

³Entenda-se pelo termo "mundo" o ambiente em que o agente está inserido.

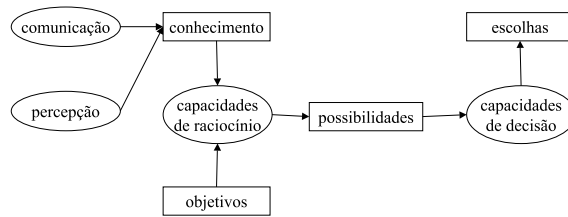


Figura 3: Arquitetura de Demazeau para Agentes Funcionais [OLI96].

- *cabeça* - controle as ações, portanto englobando as capacidades reativas, racionais e cooperativas;
- *comunicador* - implementa as capacidade de comunicação do agente;
- *corpo* - encarregado da execução das ações e observação do ambiente.

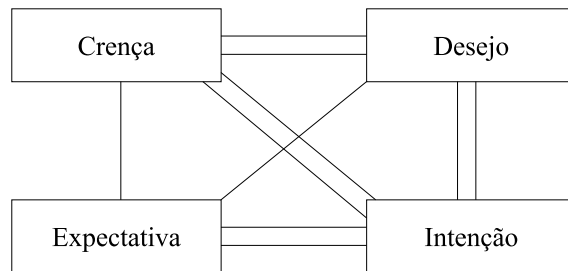


Figura 4: Arquitetura Baseada em Estados Mentais [COR94].

Arquiteturas Baseadas em Estados Mentais Numa abordagem de visão psicológica, as arquiteturas baseadas em estados mentais definem o estado de um agente como um conjunto de componentes mentais, tais como: crenças, capacidades, escolhas e compromissos. Shoham [SHO93] afirma que, para gerenciar sua autonomia, os agentes possuem inteligência que baseia-se, geralmente, nos próprios estados mentais. Em função destes estados mentais, um agente define seu comportamento autônomo, que consiste em decidir a ação seguinte que deve executar para satisfazer algum de seus objetivos, quando e para quem solicitar colaboração, que mudanças no ambiente determinam mudanças nas suas crenças ou objetivos, e assim por diante [AMA97].

Dentro das arquiteturas baseadas em estados mentais situam-se as arquiteturas BDI, que são examinadas a seguir.

Arquiteturas BDI Este tipo de arquitetura considera três estados mentais: Crença, Desejo e Intenção (Belief, Desire and Intention). Segundo [WOO99], estas arquiteturas surgem do processo de decidir, momento a momento, qual ação desempenhar na direção de seus objetivos. Este processo envolve dois processos importantes: decidir quais objetivos queremos atingir, e como iremos atingir estes objetivos.

Basicamente, crenças são a representação do mundo mantida pelo agente. As intenções são desejos eleitos para execução. Alterações percebidas pelo agente podem significar alterações em suas crenças e, possivelmente, mudanças em suas intenções, pois seus desejos podem não ser mais os mesmos para este novo conjunto de crenças.

Por considerar apenas três estados mentais, podemos dizer que a abordagem BDI é reducionista sob o ponto de vista de modelagem do comportamento humano, pois desconsidera estados mentais que representam aspectos emocionais.

Arquiteturas Reativas ou Não-Deliberativas

[COR94] define que, em uma arquitetura reativa, o processo de tomada de decisão de um agente ocorre em tempo real, em resposta a estímulos do ambiente, captados por seus sensores, ou a mensagens enviadas por outro agente. Neste tipo de agente, o mecanismo de controle é, geralmente, implementado por um conjunto de regras evento-ação - chamado por [BAS98] pelo termo estímulo-resposta - ou por máquinas de estados finitos (autômatos finitos).

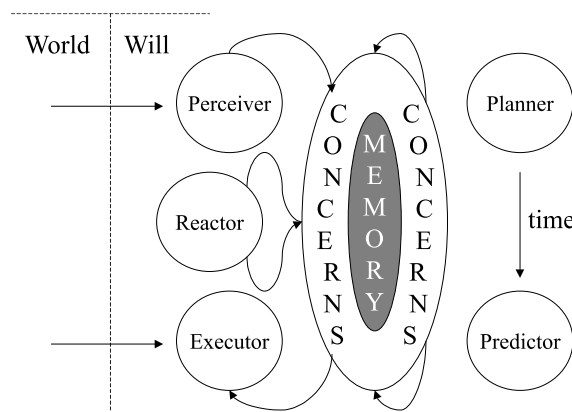


Figura 5: Arquitetura Completa do Agente Will [MOF95].

Segundo [MOF95] um agente é dito "reativo" se não necessariamente planeja tudo, mas pode algumas vezes somente reagir apropriadamente a certos estímulos. Devemos

distinguir esta reatividade forte da reatividade fraca, onde o agente planeja sua resposta, mas faz imediatamente, interrompendo ou parando qualquer outro processo em andamento que seja menos importante.

Segundo [WOO95], uma arquitetura não guarda qualquer representação simbólica do mundo e também não utiliza raciocínio simbólico complexo.

