



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

**Engenharia de Sistemas Multiagentes:
Uma Investigação sobre o Estado da Arte**

Murilo Juchem e Ricardo Melo Bastos

TECHNICAL REPORT SERIES

Number 014
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

Engenharia de Sistemas Multiagentes: Uma Investigação sobre o Estado da Arte

Relatório Técnico 014/2001

Murilo Juchem*
Ricardo Melo Bastos, Dr.†

1 Introdução

Adotar a abordagem de multiagentes para a resolução de problemas têm sido um assunto muito explorado nas últimas duas décadas. Inúmeros trabalhos têm apresentado conceitualizações, formalizações, protocolos, técnicas e métodos para aplicação deste tipo de abordagem na concepção de software. Isso tem acontecido pelo fato da abordagem multiagente possuir algumas características que viabilizam a resolução de problemas de outra forma que não a tradicional, adequando-se à problemas complexos [JEN99, JEN98] e de natureza descentralizada [MOU96].

Este paradigma adota o conceito de *agente* para caracterizar uma unidade autônoma de resolução de problemas [WOO95]. A partir disso, uma solução é criada através do agrupamento de agentes que trabalham cooperativamente, cada um deles resolvendo parte do problema. A este agrupamento é dado o nome de *Sistema Multiagente* (SMA). Segundo Jennings [JEN96a], SMA também refere-se à sub-área da Inteligência Artificial Distribuída (IAD) que investiga o comportamento de um conjunto de agentes autônomos objetivando a solução de um problema que está além das capacidades de um único agente.

Sob o ponto de vista da Engenharia de Software, a construção de software de alta qualidade de maneira produtiva é viabilizada por um conjunto de métodos, ferramentas

*juchem@inf.pucrs.br

†bastos@inf.pucrs.br

e procedimentos. O caminho para a evolução no desenvolvimento de software passa por uma combinação de métodos abrangentes para todas as fases de desenvolvimento do software, melhores ferramentas para automatizar estes métodos, blocos de construção mais poderosos para a implementação do software, melhores técnicas para a garantia da qualidade do software e uma filosofia de coordenação predominante, controle e administração [PRE95] (vide também [DAV85]).

Inserido nesta idéia, é necessário que, a exemplo do que ocorre com o paradigma Orientado a Objetos (OO) [SHL90, MAR95], o paradigma Orientado a Agentes (OA) seja também suportado por metodologias e técnicas que garantam a qualidade do processo de produção do software.

Estudos foram feitos com o objetivo de verificar a existência de um paralelo entre os paradigmas OO e OA [AMA97, JEN99, ODE99] que permita a aplicação das metodologias existentes para OO para construção de um SMA. Segundo Bastos [BAS98], é possível modelar um agente utilizando uma abordagem OO. No entanto, persiste a problemática da modelagem da sociedade, pois, por não adequar-se ao modelo conceitual, a abordagem OO não viabiliza a modelagem de todos os aspectos envolvidos em um SMA [IGL98a, WOO99a, KEN96, TAV99].

2 Sistemas Multiagentes

Um *agente* é uma entidade de software que exhibe um comportamento autônomo¹ e pró-ativo orientado a objetivos, que está situado em algum ambiente sobre o qual é capaz de realizar ações para alcançar seus próprios objetivos de projeto e a partir do qual percebe alterações [WOO95, O'HA96, ZAM00].

Wooldridge [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***² - executam a maior parte de suas ações sem interferência direta de

¹Segundo Wooldridge [WOO99], há um consenso geral de que autonomia é a idéia central da noção de agência. Entende-se autonomia como a capacidade do agente agir por seus próprios objetivos sem a intervenção de outrem [MOF95] [RUS95].

²Para ter autonomia, o agente deve ter um certo grau de inteligência, o que capacita-o a sobreviver em um ambiente dinâmico e por vezes não benigno [COR94]. Segundo Russell [RUS95], o comportamento de um agente pode ser baseado tanto em sua própria experiência, quanto em seu conhecimento embutido

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 tenham capacidade para comunicar seus requisitos aos outros e um mecanismo decisório interno que defina quando e quais interações são apropriadas;
- ***capacidade de reação*** - percebem e reagem à alterações no ambientes em que estiverem inseridos.
- ***capacidade pró-ativa*** - 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.

Agentes operam e existem em um ambiente, que pode conter ou não outros agentes. Em algumas situações, agentes podem executar tarefas por si próprios, mas, em outras, têm a necessidade do auxílio de outros agentes. Isso acontece a todo o momento no mundo real, pois quando alguém tem a necessidade de resolver algum problema que envolva algum conhecimento ou habilidade que não possua, necessita o auxílio de outra pessoa apta a fazê-lo.

Devido ao grande crescimento da utilização de computadores em redes - e podemos tomar a Internet como um importante exemplo prático deste fenômeno -, onde a informação está distribuída através dos diversos nodos que a compõe, situações onde uma entidade computacional possui todo o conhecimento necessário para resolver problemas sem o auxílio de outras estão tornando-se cada vez mais raras. Problemas como aumento da complexidade dos sistemas e a necessidade de tratamento de grandes massas de dados para a resolução de problemas têm levado pesquisadores a buscar métodos de resolução baseados em arquiteturas distribuídas. Isso tem acontecido porque as arquiteturas distribuídas vêm se mostrando muito úteis para resolução de problemas onde a própria natureza é distribuída. Pode-se pensar na aplicação de um modelo monolítico para a uso na construção do agente para um ambiente em particular. 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.

resolução de problemas em uma realidade distribuída, onde os eventos ocorrem concorrentemente. No entanto, isto requer uma complexa etapa intermediária de mapeamento que, muitas vezes, resulta em um algoritmo não-computável. Da mesma forma, a distribuição pode levar à descoberta de algoritmos computacionais que talvez não teriam sido descobertos com uma abordagem centralizada. Em outros casos, como negociação entre empresas independentes, uma abordagem centralizada é inviável, já que cada empresa deseja manter suas informações em um âmbito privado por razões mercadológicas [HUH99].

Sendo um agente uma entidade que encapsula conhecimento sobre algum domínio, nada mais natural do que agrupar agentes que possuam parte do conhecimento envolvido na estratégia de resolução de um problema e que, a partir disso, interajam com o objetivo de complementarem suas habilidades. Assim, da mesma forma que no mundo real existem empresas com funcionários possuidores de diferentes habilidades e que, utilizando essas habilidades, desenvolvem parte das atividades necessárias ao processo produtivo, podemos compor uma sociedade de agentes onde para cada agente seja alocada um subconjunto das habilidades requeridas pela estratégia de solução onde a cada um seja designado parte das tarefas a serem cumpridas, de acordo com sua disponibilidade de recursos (computacionais, materiais, tempo, etc.).

Podemos distinguir duas principais classes de sistemas com múltiplos agentes [ZAM00]:

- **Sistemas de Resolução Distribuída de Problemas**, nos quais os agentes envolvidos são explicitamente projetados para, de maneira cooperativa, atingirem um dado objetivo, considerando-se que todos os eles são conhecidos à priori e supondo que todos são benevolentes, existindo desta forma confiança mútua em relação às suas interações; e
- **Sistemas Abertos**, nos quais os agentes não são necessariamente projetados para atingirem um objetivo comum, podendo ingressar e sair do sistema de maneira dinâmica. Neste caso, a chegada dinâmica de agentes desconhecidos precisa ser levada em consideração, bem como a possível existência de comportamento não benevolente no curso das interações.

Dentro desta segunda classificação, estão inseridos os **Sistemas Multiagentes**. Neste tipo de sistema, investiga-se o comportamento de um conjunto de agentes autônomos, possivelmente pré-existent, que interagem objetivando a resolução de um problema que

está além das capacidades de um único indivíduo [O'HA96]. Desta forma, o *comportamento global* do sistema deriva da interação entre os agentes que fazem parte do sistema [ZAM00]. A partir disso, 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 onde se faça necessária a cooperação entre os agentes [BON88]. Nestes casos, diz-se que o agente exibe um *comportamento social* [ZAM00].

Moulin e Chaib-Draa [MOU96] evidenciam as características que constituem vantagens significativas dos Sistemas Multiagente 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 que são dinamicamente agrupados para resolver problemas; e
- aumento da segurança pela possibilidade de agentes assumirem responsabilidades de agentes que falham.

A justificativa de aplicação da tecnologia de agentes na concepção de Sistemas de Informação é justificada quando o problema possui as seguintes características [JEN96]:

- o domínio envolve distribuição intrínseca dos dados, capacidades de resolução de problemas e responsabilidades;
- necessidade de manter a autonomia de subpartes, sem a perda da estrutura organizacional;
- complexidade nas interações, incluindo negociação, compartilhamento de informação e coordenação;
- impossibilidade de descrição da solução do problema a priori, devido à possibilidade de perturbações em tempo real no ambiente (p.ex.: falhas de equipamento) e processos de negócio de natureza dinâmica.

3 Aplicabilidade de Agentes em Sistemas Complexos

Esta seção apresenta uma justificativa para aplicabilidade da abordagem do processo de desenvolvimento de software vinculada à idéia de Sistemas Multiagentes, denominado Engenharia de Software Orientado a Agentes [JEN99], como um paradigma adequado para o desenvolvimento de soluções de software para problemas complexos (vide figura 1), tais como aqueles encontrados em aplicações industriais.

3.1 Características de Sistemas de Software Complexos

Os problemas encontrados em aplicações industriais são complexos por natureza. 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 [JEN99]:

- *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 um do outro.

As principais ferramentas existentes para o gerenciamento desta complexidade são:

- **Decomposição** é a 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.

