

**Universidade Federal de Santa Catarina**

**Programa de Pós-Graduação em  
Engenharia de Produção**

**UTILIZAÇÃO DA AGENT UNIFIED  
MODELING LANGUAGE NA  
MODELAGEM DA ESTRUTURA  
E ITERAÇÃO ENTRE AGENTES**

**José Antônio da Cunha**

Dissertação apresentada ao  
Programa de Pós-Graduação em  
Engenharia de Produção da  
Universidade Federal de Santa Catarina  
como requisito parcial para obtenção  
do título de Mestre em  
Engenharia de Produção.



03387571

**Florianópolis  
2001**

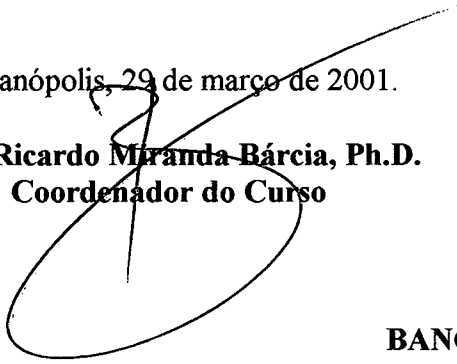
**José Antônio da Cunha**

**UTILIZAÇÃO DA AGENT UNIFIED  
MODELING LANGUAGE NA  
MODELAGEM DA ESTRUTURA  
E ITERAÇÃO ENTRE AGENTES**

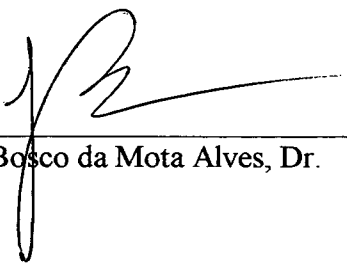
Esta dissertação foi julgada e aprovada para a  
obtenção do título de **Mestre em Engenharia de  
Produção no Programa de Pós-Graduação em  
Engenharia de Produção da  
Universidade Federal de Santa Catarina**

Florianópolis, 29 de março de 2001.

**Prof. Ricardo Miranda Bácia, Ph.D.**  
Coordenador do Curso




**BANCA EXAMINADORA**




---

Prof. João Bosco da Mota Alves, Dr.



---

Prof. Fernando Alvaro Ostuni Gauthier, Dr.  
**Orientador**



---

Prof. Luiz Fernando Jacinto Maia, Dr.

A minha esposa, Eudocia  
pelo apoio constante.  
A minha filha Larissa.  
Em homenagem à adorável memória de meu pai.

## Agradecimentos

Ao centro Federal de Educação Tecnológica  
do Rio Grande do Norte – CEFET-RN.  
Ao pessoal do Laboratório de Ensino a Distância/LED.  
Ao orientador Prof. Fernando Álvaro Ostuni Gauthier, Dr.,  
pelas dicas fornecidas nos momentos certos.  
E a todos que direta ou indiretamente  
contribuíram para a realização desta pesquisa.

## SUMÁRIO

<b>Lista de Figuras</b> .....	p.ix
<b>Lista Tabelas</b> .....	p.xi
<b>Resumo</b> .....	p.xii
<b>Abstract</b> .....	p.xiii
<b>1 <u>Capítulo 1: Introdução</u></b> .....	p.4
1.1 Considerações iniciais.....	p.4
1.2 Objetivos.....	p.6
1.2.1 Objetivos gerais.....	p.6
1.2.2 Objetivos específicos.....	p.6
1.3 Justificativas.....	p.6
1.4 Estrutura do trabalho.....	p.7
<b>2 <u>Capítulo 2: Turismo</u></b> .....	p.8
2.1 Introdução.....	p.8
2.2 Definições de Turismo.....	p.9
2.2.1 Definições Econômicas.....	p.10
2.2.2 Definições Técnicas.....	p.10
2.2.3 Definições Holísticas.....	p.10
2.3 Internet.....	p.11
2.3.1 Histórico.....	p.13
2.3.2 Histórico da Internet no Brasil.....	p.13
2.3.3 Serviços básicos da Internet.....	p.14
2.3.4 Correio Eletrônico.....	p.14
2.3.5 Telnet.....	p.14
2.3.6 FPT.....	p.14
2.3.7 World Wide Web – WWW.....	p.15
2.4 Comércio Eletrônico.....	p.15
2.4.1 As formas de Comércio Eletrônico.....	p.16
2.5 Considerações Finais.....	p.17
<b>3 <u>Capítulo 3: Agentes Inteligentes</u></b> .....	p.18
3.1 Introdução.....	p.18
3.2 Inteligência Artificial Distribuída.....	p.19
3.2.1 Resolução Distribuída de Problemas (RDP).....	p.19
3.2.2 Sistemas Multiagentes.....	p.20
3.3 Agentes.....	p.21
3.4 Propriedades dos Agentes.....	p.22
3.5 Anatomia de um Agente de Software.....	p.23
3.6 Tipos de Agentes.....	p.24
3.6.1 Agentes Colaborativos.....	p.25
3.6.2 Agente Interface.....	p.25
3.6.3 Agente Reativo.....	p.25

3.6.4 Agentes Móveis.....	p.25
3.6.5 Agentes de Informação.....	p.25
3.6.6 Agentes Heterogêneos.....	p.26
3.6.7 Agentes Econômicos.....	p.26
3.7 Corrente uso de Agentes.....	p.26
3.7.1 Redes e Gerenciamento de Sistemas.....	p.26
3.7.2 Suporte a decisões e Logística.....	p.27
3.7.3 Assistente de Usuários.....	p.27
3.7.4 Agente Representando uma estrutura Organizacional.....	p.27
3.7.5 Agente para interesse Competitivo.....	p.27
3.8 Comunicação entre Agentes.....	p.27
3.8.1 Arquitetura de Comunicação entre Agentes.....	p.28
3.8.2 A Comunicação para Agente de Software.....	p.30
3.9 Cliente/Servidor X Cliente/Servidor para Agente de Software.....	p.31
3.10 Tecnologias que Habilitam Agentes.....	p.32
3.10.1 Tecnologias que habilitam a Comunicação entre Agentes.....	p.32
3.10.2 Linguagens utilizadas para habilitar a Comunicação entre Agentes.....	p.32
3.10.3 Mecanismo de Transferência de Mensagem.....	p.33
3.10.4 Ontologia de Comunicação.....	p.33
3.11 Diferenças entre as tecnologias: Agentes e Objetos.....	p.34
3.12 Considerações Finais.....	p.35
<b>4 <u>Capítulo 4: Protocolo de Comunicação entre Agentes usando AUML</u></b> .....	<b>p.36</b>
4.1 UML – Um Breve Histórico.....	p.36
4.2 Agent Unified Modeling Language (AUML).....	p.37
4.3 Especificando Protocolos.....	p.38
4.3.1 Diagrama de Protocolo.....	p.38
4.3.2 Semântica.....	p.38
4.3.3 Notação.....	p.38
4.3.4 Opções de representação.....	p.39
4.4 O Papel do Agente.....	p.40
4.4.1 Semântica.....	p.41
4.4.2 Notação.....	p.41
4.4.3 O Ciclo de Vida do Agente.....	p.41
4.5 Encadeamentos de Interações.....	p.42
4.5.1 Notação.....	p.42
4.6 Mapeando as Mensagens.....	p.43
4.6.1 Semântica.....	p.43
4.6.2 Notação utilizada na representação do mapeamento de Mensagens.....	p.43
4.8 Opções de representação de Mensagem.....	p.44
4.8.1 Cardinalidade.....	p.44
4.8.1.1 Mensagem Assíncrona.....	p.44
4.8.1.2 Mensagem Síncrona.....	p.44
4.8.1.3 Mensagem de Intervalo.....	p.44
4.9 Protocolos Aninhados.....	p.45
4.9.1 Semântica.....	p.45

4.9.2 Representação de um Protocolo Aninhado.....	p.46
4.10 Protocolos Aninhados Complexos.....	p.47
4.11 Encadeamento Mensagens dentro e fora de protocolo aninhado.....	p.48
4.11.1 Notação.....	p.48
4.12 Protocolos Parametrizados.....	p.49
4.12.1 Representação.....	p.49
4.13 Diagrama de Seqüência.....	p.53
4.14 Diagrama de Colaboração.....	p.53
4.15 Diagrama de Atividade.....	p.55
4.16 Gráfico de Estado.....	p.56
4.17 Outras Considerações AUML.....	p.58
4.17.1 Especificações de Papéis.....	p.58
4.17.2 Representando Papéis.....	p.59
4.18 Pacotes.....	p.60
4.19 Diagrama de Implementação.....	p.60
4.20 Considerações Finais.....	p.62
<b>5 <u>Capítulo 5: Protótipo do Sistema Portal de Turismo</u></b> .....	<b>p.65</b>
5.1 Introdução.....	p.65
5.2 O Protocolo de Comunicação.....	p.67
5.3 Estrutura do Sistema.....	p.73
5.4 Planejamento do Protótipo Portal de Turismo do Rio Grande do Norte.....	p.74
5.4.1 Encontrando os Atores.....	p.74
5.4.2 Identificando os Casos de uso para o Protótipo.....	p.74
5.4.3 Identificando as Classes para o Caso de Uso “Obter Sugestões”.....	p.80
5.4.4 Diagrama de Seqüência para o caminho primário do Caso de Uso “Obter Sugestões”.....	p.82
5.4.5 Infra-estrutura em nível de Componentes para o Protótipo.....	p.84
5.5 Considerações Finais.....	p.85
<b>6 <u>Capítulo 6: Implementação do Modelo</u></b> .....	<b>p.86</b>
6.1 Introdução.....	p.86
6.2 Ambiente de Desenvolvimento.....	p.86
6.2.1 Banco de Dados.....	p.87
6.3 Estrutura de Implementação dos Agentes.....	p.87
6.3.1 Agente Coordenador.....	p.87
6.3.2 Agentes: Hospedagem, Passeio, Restaurante e Lazer.....	p.88
6.4 A Arquitetura do Sistema.....	p.89
6.5 Implementação.....	p.90
6.5.1 Interface do Sistema.....	p.91
6.6 Simulação da Utilização do Protótipo.....	p.93
6.7 Considerações Finais.....	p.93

7 <b><u>Capítulo 7: Conclusões e Recomendações</u></b> .....	p.94
8 <b>Anexo</b> .....	p.96
9 <b><u>Bibliografia</u></b> .....	p.101