Arquiteturas Híbridas

Segundo [BAS98], arquiteturas híbridas são provenientes das deficiências encontradas nas arquiteturas deliberativas e reativas, reunindo propriedades de ambas. As arquiteturas reativas têm dificuldades para modificar seus planos de ação a partir do momento em que a situação passa a divergir de seus objetivos iniciais. No caso de arquiteturas deliberativas, [COR94] afirma que têm dificuldade de lidar com situações imprevistas que exigem decisões rápidas. Na visão de Bastos, estas arquiteturas (híbridas) devem definir agentes dotados de capacidades reativas, de raciocínio e planejamento, resolvendo as limitações provenientes das abordagens mais "puras".

Segundo [STE96]:

Em um mundo incerto, um agente racional precisa ser também reativo: certas mudanças no ambiente (situações) podem ativar objetivos os quais precisam ser atingidos imediatamente.

3.2 Classificação de Russel & Norvig

O trabalho de Russel e Norvig [RUS95] apresenta uma classificação para arquiteturas de agentes que considera os seguintes tipos de agente: agente reflexivo simples, agente que guarda caminho do mundo, agente baseado em objetivos, agente baseado em utilidades. Esta classificação apresenta um maior detalhamento em relação à dicotomia reativo/deliberativo no que tange à maneira como o agente atua no ambiente no qual está inserido. Os autores apresentam também considerações acerca das implicações do tipo de ambiente no qual o agente será inserido sobre arquitetura do agente. Algoritmos são apresentados para melhor entendimento da forma de funcionamento das arquiteturas.

Agente Reflexivo Simples Nesta arquitetura o agente encontra uma regra cuja condição corresponde à situação atual (definida em percept) e executando a ação associada com esta

regra.

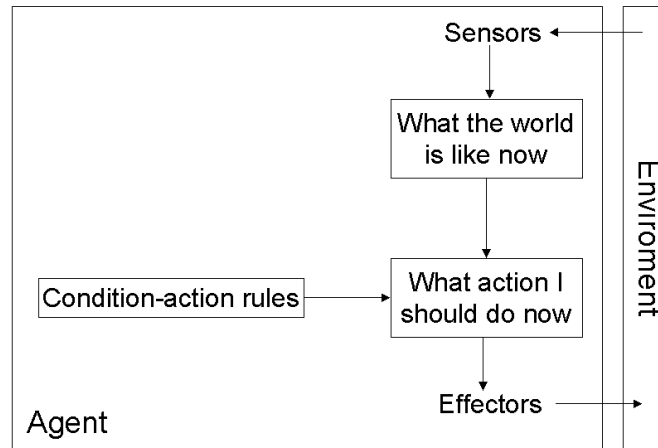


Figura 6: Agente Reflexivo Simples [RUS95].

O algoritmo de tomada de decisão descrito para este tipo de agente prevê a existência das funções `INTERPRET-INPUT` - que gera uma descrição abstrata do estado atual a partir da percepção - e `RULE-MATCH` - que retorna a primeira regra de um conjunto de regras que corresponde à descrição de estado dada. A variável estática `RULES` representa um conjunto de regras condição-ação.

Algorithm 1 Agente Reflexivo Simples [RUS95].

```
function Simple-Reflex-
Agent(percept) returns action
    static: rules
    state <= Interpret-Input(percept)
    rule <= Rule-Match(state, rules)
    action <= Rule-Action[rule]
    return action
```

Este tipo de agente é aplicável em casos em que a decisão correta pode ser tomada baseada apenas na percepção atual.

Agente Reflexivo com Estado Além das características do agente reflexivo simples, este tipo de agente possui um estado interno que é utilizado para o processo de tomada de decisão e que pode ser atualizado.

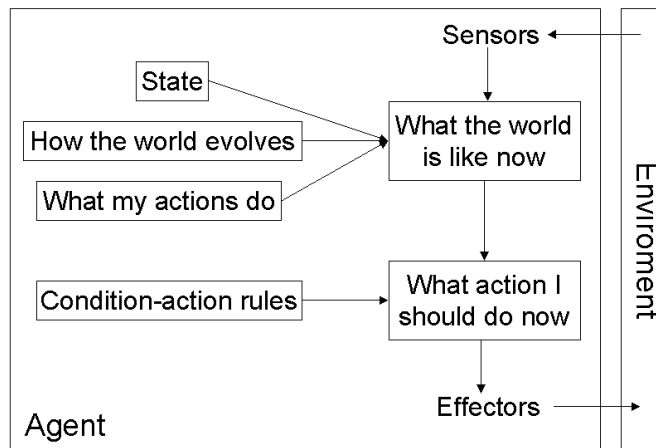


Figura 7: Agente Reflexivo com Estado [RUS95].

O algoritmo descrito para este tipo de agente utiliza, além das funções descritas para o agente reflexivo simples, a função UPDATE-STATE, que é responsável por criar uma nova descrição do estado interno. A variável estática STATE mantém uma descrição do estado atual do mundo.