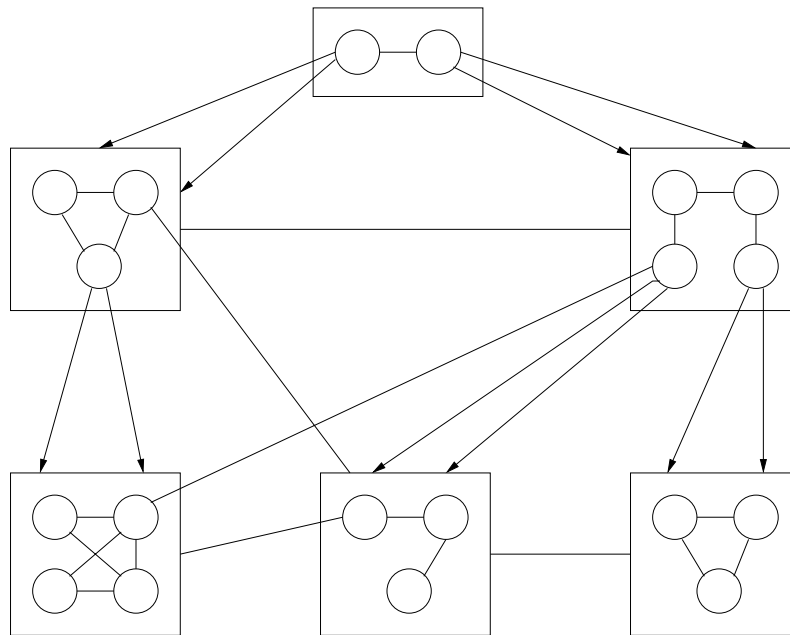


Figura 1: Visão de um Sistema Complexo [JEN99].

- **Abstração** é uma técnica que visa considerar detalhes e propriedades relevantes ao 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.

Considerando a natureza do problema em questão, 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.

Kornfield e Hewitt Apud [JEN96a] postulam que a busca cooperativa resulta no fenômeno da “*implosão combinatória*”, que viabiliza a substituição de soluções cuja complexidade resulte numa explosão combinatória, inviabilizando o processo do ponto de vista computacional, por soluções descentralizadas e distribuídas.

3.2 Software Orientado a Agentes

Algumas das características de agentes são [JEN99]:

- *entidades de resolução de problemas claramente identificáveis* com limites e interfaces bem definidos;
- *entidades situadas em um ambiente em particular* do qual recebem entradas correspondentes ao estado deste e sobre o qual intervêm através de atuadores;
- *projetados para realizar um papel específico*, tendo objetivos particulares a atingir;
- *autônomos*, têm controle sobre seu estado interno e sobre seu próprio comportamento;
- *capazes de exibir um comportamento flexível para a resolução de problemas*, necessitam ser reativos (responder imediatamente à alterações do ambiente) e pró-ativos (para agir de acordo com seus objetivos).

Ao adotar-se uma visão de mundo orientada a agentes (vide figura 2), 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 do problema para que seja representada 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 o acesso aos recursos do ambiente. 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; e
- agentes são operam em um ambiente sobre o qual têm controle parcial.

Entende-se que interação social entre agentes significa a possibilidade de evolução dos relacionamentos existentes e criação de novos relacionamentos.

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.

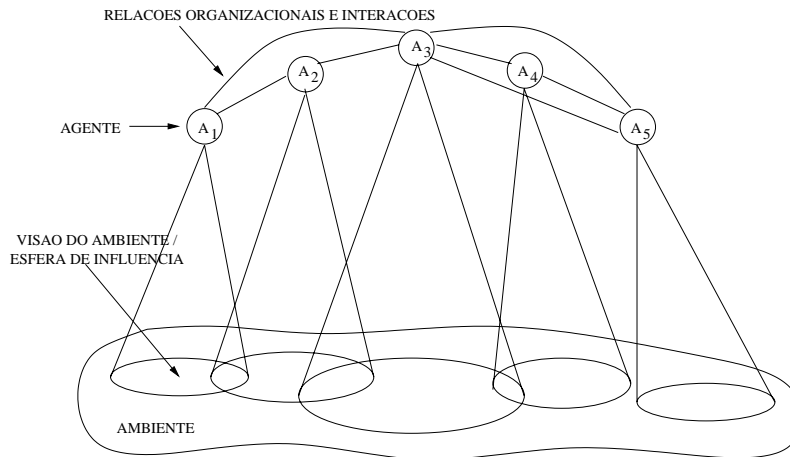


Figura 2: Visão de um Sistema Multiagente [JEN99].

3.3 Aplicação da Abordagem Orientada a Agentes

As técnicas que adotam uma abordagem orientada a agente são bem adaptadas para o desenvolvimento de sistemas complexos pelas seguintes razões [JEN99]:

- 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 Aspectos a Considerar no Desenvolvimento de um SMA

As características da abordagem multiagente impõem necessidades que devem ser viabilizadas para que o sistema possa ser considerado eficaz. Considerando o exemplo de uma organização empresarial tradicional como sendo uma sociedade de pessoas que combinam esforços para a resolução de um problema comum, podemos dizer que esta constitui um exemplo de sistema de resolução de problemas de natureza distribuída. Neste caso, para obter-se resultados favoráveis nesta empresa devem ser criadas regras que definam uma

estrutura que viabilize o alcance dos seus objetivos e implementação de suas estratégias de funcionamento. Em um SMA, devemos considerar critérios que viabilizem e garantam a coerência das ações dos agentes com vistas a atingir de maneira efetiva os objetivos do SMA.

Moulin e Chaib-Draa [MOU96] propõem um framework que fornece uma estrutura de análise e classificação da maior parte das atividades de pesquisa em Sistemas Multiagentes, do qual podemos citar duas perspectivas:

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

Dentro da perspectiva de grupo, podemos definir três grandes grupos de aspectos a serem considerados no projeto de um SMA:

- ***aspectos fundamentais*** - definem as características que devem ser viabilizadas para a garantia da compatibilidade entre as ações dos agentes que constituem o SMA.
- ***aspectos arquiteturais*** - definem as características que devem ser providas pela arquitetura a ser adotada para viabilização dos aspectos fundamentais dentro do SMA.
- ***aspectos ambientais*** - definem as características do ambiente no qual os agentes do SMA estarão inseridos, para que se possa determinar os tipos de técnicas de percepção que devam ser utilizadas por estes agentes.

4.1 Aspectos Fundamentais

A coerência de um SMA é viabilizada pela garantia de um comportamento coerente de seus agentes, ou seja, as ações dos agentes fazem sentido em relação aos objetivos comuns do grupo [MOU96]. Dentro da perspectiva de grupo, serão examinados a seguir os aspectos fundamentais a serem considerados para o desenvolvimento de Sistemas Multiagentes para a garantia da compatibilidade das ações dos agentes.

Estrutura

Atualmente, adotar a idéia de *organização* e de *mudanças organizacionais* é importante para criar SMA mais adaptáveis. No entanto, é necessário notar a distinção entre os termos *estrutura* e *organização* [MOU96].

Entende-se *estrutura* como sendo o padrão de relações de informação e controle entre agentes, bem como a distribuição das habilidades entre eles. Assim, a estrutura fornece uma visualização de como os problemas são resolvidos pelo grupo e o papel que cada agente desempenha dentro desta estrutura, ou seja, define os papéis e os relacionamentos para atender às seguintes condições [MOU96]:

- **Cobertura** - qualquer habilidade necessária para a resolução do problema deve estar inserida no rol de habilidades de ao menos um agente.
- **Conectividade** - agentes devem interagir de maneira a permitir que suas habilidades sejam integradas e desempenhadas no sentido de contribuir para uma solução global.
- **Potencialidade** - cobertura e conectividade devem ser atingíveis dentro de limitações computacionais e de comunicação, assim como as especificações de confiabilidade do grupo.

Desta maneira, cada agente desempenha um ou mais papéis específicos no sistema. Define-se como papel aquilo que é esperado que o agente faça dentro da organização, ou seja, um conjunto de responsabilidades bem definidas dentro do contexto global do sistema que o agente pode cumprir com um certo grau de autonomia [ZAM00].

Esta perspectiva pode tornar menos complexo e de gerenciamento mais simples o projeto de sistemas paralelos e distribuídos em relação à maioria das metáforas tradicionais de sistemas concorrentes, considerando-se que [ZAM00]:

- Cada agente tem controle próprio sobre seu processamento, sendo totalmente responsável em cumprir seu papel;
- As interdependências entre os componentes do sistema são reduzidas, uma vez que cada agente têm embutida a maior parte da funcionalidade para cumprir seu papel. Essa independência facilita o projeto por imprimir uma clara separação entre os níveis de componente (intra-agente ou *micro-nível*) e do sistema (inter-agente ou *macro-nível*);

- Em muitos casos, SMAs pretendem suportar ou controlar alguma organização existente na realidade, pois sua adoção reduz a distância entre os sistemas de software e os sistemas de mundo real, o que simplifica o desenvolvimento do sistema.

Em SMAs, o comportamento autônomo e pró-ativo exibido pelos agentes constituintes sugerem que aplicações podem ser projetadas tomando-se como exemplo o comportamento e a estrutura de sociedades humanas [ZAM00].

Organização

O conceito de *organização* refere ao conjunto de compromissos globais, crenças mútuas, e intenções comuns aos agentes quando agem juntos para atingir um dado objetivo [MOU96]. Estes elementos definem um conjunto de diretrizes a serem seguidas por cada um dos agentes do SMA, descrevendo uma política de interação entre os agentes (p.ex.: o agente *X* acredita que o agente *Y* tem autoridade para controlar determinado recurso).

Estas diretrizes podem evitar a ocorrência de situações caóticas no SMA, criando padrões comportamentais que evitem que o SMA se torne uma “terra sem lei”, facilitando ou mesmo viabilizando a realização dos objetivos globais da sociedade.

Coordenação

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.

“*Coordenação* é o processo pelo qual um agente raciocina sobre suas ações locais e as ações (antecipadas) de outros agentes com o objetivo de garantir que a comunidade funcione de maneira coerente³” [JEN96a]

A definição de estratégias que conciliem os interesses individuais de cada agente para que as atividades relacionadas desenvolvam-se de modo coordenado é um dos aspectos fundamentais a serem considerados no projeto de Sistemas Multiagentes [SEG96].