## Lista de figuras

Figura 1: Anatomia de um agente.....	p.30
Figura 2: Comunicação entre agentes.....	p.34
Figura 3: Arquitetura de comunicação entre agentes.....	p.34
Figura 4: Comunicação à KQML.....	p.35
Figura 5: Arquitetura three-tier cliente/server.....	p.37
Figura 6: Arquitetura three-tier client/server para agente de software.....	p.38
Figura 7: Representação do diagrama do protocolo FIPA contractNet.....	p.45
Figura 8: Representando o fluxo de mensagens.....	p.48
Figura 9: Representação de encadeamentos de interações.....	p.48
Figura 10: Mensagens assíncronas.....	p.50
Figura 11: Mensagens síncronas.....	p.50
Figura 12: Mensagens de intervalos.....	p.50
Figura 13: Representando uma mensagem.....	p.51
Figura 14: Representação de um protocolo aninhado.....	p.52
Figura 15: Representação de um protocolo aninhado entrelaçado.....	p.53
Figura 16: Representação de um protocolo aninhado complexo.....	p.54
Figura 17: Encadeamento de interações e mensagens em protocolos encadeados..	p.55
Figura 18: Representação de um protocolo parametrizado.....	p.56
Figura 19: Instância do protocolo FIPA-ContractNet.....	p.58
Figura 20: Sintaxe básica para representação de diagrama de sequência.....	p.59
Figura 21: Sintaxe básica para diagrama de colaboração.....	p.60
Figura 22: Diagrama de colaboração representando um protocolo de Interação entre agentes.....	p.60
Figura 23: Sintaxe básica para representação de diagrama de atividade.....	p.61
Figura 24: Diagrama de atividade para um protocolo de venda entre vários Agentes.....	p.62
Figura 25: Sintaxe básica para representação de gráfico de estado.....	p.63
Figura 26: Gráfico de estado para um protocolo de pedido.....	p.64
Figura 27: Representando mudança de papel usando diagrama de classes.....	p.65
Figura 28: Extensão de pacotes para agentes.....	p.66
Figura 29: Diagrama de implementação.....	p.67
Figura 30: Exemplo de um diagrama de implementação usando UML.....	p.67
Figura 31: Extensão de diagrama de implementação UML para agentes móveis...	p.68
Figura 32: Resolução distribuída para o protótipo turismo.....	p.71
Figura 33: Protocolo ao nível do agente coordenador(AC).....	p.73
Figura 34: Protocolo ao nível do agente hospedagem.....	p.74
Figura 35: Protocolo de comunicação do agente passeio.....	p.75
Figura 36: Protocolo de comunicação do agente restaurante.....	p.76
Figura 37: Protocolo de comunicação do agente lazer.....	p.77
Figura 38: Estrutura do sistema de agentes.....	p.78
Figura 39: Diagrama de caso de uso do protótipo.....	p.80

Figura 40: Diagrama de classes do protótipo.....	p.86
Figura 41: Diagrama de seqüência do protótipo.....	p.87
Figura 42: Diagrama de componentes do protótipo.....	p.88
Figura 43: Arquitetura geral do sistema para o protótipo.....	p.94
Figura 44: Formulário de entrada do perfil.....	p.95
Figura 45: Solução retornada ao visitante do site.....	p.96

## Lista de tabelas

Tabela 1: Formato da mensagem do protocolo de comunicação.....	p.72
Tabela 2: Notação dos elementos do protocolo (agentes).....	p.72
Tabela 3: Atores propostos para o portal de turismo do RN.....	p.79

## Resumo

---

---

CUNHA, José Antônio da. **Utilização da Agent Unified Modeling Language (AUML) na modelagem da estrutura e interação entre agentes.** Florianópolis, 2001. 108p. Dissertação (mestrado em Mídia e Conhecimento) – Programa de pós-Graduação em Engenharia de Produção, UFSC, Florianópolis.

O comércio eletrônico baseado na Internet está em plena expansão no mundo. O Brasil ainda enfrenta muitos desafios para ter uma expansão significativa do comércio eletrônico. A própria situação socioeconômica e o ainda pouco volume de computadores conectados à rede são alguns dos entraves que se pode citar. No entanto, está se presenciando uma expansão dessa nova prática comercial, pelo fato de que as empresas estão vendo nessa prática, a possibilidade de compradores em potencial, de maneira rápida e fácil, realizarem suas compras simplesmente navegando pela rede. Seus impactos serão muito grandes e, provavelmente, ainda não se compreende toda sua extensão. As próprias práticas comerciais serão modificadas.

O presente trabalho aborda um estudo sobre a tecnologia de agentes inteligentes, bem como um estudo sobre a expansão da UML para a modelagem desses agentes, levando-se em conta também os protocolos de comunicação nas trocas de mensagens entre esses agentes, com base nos diagramas UML.

Nessa perspectiva, será proposto um modelo de agentes que, dado um problema inicial, este será subdividido em problemas menores e, distribuídos entre os Agentes, tornando a busca pela solução mais simples e rápida.

---

---

Palavras-chave: Agentes Inteligentes, UML, Protocolo de comunicação, Mensagem, Agent Unified Modeling Language, Turismo.

## Abstract

---

---

CUNHA, José Antônio da. **Utilização da Agent Unified Modeling Language (AUML) na modelagem da estrutura e interação entre agentes.** Florianópolis, 2001. 108p. Dissertação (mestrado em Mídia e Conhecimento) – Programa de pós-Graduação em Engenharia de Produção, UFSC, Florianópolis.

The based e-commerce in the Internet is in full expansion in the world. Brazil still faces many challenges to have a significant expansion of the electronic commerce. The proper situation partner-economica and still the little volume of computers hardwired to the network are some of the impediments that we can cite. However, we are witnessing an expansion of this new practical advertising, for the fact of that the companies are sell in this practical advertising, the possibility of purchasers in potential, in fast and easy way, to carry through its purchases simply sailing for the network. Its impacts will be very great and probably still we do not understand all its extension. Proper practical the commercial ones will be modified.

The present work approaches a study on the technology of intelligent agents, as well as a study on the expansion of the UML for the modeling of these agents, taking itself in account also the protocols of communication in the swaps of messages enter these agents, on the basis of diagrams UML.

In this perspective, a model of agents will be considered that, given an initial problem, this is subdivided in smaller problems and distributed between exactly, becoming the fetching for the fast and simplest solution.

---

---

Key-words: Intelligent Agent , UML, protocol of communication, mesage, Agent Unified Modeling Language e tourism

## Capítulo 1

### INTRODUÇÃO

#### 1.1 Considerações iniciais

No final da década de 90, a ampla disponibilidade de computadores de baixo custo e o amplo acesso à conectividade global inflamaram a demanda por um outro tipo de aplicativo: o aplicativo altamente distribuído. Aplicativos altamente distribuídos são utilizados por um número cada vez maior de pessoas espalhadas pelo mundo, que estão conectadas a muitos aplicativos e servidores de dados através da Internet. Portanto, à medida que a Internet cresce, aumenta a necessidade de aplicação distribuída. Estão sendo esgotadas as capacidades das aplicações tradicionais. Isto faz com que o desenvolvedor de aplicativos, mude os seus paradigmas de desenvolvimento, assimilando novas tecnologias e novas ferramentas. Dentre as quais, podemos citar inúmeras, tais como: CORBA, COM, DCOM, MTS, ASP, JAVA, HTML, DHTML, PERL, JAVABEANS e muitas outras.

O crescimento da Internet provocou uma profunda revolução nas tecnologias de transações eletrônicas. Toda empresa, pequena ou grande, agora dispõe dos recursos necessários para interagir diretamente com seus clientes finais. Toda empresa, qualquer que seja seu grau de descentralização, agora tem a possibilidade de consolidar as informações sobre seus clientes e obter um diagnóstico mais fiel de quem são eles, que produtos e serviços compram e como gostam de ser atendidos. Uma empresa pode comprar ou vender produtos ou serviços para empresas ou pessoas localizadas remotamente, sem ter que se deslocar.

Com a Internet surgiram diversos serviços que são prestados através da mesma. Dentre os serviços básicos da Internet, podemos citar Correio Eletrônico (*E-mail*) - ele permite enviar mensagens para outras pessoas e/ou instituições, *TELNET* - permite ao usuário conectar-se a um computador remoto, *FTP* - permite o intercâmbio de arquivos de domínio público entre computadores e, a *WWW* - a *Web* como é chamada, permite divulgar informações por meio de um sistema multimídia de comunicação.

A *Web* é indiscutivelmente a área mais promissora da *Internet* e, com certeza, a de maior crescimento. Através de seu sistema de comunicação, ela permite que diferentes tipos de computadores se comuniquem através de hipertextos (textos com imagens, animações, vídeos e sons). Desse modo, utilizando o sistema de comunicação da *Web*, é possível realizar inúmeras aplicações pela *Internet*, tais como: apresentação de sites pessoais, de pesquisas, de comércio eletrônico e, muitos outros. Dentre as aplicações realizadas pela Internet usando a *Web*, o comércio eletrônico é o mais promissor potencialmente.

Esse tipo de comércio usa a capacidade das informações digitais para entender as necessidades e preferências de cada cliente e parceiros, a personalização de produtos e serviços para eles, e o fornecimento dos produtos e serviços da maneira mais rápida possível. Utilizando os serviços automatizados e personalizados, as empresas podem aumentar os lucros, reduzir os custos e, principalmente, estabelecer e fortalecer as relações com os clientes e parceiros.

O termo comércio eletrônico é bastante amplo. Ele pode significar, por exemplo, marketing direto, vendas e serviços ao cliente, transações bancárias, faturamento on-line, distribuição segura de informações, intercâmbio de cadeia de valores e compras corporativas.

Embora os benefícios dos sistemas de comércio eletrônico sejam atraentes, o desenvolvimento, distribuição e gerenciamento desses sistemas não são sempre fáceis. Além de quase sempre as empresas necessitarem de reestruturar seus processos comerciais, elas precisam adotar uma nova tecnologia.

As empresas que usam o comércio eletrônico, estão sempre preocupadas com a satisfação plena do cliente, como uma forma de cultivar esse comércio. Para isso, elas precisam de aplicações que sejam rápidas, eficazes e, personalizadas. Diante desta perspectiva, a tecnologia de Agente Inteligente de *Software*, está sendo bastante promissora, dentro do domínio do comércio eletrônico, prestando serviço ao cliente como, procurar produtos, negociar produtos e serviços, procurar e recuperar informações, filtrar informações, etc.

No presente trabalho, pretende-se desenvolver um protótipo usando a tecnologia de Agentes para uma subárea do comércio eletrônico, o turismo.

Um dos setores mais significativos e dinâmicos do mundo atual é o turismo. “A Organização Mundial do Turismo (OMT) estima um crescimento de 4 a 5 por cento ao ano da área de turismo até o início do século XXI. Esse crescimento, segundo a OMT, é motivado principalmente, pelo interesse por produtos turísticos novos ou renovados”. (Marques, Bissoli, 1999).

Atualmente temos presenciado uma verdadeira explosão do turismo no Brasil, mas por outro lado, a oferta turística não tem seguido o mesmo ritmo de crescimento da demanda, acarretando um estrangulamento no processo de desenvolvimento turístico brasileiro. “Isto deve-se a vários fatores como a falta de planejamento, a falta de informações atualizadas, ordenadas e confiáveis, dificultando a obtenção de benefícios e investimentos, que só são concedidos mediante projetos amplamente justificados.”(...) “Neste contexto, as novas tecnologias têm um papel fundamental, no desenvolvimento de sistemas de informação, podendo agilizar e facilitar de maneira eficaz o processo de planejamento turístico”. (Marques, Bissoli, 1999).