Algorithm 2 Agente Reflexivo com Estado [RUS95].

```
function Reflex-Agent-With-
State(percept) returns action
    static: state, rules
    state <= Update-State(state, percept)
    rule <= Rule-Match(state, rules)
    action <= Rule-Action[rule]
    state <= Update-State(state, action)
    return action
```

Agente Baseado em Objetivos Em algumas aplicações, conhecer o estado atual do ambiente não é suficiente para saber o que fazer. Isso pode ocorrer quando um proble-

ma requer uma seqüência de passos para atingir sua resolução. Neste caso, a busca e o planejamento são sub-áreas da IA que tratam de achar seqüências de ações para atingir objetivos.

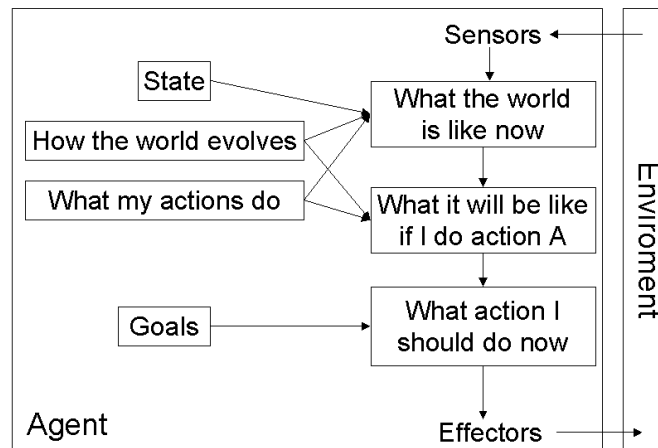


Figura 8: Agente Baseado em Objetivos [RUS95].

Segundo [STE96], um agente direcionado a objetivos representa o conhecimento da maneira como o mundo pode ser transformado através da execução de ações.

Agente Baseado em Utilidade Um comportamento de alta qualidade não é o resultado de objetivos isolados. Considerando o exemplo de um taxi que pretende chegar a um destino, existem muitas seqüências de ações através das quais se atinge este objetivo, mas alguns são mais rápidos, seguros, mais confiáveis, ou baratos que outros. Objetivos definem tão somente distinções entre o que seja estados "felizes" e "infelizes". O grau de "felicidade" do agente é descrito por uma função que faz o mapeamento de um estado em um número real. O fornecimento de uma especificação completa da função de utilidade permite decisões racionais em dois tipos de caso onde objetivos têm problemas: (1) quando houver objetivos conflitantes, somente um deles pode ser atingido (ex.: velocidade e segurança), a função de utilidade especifica a troca apropriada; (2) quando existem muitos objetivos pretendidos pelo agente, nenhum deles pode ser atingido com certeza, a função de utilidade fornece um caminho na qual a probabilidade de sucesso possa ser pesada frente à importância dos objetivos.

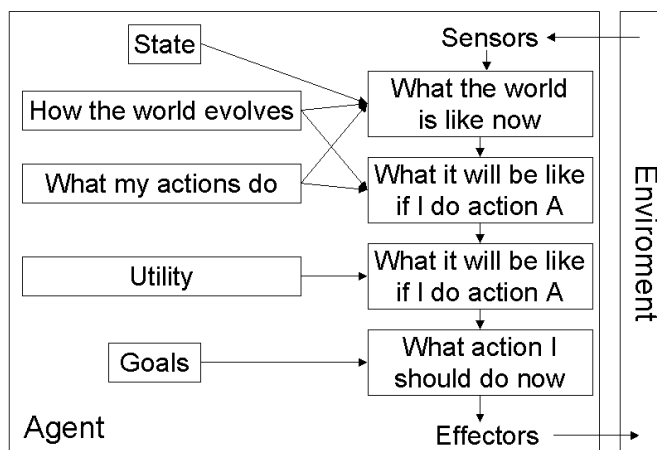


Figura 9: Agente Baseado em Utilidade [RUS95].

Ambiente

Como visto na definição de Wooldridge [WOO95], um agente está situado em algum ambiente (vide figura 10). [D’AM95] entende que o ambiente pode ser visto como tudo o que envolve um agente. Através do ambiente ocorre a dispersão do controle, dos dados e do conhecimento pela comunidade de agentes. Mundo refere-se à descrição completa e instantânea do ambiente em que um agente encontra-se.

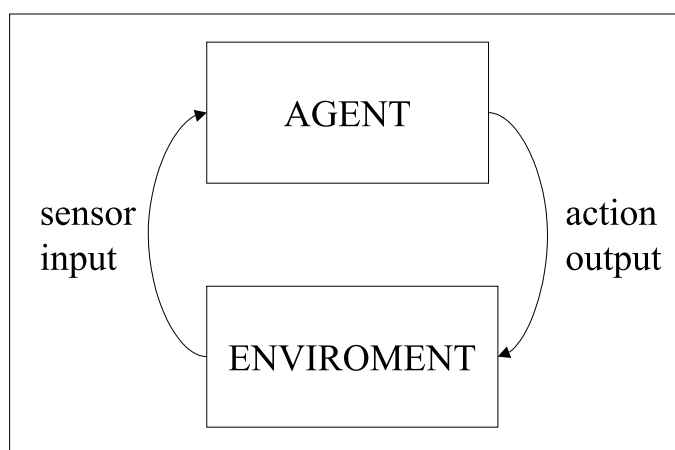


Figura 10: Interação Agente-Ambiente [WOO99].