A necessidade de coordenação entre múltiplos agentes surge do fato de que [JEN96a]:

³Coerência significa quão bem um sistema comporta-se como uma unidade [HUH99].

- existem dependências entre as ações dos agentes, ou seja, a ação de um agente pode ser pré-requisito da ação de outro agente; e
- nenhum indivíduo tem competência, recursos ou informação suficientes para resolver um problema completo de forma independente; onde
- deve ser garantido o respeito às restrições globais à solução do problema; e
- devem ser viabilizados procedimentos que garantam a harmonia quando da execução de uma tarefa de forma conjunta por mais de um agente.

A coordenação é um fator vital para o funcionamento de um Sistema Multiagente, pois sem coordenação os benefícios advindos da resolução distribuída de problemas desaparecem e a comunidade pode degenerar em uma caótica coleção de indivíduos que agem de forma incoesa em relação ao sistema como um todo [MOU96, JEN96a]. Para isso, basta que um único agente tenha uma visão parcial ou imprecisa do sistema e que suas ações possam interferir nas ações de outros agentes ao invés de suportá-las [MOU96].

Segundo Lesser e Corkill Apud [JEN96a], os objetivos do processo de coordenação visam garantir que:

- todas as partes necessárias ao sistema estejam inseridas nas capacidades funcionais de ao menos um agente;
- os agentes interajam de maneira a permitir que suas atividades sejam desenvolvidas e integradas em uma solução global;
- os membros da sociedade atuem com propósitos e consistentemente;
- todos esses objetivos sejam atingíveis dentro das limitações computacionais impostas e dos recursos disponíveis.

Durfee e outros [DUR96] identificam três fatores básicos que devem estar presentes para uma coordenação com sucesso:

- existência de uma *estrutura* que permita aos agentes interagirem de forma preditiva;
- *flexibilidade* nas interações de tal forma que os agentes possam operar em ambientes dinâmicos e agir satisfatoriamente com uma visão parcial e imprecisa da sociedade;

- os agentes devem possuir *conhecimento e capacidade de raciocínio* suficientes para explorar esta estrutura e flexibilidade.

A maneira mais fácil de garantir um comportamento coerente dentro de um Sistema Multiagente é implantar um agente que tenha uma perspectiva mais ampla do sistema, atuando como um *coordenador* que, reunindo informações sobre toda a sociedade, seria responsável por criar planos e atribuir tarefas aos membros desta sociedade [MOU96, JEN96a]. No entanto, esta não é uma abordagem prática em sistemas reais por ser muito difícil criar um agente que mantenha-se informado sobre todas as intenções e crenças de todos os agentes do grupo [MOU96]. Além disso, um coordenador centralizado tornaria-se um gargalo⁴ de comunicação, que degradaria a performance do sistema. Também devemos considerar o fato de que, nesta abordagem, uma falha do agente coordenador comprometeria o funcionamento de todo o sistema [JEN96a], apesar de que, neste caso, poderiam ser adotados mecanismos de tolerância à falhas onde outro agente poderia assumir o papel de coordenador.

A partir disso, uma problemática apresentada para o projeto de Sistemas Multiagentes passa a ser a manutenção da coerência global sem um controle global explícito [HUH99]. Segundo Oliveira [OLI96], “os agentes devem raciocinar a respeito das ações, mas também sobre o processo de coordenação em si”. Muitas pesquisas têm seu foco no desenvolvimento de comunidades nas quais o controle e os dados são distribuídos⁵.

Uma desvantagem advinda da distribuição do controle e dos dados é a dificuldade de se ter conhecimento sobre o estado global do sistema, que está disperso através da comunidade, sendo que cada indivíduo possui uma visão parcial e imprecisa desta perspectiva.

Os principais requisitos para a coordenação são [SEG96]:

- comunicação entre os agentes;
- reconhecimento das interações potenciais entre planos; e
- negociação entre os agentes.

Consideraremos três *processos de coordenação* fundamentais (segundo Mintzberg Apud [MOU96]):

⁴Adotaremos como conceito de “gargalo” as situações de degradação de performance em que um processo é sobrecarregado por muitas requisições de outros que aguardam o atendimento destas.

⁵*Controle Distribuído* significa que indivíduos têm um grau de autonomia na geração de novas ações e na decisão sobre qual tarefa executar primeiro [JEN96a].

- **Ajuste mútuo** é a forma mais simples de coordenação, pressupondo dois ou mais agentes que concordam em compartilhar recursos para atingir algum objetivo comum.
- **Supervisão direta** prevê que uma relação já tenha sido estabelecida entre dois ou mais agentes na qual um agente mantém algum controle sobre outros.
- **Padronização** estabelece uma relação entre dois ou mais agentes na qual um agente mantém o controle sobre os outros estabelecendo procedimentos padronizados a serem seguidos por seus supervisionados em determinadas situações.

Utilizando-se destes três processos de coordenação, é possível descrever sistemas de coordenação sofisticados a partir do qual destacam-se (segundo Malone Apud [MOU96]):

- **Hierarquias** - baseado no processo de Supervisão Direta, este sistema aglutina agentes em pequenos grupos coordenados por um supervisor, dispostos de forma hierárquica, onde um supervisionado pertencente a um grupo pode atuar como supervisor de outro. Este tipo de estratégia permite dividir um grupo grande em vários sub-grupos, distribuindo os fluxos de coordenação entre vários níveis de supervisão (vide item 4.2).
- **Mercados** - baseado no processo de Ajuste Mútuo, este sistema pressupõe que agentes controlam recursos escassos e concordam em compartilhá-los com outros agentes para atingir um objetivo comum. A estes recursos são vinculados preços e, uma vez que um contrato foi fechado, há uma aceitação de que o comprador torna-se o supervisor do fornecedor.

O conceito de coordenação define aspectos gerais de interação entre agentes de forma a viabilizar a coesão entre seus comportamentos e ações em relação aos objetivos globais do sistema. A partir disso, apresenta-se uma taxonomia para os processos de coordenação (vide figura 3) que apresenta duas abordagens principais [HUH99]:

- **Cooperação** - é a coordenação entre agentes não-antagônicos⁶;
- **Negociação** - é a coordenação entre agentes competitivos ou que agem em interesse próprio.

⁶O conceito de não-antagônico define que os interesses dos agentes não são conflitantes.

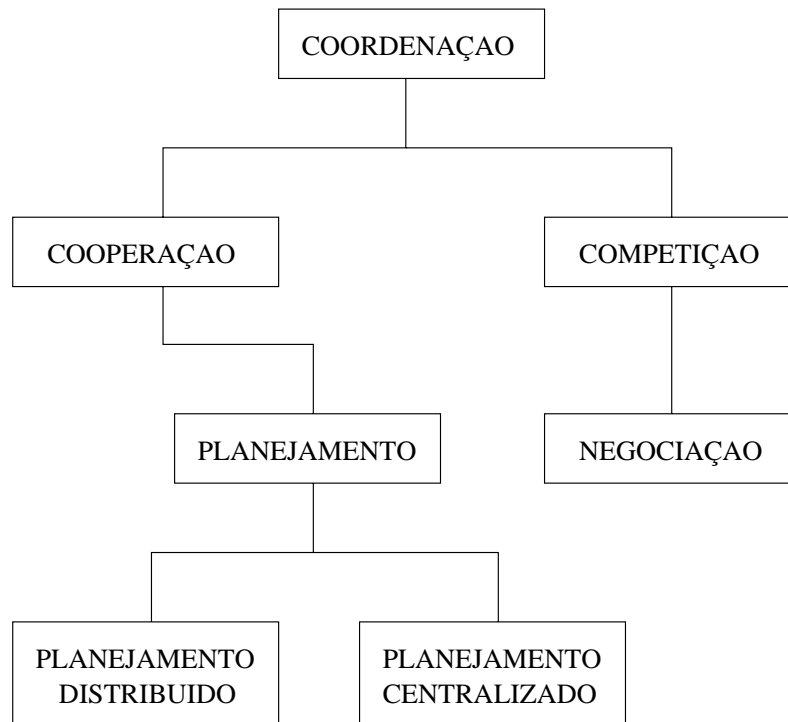


Figura 3: Taxonomia da coordenação [HUH99].

Cooperação Lux e Steiner [LUX95] consideram que a “cooperação acontece quando vários agentes planejam e executam suas ações de uma forma coordenada”, sendo que é requerida quando:

- o agente não consegue encontrar um plano local que contemple o objetivo;
- o plano adequado ao objetivo envolve ações de outros agentes;
- o agente considera que um plano pode ser melhor (de menor custo ou mais eficiente) do que um plano local;
- durante a fase de planejamento:
 - o agente encontra planos incompletos, que podem ser completados em cooperação com outros agentes; ou
 - o agente encontra eventos para os quais não está habilitado a responder, mas sabe que outros agentes estão.

Os objetivos genéricos para a cooperação entre agentes são [MOU96]:

- diminuição do tempo de execução de uma tarefa através do paralelismo;
- aumento do escopo de tarefas executáveis através do compartilhamento de recursos;
- maior probabilidade de finalização de uma tarefa em função de sua dupla incumbência, a ser realizada possivelmente através de distintos métodos de execução; e
- diminuição da interferência entre tarefas evitando interações prejudiciais.

O processo de *planejamento* constitui uma forma especializada do processo de cooperação que tem como produto um conjunto de atividades organizados com um curso de ação definido, em que estas atividades são distribuídas entre agentes capacitados a executá-las. Este planejamento pode ocorrer de duas formas:

- *centralizada* - um único agente constrói o plano; ou
- *distribuída* - pressupõe que o plano é construído por mais de uma agente, sendo considerada quando um único agente não possui uma visão global das atividades do grupo.