A informática está presente em muitas áreas das atividades cotidianas e claramente não poderia deixar de estar no setor de turismo, onde pode facilitar as diversas atividades relacionadas ao turismo, desde o planejamento até os serviços, como reservas de hotéis, passagens aéreas, informações de horários, cotação de tarifas de serviços turísticos em todo mundo, venda e emissão de bilhetes e muitos outros.

Não é pretensão deste trabalho fazer um estudo aprofundado sobre turismo, mas, como citado anteriormente, pretende-se desenvolver um protótipo para demonstração da utilização da tecnologia de agentes na prestação de serviços turísticos ao cliente, atendendo aos objetivos da facilidade de obtenção desses serviços, bem como a rapidez de acesso a eles. Contudo, apresentar-se-á no próximo capítulo algumas definições para o termo turismo e faremos algumas considerações sobre o comércio eletrônico.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Estudar a utilização da AUML (UML estendida para modelar agentes) na modelagem da estrutura de agentes e interação entre agentes, mais especificamente no domínio do comércio eletrônico.

### 1.2.2 Objetivos Específicos

- Fazer um estudo sobre o estado da arte de agentes inteligentes e protocolos de interação entre eles;
- Estudar uma ferramenta que auxilie na construção dos agentes;
- Estudar uma ferramenta que auxilie no estabelecimento das interações dos agentes e modelagem destes;
- Desenvolver um protótipo, utilizando uma estrutura de agentes para o processo de interação no auxílio ao cliente, para facilitar na prestação de serviços turísticos;
- Implementar o protótipo, considerando a estrutura de agentes apresentada;
- Testar o funcionamento do protótipo.

## 1.3 Justificativa

Em uma aplicação *stand-alone*, o acesso ao aplicativo é feito por um número limitado de usuários, mas isto não é o que ocorre em uma aplicação *on-line*, como é o caso da Internet, onde milhares de usuários podem estar acessando o sistema simultaneamente. Dessa forma, à medida que o número de usuários acessando o sistema aumenta, o número de interações também aumenta e o tempo despendido na comunicação também. Nesse



contexto, a tecnologia de agentes pode ser utilizada para solucionar o problema do tempo gasto na interação entre cliente e sistema. Nesse presente trabalho, pretende-se utilizar a tecnologia de agentes para solucionar esse problema.

Sendo assim, o modelo apresentado e implementado no protótipo é utilizado alternativamente como uma integração entre clientes e empresas prestadoras de serviços turísticos através da tecnologia de agentes que participam de forma autônoma no processo de interação/negociação entre as duas partes.

#### 1.4 Estrutura do Trabalho

Este trabalho está estruturado em sete capítulos, assim divididos, dadas as especificidades, afim de facilitar a compreensão do tema apresentado.

O **primeiro capítulo** faz uma introdução, mostra os objetivos, apresenta as limitações encontradas no estudo realizado e justifica o trabalho.

O **segundo capítulo** faz uma abordagem sobre o setor turístico e o comércio eletrônico, mostrando definições e fazendo algumas considerações sobre o uso de tecnologias, tanto em um setor como no outro.

No **terceiro**, faz-se um levantamento do estado da arte da tecnologia de agentes inteligentes, abordando definições, estrutura, propriedades e, além do mais, procurasse fazer uma relação entre as tecnologias de agente e de objeto.

O **quarto**, traz um estudo da utilização da UML estendida (AUML) na modelagem da estrutura de agentes inteligentes, mais especificamente no estabelecimento do protocolo de interação entre os agentes.

O **quinto**, trata do modelo do protótipo no auxílio ao cliente e na prestação de serviços turísticos pela Internet.

O **sexto capítulo**, trata da implementação do protótipo sugerido.

O **sétimo**, apresenta a conclusão e as recomendações para trabalhos futuros com relação ao tema abordado

Finalmente, é apresentada a **bibliografia** utilizada para o desenvolvimento do presente trabalho.

## Capítulo 2

### REVISÃO: TURISMO, INTERNET E COMÉRCIO ELETRÔNICO

#### 2.1 Introdução

O turismo é um dos setores mais significativos e dinâmicos do mundo atual. A Organização Mundial de Turismo (OMT) estima um crescimento de 4 a 5 por cento ao ano da área de turismo até o início do século XXI. “Esse crescimento é motivado, principalmente, pelo interesse por produtos turísticos novos ou renovados, tais como o turismo rural, de aventura e ecológico. Tal conclusão está fundamentada no estudo da evolução das variáveis de mercado (demanda, oferta de atrativos naturais e culturais, equipamentos turísticos e infra-estrutura) e em variáveis exógenas (demográficas, sociais, econômicas, políticas e legislativas)”. [Marques, Bissoli, 2000].

Segundo Caio Luiz de Carvalho, presidente da Empresa Brasileiro de Turismo (EMBRATUR), “a indústria do turismo, nos dias atuais, vem crescendo de maneira extremamente veloz em todo mundo, garantindo o crescimento econômico-social das mais diversas regiões permitindo dessa forma, a expansão do mercado de trabalho, gerando empregos e propiciando de uma certa maneira uma distribuição de renda mais uniforme. Tornando-se um dos instrumentos mais importantes do desenvolvimento da economia de um país”. [www.embratur.com]

No Brasil, segundo matriz insumo-produto do IBGE, o turismo impacta 52 segmentos diferentes da economia, empregando, em sua cadeia, desde a mão-de-obra mais qualificada, em áreas que se utilizam desde alta tecnologia (transportes e comunicação) até as de menor qualificação, tanto no emprego formal quanto no informal.

O Governo Federal, consciente de que o turismo é o caminho mais curto para incrementar novos processos de produção, implantou nos últimos anos, por meio da EMBRATUR, e em parceria com entidades do setor público e privado, a Política Nacional de Turismo, que mudou radicalmente o perfil do setor. Sua concretização permitiu um salto qualitativo em várias frentes, trazendo à tona dados que, por sua expressividade, torna incontestável o consenso de que o setor turístico vem provocando uma “revolução silenciosa” no mundo, segundo Carvalho, presidente da EMBRATUR.

Segundo a EMBRATUR, estas medidas tomadas pelo Governo Federal, geraram um faturamento da ordem de 38 bilhões de dólares, aumentando a demanda de emprego em cinco milhões. Isto tem gerado um impacto na arrecadação de impostos diretos e indiretos da ordem de 7 bilhões de dólares, impulsionando o crescimento econômico de cidades turísticas já conhecidas, bem como dos pequenos municípios que foram se engajando à nova política.

De acordo com a nova política Nacional de Turismo, a atividade passou a ser prioridade estratégica para a geração de empregos e renda no país, e para a diminuição

das desigualdades entre as diversas regiões brasileiras, que através da mesma, está tentando alavancar o turismo das regiões Nordeste, da Amazônia e do Centro Oeste, com investimentos para a melhoria da infra-estrutura turística dessas regiões. Só no Nordeste, foram investidos 8 bilhões de dólares em obras de infra-estrutura básica, foram implementadas ações nas áreas de capacitação profissional, adequação da legislação, marketing e implantação de metodologias da gestão turística pelo Programa Nacional da Municipalização do Turismo (PNMT), acarretando na melhoria dos serviços prestados. Com esse investimento foram construídos no Nordeste, sete novos aeroportos, 22 mil metros quadrados de patrimônio históricos foram restaurados, 17 projetos de saneamento básico foram executados e 280 quilômetros de estradas foram construídas. A estimativa é de que, mais 6 bilhões de dólares serão investidos pela iniciativa privada em novos empreendimentos turísticos até 2002. [Carvalho, EMBRATUR].

Todos esses esforços acarretaram no aumento do fluxo turístico interno, o que pode ser observado através do número de autorização de vôos *charters* concedidos pelo Departamento de Aviação Civil (DAC). Em um ano, 1200 autorizações foram feitas. Apesar do alto número de autorizações concebidas, a participação desse tipo de transporte no Brasil ainda é pequena. Por isso, a EMBRATUR estabeleceu como meta para o ano 2000 a implementação de uma política de *charters*. No Brasil apenas 7% do transporte de turista é feito dessa forma, enquanto na Europa esse percentual chega a 56%. [Carvalho, EMBRATUR].

Como pode-se ver, a política para o turismo no Brasil tem mudado bastante nos últimos cinco anos. Existe a preocupação de incentivo ao desenvolvimento do turismo em todas as regiões do país, com investimentos em infra-estrutura nos diversos setores turísticos bem como no marketing interno e externo para divulgação desses pólos turísticos. Para se ter uma idéia, segundo Caio Luiz de Carvalho, presidente da EMBRATUR, essa empresa pretende investir milhões de dólares na promoção do turismo, junto aos países do Mercosul. Para isso, foi lançada uma campanha com o *slogan* "Se viajar é sua paixão, o Brasil é seu destino."

## 2.2 Definições de Turismo

"O fenômeno turístico é tão grande e complexo que é praticamente impossível expressá-lo corretamente; ocorre em diferentes campos de estudo, em que é explicado conforme diferentes correntes de pensamento e verificado em vários contextos da realidade social". [Beni, 1998].

Beni (1998) apresenta três tendências de definições: econômicas, técnicas e holísticas. [Marques e Bissoli, 2000].

### 2.2.1 Definições Econômicas

- Soma das operações, principalmente de natureza econômica, diretamente relacionadas com a entrada, a saída e a permanência de estrangeiros em um país, cidade ou região. [Herman von Schullard, 1910].
- Turismo é um setor importante, nacionalmente identificável. Compreende um amplo corte transversal de atividades componentes, incluindo a provisão de transporte, alojamento, recreação, alimentação e serviços afins. [Departamento de Turismo e Recreação, 1975].
- Turismo refere-se à provisão de transporte, alojamento, recreação, alimentação e serviços relacionados a viajantes domésticos e do exterior. Compreende a viagem para todos os propósitos, desde recreação até negócios. [Ansett Airlines of Australia, 1977].
- Turismo pode ser definido como a ciência, a arte e a atividade de atrair e transportar visitantes, alojá-los e, de maneira cortês, satisfazer suas necessidades e desejos. [Robert MacIntosh, 1977].

### 2.2.2 Definições Técnicas

- Turista é a pessoa que visita um país que não seja o de sua residência por um período de pelo menos 24 horas. [Comissão de Estatística da Liga das Nações, 1937].
- Para propósito estatístico, o termo ‘visitante’ descreve a pessoa que visita um país que não seja o de sua residência, por qualquer motivo, e que nele não exerça ocupação remunerada. Turistas são visitantes temporários que permanecem pelo menos 24 horas no país visitado e cuja finalidade da viagem pode ser classificada sob um dos seguintes tópicos: lazer (recreação, férias, saúde, estudo, religião e esporte), negócio, família, missões e conferências. Excursionistas são visitantes temporários que permanecem menos de 24 horas no país visitado – incluindo viajantes de cruzeiros marítimos. [Nações Unidas, 1963; aprovada pela OMT em 1968].