O tipo de ambiente no qual o agente está situado pode determinar a maneira pela qual o agente deve ver o mundo, ou seja, determinar sobre qual tipo de representação do mundo que o agente deverá trabalhar, bem como sua maneira de atuar e de perceber as alterações no ambiente.

O mundo real é um lugar barulhento, e muito do que acontece nele é irrelevante. Um agente necessita de sensores inteligente para focalizar o que precisa ver, e ignorar o resto. [MOF95]

Propriedades de Ambientes Segundo Russel e Norvig [RUS95], as principais classificações a serem feitas em um ambiente são: acessível ou inacessível; determinístico ou não-determinístico; episódico ou não-episódico; estático ou dinâmico; e discreto ou contínuo.

- *Acessível x Inacessível* - essa característica determina se o aparato sensorial do agente lhe fornece um estado completo do ambiente. Se isto ocorre, o ambiente é considerado acessível; caso contrário, é considerado inacessível. Um ambiente é considerado efetivamente acessível se os sensores detectam todos os aspectos relevantes para a escolha da ação. No caso de um ambiente ser acessível, não é necessário que o agente mantenha qualquer representação interna do mundo.
- *Determinístico x Não-Determinístico* - se o estado do ambiente é determinado somente por seu estado atual e as atuações dos agentes, podemos dizer que este ambiente é determinístico.
- *Episódico x Não-Episódico* - em um ambiente episódico a experiência do agente é dividida em episódios. Cada um deles consistem em percepções e ações dos agentes, e a qualidade de cada ação depende somente do episódio em si.
- *Estático x Dinâmico* - um ambiente é dinâmico se pode mudar enquanto o agente está deliberando; caso contrário é estático. Um ambiente é dito semi-estático quando não muda com a passagem do tempo, apenas com as ações desempenhadas por agentes.
- *Discreto x Contínuo* - é chamado discreto o ambiente que tem um número limitado de percepções e ações distintas e claramente definidas.

3.3 Arquitetura M-DRAP

O M-DRAP é um modelo proposto por [BAS98] para a resolução do problema da alocação dinâmica de recursos concebido através do paradigma multiagente. O modelo do M-DRAP é baseado nas classes propostas pelo CIMOSA, que é um padrão aceito internacionalmente para a construção de aplicações de manufatura.

A arquitetura de referência utilizada nos agentes do M-DRAP (vide figura 11) é composta pelos seguintes componentes:

- *Receptor de Eventos* - tem como finalidade a percepção dos eventos destinados ao agente, identificação de sua natureza e encaminhamento de cada um deles ao módulo habilitado para o seu tratamento. Outro papel deste componente é completar a mensagem original com informações do componente Biblioteca.
- *Biblioteca* - armazena a lista dos agentes que participam de um plano e suas Regras de Procedimento.
- *Planejamento* - realiza o planejamento das atividades a serem realizadas pelo agente. É responsável também por abrir licitações, requisitar a colaboração de outros agentes, e encaminhar propostas de sua competência. Se necessário, este componente negocia com outros agentes com o intuito de encontrar soluções que viabilizem a elaboração de suas próprias propostas.
- *Negociador* - realiza a avaliação das propostas recebidas, a contratação de seus serviços, e o estabelecimento de compromissos quando contratado. Cabe ao Negociador tentar contornar os problemas causados por perturbações, e buscar soluções alternativas que evitem prejuízos na execução dos compromissos afetados por estas.
- *Planejamento* - armazena todas as propostas e compromissos do agente.
- *Controlador* - é responsável pelo acompanhamento da execução de cada atividade, e por comunicar o seu término aos agentes responsáveis pelas atividades seguintes. No caso de detecção de atrasos, informa este ao componente Negociador.

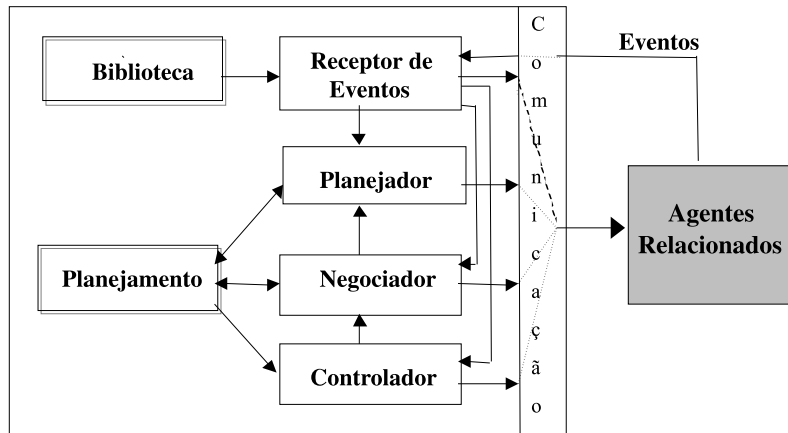


Figura 11: Arquitetura de Referência dos Agentes M-DRAP [BAS98].