O planejamento em SMA apresenta-nos questões sobre a forma através da qual um agente poderia [DUR96] (i) formular um curso de ação levando em conta as ações a serem executadas paralelamente por outros agentes, (ii) descobrir o curso de ação de outros agentes e (iii) identificar de que forma um agente poderia comprometer-se com seus próprios modelos.

Um modelo de planejamento distribuído deve considerar [SEG96] (i) a concorrência existente entre os agentes em um ambiente compartilhado, e (ii) as alterações ocorridas no ambiente entre o planejamento e a execução dos planos resultantes deste.

Ajustes nos planos de um agente podem ser requeridas por [DUR96] (i) resultados de suas próprias ações, (ii) resultados de ações de outros agentes, (iii) mudanças no ambiente, (iv) mudanças de objetivos, ou (v) alterações na percepção do agente quanto ao contexto multiagente no qual está inserido.

Segundo Bond e Gasser [BON88], alinhar o comportamento de agentes em direção à objetivos comuns, com a divisão explícita de tarefas, pode viabilizar a realização de grandes objetivos. O autor afirma que técnicas como planejamento centralizado para

múltiplos agentes, reconciliação de planos, planejamento distribuído, análise organizacional, são caminhos que auxiliam o alinhamento das atividades dos agentes por determinar tarefas após raciocinar sobre as consequências destas em certas organizações.

Segundo Agre e Chapman, a utilização de planos pode ser vista de duas maneiras [AGR89]:

- ***Plano como Programa*** - O uso de plano é a execução de determinado procedimento, sendo que esta visão implica na execução de planos de forma independente do domínio, e a construção de planos pode ser independente ou dependente do domínio, algorítmica ou baseada em casos, formalmente correta ou heurística. Nesta visão, o plano pode ser decomposto em ações primitivas que podem ser simplesmente executadas através de um dispositivo independente do domínio. As razões para duvidar da visão de plano como programa são :
 - possui problemas intratáveis computacionalmente;
 - é inadequada para um mundo caracterizado por eventos imprevisíveis, tais como as ações de outros agentes;
 - requer que planos sejam muito detalhados;
 - falha ao tratar problemas de relacionamento do texto do plano para a situação concreta.
- ***Planos como Comunicação*** - O uso de plano é seguir instruções em linguagem natural, sendo que um plano não determina diretamente as ações de um agente. Ao invés disso, um plano é um recurso que o agente utiliza para decidir o que fazer. Nesta visão, resolver qual atividade que um plano sugere requer um esforço interpretativo contínuo. A única situação completa conhecida é a situação inicial passada ao planejador. Durante a execução, as circunstâncias que aparecem podem somente determinar ramificações condicionais ou causar retorno do controle ao planejador se algo está obviamente errado.

Negociação A negociação representa um papel fundamental em atividades cooperativas dentro de sociedades humanas, permitindo que pessoas resolvam conflitos que possam interferir com comportamento cooperativo [MOU96].

O processo de negociação atua sobre o melhoramento de concordância acerca de “pontos de vista comuns ou planos através de compartilhamento de estruturas de informações relevantes” [MOU96], ocorrendo entre agentes com objetivos diferentes no qual “decisão conjunta é alcançada por dois ou mais agentes, cada um tentando alcançar seus objetivos individuais” [HUH99].

Os principais elementos utilizados pelos agentes envolvidos no processo de negociação são [HUH99]:

- *linguagem*;
- *protocolo*, que define a maneira através do qual eles negociam;
- *processo de decisão*, que determina suas posições, concessões, e critérios utilizados para acordos.

As abordagens de negociação utilizadas podem ser [HUH99]:

- *centradas no ambiente* - foca o problema de “como podem as regras do ambiente serem desenvolvidas para que os agentes nele envolvidos, independente de suas origens, capacidades, ou intenções, interajam produtivamente e razoavelmente?”. Neste caso, o mecanismo de negociação resultante deve possuir as seguintes propriedades:
 - *eficiência* - os agentes não devem desperdiçar recursos para chegar a um acordo;
 - *estabilidade* - nenhum agente deve ter um incentivo para desviar das estratégias acordadas;
 - *simplicidade* - deve impôr baixas demandas computacionais e de largura de banda⁷ sobre os agentes;
 - *distribuição* - não deve requerer um tomador de decisões centralizado;
 - *simetria* - não deve haver diferenciação no tratamento dos agentes por razões arbitrárias ou inapropriadas.

⁷Define-se largura de banda como sendo a capacidade de transmissão de um canal de comunicação.

- **centradas no agente** - assume que os agentes são economicamente racionais, o conjunto de agentes deve ser pequeno, necessitam ter uma linguagem e abstração do problema comuns, e têm que alcançar uma solução comum. A partir desses pressupostos, foi desenvolvido um protocolo de negociação unificado (vide [ROS94]), no qual agentes criam um *acordo*, que constitui um plano conjunto entre agentes que satisfarão todos os seus objetivos, cuja **utilidade** para um agente é a quantidade que ele pagará subtraindo o custo do acordo. Agentes discutem um **conjunto de negociações**, que é o conjunto de todos os acordos que têm uma utilidade positiva para todo agente. Do processo de negociação, três situações podem surgir:
 - **conflito** - o conjunto de negociações está vazio;
 - **compromisso** - agentes preferem trabalhar de maneira isolada, mas se isto não for possível, chegarão um acordo negociado;
 - **cooperativo** - todos acordos no conjunto de negociações são preferidos por ambos agentes prioritariamente à realização seus objetivos de forma isolada.

Uma das abordagens propostas mais utilizadas em SMA é o protocolo de **redes de contrato** [SMI81], que é inspirado nos processos de contratação existentes em organizações humanas. Neste processo, agentes coordenam suas ações através de contratos para cumprir seus objetivos específicos, onde existe um agente que atua como **gerente**, decompondo seus contratos em subcontratos a serem realizados por outros potenciais agentes **empreiteiros**. Da perspectiva do gerente, o processo consiste em [HUH99]:

- Anunciar uma tarefa que precisa ser executada;
- Receber e avaliar ofertas de empreiteiros em potencial;
- Concessão de um contrato para um empreiteiro apropriado;
- Receber e sintetizar resultados.

A partir da perspectiva do empreiteiro, o processo é:

- Recebe anúncios de tarefa;
- Avalia a própria capacidade de resposta;

- Responde (recusa, oferta);
- Executar a tarefa se a oferta enviada foi aceita;
- Enviar resultados ao gerente.

O protocolo de rede de contratos oferece a vantagem de degradação suave do desempenho. Se um empreiteiro não está apto a prover uma solução considerada satisfatória, o gerente pode procurar outro empreiteiro potencial para a tarefa.

Interação

A interação propicia a combinação de esforços entre um conjunto de agentes na busca de soluções para problemas globais, pressupondo ações de coordenação de ao menos dois agentes.

Os aspectos a serem considerados no processo de interação de agentes são [BON88]:

- entre quem a interação deve ocorrer;
- quando a interação deve ocorrer;
- qual deve ser o conteúdo da interação ou comunicação;
- como a interação deve ser realizada, definindo os processos e recursos a serem utilizados;
- porque a interação é necessária; e
- com base em que mecanismo será estabelecida a compreensão mútua (linguagem comum, interpretação baseada no contexto, etc).

A interação entre agentes pode ocorrer através de ações [MOU96]:

- lingüísticas explícitas (ou **comunicação**) [COH95]; ou
- não lingüísticas (através da percepção de alterações no ambiente de atuação).

Comunicação A capacidade de comunicação de um agente o habilita a intercambiare informações com outros agentes, sendo esta parte (i) *percepção* (recepção de mensagens) e (ii) *ação* (envio de mensagens). A comunicação pode permitir aos agentes coordenarem suas ações e comportamento, resultando em sistemas mais coerentes [HUH99].

A comunicação pode ser feita através de um *protocolo binário*, que envolve um único remetente e um único destinatário, ou de um *protocolo n-ário*, que envolve um único remetente e vários destinatários (por exemplo, *multicast* e *broadcast*), sendo que nos dois casos é especificado por uma estrutura de dados contendo elementos básicos [HUH99]:

- remetente;
- destinatário(s);
- linguagem;
- funções de codificação; e
- ações a serem tomadas pelo receptor.

Atos de Fala A comunicação falada humana, formalizada pela *teoria dos atos de fala* [AUS62, SEA70], têm sido utilizada como modelo para a comunicação entre agentes computacionais [COH95]. A teoria dos atos de fala visualiza a linguagem natural humana como ações, tais como requisições, sugestões, compromissos, e respostas, sendo possui três aspectos [HUH99]:

- *locução* - expressão física pelo locutor;
- *ilocução* - significado pretendido da expressão pelo locutor;
- *perlocução* - a ação que resulta da locução.

A intenção da mensagem presente em uma comunicação entre humanos nem sempre é de fácil identificação, porém devemos garantir que não haverá dúvidas sobre o tipo de mensagem em uma comunicação entre agentes computacionais [HUH99].

Blackboard Outra maneira de viabilizar a comunicação entre agentes é disponibilizar um repositório de dados compartilhados no qual informações podem ser postadas e recuperadas [MOU96] (vide também [HUH99]).

4.2 Aspectos Arquiteturais

Uma taxonomia pode ser utilizada para classificar as possíveis formas de dispor os relacionamentos existentes entre os agentes [SHE98]:

- **Hierárquica** - Dispõem os agentes em uma estrutura hierárquica na qual a comunicação ocorre de forma hierárquica. Nestes casos, cada agente pode comunicar-se apenas com os agentes por ele supervisionados ou por seu próprio supervisor. Este tipo de estrutura: (i) dispensa mecanismos para localização de agentes e (ii) reduz significativamente a quantidade de comunicação no sistema, porém (iii) apresenta uma estrutura que não permite que os agentes organizem-se dinamicamente para melhor atender as necessidades de uma tarefa específica (iv) onde os níveis mais baixos dependem dos níveis mais altos, e os níveis mais altos podem ter controle parcial ou total em relação aos níveis mais baixos.
- **Nivelada** - Implica que cada agente possa contactar diretamente qualquer dos outros agentes. Desta forma, podem criar estruturas de maneira dinâmica para realizar uma tarefa específica.
- **Agentes Compostos por Agentes** - Pressupõe a existência de alguns agentes que são componentes de outros agentes.
- **Sistemas Multiagentes Compostos por Sistemas Multiagentes** - Considera um SMA como sendo um agrupamento de vários módulos que, por sua vez, são vistos isoladamente como SMAs.