### 2.2.3 Definições Holísticas

- A soma dos fenômenos e das relações resultantes da viagem e da permanência de não-residentes, na medida em que não leve à residência permanente e não esteja relacionada a nenhuma atividade remunerada. [Hunziker e Krapf, 1942].
- É o estudo do homem longe de seu local de residência, do setor que satisfaz suas necessidades e dos impactos que ambos, ele e o setor, geram sobre os

ambientes físicos, econômicos e sociocultural da área receptora. [Jafar Jafari, 1977].

A OMT adota a seguinte definição para o turismo:

“Turismo compreende as atividades que as pessoas realizam durante suas viagens e estadas em lugares distintos aos de sua residência habitual, por um período de tempo consecutivo inferior a um ano, com fins de lazer, de negócios e outros”. [Marques e Bissoli, 2000].

Foram considerados alguns aspectos para esse novo marco conceitual do turismo, segundo Huéscar,(1993) e Sancho, (1998) [Marques e Bissoli, 2000].

- Aumento contínuo das viagens, dos movimentos turísticos e dos gastos dos turistas;
- Aumento da concorrência entre as regiões ou países de destino;
- Melhor conhecimento e maior atenção em relação aos crescentes efeitos do turismo, sejam eles econômicos, socioculturais ou ambientais;
- Maior informação por parte do consumidor em relação aos destinos turísticos e às opções de viagem – e maior exigência na escolha entre os diversos produtos e serviços turísticos;
- Domínio das tecnologias de transações do mercado de turismo, especialmente nos aspectos de informação transmitida por computador e sistemas de reservas.

### 2.3 Internet

“A tecnologia tem invadido praticamente todos os setores da economia, e um dos setores da economia que teve especial destaque, sem dúvida nenhuma, foi a área de prestação de serviços, na qual pode-se observar alterações como o emprego de tecnologia na busca de informações mais precisa; pela melhoria dos recursos humanos nas tarefas que requerem iniciativa e rapidez na comunicação; pela melhoria e rapidez nas tomadas de decisão, e também não podemos deixar de observar a exploração de escritórios espalhados pelo mundo inteiro, deslocando o local de trabalho no espaço e no tempo”. [Marques e Bissoli, 2000].

A informatização da sociedade é um processo realmente irreversível, segundo Trigo, (1993). [Marques e Bissoli, 2000].

Como não poderia deixar de ser, o setor de turismo está sempre buscando melhoria na prestação de serviços, redução de custos, incremento da produtividade e

Somente em 20 de dezembro de 1994 é que a EMBRATEL lança o serviço experimental a fim de conhecer melhor a Internet. [Marcondes, 1998].

No Brasil, a abertura da Internet ao setor privado, para exploração comercial, só acontece em 1995, pela iniciativa do Ministério das Telecomunicações e Ministério da Ciência e Tecnologia. [Marcondes, 1998].

A responsável pela infra-estrutura de interconexão e informação em nível nacional, é a RNP, que controla o *backbone* (Coluna dorsal de uma rede, backbone representa a via principal de informações transferidas por uma rede, nesse caso, a Internet). [Marcondes, 1998].

### 2.3.3 Serviços Básicos da Internet

A Internet fornece vários serviços, dentre eles pode-se destacar: Correio Eletrônico (E-mail), TELNET, FTP (File Transfer Protocol), a WWW, etc.

### 2.3.4 Correio Eletrônico

O sistema de comunicação por via de E-MAIL é a forma mais antiga e popular de comunicação pela Internet. Por meio do E-MAIL, pode-se enviar mensagens a um ou vários destinos, utilizar listas de discussão, anexar às mensagens arquivos de som, imagens, textos e entre outras coisas. [Marcondes, 1998].

### 2.3.5 Telnet

Este tipo de sistema permite ao usuário conectar-se a um computador da rede. Após feito a conexão, o usuário pode executar comandos e usar recursos do computador remoto. [Marcondes, 1998].

Portanto, feita a conexão, o usuário pode acessar E-MAILs, Bancos de Dados, Catálogos de bibliotecas, Ferramentas de procura de informações, etc. Isso tudo de forma direta e em tempo real. [Marcondes, 1998].

### 2.3.6 FPT

Um outro serviço bastante utilizado na rede é o FTP (File Transfer Protocol) – protocolo de transferência de arquivos. Com ele, pode-se fazer cópia de arquivo disponibilizado pela Internet (*download*). Os arquivos que podem ser baixados pela rede são chamados de *freeware* (programas que podem ser copiados e utilizados gratuitamente por qualquer pessoa). [Marcondes, 1998].

### 2.3.7 World Wide Web – WWW

A World Wide Web ou simplesmente Web como é mais conhecida, é o serviço gráfico da Internet que oferece uma rede de documentos interativos e o *software* para acessá-los. Ela está baseada em documentos chamados de páginas Web, que combinam texto, formulários, imagens, sons, animação e *links* de hipertexto, chamados de *hyperlinks*. Ela representa uma forma revolucionária de divulgar informações por meio de um sistema multimídia de comunicação. [Marcondes, 1998].

Indiscutivelmente a WWW é a área mais promissora da Internet e, com certeza, a região de maior crescimento. Tal sucesso fez com que os serviços se multiplicassem: os bancos planejam oferecer todas as suas operações na rede; notícias são distribuídas imediatamente; Pizzarias aceitam pedidos; Livrarias e lojas de CDs vendem seus produtos pela rede; resumindo: vende-se de tudo pela rede. Esse comércio, feito através da rede, é chamado de comércio eletrônico. [Marcondes, 1998].

## 2.4 Comércio Eletrônico

O poder comercial da Internet é visível nos números de bilhões de dólares e na preocupação de governos. De acordo com a NUA - Internet Surveys, o mercado criado pela Internet já conta com mais de cem milhões de usuários espelhados por todo mundo. Um mercado superior ao próprio mercado consumidor brasileiro, pois todos os cem milhões têm poder de compra – uma vez que os mesmo adquiriram computador com *modem*, telefone e serviço de um provedor. [Developers, setembro de 1998, p. 18].

À medida que o número de pessoas que acessam a Internet cresce, cresce também o número de empresas dispostas a entrar no mercado virtual, uma vez que o mesmo apresenta-se com possibilidades tão animadoras. Estima-se que o volume mundial de vendas ao consumidor via Internet no ano 2002, seja da ordem de 26 bilhões de dólares (EMARKETER). [Developers, setembro de 1998, p. 18].

Portanto, considerando os dados acima citados e levando-se em conta o fato de que a população mundial é em torno de cinco bilhões, apenas 2% a 3% do total de pessoas no mundo acessam a Internet, o que permite prever que o crescimento não deverá cair tão cedo, seja no mundo ou no Brasil.

A Internet pode, sob muitos aspectos, ser comparada a uma grande vitrine eletrônica. A falta de interação presencial afasta a inibição e o constrangimento, liberando a curiosidade despreziosa que pode induzir a compra. Boa parte das visitas aos sites comerciais da rede deve-se, certamente, a esse tipo de curiosidade, a qual pode ser satisfeita independente do horário.

Portanto, o comércio eletrônico desponta como o caminho natural da evolução da Internet. Ele está revolucionando a geração de demanda de produtos e serviços, as formas de se atingir diversos tipos de consumidores, além de aprimorar a divulgação e a oferta desses itens de consumo. Além do mais, ele amplia e dá agilidade ao processo de busca e aquisição de produtos e serviços.

### 2.4.1 As formas de Comércio Eletrônico

Tradicionalmente, quando se fala em comércio eletrônico, pensa-se de imediato na venda para consumidores finais, ou o chamado *Business to Consumer*. Entretanto, as maiores oportunidades para quem desenvolve soluções e aplicações em comércio eletrônico estão no relacionamento comercial entre empresas, o chamado *Business to Business*. [Developers, setembro de 1998, p. 28].

*E-Business*, é a soma dos recursos da Internet com os sistemas de tecnologias da informação das empresas, integrando-os e otimizando a automatização dos processos, ampliando o alcance e a abrangência dos sistemas corporativos e permitindo a criação de novas aplicações. Adicionando aos sistemas corporativos os recursos da Web em suas diversas formas – Internet, Intranet, Extranet, abre-se o acesso aos dados e informações da empresa, permitindo a interação com clientes, fornecedores e revendedores.

No entanto, ao se pretender implantar um sistema de *E-business*, deve-se ter algumas preocupações, dentre as quais pode se citar: melhorar a produtividade, diminuir o tempo de acesso, integrar os sistemas existentes, permitir escalabilidade, garantir a segurança dos sistemas, gerenciar as aplicações, manter homogeneidade entre usuários, diminuir custos administrativos e atender a demanda por desempenho e novas tecnologias.

“Entende-se como comércio eletrônico via Internet o relacionamento comercial entre organizações e/ou entre organizações e consumidores através da troca eletrônica de informações na *Web*, no que se refere aos processos de compra /venda de produtos, serviços e logística”. [Herson, Developers, setembro de 1998, p. 12].

O comércio eletrônico é, antes de tudo, um comércio que utiliza a informática e as telecomunicações como suporte e, como tal, exige uma boa estratégia comercial para se sobressair em relação aos concorrentes. Isso significa que não basta uma boa equipe de informática cuidando da criação e do gerenciamento da loja virtual, mas de uma equipe que consiga integrar a tecnologia com as necessidades dos clientes. [Vaz, Developers, setembro de 1998, p. 18].

O comércio eletrônico apresenta algumas vantagens sobre o tradicional uma vez que, o atendimento virtual do eletrônico pode ser personalizado, favorecendo o *cross selling* (mecanismo que mostra ao cliente seu último pedido e apresenta produtos complementares aos já adquiridos) ou o *upselling* (oferece mercadorias de maior valor as quais poderiam ter passado despercebidas ao cliente). Uma outra vantagem apresentada pelo comércio eletrônico é a capacidade de gerar vendas adicionais, reduzindo o custo de venda e aumentando a eficiência da transação além disso, sua flexibilidade e velocidade para desencadear promoções instantâneas, respondendo com rapidez incrível aos estímulos do mercado é uma das grandes vantagens desse comércio. [Pombo, Developers, setembro de 1998, p. 28].

No entanto, apesar de todas essas vantagens, quando desenvolve-se uma aplicação de comércio eletrônico, deve-se avaliar de perto cada um de seus



componentes, pois eles formam um círculo integrado e produtivo. A aplicação deve garantir que a transação comercial ocorra de forma segura, com o rigoroso acompanhamento da entrega do produto. Todo esse processo deve ser analisado, para que se aprimorem os sistemas comerciais, a qualidade gráfica e a personalização das páginas para a crescente fidelidade dos clientes e a conseqüente geração de novas transações e análises. Essa capacidade de atrair clientes potenciais para a *Web site* garante o sucesso do projeto. [Pombo, Developers, setembro de 1998, p. 29].