3.4 Arquitetura ADEPT

ADEPT [JEN96c] é um projeto cujo objetivo é criar uma infra-estrutura de gerenciamento de processos de negócio utilizando agentes. Este projeto envolve tecnologias de negociação, fornecimento de serviços e agentes autônomos. A figura 12 apresenta a arquitetura dos agentes ADEPT.

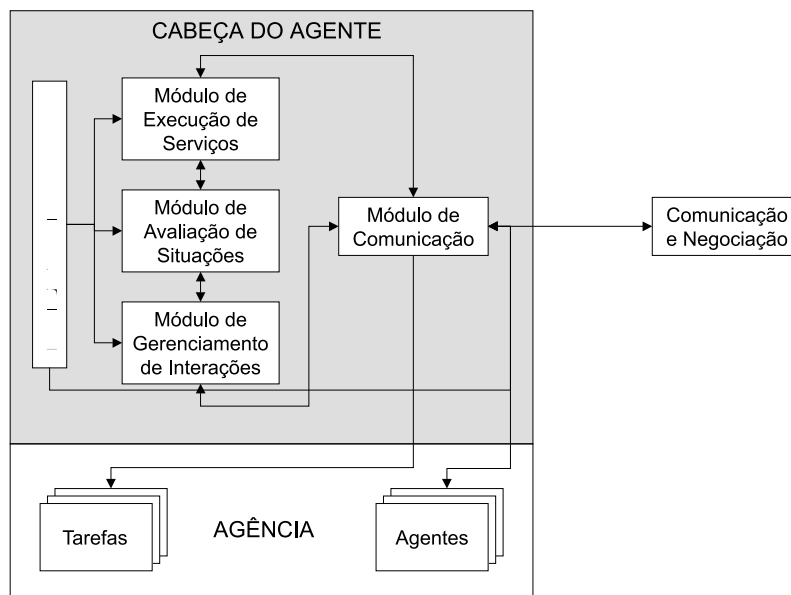


Figura 12: Arquitetura dos Agentes ADEPT [JEN96c].

Todos os agentes ADEPT seguem uma arquitetura básica. Esta arquitetura considera

um componente chamado cabeça do agente responsável pelo gerenciamento das atividades do agente e pela interação com outros agentes e uma agência que representa os recursos de resolução de problemas do domínio do agente.

A cabeça do agente agrupa componentes funcionais responsáveis por cada uma das atividades principais - comunicação, execução de serviços, avaliação de situações, e gerenciamento de interações - sendo eles:

- *Módulo de Comunicação* - responsável pelo roteamento das mensagens entre um agente e sua agência, e entre agentes.
- *Módulo de Gerenciamento de Interações* - fornecimento de serviços através da negociação.
- *Módulo de Avaliação de Situações* - responsável pela avaliação e monitoração das habilidades dos agentes conhecer os contratos já firmados e potenciais contratos que poderá firmar no futuro.
- *Módulo de Execução de Serviços* - responsável por gerenciar serviços durante a execução. Atende três papéis principais: gerenciamento de execução de serviços, gerenciamento de informação, e tratamento de exceções.
- *Modelos de Relacionamento* - mantém e provê acesso à contratos firmados com outros agentes e uma lista de agentes que podem fornecer serviços de interesse.
- *Modelo Próprio* - local de armazenamento primário para contratos firmados, descrições dos serviços que o agente pode fornecer, informação de aplicação/serviços em tempo de execução e informações genéricas do domínio.

4 Engenharia de Software Orientada a Agentes

[JEN00] apresenta uma abordagem do processo de desenvolvimento de software vinculada à idéia de MAS, a qual denominam Engenharia de Software Orientado a Agentes, como um paradigma⁴ adequado para o desenvolvimento de soluções de software para problemas complexos, tais como aqueles encontrados em aplicações industriais.

⁴Segundo [?], o termo paradigma pode ser entendido como modelo, exemplo ou tipo de conjugação ou declinação.