Propriedades Arquiteturais

É possível distinguir três atributos principais dos *protocolos de comunicação* que são relevantes para um SMA [SHE98]:

- **Simetria** - uma característica dos protocolos cliente/servidor é o fato de que eles implicam assimetria entre as entidades comunicativas, ou seja, um está no controle da comunicação enquanto a outra parte pode somente responder às questões requisitadas, e não pode iniciar a comunicação. Projetistas de SMA têm compreendido que tais assimetrias não são apropriadas para estes sistemas, e implementado significados simétricos de comunicação.

- **Recipientes de Mensagem** - mensagens em uma rede podem ser enviadas para um único endereço, múltiplos endereços (*multicast*) e para todos os endereços de um ambiente (*broadcast*). Em sistemas abertos, a utilização de *broadcast* facilita o envio de mensagens, porém o fato de se enviar mensagens irrelevantes à grande parte dos agentes pode incorrer em um tráfego desnecessário na rede e uma conseqüente perda de performance.
- **Tipo de Conexão** - em relação ao tipo de conexão a ser utilizada entre agentes de um SMA, um protocolo de comunicação pode ser classificado em orientado à conexão ou sem conexão. Uma comunicação orientada à conexão é preferida quando tarefas dependentes são executadas paralelamente por mais de um agente, e coordenação é necessária durante a execução.

Um SMA também pode ser classificado quanto ao seu *nível de abertura* em relação ao ingresso de novos agentes [SHE98]:

- **Abertura Dinâmica** - isto significa que agentes podem ingressar e sair do sistema à qualquer momento e sem a necessidade de adotar qualquer procedimento de notificação dos outros agentes. Esta característica permite uma maior flexibilidade de adaptação do sistema em relação a mudanças no ambiente. No entanto, serviços e computação adicionais são requeridos para suportar o fato de que agentes podem aparecer e desaparecer de forma imprevisível, sendo que um robusto mecanismo de localização de agentes torna-se necessário.
- **Abertura Estática** - ainda considera-se aberto, porém menos dinâmico, o caso onde agentes podem ingressar no sistema sem reiniciá-lo, mas todos os agentes são notificado de tal ingresso, ou todos eles mantêm uma lista de prováveis agentes a ingressar.
- **Abertura Off-Line** - o tipo mais restrito de abertura é a que permite o ingresso de novos agentes somente em off-line, através da desativação do sistema, adicionando agentes, atualizando algumas informações de conexão, e reiniciando o sistema.

Alguns SMA necessitam de *serviços de infraestrutura* para que possam funcionar corretamente. Estes serviços podem ser [SHE98] (veja também *Agent Directory Service* [STE96]):

- *serviço de nomes de agente*, que evita que dois agentes possuam o mesmo nome, evitando confusões em consequência disto.
- *serviço de localização de agente*, que é necessário quando a existência ou disponibilidade dos agentes não são de conhecimento comum. Este tipo de serviço (por exemplo, *brokering* ou *matchmaking* [FON97, KUO95]) é, algumas vezes implementado de forma centralizada, o que facilita a implementação e a manutenção da base de localização, porém o sistema torna-se mais vulnerável e cria um único ponto de falha. De outra forma, mecanismos de localização distribuídos são mais complicados de se projetar, implementar e manter, porém mais confiável e robusto.
- *serviço de segurança*, que é mais adequado à SMA aberto quando não se tem certeza sobre a verdadeira identidade e a confiabilidade de outros agentes. Nestes casos, mecanismos de segurança podem reduzir os riscos derivados desta incerteza. Estes serviços podem introduzir o conceito de *autoridades de certificação* eletrônicos, que são emissores de certificados de autenticidade tidos como confiáveis. Isto gera, inevitavelmente, um *overhead* de computação e de comunicação, podendo representar um “gargalo” do sistema.
- *serviços de mobilidade*, que é requerido por alguns sistemas para viabilizar o funcionamento de agentes móveis [GRA00]. A maneira mais comum de prover este serviço é através de servidores de mobilidade (também chamados *agent dock*), que permanecem em execução nas máquinas onde é permitida a chegada de agentes móveis.

Um aspecto importante a ser considerado é a *robustez do sistema*, onde a falha de um agente não necessariamente implica em uma falha de todo o SMA. A robustez é viabilizada através de capacidades replicadas, ou seja, existem múltiplos agentes com as mesmas capacidades ou capacidades similares, o que permite que agentes possam assumir responsabilidades de outros agentes que estão indisponíveis. A desvantagem desta replicação reside na redundância resultante [SHE98].

Agentes Federados Genesereth [GEN94] apresenta uma organização de sistema de agentes federados (vide figura 4), que consiste de agentes e *facilitadores* como um meio de atingir a interoperabilidade entre estes. Uma organização federada sugere que agentes

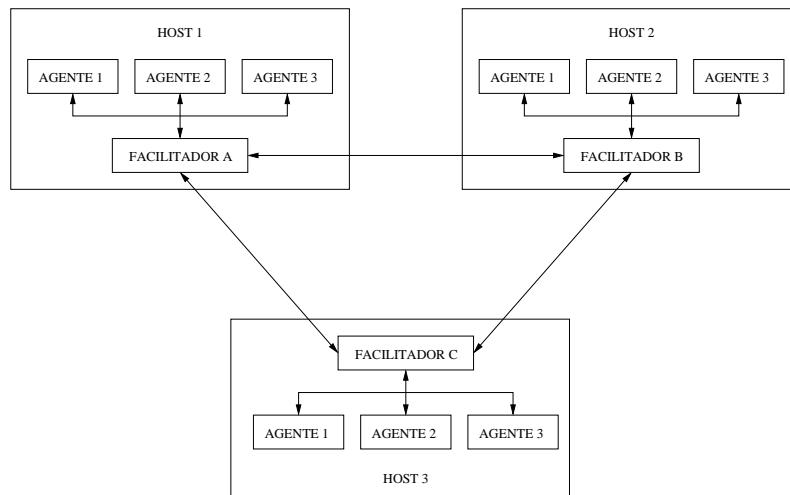


Figura 4: Organização Multiagente Federada [GEN94].

não comunicam-se diretamente com outros, mas cada agente comunica-se com seu facilitador local que, por sua vez, comunica-se com outros facilitadores. Desta forma, cada grupo de agentes ligado a um único facilitador é uma *federação*, onde um agente rende parte de sua autonomia ao facilitador.

Esta abordagem pode ser utilizada para agrupar os agentes nos diversos processadores de um ambiente distribuído. Desta maneira, cada federação corresponderia a um processador onde o respectivo facilitador teria controle sobre a conexão de rede.

4.3 Aspectos Ambientais

Como visto na definição de Wooldridge [WOO95], um agente está situado em algum ambiente. D'Amico e outros [D'AM95] entendem que o ambiente pode ser visto como sendo tudo o que envolve um agente. Assim sendo, o ambiente constitui o contexto onde todas as interações entre os agentes ocorrem. Através do ambiente ocorre a dispersão do controle, dos dados e do conhecimento pela comunidade de agentes.

“Uma organização garante a sua sobrevivência a partir de sua capacidade de aprendizado e de ajuste às características e mudanças no mundo⁸” [BAS98].

Portanto, é imprescindível que os agentes de uma organização possuam conhecimento sobre as alterações ocorridas no ambiente, pois a ausência desta capacidade inviabiliza a

⁸Mundo refere-se à descrição completa e instantânea do ambiente em que um agente encontra-se.

adaptação à nova realidade.

No entanto, um fator importante a considerar no projeto de um Sistema Multiagente é o tipo de ambiente no qual seus agentes estarão situados, que determina qual o tipo de representação de ambiente sobre a qual cada um dos agentes deverá trabalhar, bem como a maneira de atuar e de perceber as alterações no ambiente .

As principais classificações a serem feitas em um ambiente são [RUS95]:

- ***Acessível*** × ***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*** × ***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*** × ***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*** × ***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*** × ***Contínuo*** - é chamado discreto o ambiente que tem um número limitado de percepções e ações distintas e claramente definidas.

5 Técnicas para Desenvolvimento de SMA

Sob o ponto de vista da Engenharia de Software, a construção de software de alta qualidade de maneira produtiva é viabilizada por um conjunto de métodos, ferramentas e procedimentos, sendo que o caminho para a evolução no desenvolvimento de software passa por uma combinação de métodos abrangentes para todas as etapas de desenvolvimento

do software, melhores ferramentas para automatizar estes métodos, blocos de construção mais poderosos para a implementação do software, melhores técnicas para a garantia da qualidade do software e uma filosofia de coordenação predominante, controle e administração [PRE95] (vide também [DAV85]).

Considera-se que o processo de produção de software engloba três etapas [MAR95, SHL90, RUM97, COA90]:

- **análise** constitui a etapa de modelagem do problema para uma representação padronizada (geralmente diagramas), sendo o resultado do processo de levantamento de requisitos funcionais e de aspectos relacionados.
- **projeto** caracteriza os procedimentos necessários à adequação do modelo de análise às características existentes no ambiente computacional disponível para implementação da solução, considerando as limitações de recursos impostas pelo mesmo.
- **implementação** envolve procedimentos de mapeamento do modelo de projeto para uma solução física.