## 2.5 Considerações Finais

Nesse capítulo procurou-se dar um enfoque ao setor turístico e ao comércio eletrônico, ressaltando suas definições, bem como alguns aspectos do turismo e comércio eletrônico no âmbito nacional e internacional.

Como foi descrito nos parágrafos acima, tanto o turismo como, o comércio eletrônico está em plena expansão no contexto mundial e tudo isso está ocorrendo, devido ao crescimento cada vez mais expressivo da rede mundial. O uso crescente da Internet e o surgimento de novas tecnologias que facilitam mais ainda o desenvolvimento de aplicações para a Internet faz com que o comércio eletrônico seja um dos serviços realizado por essa rede, mais promissor. Dentro do comércio eletrônico, o setor turismo eletrônico vem se destacando como uma das áreas que mais progride.

Vimos também que, as principais preocupações do comércio eletrônico são: a rapidez no atendimento ao cliente, a segurança nas transações comerciais, a personalização do cliente a flexibilidade e a velocidade nas transações comerciais. Tudo isso com o intuito de cada vez mais melhorar o atendimento ao cliente. Portanto, diante desse contexto, a tecnologia de agentes inteligentes de software, pode ser utilizada, uma vez que um agente serve para favorecer os seres humanos e as organizações na obtenção de informações e a realizar tarefas, tais como: localizar e acessar informações de várias fontes, filtrar informações indesejadas e fornecer suporte a decisões, por exemplo. A proposta aqui sentada é desenvolver uma estrutura de agentes que interajam entre si, de forma que forneçam informações de serviços turísticos aos clientes de acordo com o perfil desses, de uma maneira transparente.

No próximo capítulo far-se-á algumas considerações sobre agentes, ressaltando alguns aspectos como definições, propriedades, estruturas e interações entre os agentes.

## Capítulo 3

### AGENTES INTELIGENTES

#### 3.1 Introdução

Um dos avanços mais significativos na tecnologia moderna é a combinação da Internet com a *World Wide Web*. A natureza onipresente da Internet com a *Web* e o acesso através de preços acessíveis, faz dela uma excelente plataforma para comunicação mais efetiva entre as pessoas. A cada momento surgem novas empresas cuja única presença – pelo menos para os clientes – ocorre por meio da *Web*. No entanto, essas empresas tiveram que, enfrentar novos desafios, como, por exemplo, descobrir como atrair os clientes, manter esses clientes, reduzir o tempo de atendimento ao cliente, o custo dos serviços, tornar a transação segura para o cliente. Como pode-se ver, o comércio eletrônico pela Internet precisa ser parte de uma estratégia mais ampla – uma estratégia que englobe todas as maneiras pelas quais o cliente possa negociar eletronicamente com eficiência.

Essa convivência digital entre pessoas, conectadas por uma estrutura virtual em constante modificação, estabeleceu diferentes perspectivas sobre os fundamentos do processo interativo, fazendo com que a tecnologia de agentes tivesse um grande avanço no desenvolvimento de aplicações para a Internet.

Como já foi dito no capítulo anterior, dentre as aplicações para a Internet uma subárea mais promissora é a do comércio eletrônico. Dentro da subárea comércio eletrônico, o setor mais promissor, segundo pesquisa realizada nos Estados Unidos e divulgada na Info Exame, é o de turismo. [Info Exame Nº 176, p 42].

A tecnologia de agentes tem sido bastante utilizada no comércio eletrônico, auxiliando os clientes na busca de informações, na localização e na negociação de produtos, assim como no setor específico de turismo, ela pode auxiliar o cliente em diversos serviços, como, por exemplo, na escolha de hospedagens e na escolha de pacotes turísticos, restaurantes, lazer, dentre outros, de uma forma racional e rápida, de maneira que, o cliente economize tempo na seleção dessas informações, libertando-o para outras atividades.

Neste capítulo, pretende-se fazer uma abordagem sobre os agentes de *software* nos seguintes aspectos: definições, propriedades, tipos, arquiteturas, protocolos de interação entre agentes e, para finalizar, far-se-á uma breve comparação entre objetos e agentes. Não se pode falar de agentes sem antes se falar de Inteligência Artificial (IA), uma vez que a IA é o começo de tudo. Portanto, na próxima seção apresentar-se-á inicialmente alguns conceitos relacionados com a IA.

## 3.2 Inteligência Artificial Distribuída

Tradicionalmente, a Inteligência Artificial (IA) é apresentada como a parte da ciência da computação cuja ênfase está no estudo de sistemas de computação inteligentes, ou seja, sistemas que exibam características associadas à inteligência no comportamento humano – compreensão de linguagem, aprendizado, raciocínio, resolução de problemas, e assim por diante [Tomanaga, 96]. Segundo os pesquisadores, as pesquisas em IA são caracterizadas por tentativas de se entender e desenvolver uma única entidade inteligente.

Devido à proliferação de computadores de processamento paralelos poderosos, proliferação de redes de computadores e, conseqüentemente a grande expansão da Internet, fazendo com que cada vez mais as ações das pessoas e resolução de problemas sejam distribuídas. As novas pesquisas em IA passaram a dar ênfase ao comportamento social de sistema, ao invés do estudo de apenas uma entidade inteligente. Surgindo, então, a Inteligência Artificial Distribuída (IAD). A IAD é uma subárea da IA preocupada com a cooperação, coordenação e interação da IA evoluindo da visão individual para a social.

Pode-se distinguir duas principais áreas da IAD: Resolução Distribuída de Problemas (RDP) e Sistemas Multiagentes (SMA). Na primeira considera-se o trabalho de se resolver um problema particular por dividindo-o em muitos módulos que cooperam e compartilham conhecimento sobre o mesmo e sobre o desenvolvimento da solução. No outro lado estão os SMA onde não se parte de um problema específico, como no caso anterior, mas almeja-se coordenar um comportamento inteligente entre um conjunto de agentes autônomos, possivelmente pré-existent: como eles podem coordenar seus conhecimentos, metas, habilidades e planos conjuntamente para realizar uma ação ou resolver um problema. Nos SMA os agentes interagem entre si para resolver problemas buscando uma solução global. A interação entre agentes acontece ou pela troca de mensagens ou por quadros negros (áreas comuns de armazenamento de dados utilizadas pelos agentes).

As aplicações nessa área são inúmeras e sofisticadas, [Tomanaga, (96)]: robótica, tomada de decisão administrativa, administração de balcões bancários, controle de tráfico aéreo, gerência de redes de telecomunicações, automação de escritórios e sistemas de atendimento ao público, etc.

### 3.2.1 Resolução Distribuída de Problemas (RDP)

Segundo Álvares e Sichman (97), a motivação inicial de uma resolução distribuída de problemas é um problema inicial que precisa ser solucionado e que a estratégia de resolução apresenta as seguintes características:

- O problema é resolvido por um conjunto de agentes e cada agente é concebido para solucionar um determinado problema particular;

- A interação entre os agentes é realizada seja por troca de mensagens seja por compartilhamento de dados comuns;
- Os agentes são executados de modo concorrente, para aumentar a velocidade de resolução;
- Os agentes cooperam, dividindo entre si as diversas partes do problema original.

Segundo Álvares e Sichman (97), do ponto de vista da concepção do sistema, os agentes não existem a priori: sua concepção, bem como sua interação, é realizada face à existência de um problema que o sistema deve solucionar. Não existe a preocupação da reutilização dos agentes num outro contexto.

### 3.2.2 Sistemas Multiagentes

Por outro lado, nos SMA, não existe um problema definido a priori para ser resolvido. O objetivo da área é estudar modelos genéricos a partir dos quais se podem conceber agentes, organizações e interações, de modo a poder instanciar tais conceitos quando se desejar resolver um problema particular.

Do ponto de vista da concepção de um sistema SMA, os agentes, suas interações e suas organizações são concebidas independentemente de um problema particular a ser resolvido, contrariamente à concepção RDP. Tornando-se possível a reutilização de tais componentes quando se desejar desenvolver aplicações similares. Na concepção SMA, quando um problema é apresentado ao sistema, então as organizações e as interações são instanciadas dinamicamente pelos agentes. [Álvares e Sichman 97].

Segundo esses autores, RDP e SMA são pesquisas que se relacionam, uma vez que a RDP pode ser considerado um subconjunto de SMA, ou seja, pode-se considerar a RDP como um caso particular de SMA. Além do mais, o SMA fornece uma base para a RDP, sendo assim, RDP e SMA são complementares.

Independente da abordagem, os sistemas multiagentes estão revolucionando a forma de resolver problemas, passando de uma visão global, de atitudes baseadas em uma abordagem centralizada, a uma visão local, em níveis de agentes (componentes), onde o comportamento coletivo emerge das interações dos agentes.

A *Internet* tem sido a grande vitrine para o desenvolvimento de sistemas multiagentes, em diversos serviços prestados por ela, como por exemplo, a busca e a filtragem de informações, a prestação de serviços e o comércio eletrônico.

No entanto, apesar das comunidades de pesquisas nessa área estarem sempre atualizando o conceito de agentes, ainda não existe um consenso. Na próxima seção apresentar-se-á alguns conceitos sobre agentes.

### 3.3 Agentes

De acordo com as pesquisas realizadas, percebe-se que existem diversas definições para o termo “agente”, as quais estão sempre relacionadas, com suas características e sua implementação. A seguir serão enumeradas algumas definições mais comuns do termo “agente”.

O termo “agente” é usado, e mal usado, amplamente para descrever uma grande faixa de entidades computacionais. [Dennett, 99].

Um “Agente” pode ser uma pessoa, uma máquina, um *software*, ou uma variedade de outras coisas. A definição básica de agentes é algo que age. [Bauer, 1999]. Embora, para os desenvolvedores de sistema de Tecnologia de Informação (TI), tal definição é muito genérica. Necessita-se que sejam adicionadas propriedades a esses agentes. As propriedades dos agentes serão apresentadas a seguir.

Uma definição aplicável a “agentes” em geral, quer sejam naturais ou artificiais, de *software* ou *hardware* é: “agente é uma entidade que reside em um ambiente onde interpreta dados sensoriais que reflete eventos do ambiente e executa comandos motores, que produzem efeitos no ambiente”. [Fernandes, 97].

Uma definição mais associada com agentes inteligentes de *software* é: “entidade que executa um conjunto de operações em intenção de um usuário ou outro programa, com algum grau de independência ou autonomia e, executando estas operações, emprega algum conhecimento dos objetivos ou desejos do usuário”. [Fernandes, 97].