4.1 Características de Sistemas de Software Complexos

Os problemas encontrados em aplicações industriais são complexos por natureza (vide figura 13). Esta classe de aplicações é tipicamente caracterizada por um grande número de partes com muitas interações. Portanto, a finalidade do processo de engenharia de software é prover estruturas e técnicas para mais facilmente tratar esta complexidade. No entanto, esta complexidade geralmente apresenta importantes características [JEN00]:

- *a complexidade toma forma hierárquica, ou seja, o sistema é composto de sub-sistemas inter-relacionados* que, por sua vez, apresentam também uma hierarquia, e assim por diante;
- *a escolha de quais componentes no sistema são primitivos é geralmente arbitrária*, e esta definição depende muito dos objetivos que determinam o ângulo de visão através do qual o observador efetua a análise do problema;
- *a evolução ocorre mais rapidamente em sistemas hierárquicos do que em não-hierárquicos de mesmo tamanho*, ou seja, sistemas complexos resultarão do processo evolutivo de sistemas simples mais rapidamente se houverem formas estáveis intermediárias;
- *é possível fazer a distinção entre interações entre sub-sistemas e interações dentro de sub-sistemas*, o que permite tratar os sub-sistemas quase como se fossem independentes do outro.

As principais ferramentas existentes para o gerenciamento desta complexidade são:

- *Decomposição* - trata-se da técnica mais básica para resolução de grandes problemas - comumente conhecida pelo jargão "dividir para conquistar" - que trata de dividi-los em problemas menores e, potencialmente, mais facilmente gerenciáveis.
- *Abstração* - é uma técnica que visa considerar detalhes e propriedades relevantes escopo do problema em questão com o objetivo de gerar um modelo simplificado da realidade.
- *Organização* - trata de identificar e gerenciar os inter-relacionamentos entre os componentes de resolução do problema.

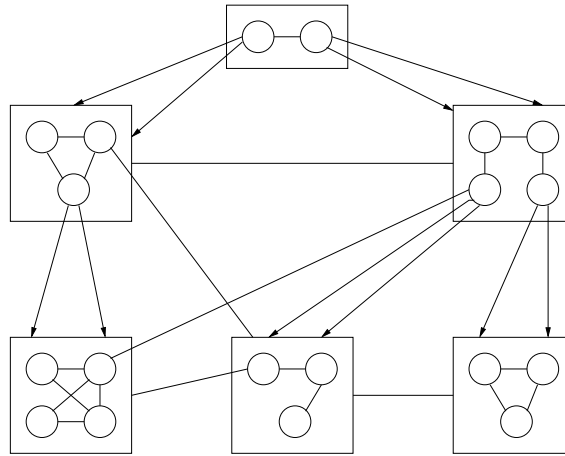


Figura 13: Visão de um Sistema Complexo [JEN00].

Considerando a natureza do problema em questão e a maneira através da qual estas ferramentas são implementadas para sua resolução varia entre os diferentes paradigmas de software. Isso nos leva a concluir que a adequação de um dado paradigma à resolução de um dado problema depende da forma pela qual este paradigma implementa estas ferramentas.

4.2 Software Orientado a Agentes

Ao adotar-se uma visão de mundo orientada a agentes, percebe-se que um simples agente é insuficiente para resolver a maioria dos problemas. Portanto, nestes casos, envolve-se múltiplos agentes no processo de resolução para que sejam representadas sua natureza descentralizada, as diversas perspectivas do mundo, ou os interesses conflitantes. Além disso, agentes precisam interagir com outros para atingir seus objetivos individuais ou para resolver as necessidades que surgem do fato de estarem situados em um ambiente comum. Duas importantes considerações devem ser feitas:

- estas interações ocorrem por meio de uma linguagem de alto-nível (declarativa) e, assim sendo, geralmente são conduzidas ao nível de conhecimento;
- agentes são flexíveis e operam em um ambiente sobre o qual têm controle parcial.

[JEN00] entende que interação social entre agentes significa a possibilidade de evolução dos relacionamentos existentes e criação de novos relacionamentos (vide figura 14).

Analisando-se os pontos acima abordados de maneira conjunta, é possível constatar que aplicar uma abordagem orientada a agentes para a resolução de um problema significa decompô-lo em múltiplos componentes autônomos com objetivos particulares e que se inter-relacionam. Com isso, podemos enumerar as três palavras-chave desta abordagem: agentes, interações e organizações.

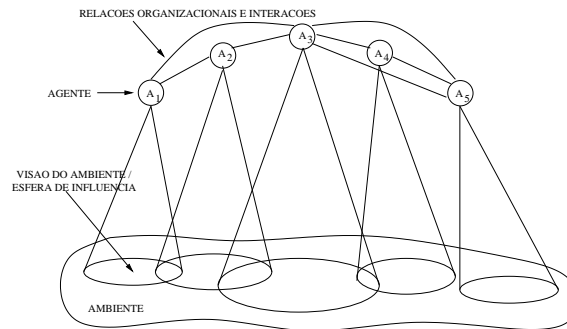


Figura 14: Visão de um Sistema Multi-agente [JEN00].

4.3 Aplicação da Abordagem Orientada a Agentes

[JEN00] considera que as técnicas que adotam uma abordagem orientada a agente são bem adaptadas para o desenvolvimento de sistemas complexos pelas seguintes razões:

- as decomposições da orientação a agentes são um caminho efetivo para particionar a problemática de um sistema complexo;
- as abstrações da orientação a agentes são uma abordagem natural para modelar sistemas complexos;
- a filosofia orientada a agentes para identificar e gerenciar relacionamentos organizacionais é apropriada para a representação das dependências e interações que existem em um sistema complexo.