Inserido neste ponto de vista, é necessário que, a exemplo do paradigma Orientado a Objetos (OO) [MAR95, SHL90, RUM97, COA90, COA91, BOO83], o paradigma Orientado a Agentes (OA) seja também suportado por técnicas e métodos que garantam a qualidade do processo de produção do software.

Inúmeros trabalhos têm apresentado conceitualizações, formalizações, técnicas e metodologias [JEN99, WOO00, IGL98a, KEN96, TAV99, ODE00, BAS00], porém, de acordo com o estudo de Bastos [BAS00], apesar da existência de diversos trabalhos aplicando a abordagem multiagente para a modelagem de Sistemas de Informação, não existe ainda um consenso sobre metodologias para modelar este tipo de sistema.

5.1 Projeto OA e Projeto OO

A distinção entre os requisitos envolvidos no projeto de um SMA e um Sistema OO é evidenciada pelas próprias diferenças existentes entre agentes e objetos.

Segundo a visão de Amandi [AMA97], o conceito de agente apresenta semelhanças estruturais em relação ao conceito de objeto. 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.

Segundo Shoham [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.

Entretanto, 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 (i) encapsulam seus estados, (ii) são aptas a desempenharem ações considerando estes estados, e (iii) comunicam-se através da passagem de mensagens. Por outro lado, existem distinções significativas entre agentes e objetos:

- **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 pelo slogan “objetos fazem gratuitamente; agentes fazem por dinheiro” [WOO99]. É preciso observar que nada impede a implementação da característica de autonomia em agentes utilizando-se técnicas OO, porém é necessário notar que a característica de autonomia não é um componente básico do paradigma OO.
- Agentes apresentam um **comportamento autônomo flexível (reativo, pró-ativo, social)**. Os modelos existentes no paradigma OO não especificam nenhuma maneira de representar estes tipos de comportamento, embora isto possa ser implementado utilizando-se técnicas OO.
- Quanto inserido em um SMA, **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 Odell [ODE99], a invocação de operações ocorre de forma distinta entre objetos e agentes. A invocação de operações em objetos ocorre de forma externa, ou seja, um

objeto executa uma operação em resposta à uma mensagem recebida. Uma mensagem enviada a um agente passa por um mecanismo de deliberação interno ao agente que utiliza regras e objetivos para decidir qual o tratamento adequado àquela mensagem (executar ações, ignorar mensagem, alterar objetivos, etc.).

Estudos foram feitos com o objetivo de verificar a existência de um paralelo entre os paradigmas OO e OA [AMA97, JEN99, ODE99] que permita a aplicação das metodologias existentes para OO para construção de um SMA. Segundo Bastos [BAS98], é possível modelar um agente utilizando uma abordagem OO. No entanto, persiste a problemática da modelagem da sociedade, pois, por não adequar-se ao modelo conceitual, a abordagem OO não viabiliza a modelagem de todos os aspectos envolvidos em um SMA [IGL98a, WOO99a, KEN96, TAV99].

Os trabalhos existentes na literatura que abordam o processo de engenharia de um SMA [IGL98a, TAV99, BAS00] restringem sua pesquisa à fase de análise, com excessão de [WOO00, IGL98], que também abordam a fase de projeto, porém não são suficientemente detalhados, além de não apresentarem uma discussão acerca dos requisitos de atividades de projeto de SMA.

A UML [UML99] é uma linguagem de modelagem desenvolvida para dar suporte ao desenvolvimento OO, sendo reconhecida como padrão pela OMG e por grande parte dos engenheiros de software. O trabalho desenvolvido por Odell na definição da AUML (*Agent Unified Modelling Technique*) [ODE00] (vide item 5.2) constitui um esforço de adaptação da linguagem UML às características inerentes ao paradigma OA. Porém, por tratar-se de uma proposta bastante recente, é natural que não exista consenso sobre a sua real adequação para modelagem de SMA.

5.2 Propostas para Modelagem de SMA

A seguir faremos uma revisão bibliográfica acerca das propostas de modelagem de SMA existentes.

Metodologia Gaia [WOO99a, WOO00]

O processo de projeto Orientado a Agentes (OA) proposto por Wooldridge e outros [WOO00], denominado *Metodologia Gaia*, difere dos processos tradicionais, que visam transformar um modelo derivado do processo de análise em modelos de mais baixo nível de abstração

que possam ser implementados, objetivando derivar um modelo de análise em um modelo com baixo nível de abstração sobre o qual possam ser aplicadas técnicas de projeto tradicionais (incluindo técnicas OO).

Esta proposta apresenta dois modelos para a fase de análise:

- **Modelo de Papéis** - identifica os papéis existentes no sistema, sendo estes descrições abstratas de uma função pretendida por uma entidade. Este modelo descreve dois elementos básicos: (i) permissões/direitos associados ao papel, que relacionam o tipo e a quantidade de recursos que podem ser explorados; e (ii) responsabilidades do papel, que determinam a funcionalidade do agente dentro da sociedade⁹.
- **Modelo de Interação** - é constituído de um conjunto de definições de protocolo¹⁰, um para cada tipo de interação entre papéis. As definições de protocolo consistem da definição dos seguintes atributos: (i) propósito, que é uma breve descrição da natureza da interação; (ii) iniciador, que é o papel responsável por iniciar a interação; (iii) respondedor¹¹, papel com o qual o iniciador interage; (iv) entradas, que é a informação utilizada pelo iniciador enquanto executando o protocolo; e (v) saídas, que é informação fornecida pelo/para o respondedor durante o curso de interação.

Desta forma, o que de fato se almeja é garantir que aspectos como estrutura, organização, cooperação e interação sejam definidos no nível de modelagem. Estes aspectos dependeriam da forma pela qual a rede de relações entre os agentes estaria estabelecida. Devido à autonomia ser uma característica intrínseca ao próprio conceito de agente, a forma pela qual as interações se processam dentro do sistema depende fundamentalmente das atitudes tomadas por cada um dos agentes que o compõem, sendo que a garantia da coesão do grupo depende fundamentalmente dos objetivos adotados por cada um de seus indivíduos, que refletem seus desejos e crenças.

A partir disso, se a etapa de projeto do sistema tem por objetivo viabilizar a aplicação de diretrizes definidas durante o processo de modelagem, projetar um Sistema Multiagente consiste em definir quais tipos de agentes o comporão, o que passa pela definição de estereótipos (ou papéis) para cada um deles.

⁹A funcionalidade do papel de um agente na sociedade é definida por suas responsabilidades. [WOO99a]

¹⁰Protocolo é um padrão de interação institucionalizado.

¹¹Tradução livre para a palavra da língua inglesa *responder*, já que não achou-se outro termo para este contexto.

Na pesquisa de Wooldridge e outros [WOO99a], estes estereótipos são definidos por três modelos:

- **Modelo de Agente** - define os *tipos de agente* que irão fazer parte do sistema, e as *instâncias de agente* que serão instanciadas a partir destes;
- **Modelo de Serviços** - define os principais serviços que serão atribuídos a cada tipo de agente;
- **Modelo de Conhecimento** - define conhecimentos para cada tipo de agente.

Metodologia MAS-CommonKADS [IGL98]

A metodologia MAS-CommonKADS [IGL98] é uma extensão multiagente da principal metodologia estruturada de suporte à engenharia do conhecimento, chamada CommonKADS, e propõe os seguintes modelos:

- **Modelo do Agente (AM)** - especifica as características do agente, tais como capacidade de raciocínio, percepção e ação, serviços, grupos e hierarquias do agente (modeladas no modelo organizacional).
- **Modelo de Tarefas (TM)** - descreve as tarefas que o agente pode realizar.
- **Modelo de Habilidades (EM)** - descreve o conhecimento que os agentes necessitam para atingir os seus objetivos.
- **Modelo de Organização (OM)** - descreve a organização na qual o SMA está sendo introduzido e a organização social da sociedade de agentes.
- **Modelo de Coordenação (CoM)** - descreve as conversações entre agentes (interações, protocolos e capacidades requeridas).
- **Modelo de Comunicação (CM)** - descreve detalhes das interações entre humanos e agentes de software, e fatores humanos a serem considerados no desenvolvimento de interfaces de usuário.
- **Modelo de Projeto (DM)** - coleta os modelos anteriores e consiste de três submodelos:

- *Projeto da Rede* - descreve aspectos relevantes da infraestrutura da rede de agentes. Esta infraestrutura pode fornecer algumas facilidades para os agentes:
 - * *Facilidades de Rede* - serviço de nomes de agente, páginas amarelas (yellow pages), serviço de registro, nível de segurança, encriptação e autenticação, protocolo de transporte e aplicação, serviço de contas, etc.
 - * *Facilidades de Conhecimento* - servidores de ontologia, tradutores de linguagem de representação de conhecimento, etc.
 - * *Facilidades de Coordenação* - protocolos e primitivas de coordenação, servidores de protocolo, facilidades de gerenciamento de grupo, facilidades para assistência em coordenação de objetivos compartilhados, agentes de policiamento para detecção de comportamentos indesejados e controle de utilização de recursos compartilhados, etc.
- *Projeto do Agente* - divide ou compõe agentes de análise, de acordo com critérios pragmáticos, e seleciona a arquitetura mais adequada para cada agente.
- *Projeto da Plataforma* - seleciona a plataforma de desenvolvimento de agentes para cada arquitetura.

Proposta de Kendall [KEN00]

O trabalho de Kendall propõe um *modelo de papéis* que identifica e descreve uma estrutura de arquétipo ou recorrente de entidades utilizando *papéis*. Um papel [BID79] define uma posição e um conjunto de responsabilidades dentro de um modelo de papéis, tendo colaboradores que são os outros papéis que interagem com este. Um papel possui um conjunto de serviços e atividades que são disponibilizados através de uma interface externa.