Odell (2000), define agente como “uma extensão de objetos ativos, exibindo autonomia dinâmica (a habilidade para iniciar uma ação solicitada externamente) e autonomia determinística (a habilidade para rejeitar ou modificar uma solicitação externa)”. Dessa maneira, para ele, a definição básica para agente é “um objeto que pode dizer ‘vá’ (autonomia dinâmica) e ‘não’ (autonomia determinística)”.

Agentes inteligentes de *software* é uma entidade autônoma de *software* que pode interagir com seu ambiente”. [Odell, 2000]. Em outras palavras, eles são implementados usando *software*, são autônomos, e podem reagir a outras entidades – incluindo humanos, máquinas e outros *softwares* agentes em vários ambientes e plataformas.

Como pode-se ver essas definições são bastantes gerais e que, para realmente representar um agente inteligente, é necessário acrescentar algumas propriedades. A seguir descrever-se-á algumas das propriedades dos agentes.

### 3.4 Propriedades dos Agentes

Algumas das propriedades comuns dos agentes são:

- **Autonomia** – capacidade de agir sem a intervenção externa. Ele tem algum grau de controle sobre seu estado interno e sobre as ações baseado na sua própria experiência. [Bauer, 1999].
- **Interatividade** – comunica-se com o ambiente e outros agentes. [Bauer, 1999].
- **Adaptabilidade** – capacidade de responder a outros agentes e/ou ao seu ambiente. A mais avançada forma de adaptação permite a um agente modificar suas ações baseado em suas experiências. [Bauer, 1999].
- **Sociabilidade** – interação que é marcada pela amizade ou relação social amável, isto é, o agente é afetuoso, companheiro, ou amigável. [Bauer, 1999].
- **Mobilidade** – habilidade para transporta-se de um ambiente para outro. [Bauer, 1999].
- **Representatividade** – pode agir em nome de alguém ou de alguma coisa, isto é, agir para o interesse, benefício de alguma entidade. [Bauer, 1999].
- **Pró-atividade** – orienta-se para atingir objetivos. Ele não simplesmente reage ao ambiente, ele toma a iniciativa da ação. [Bauer, 1999].
- **Inteligência** – o estado de inteligência é formalizado pelo conhecimento isto é, crenças, objetivos, planos e suposições, e interage com outros agentes usando linguagem simbólica. [Bauer, 1999].
- **Racionalidade** – habilidade para escolher uma ação baseada em um objetivo interno e no conhecimento de que uma ação particular irá conduzi-lo do início ou ao final do objetivo. [Bauer, 1999].
- **Temporariamente contínuos** – é um processo que roda continuamente. [Bauer, 1999].
- **Personalidade** – personalidade e estado emocional acreditável. [Bauer, 1999].

- **Transparente e responsável** – deve ser transparente quando requerido, deve fornecer uma lista de suas atividades. [Bauer, 1999].
- **Coordenação** – habilidade de realizar algumas atividades em um ambiente compartilhado com outros agentes. As atividades são frequentemente coordenadas via um plano, **workflows**, ou algum outro mecanismo de gerenciamento de processo. [Bauer, 1999].
- **Cooperação** – habilidade para cooperar com outros agentes para agir em propósito comum; ou eles têm sucesso ou falham juntos. (Colaboração é outro termo que é usado como sinônimo de cooperação). [Bauer, 1999].
- **Competição** – habilidade de cooperar com outros agentes, apesar que, o sucesso de um agente, implica na falha de outros ( o oposto de cooperativo). [Bauer, 1999].
- **Robustez** – habilidade para proceder com erros e dados incompletos. [Bauer, 1999].
- **Confiabilidade** – segue as leis da robótica e é real. [Bauer, 1999].
- **Não Prognóstico** – habilidade de agir de forma não completamente planejada, mesmo quando todas as condições iniciais são conhecidas. [Bauer, 1999].

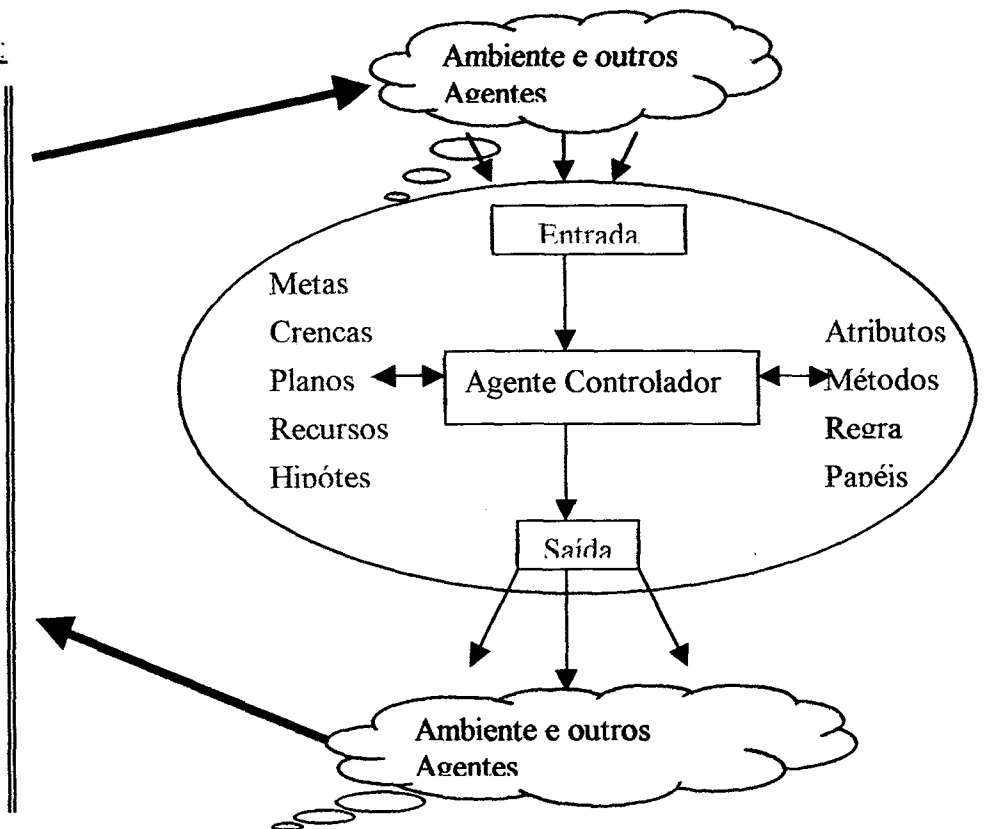
Para a indústria de Tecnologia de Informação - TI, a definição de “agentes” ainda não emergiu. Muitos concordam que “agentes” deve ter, pelo menos, três das propriedades descritas acima. Outros requerem todas as propriedades listadas acima. No mínimo, um agente para TI é uma entidade autônoma que pode interagir com seu ambiente. Em outras palavras, ele deve ser capaz de perceber seu ambiente através de sensores e agir produzindo efeito no ambiente. [Bauer, 1999].

### 3.5 Anatomia de um Agente de Software

Segundo Odell (2000), uma possível Anatomia de um Agente é mostrada na Figura 3.1, a seguir, onde é claramente percebido a interação do agente com o ambiente e/ou outros agentes. Pode-se perceber também que a comunicação é realizada através de mensagens via declarações, consultas, comandos, mudanças de estado.

**Mensagens via**  
**Ocorrência de eventos:**

- Declarações  
Consultas  
Comandos  
Mudanças de estado  
Outras
- Notificar
  - Monitorar
  - Registrar
  - Cancelar
  - Subscrever
  - Recrutar
  - Negociar
  - Informar



**Figura 3.1 Anatomia de um Agente [Odell, 2000]**

### 3.6 Tipos de Agentes

Podemos dividir as pesquisas em agentes entre as três linhas: [Dennett, 2000]

- Inteligência Artificial Distribuída (IAD) – Sistemas de multi-agentes (SMA);
- Larga noção de “agente” (dos 90 até agora);
- Modelo baseado em agentes e ainda outros tipos de agentes (o futuro).

Na década de oitenta surgem a Inteligência Artificial Distribuída e os Sistemas Multiagentes e na década de noventa ampliam-se a noção de agentes, com diversos tipos de agentes, tais como: agente interface, reativo, móvel, agente de informação, etc. no final da década de noventa surge a idéia dos modelos baseados em agentes, a novas espécies de agentes (agentes econômicos). [Dennett, 2000].



A seguir ir-se-á listar alguns tipos de agente.

### **3.6.1 Agentes Colaborativos. [Dennett, 2000].**

Os agentes colaborativos representam o começo das pesquisas em agentes. Eles podem ser utilizados para resolver conflitos em sistema de negociação, para integrar informações, prover soluções para problemas distribuídos inerentes, tais como: controle de tráfico aéreo, gerenciamento de redes de telecomunicações, etc.

### **3.6.2 Agente Interface. [Dennett, 2000].**

A Segunda linha de pesquisa em agentes. Eles podem ser utilizados para dar suporte e prover assistência, cooperar com o usuário em alguma tarefa em uma aplicação. Um de suas características é que eles aprendem observando e imitando o usuário, recebendo *feedback* completo do usuário, recebendo instruções explícitas ou pedindo conselho a outros agentes.

### **3.6.3 Agentes Reativos. [Dennett, 2000].**

Os agentes reativos não têm modelo simbólico interno. Eles respondem a ação por estímulo para o corrente estado do ambiente. Sua arquitetura é baseada em contribuição. Cada agente reativo é simples e interage com outros de uma maneira básica e seus principais benefícios são robustez, tempo rápido de resposta. Um desafio na implementação desse tipo de agente é a escalabilidade.

### **3.6.4 Agentes Móveis. [Dennett, 2000].**

Um agente móvel tem com sua principal característica mover-se de um *host* para outro onde pode ser executado. Do ponto de vista conceitual, agente móvel pode também ser considerado como itinerante, dinâmico, viajante ou migrante. Uma desvantagem de tal mobilidade é que o *site* remoto deve fornecer CPU capaz de suportar o processamento do agente móvel. São utilizados em aplicações para recuperar informações distribuídas e rota de redes de telecomunicações.

### **3.6.5 Agentes de Informação. [Dennett, 2000].**

Os agentes de informação são utilizados para gerenciar o explosivo crescimento de informações, manipular ou coletar informações de muitas fontes distribuídas. Eles podem ser móveis ou estáticos. A seguir listar-se-á alguns exemplos de agentes de informação.

- *BargainFinder* compara lojas na Internet para armazenar CDs;
- *Jasper* trabalha em função de um usuário ou de comunidades de usuários e armazena, recupera e informa outros agentes acerca de informações úteis na WWW;
- *Internet SofBot* quando a pesquisa é de alto nível, ele seleciona quem mais facilita a pesquisa na Internet (*finger, ftp, gopher*).