4.4 Agentes e Objetos

As abordagens aqui apresentadas tentarão mostrar a Orientação a Objetos (OO) como ferramenta para construção de agentes⁵. A idéia fundamental aqui desenvolvida é a de

⁵Deve-se ter presente o fato de que não se trata da modelagem de sociedades e sim de indivíduos.

que é possível construir um agente modelando-se cada um de seus módulos funcionais como objetos ou agregações destes. O que justifica a intenção de se modelar um agente utilizando-se o paradigma orientado a objetos é a existência de inúmeras ferramentas de projeto, implementação e depuração para este paradigma.

Segundo a visão de [AMA97], o conceito agente apresenta semelhanças estruturais em relação ao conceito de objetos. O fato de que um agente é uma entidade que possui capacidades comportamentais e conhecimento privado, e que um objeto também possui esta mesma estrutura, demonstra a existência de características comuns entre estes conceitos. A partir disso, é possível afirmar que o que difere um agente e de um objeto é fundamentalmente o conceito de autonomia.

Amandi traça um paralelo entre os conceitos de agente e objeto mapeando o estado mental do agente para o estado interno do objeto e as capacidades comportamentais do agente para os métodos de um objeto. No entanto, [JUC98] realiza o mapeamento de cada um dos módulos funcionais do agente para um ou mais objetos, mapeamento este que se mostra mais conveniente para modelagem de agentes com arquiteturas funcionais.

Na visão de Wooldridge [WOO99], programadores do paradigma orientado a objetos geralmente cometem erros ao fazer comparações entre as propriedades relativas a agentes e objetos. Isso acontece porque tendem a pensar que são conceitos equivalentes por ambos serem entidades que encapsulam seus estados, serem aptas a desempenharem ações considerando estes, e comunicarem-se através da passagem de mensagens.

Porém, da mesma forma que similaridades existem, também há distinções significativas entre agentes e objetos. Primeiramente, agentes diferem de objetos por serem autônomos. Objetos têm autonomia sobre seu estado interno, mas não exibem controle sobre seu comportamento. Assim sendo, objetos têm controle sobre como as coisas são feitas, mas não têm nenhum poder de decidir se determinada solicitação vai ou não ser atendida. Esta distinção tem sido resumida no seguinte slogan: "Objetos fazem gratuitamente; agentes fazem por dinheiro".

É preciso observar que nada impede a implementação de agentes utilizando-se técnicas OO. Mas é necessário notar que a característica de autonomia não é um componente básico do modelo OO.

Uma segunda importante diferença entre sistemas OO e sistemas orientados agentes diz respeito à noção de comportamento autônomo flexível (reativo, pró-ativo, social). O modelo OO padrão não especifica nenhuma maneira de construir sistemas que integram estes tipos de comportamento, embora isto possa ser implementado utilizando-se técnicas

OO.

A terceira diferença importante diz respeito ao fato de que, dentro de um Sistema Multiagente, cada agente possui necessariamente sua própria linha de execução. Assim sendo, mais que uma estrutura composta por métodos e estado interno, cada um dos agentes de uma sociedade pode ser visto como sendo um processo computacional.

Segundo [SHO93], agentes podem ser vistos como objetos ativos com estados mentais, traçando semelhanças entre os dois conceitos, tais como a existência de troca de mensagens entre as entidades para a troca de informações e solicitação de serviços, e os conceitos de herança e agregação.

5 Conclusão

Este trabalho apresentou as diversas conceitualizações dadas para o termo agente, bem como para os termos relacionados, fazendo surgir a inexistência de um consenso sobre o que seja um agente. Porém, pôde-se notar a existência de características comuns aos conceitos apresentados, sendo que a mais evidente delas, segundo Wooldridge [WOO95] [WOO99], é a de que um agente está situado em um ambiente sobre o qual atua de maneira autônoma.

A seguir, examinando algumas das classificações existentes para os diversas arquiteturas de agentes, pôde-se verificar uma classificação que representa-se, em linhas gerais, um consenso acerca do vêm sendo utilizado para tal fim. Considerando isso, pode-se identificar dois extremos entre os quais uma arquitetura de agente pode estar situada: reativas e deliberativas.

- *Arquiteturas deliberativas* são aquelas que baseiam seu processo decisório em raciocínio lógico (explícito) que opera sobre uma representação simbólica (interna ao agente) do mundo. Não há consenso de que capacidades de planejamento sejam necessárias para que uma arquitetura seja considerada deliberativa, mas existem classificações que definem esta restrição.
- *Arquiteturas reativas* definem que o processo de tomada de decisão ocorre em resposta a estímulos do ambiente ou por mensagens enviadas por outros agentes, tendo como base um conjunto de regras evento-ação.

- *Arquiteturas híbridas* adotam a idéia de combinar características reativas à uma arquitetura deliberativa, ou vice-versa.