A autora apresenta modelos de papéis como sendo uma extensão dos diagramas de colaboração utilizados em UML [UML99]. Vários aspectos precisam ser considerados em papéis de agente além dos aspectos representados em papéis de objetos:

- *responsabilidades* - objetivos, obrigações e interdições.
- *especialidade* - ontologia, inferência, conhecimento para resolução de problemas.
- *coordenação e negociação* - protocolos, resolução de conflitos, conhecimento sobre porque está relacionado com outros papéis, permissões.

- *outros* - recursos, aprendizado/adaptabilidade.

Uma vez tendo modelos de papéis básicos, pode-se construir novos modelos. Um modelo de papéis pode ser construído a partir da agregação de outros. Da mesma forma, um novo modelo de papéis pode ser obtido a partir de um ou mais modelos básicos, refinando-os.

Proposta de Bastos, Oliveira e Oliveira [BAS00] [BAS98]

Esta proposta considera a definição de um conjunto de modelos de representação e atividades para o desenvolvimento de SMAs.

Definição de Requisitos Uma primeira atividade seria a definição dos requisitos do sistema considerando o domínio de aplicação. Esta atividade é abordada como uma fase de obtenção de conhecimentos acerca do domínio do problema, na qual são identificadas restrições, características, e resultados esperados.

Modelagem da Solução usando Multiagentes Em uma segunda etapa, seria realizada a definição de uma solução através da utilização do paradigma OA. Isto é feito através de um processo de identificação e derivação dos agentes que compõe a sociedade, que pode ser facilitado através do uso de um modelo de referência OO que represente o domínio do problema, pois considera-se que agentes e objetos possuem características comuns [WOO99]. Assim, pretende-se que, em um primeiro momento, as classes de objeto deste modelo tornem-se classes de agentes.

Modelagem da Sociedade de Agentes A terceira etapa seria a modelagem da sociedade de agentes, cujo objetivo é representar a perspectiva da sociedade, na qual seriam identificadas as relações de interação existentes entre os agentes. O diagrama utilizado para este propósito é uma extensão do diagrama de objetos proposto pela UML [UML99]. Este diagrama, chamado *Diagrama da Sociedade*, cada retângulo representa uma classe de agentes e os arcos representam os canais de comunicação entre elas. Nesta etapa, é definida a base de conhecimento de cada classe de agente, que é composta por (i) *conhecimentos herdados* da classe de objeto que a originou, e por (ii) *conhecimentos específicos* que compreendem os conhecimentos incorporados pela classe de agentes que habilitam esta a executar suas atribuições.

Um segundo modelo utilizado para a modelagem da sociedade é o **Modelo de Funções**, que define os papéis e a base de conhecimento de cada classe de agente. Um papel define atribuições e restrições que devem ser observadas pelo agente, sendo que as atribuições estabelecem todas as capacidades funcionais do agente.

O terceiro modelo, chamado **Modelo de Interação**, representa o fluxo de interação existente entre os agentes, sendo composto por um ou mais diagramas de interação, que representam o comportamento dos agentes de acordo com o seu papel na sociedade.

Modelagem dos Agentes A última atividade deste processo é a modelagem de cada agente da sociedade e a definição de sua respectiva arquitetura (vide [ROZ00]).

Agent Unified Modelling Language [ODE00]

Os diagramas de UML (*Unified Modelling Language*) [UML99, RUM99] foram inicialmente projetados para suportar a descrição dos vários aspectos inerentes a sistemas OO. Na visão de [ODE00], um agente é uma extensão do conceito de objeto, o que permite explorar extensões de linguagens de modelagem OO com o objetivo de adaptá-las às características do paradigma OA. A partir disso, o autor propõe extensões à UML, descrevendo os requisitos mais comuns para modelagem de agentes e SMAs, criando assim a AUML (*Agent UML*).

A proposta inicial da AUML é representar protocolos de interação de agentes¹². Nos diagramas da AUML, os fluxos de controle e informação são considerados **atos de comunicação** (mensagens). O protocolo como um todo é tratado como sendo uma entidade, colocando-se um diagrama que represente o fluxo de mensagens entre os agentes (diagramas de seqüência, de colaboração, de atividade e de estados) dentro de um pacote¹³, constituindo uma agregação conceitual de seqüências de interação. A partir disso, o protocolo pode ser tratado como um padrão generalizado para problemas de domínio análogos através da especificação de parâmetros de entidade dentro do pacote¹⁴, que são particularizados quando o pacote é instanciado. Isto fornece soluções reutilizáveis que podem ser aplicadas para várias situações onde um mesmo tipo de protocolo é requerido.

¹²AIP (*Agent Interaction Protocol*) descreve um padrão de comunicação como sendo uma seqüência de mensagens permitida entre agentes e as restrições sobre o conteúdo destas mensagens.

¹³Representados através de *packages* da UML.

¹⁴Representados através de *templates* de pacotes da UML.

No caso dos *diagramas de interação*, são representados padrões estruturais de interações entre os agentes. O *diagramas de seqüência* e o *diagrama de colaboração* são subtipos do diagrama de interação, sendo semanticamente equivalentes. A disposição dos elementos gráficos do diagrama de seqüência enfatiza a seqüência cronológica de comunicações, enquanto no diagrama de colaboração são enfatizadas as associações entre agentes, onde a seqüência das interações é representada através da numeração das mensagens.

O *diagrama de atividade* e o *diagrama de estados* dão ênfase ao fluxo de processamento. O diagrama de atividade representa operações e os eventos que ativam estas, diferindo dos diagramas de interação por representar de maneira explícita as linhas de execução dos fluxos de controle, o que é particularmente útil para protocolos de interação complexos envolvendo paralelismo de processamento. Diagramas de estado não são normalmente utilizados para expressar protocolos de interação, pois têm uma visão centrada nos entados, ao invés de uma visão centrada no agente ou no processo. Este tipo de diagrama é melhor aplicável como um mecanismo de restrições para o protocolo, sendo incorporado aos agentes para que as restrições sejam conhecidas.

6 Considerações Finais

A partir deste trabalho foi possível conhecer, em linhas gerais, os requisitos envolvidos no processo de Engenharia de Sistemas Multiagentes, através do estudo dos vários aspectos relacionados ao paradigma OA. Este estudo permitirá que trabalhos futuros discorram sobre quais características necessitarão ser representadas na etapa de projeto do sistema, viabilizando a criação de modelos que representem todos os aspectos envolvidos em um SMA.

Também foi realizada uma investigação acerca das propostas para modelagem de SMA existentes na literatura, o que permitiu que fossem identificados pontos comuns em relação às características a serem representadas nos modelos:

- organização social;
- distribuição de responsabilidades (serviços, tarefas e objetivos) para cada um dos papéis;
- protocolos de interação;

- conhecimento dos agentes; e
- infraestrutura de rede, no caso da etapa de projeto.

Uma questão importante a ser considerada no nível de projeto, e não resolvida em nenhum dos trabalhos, é a maneira através da qual far-se-á com que os agentes incorporem conhecimentos sobre a estrutura da sociedade na qual coexistem para que possam agir de acordo com objetivos, restrições e padronizações estabelecidas por ela.

Este estudo representa um esforço em direção à pesquisa objeto da dissertação de mestrado, cujo objetivo será identificar as propriedades a serem consideradas para o projeto de um SMA no nível das interações entre os agentes que compõem a sociedade, propondo atividades a serem executadas durante a etapa de projeto e uma linguagem para representação dos artefatos de projeto.

Referências

- [AGR89] AGRE, Philip E.; CHAPMAN, David. **What are plans for?** [S.l.: s.n.], 1989. n.AI-MEMO 1050a.
- [AMA97] AMANDI, A. A. **Programação de agentes orientada a objetos.** Porto Alegre: CPGCC da UFRGS, 1997. Tese de Doutorado.
- [AUS62] AUSTIN, John L. **How to do Things with Words.** Clarendon, Oxford, UK: [s.n.], 1962.
- [BAS98] BASTOS, R. M. **O Planejamento de Alocação de Recursos Baseado em Sistema Multi-Agentes.** Porto Alegre: CPGCC da UFRGS, 1998. Tese de Doutorado.
- [BAS00] BASTOS, Ricardo Melo; OLIVEIRA, José Palazzo Moreira de. A conceptual modeling framework for multi-agent information systems. In: PROCEEDINGS OF THE XIX INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING, 2000, Salt Lake City, USA. **Anais...** 2000. v.1920.
- [BID79] BIDDLE, B. J.; THOMAS, J. **Role Theory: Concepts and Research.** New York: R. E. Krieger Publishing Co., 1979.

- [BON88] BOND, A. H.; GASSER, L. **Readings in distributed artificial intelligence**. San Mateo: Morgan Kaufmann, 1988.
- [BOO83] BOOCH, G. Object-oriented design. In: FREEMAN, P.; WASSERMAN, A. I. (Eds.). **Tutorial: software design techniques**. [S.l.]: IEEE Comp. Soc. Press, 1983. p.420–436.
- [COA90] COAD, P.; YOURDON, E. **Object-Oriented Analysis**. Englewood Cliffs, NJ, USA: Yourdon Press, 1990.
- [COA91] COAD, Peter; YOURDON, Edward. **Object-oriented design**. Englewood Cliffs , NJ , USA: Prentice-Hall, 1991.
- [COH95] COHEN, Philip R.; LEVESQUE, Hector J. Communicative actions for artificial agents. In: PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, 1995, San Francisco, CA. **Anais...** MIT Press, 1995. p.65–72.
- [COR94] CORREA, M. A. Filho. **Arquitetura de diálogos entre agentes cognitivos distribuídos**. Rio de Janeiro: COPPE da UFRJ, 1994. Tese de Doutorado.
- [D'AM95] D'AMICO, C. **Inteligência artificial: uma abordagem de agentes**. Porto Alegre: CPGCC da UFRGS, 1995.
- [DAV85] DAVID, J. Smith; KENNETH, B. Wood. **Engineering quality software**. New York: Elsevier Science Publishers Ltda., 1985.
- [DUR96] DURFEE, Edmund H. Planning in distributed artificial intelligence. In: O'HARE, Greg; JENNINGS, Nick (Eds.). **Foundations of distributed artificial intelligence**. [S.l.]: John Wiley and Sons, 1996. cap.8.
- [FON97] FONER, L.; CRABTREE, I. B. Multi-agent matchmaking. **Lecture Notes in Computer Science**, v.1198, p.100–??, 1997.
- [GEN94] GENESERETH, Michael R.; KETCHPEL, Steven P. Software agents. **Communications of the ACM**, v.37, n.7, p.48–53, jul 1994.
- [GRA00] GRAY, Robert S.; KOTZ, David; CYBENKO, George; RUS, Daniela. **Mobile Agents: Motivations and State-of-the-Art Systems**. Hanover, NH: [s.n.], 2000. n.TR2000-365.