### 3.6.6 Agentes Heterogêneos. [Dennett, 2000].

Uma arquitetura Heterogênea de agentes engloba no mínimo dois tipos diferentes de agentes. São utilizados em suporte de demanda para interoperar aplicações *stand-alone*, e sistemas CAD. Sua implementação necessita de uma linguagem comum para comunicação entre eles. Essa linguagem de comunicação deve definir um protocolo e um formato de permuta comum e um conjunto de ontologias.

### 3.6.7 Agentes Econômicos. [Dennett, 2000].

Essa é a terceira linha de Agentes. Ele representa o último debate no desenvolvimento de agentes. eles envolvem esforços de várias áreas de pesquisa: ciências sociais, vida artificial, sistemas adaptativos complexos, programação orientada a mercado e serviços de *Internet*. Suas principais características são: emergência, evolução de normas sócias, adaptativo às mudanças e a sua eficiência.

## 3.7 Corrente uso de Agente

Segundo Dennett (2000), a industria de agente está em um estágio embrionário. Dessa maneira, o desenvolvimento de sistemas baseados em agente e de tecnologias para implementar agente apesar de poucas e isoladas e existem – e está em crescimento. Correntemente, várias classes de agente tem sido de alguma forma utilizada. A seguir ir-se-á exemplificar o uso de agentes em diferentes áreas de utilização:.

### 3.7.1 Redes e Gerenciamento de Sistemas

As companhias de telecomunicações têm sido as mais ativas nessa área. A meta dessas companhias é usar agente para auxiliar em sistemas complexos e gerenciamento de tarefas em redes, tais como carga de balanceamento, antecipação de falhas, análise de problemas e síntese de informações. [OMG Document, 2000].

### **3.7.2 Suporte a Decisões e Logística**

Muitas empresas de serviços públicos e organizações militares usam agentes para síntese de informações e suporte a decisões. Geralmente, esses sistemas são utilizados para alertar operadores de possíveis problemas ou fornecer informações para suporte de uma decisão complexa. [OMG Document, 2000].

### **3.7.3 Assistente de Usuários**

Esses agente opera à nível de interface de usuário, oferecendo informações ou fazendo recomendações. Eles são algumas vezes representados visualmente como um desenho animado. Empresas como Microsoft, Lótus e Apple têm mostrado maior interesse nessa área. O exemplo mais conhecido desse tipo de agente em uso é o help animado usado pelos produtos Microsoft Office. [OMG Document, 2000].

### **3.7.4 Agentes Representando uma Estrutura Organizacional**

Esses agentes são estruturados para operar de maneira similar as organizações humanas. Por exemplo, um sistema multiagente que represente uma cadeia de lojas de fornecimento poderia ter agente desempenhando diferentes papéis, tais como cliente, fornecedor, estoque, pedido, itens e manufatura. [OMG Document, 2000].

### **3.7.5 Agente para Interesse Competitivo**

Esse tipo de agente é bastante usado em aplicações *Web* comerciais para fazer recomendações. Por exemplo, se você gosta de um produto “x”, então você deve gostar do produto “y”.

Os exemplos listados acima não esgotam todas as características dos agentes, apenas representam algumas dessas características.

## **3.8 Comunicação entre Agentes**

Um Agente pode se comunicar com outro Agente – bem como com seu ambiente. [Odell, 2000], como mostra a figura 3.2 a seguir:

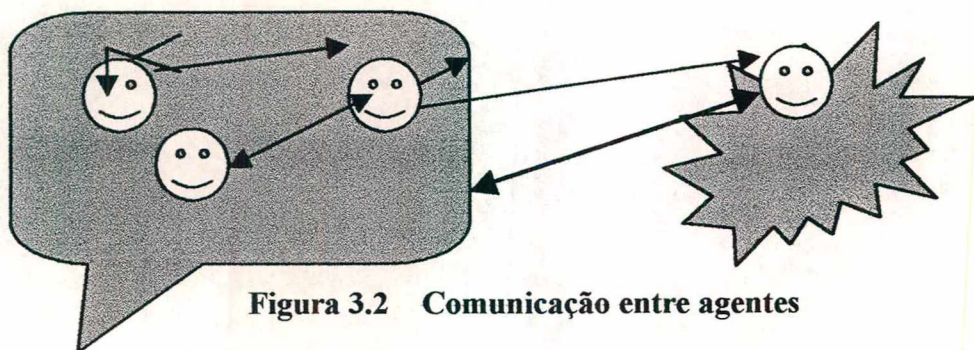


Figura 3.2 Comunicação entre agentes

### 3.8.1 Arquitetura de Comunicação ente Agentes

A Figura 3.3, a seguir mostra as diferentes arquiteturas de comunicação entre agentes e a Figura 3.4, mostra a arquitetura de comunicação entre agentes usando KQML. [Odell, 1998].

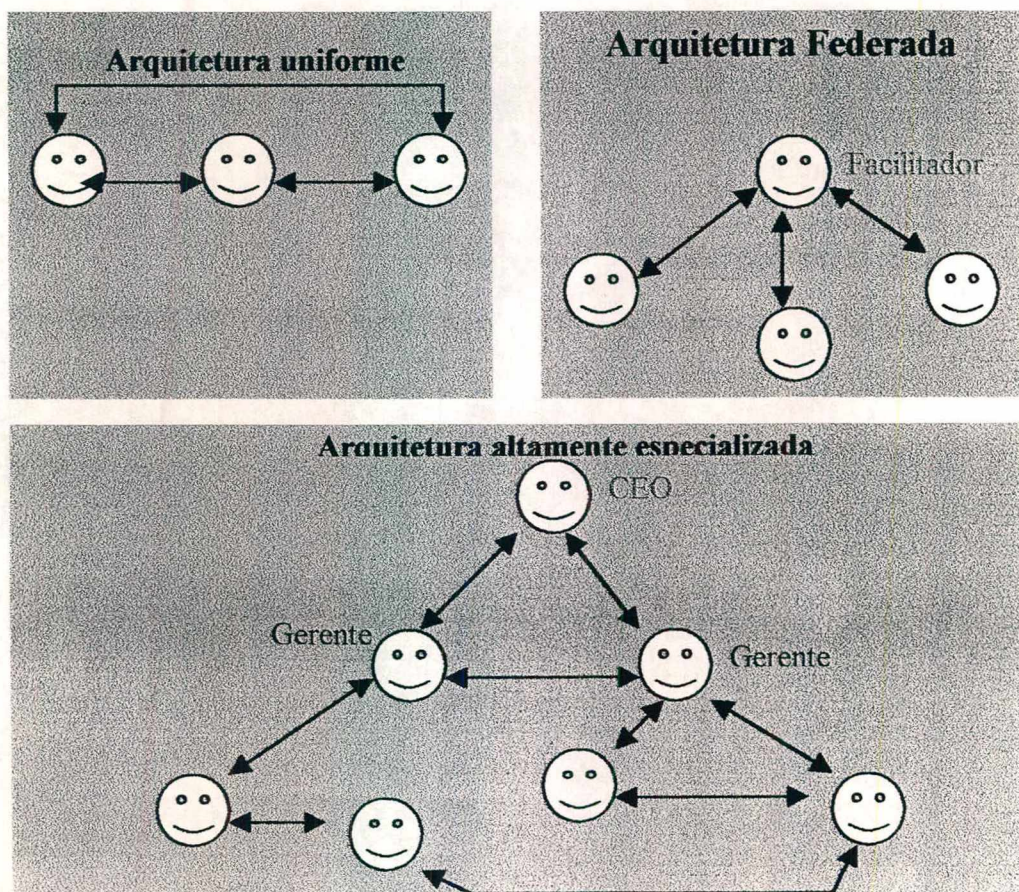


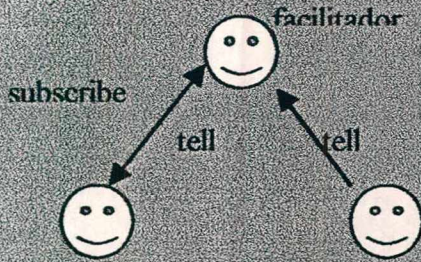
Figura 3.3 Arquitetura de comunicação ente Agentes

### Comunicação à KQML

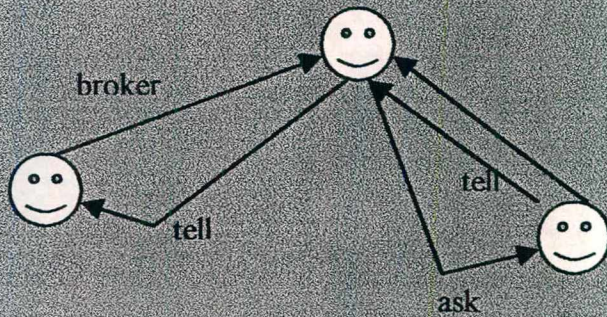
Simple point-to-point



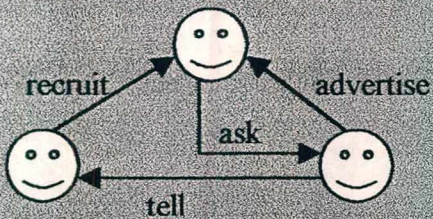
Subscribe



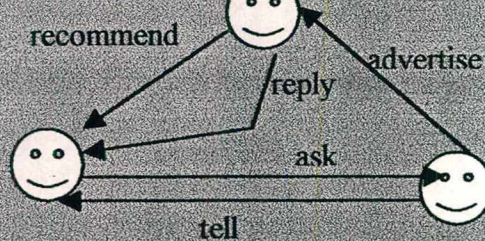
Broker



Recruit



Recommend



Broadcast

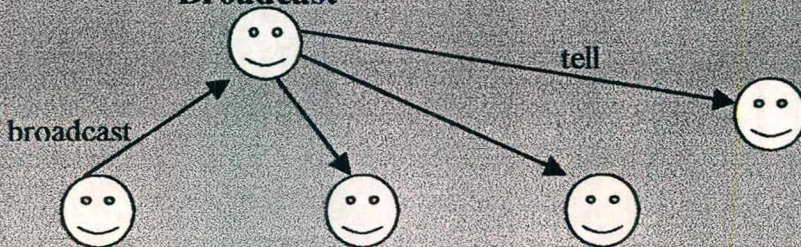


Figura 3.4 Comunicação à KQML

### 3.8.2 A Comunicação para Agente de Software

A arquitetura de comunicação para Agente de *Software* envolve: [Odell, 1998]:

- **Protocolos de Comunicação**
  - *Unicast* – envia uma mensagem quando há somente um processo remetente e um receptor;
  - *Broadcast* – envia somente uma mensagem a todos *hosts* na rede, reconhece e lê a mensagem;
  - *Multicast* – envia somente uma mensagem e todos os *hosts* que estão registrados como “de interesse” reconhecem e podem lê a mensagem.
- **Protocolos de Aplicação**
  - *Publish/subscribe* – assíncrono, comunicação dirigida a eventos, de muitos para muitos;
  - *Request/reply* – síncrono, comunicação dirigida à demanda, de um para muitos;
  - *Solicit/response* – assíncrona requisitar/responder.
- **Mensagem**
  - Baseada em assunto;
  - Baseada em conteúdo.
- **Propriedade das Mensagens**
  - Emprega formato repositório;
  - O formato se auto descreve;
  - Emprega transformação/tradução;
  - Prioridade de mensagem;
  - Expiração de mensagem.