Como estudos de caso, foram apresentadas algumas das arquiteturas existentes. Primeiramente apresentou-se um conjunto mais genérico de arquiteturas definidos por Russel [RUS95] com suas respectivas classificações. Apresentou-se também o conceito do que seja um ambiente (dentro do qual um agente se insere) e suas classificações - acessível ou inacessível; determinístico ou não-determinístico; episódico ou não-episódico; estático ou dinâmico; e discreto ou contínuo. Em um segundo momento, apresentou-se um conjunto de arquiteturas mais específicas à linha de pesquisa à qual este trabalho está vinculado - ADEPT [JEN96c] e M-DRAP [BAS98].

Finalmente, apresenta-se características que evidenciam a adequação da utilização de agentes no processo de engenharia de software. Segundo Jennings [JEN00], a complexidade dos sistemas de software apresenta regularidades. Essas regularidades dão suporte à um conjunto de ferramentas existentes para o gerenciamento desta complexidade, sendo elas: (i) decomposição; (ii) abstração; e (iii) organização. Jennings, então, evidencia características do paradigma orientado a agentes que o tornam adequado à criação de soluções para problemas complexos.

A contribuição deste trabalho foi a organização dos conceitos relativos a SMA que viabilizarão uma posterior pesquisa mais centrada em aspectos de modelagem de Sistemas Multiagentes sob o ponto de vista da engenharia de software.

Referências

- [AMA97] AMANDI, A.A. Programação de Agentes Orientada a Objetos. Porto Alegre: CPGCC da UFRGS, 1997. Tese de Doutorado.
- [BAS97] BASTOS, R.M., OLIVEIRA, J.P.M., OLIVEIRA, F.M. Uma Estratégia para Alocação de Recursos através de um Sistema Multi-Agentes. In: XXIII CONFERÊNCIA LATINOAMERICANA DE INFORMÁTICA, 1997, Valparaíso, Chile.
- [BAS98] BASTOS, R.M. O Planejamento de Alocação de Recursos Baseado em Sistemas Multiagentes. Porto Alegre: 1998. Tese de Doutorado.

- [BON88] BOND, A.H. & GASSER, L. (Eds.) Readings in distributed artificial intelligence. San Mateo: Morgan Kaufmann, 1988.
- [COR94] CORREA FILHO, M. A Arquitetura de Diálogos entre Agentes Cognitivos Distribuídos. Rio de Janeiro: COPPE da UFRJ, 1994. Tese de Doutorado.
- [D'AM95] D'AMICO, C. et ali. Inteligência artificial: uma abordagem de agentes. Porto Alegre: CPGCC da UFRGS, 1995. (RP-257).
- [JEN96] JENNINGS, N.R. Coordination techniques for distributed artificial intelligence. In: O'HARE, G.M.P.; JENNINGS, N.R. (Eds.). Foundations of distributed artificial intelligence. New York: John Wiley & Sons, 1996. p.187-210.
- [JEN96b] JENNINGS, N.R., WOOLDRIDGE, M. Software Agents. IEE Review, January, 1996, 17-20.
- [JEN96c] JENNINGS, N.R., FARATIN, P., NORMAN, T.J. O'BRIEN, P., WIEGAND, M.E., VOUDOURIS, C., ALTY, J. L., MIAH, T., MANDANI, E. H. ADEPT: Managing Business Processes using Intelligent Agents. In: Proc. BCS Expert Systems 96 Conference (ISIP Track), Cambridge, UK, 1996, 5-23.
- [JEN00] JENNINGS, N.R. On Agent-Oriented Software Engineering. 117 (2) p.227-296.
- [JUC98] JUCHEM, M. Sistema de Modelagem e Instanciação de Agentes. Porto Alegre: II da PUCRS, 1998. Trabalho de Conclusão.
- [MOF95] MOFFAT, D., FRIJDA, N.H. Where there's a Will there's an Agent, Lecture Notes in Computer Science, v.890, p.245-??, 1995.
- [OLI96] OLIVEIRA, F.M. Inteligência Artificial Distribuída. In: IV ESCOLA REGIONAL DE INFORMÁTICA, Canoas, 3, 1996. Anais: Sociedade Brasileira de Computação, 1996. 239 p.
- [RUS95] RUSSEL, S.J., NORVIG, P. Artificial Intelligence: a modern approach. Prentice-Hall, New Jersey, 1995.

- [SHO93] SHOHAM, Y. Agent-oriented programming. *Artificial Intelligence*, Amsterdam, v.60, p.51-92, 1993.
- [STE96] STEINER, D.D. IMAGINE: An Integrated Environment for Constructing Distributed Artificial Intelligence Systems. In: O'HARE, G.M.P.; JENNINGS, N.R. (Eds.). *Foundations of distributed artificial intelligence*. New York: John Wiley & Sons, 1996. p.345-364.
- [WOO95] WOOLDRIDGE, M., JENNINGS, N.R. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10 (2):115-152, 1995.
- [WOO99] WOOLDRIDGE, M. Intelligent Agents. In: WEISS, G. (Ed.) *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.