- [HUH99] HUHNS, Michael N.; STEPHENS, Larry M. Multiagent Systems and Societies of Agents. In: WEISS, Gerhard (Ed.). **Multiagent Systems - A Modern Approach**. [S.l.]: MIT Press, 1999. cap.2.
- [IGL98] IGLESIAS, C. A.; GARIJO, M.; GONZALEZ, J. C.; VELASCO, J. R. Analysis and design of multiagent systems using MAS-CommonKADS. **Lecture Notes in Computer Science**, v.1365, p.313–??, 1998.
- [IGL98a] IGLESIAS, C. A.; GARIJO, M.; GONZALEZ, J. C. A Survey of Agent-Oriented Methodology. In: PROCEEDINGS OF THE V WORKSHOP ATAL'98, 1998. **Anais...** 1998. p.317–330.
- [JEN98] JENNINGS, N. R.; WOOLDRIDGE, M. J. Applications of intelligent agents. In: JENNINGS, N. R.; WOOLDRIDGE, Michael J. (Eds.). **Agent technology: foundations, applications, and markets**. Heidelberg, Germany: Springer-Verlag, 1998. p.3–28.
- [JEN99] JENNINGS, N. R. Agent-Oriented Software Engineering. In: PROCEEDINGS OF THE 9TH EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD : MULTI-AGENT SYSTEM ENGINEERING (MAAMAW-99), 1999, Heidelberg, Germany. **Anais...** Springer-Verlag, 1999. v.1647, p.1–7.
- [JEN96] JENNINGS, Nicholas R.; FARATIN, P.; JOHNSON, M. J.; O'BRIEN, P.; WIEGAND, M. E. Using intelligent agents to manage business processes. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON PRACTICAL APPLICATIONS OF INTELLIGENT AGENTS AND MULTIA-AGENT TECHNOLOGY (PAAM-96), 1996, London, UK. **Anais...** 1996. p.345–360.
- [JEN96a] JENNINGS, Nicholas R. Coordination Techniques for DAI. In: O'HARE, Greg; JENNINGS, Nicholas (Eds.). **Foundations of distributed artificial intelligence**. [S.l.]: John Wiley and Sons, 1996. cap.6.
- [KEN96] KENDALL, E. A.; MALKOUN, M. T.; JIANG, Chong. A Methodology for Developing Agent-Based Systems. In: PROCEEDINGS OF THE FIRST

- AUSTRALIAN WORKSHOP ON DAI, 1996. **Anais...** Springer-Verlag: Heidelberg, Germany, 1996.
- [KEN00] KENDALL, Elizabeth A. Role modeling for agent system analysis, design, and implementation. **IEEE Concurrency**, v.8, n.2, p.34–??, apr 2000.
- [KUO95] KUOKKA, Daniel; HARADA, Larry. Matchmaking for information agents. In: PROCEEDINGS OF THE FOURTEENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1995, San Mateo. **Anais...** Morgan Kaufmann, 1995. p.672–678.
- [LUX95] LUX, A.; STEINER, D. Understanding cooperation: an agent’s perspective. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, 1995, Menlo Park, San Francisco. **Anais...** AAAI Press, 1995. p.261–268.
- [MAR95] MARTIN, James; ODELL, James J. **Análise e projeto orientados a objeto**. São Paulo: Makron Books Editora do Brasil Ltda., 1995.
- [MOF95] MOFFAT, D.; FRIJDA, N. Where there’s a will there’s an agent. **Lecture Notes in Computer Science**, v.890, p.245–??, 1995.
- [MOU96] MOULIN, Bernard; CHAIB-DRAA, Brahim. An Overview of Distributed Artificial Intelligence. In: O’HARE, Greg; JENNINGS, Nicholas R. (Eds.). **Foundations of distributed artificial intelligence**. [S.l.]: John Wiley and Sons, 1996. cap.1.
- [ODE00] ODELL, James; PARUNAK, H. Van Dyke; BAUER, Bernhard. Extending UML for agents. In: PROCEEDINGS OF THE AGENT-ORIENTED INFORMATION SYSTEMS WORKSHOP AT THE 17TH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI 2000), 2000. **Anais...** 2000.
- [ODE99] ODELL, James. **Objects and agents - how do they differ? (draft 2.2)**. Disponível por WWW em <http://www.jamesodell.com> (sep 1999).
- [O’HA96] O’HARE, Greg; JENNINGS, Nicholas R. **Foundations of distributed artificial intelligence**. [S.l.]: John Wiley and Sons, 1996.

- [OLI96] OLIVEIRA, F. M. Inteligência Artificial Distribuída. In: **IV Escola Regional de Informática**. Canoas, Brasil: Sociedade Brasileira de Computação, 1996. p.239.
- [PRE95] PRESSMAN, Roger S. **Engenharia de software**. São Paulo: Makron Books, 1995.
- [ROS94] ROSENSCHEIN, Jeffrey S.; ZLOTKIN, Gilad. Designing conventions for automated negotiation. **AI Magazine**, p.29–46, 1994.
- [ROZ00] ROZA, Marcelo P. **Uma proposta de extensão da metodologia M-DRAP com utilização de estados mentais**. Porto Alegre: PPGCC da PUCRS, 2000. Relatório Seminário de Andamento.
- [RUM97] RUMBAUGH, J. **Modelagem e projeto baseados em objetos**. Rio de Janeiro: Editora Campus, 1997.
- [RUM99] RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. **The unified modeling language reference manual**. Workingham, Inglaterra: Addison Wesley Longman, 1999.
- [RUS95] RUSSELL, S. J.; NORVIG, P. **Artificial intelligence. A modern approach**. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [SEA70] SEARLE, John R. **Speech Acts: An Essay in the Philosophy of Language**. Cambridge, MA: U. Press, 1970.
- [SEG96] SEGHTROUCHNI, Amal El Fallah. Rational agent cooperation through concurrent plan coordination. In: PROCEEDINGS OF THE IBEROAMERICAN WORKSHOP ON ARTIFICIAL INTELLIGENCE AND MULTI-AGENT SYSTEMS, 1996, Mexico. **Anais...** Lania, Mia Universidad Veracruzana, 1996. p.162–171.
- [SHE98] SHEHORY, Onn. **Architectural properties of multiagent systems**. Pittsburgh, PA: [s.n.], 1998. n.CMU-RI-TR-98-28.
- [SHL90] SHLAER, Sally; MELLOR, Stephen J. **Análise de sistemas orientada para objetos**. São Paulo: McGraw-Hill Ltda., 1990.

- [SHO93] SHOHAM, Yoav. Agent-oriented programming. **Artificial Intelligence**, v.60, p.51–92, mar 1993.
- [SMI81] SMITH, Reid G.; DAVIS, R. Distributed problem solving: the contract net approach. **IEEE trans. on systems, man and cybernetics**, v.SMC-11, n.1, jan 1981.
- [STE96] STEINER, D. D. IMAGINE: An integrated environment for constructing distributed artificial intelligence systems. In: O’HARE, Greg; JENNINGS, Nick (Eds.). **Foundations of distributed artificial intelligence**. [S.l.]: John Wiley and Sons, 1996. cap.13.
- [TAV99] TAVETER, K. Business Rules’ Approach to the Modelling, Design, and Implementation of Agent-Oriented Information Systems. In: PROCEEDINGS OF THE INTERNATIONAL BI-CONFERENCE WORKSHOP ON AGENT-ORIENTED INFORMATION SYSTEMS (AOIS’99), 1999. **Anais...** Springer-Verlag: Heidelberg, Germany, 1999.
- [UML99] UML 1.3 Notation Guide. Disponível por WWW em <http://www.rational.com/uml> (jun 1999).
- [WOO99] WOOLDRIDGE, M. Intelligent Agents. In: WEISS, Gerhard (Ed.). **Multi-agent Systems - A Modern Approach**. [S.l.]: MIT Press, 1999. cap.1.
- [WOO99a] WOOLDRIDGE, Michael; JENNINGS, Nicholas R.; KINNY, David. A methodology for agent-oriented analysis and design. In: PROCEEDINGS OF THE THIRD ANNUAL CONFERENCE ON AUTONOMOUS AGENTS (AGENTS-99), 1999, New York. **Anais...** ACM Press, 1999. p.69–76.
- [WOO00] WOOLDRIDGE, Michael; JENNINGS, Nicholas R.; KINNY, David. The Gaia Methodology for Agent-Oriented Analysis and Design. **Journal of Autonomous Agents and Multi-Agent Systems**, v.3, n.2000, 2000.
- [WOO95] WOOLDRIDGE, Michael; JENNINGS, Nicholas R. Intelligent Agents: Theory and Practice. **The Knowledge Engineering Review**, v.10, n.2, p.115–152, 1995.

[ZAM00] ZAMBONELLI, F.; JENNINGS, N. R.; WOOLDRIDGE, M. Organisational Abstractions for the Analysis and Design of Multi-Agent Systems. In: PROCEEDINGS OF THE 1ST INTERNATIONAL WORKSHOP ON AGENT-ORIENTED SOFTWARE ENGINEERING, 2000, Limerick, Ireland. **Anais...** 2000. p.127–141.