### 3.9 Cliente/Servidor X Cliente/Servidor para Agente de Software

Em uma aplicação *Three-Tier Client/Server* o método de acesso é frequentemente via as tecnologias de acesso a dados e as chamadas SQL para o BD (banco de dados). Comumente há uma intermediação entre os servidores e os controladores. [Odell, 1998]; como se pode ver na Figura 3.5 a seguir:

Em uma aplicação *Three-Tier Client/Server* utilizando Agente de *Software*, os Agentes podem enviar programas através da rede. Veja a Figura 3.6 a seguir:

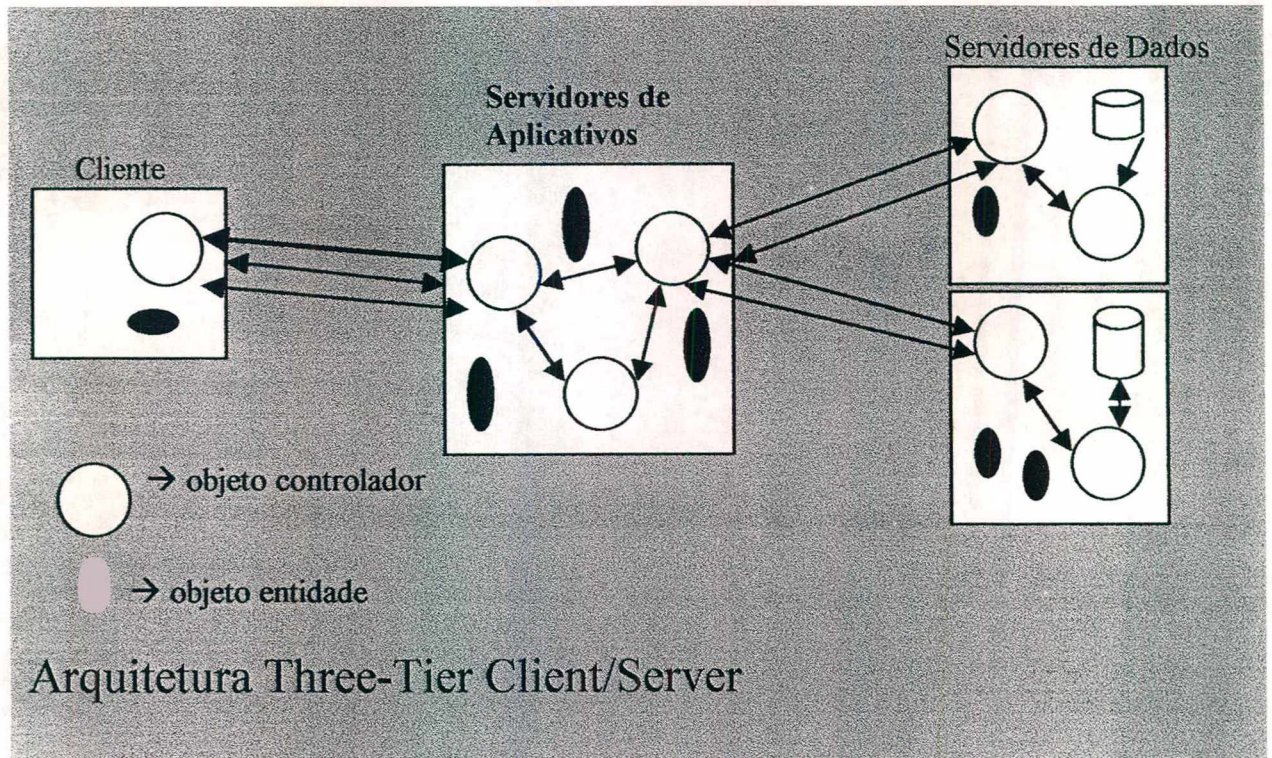
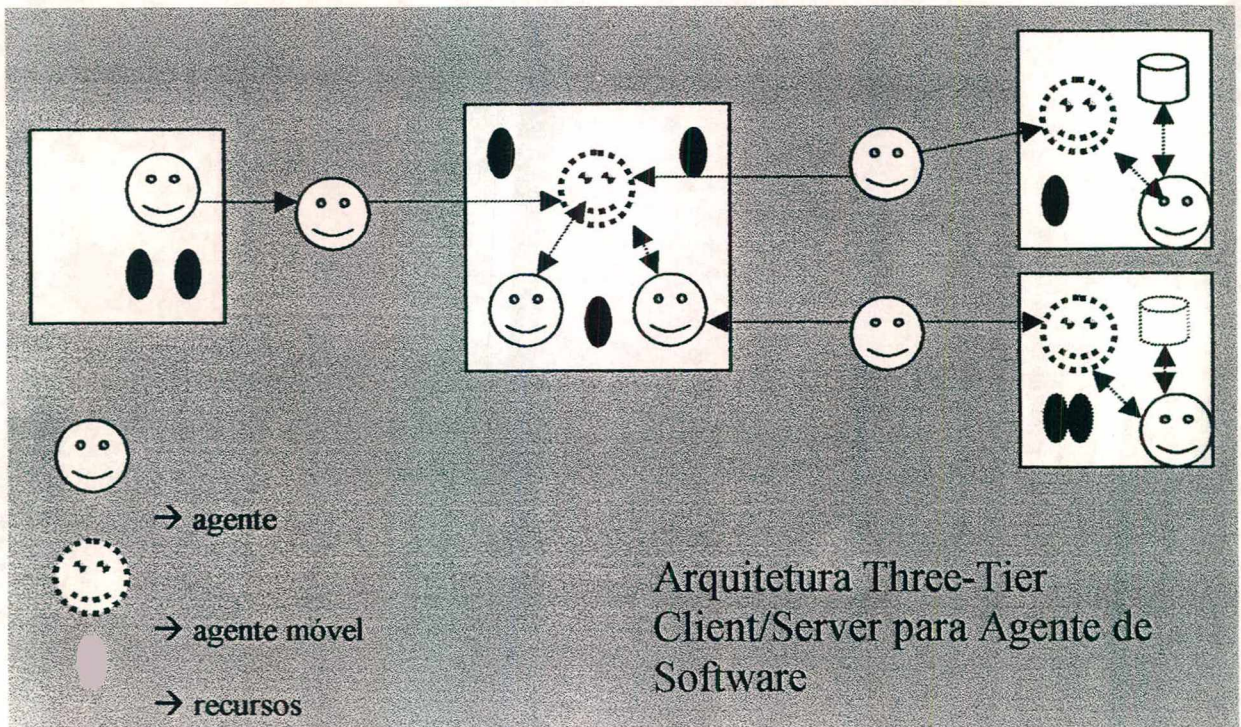


Figura 3.5 Arquitetura Three-Tier Client/Server



**Figura 3.6** Arquitetura Three-Tier Client/Server para Agente de Software

### 3.10 Tecnologias que Habilitam Agentes

Como mencionado anteriormente, a tecnologia de agentes requer que os agentes sejam autônomos, interativos e adaptativos, dentre outras propriedades. Para suportar essas características, os projetistas de agentes têm que utilizar tecnologias. Nessa seção descrever-se-á algumas tecnologias utilizadas para habilitar agentes.

#### 3.10.1 Tecnologias que habilitar a Comunicação entre Agentes

Essa é uma área que precisa de maior padronização, pois se cada desenvolvedor de projeto utilizar a comunicação entre agentes de forma diferente, o sistema de comunicação entre agentes, tornar-se-ia impraticável. Não somente o conteúdo da mensagem de comunicação poderia ser diferente, mas o significado da comunicação poderia ocorrer de diversas formas. [Bauer, 1999].

#### 3.10.2 Linguagens utilizadas para habilitar a Comunicação entre Agentes



O-338.757-1

As mensagens devem ter uma semântica bem definida. Sendo assim, necessita-se padronizar a linguagem de comunicação entre os agentes (LCA). Além disso, a linguagem utilizada deve ter uma semântica formal, mas que diferentes implementações preservem as características essenciais da LCA. Possíveis implementações: [Bauer, 1999].

- KQML;
- Arcol e FIPA;
- KIF;
- Baseado em XML.

### 3.10.3 Mecanismo de Transferência de Mensagem

Em um ambiente de agentes, mensagens podem ser agendadas, bem como orientadas a eventos. Elas podem ser enviadas de modo sincronizado ou não sincronizado. Além disso, o mecanismo de transferência de mensagens pode suportar endereçamento único bem como endereçamento baseado em papéis. Finalmente, o mecanismo deve suportar os modos de serviços Unicast, Multicast, e Braodcast. Possíveis implementações: [Bauer, 1999].

- CORBA;
- OMG Messaging Services;
- JAVA;
- RMI;
- DCOM;
- JAVABEANS

### 3.10.4 Ontologia de Comunicação

Na comunicação entre agentes os conceitos fazem parte da comunicação. Além disso, os agentes podem ter diferentes termos para o mesmo conceito, termos idênticos para diferentes conceitos, e diferentes sistemas de classes. Uma Ontologia comum, então, é requerida para representar o conhecimento de vários domínios. Possíveis implementações: [Bauer, 1999].

- UML;
- MOF;
- OKBC;
- Esquema XML.

### 3.11 Diferença entre as Tecnologias: Agente e Objeto

Existe por parte do *OMG Agent Working Group* uma preocupação em relação às tecnologias de agentes e objetos e a interoperabilidade entre agente-objeto? As tecnologias agentes e objetos têm similaridades e diferenças e elas não são sempre claras e a exata definição de agentes (e objetos) permanece indefinida. [Bauer, 1999].

Pode-se de qualquer forma concordar que ambas tecnologias são usadas para partilhar estado e ação, mas elas diferem no aspecto comunicação e talvez em outras formas. Todavia, está muito difundido o interesse por ambas tecnologias e, cada vez mais, projetistas de *software* estão usando uma ou outra, para modelar seus problemas. [Bauer, 1999].

Desse modo é inevitável que as duas tecnologias venham a interoperar ou, se não, unificar-se. Se as duas tecnologias têm necessidades similares, então, torna-se muitas vezes difícil para os projetistas de *software* decidir qual delas usar, acarretando perda de tempo e tornando o trabalho improdutivo. Dessa forma, existe uma real necessidade que as duas tecnologias coexistam e tornem-se mais integradas, de forma que os agentes possam interagir como objetos e vice-versa. Em particular o *OMG Agent Working Group* - preocupa-se com a relação entre agentes e objetos, e sua unificação, não deixando perder de vista a relação dessas tecnologias com outras tecnologias de grande aceitação, tais como Web, XML, *Email*, objetos distribuídos, etc. [Bauer, 1999].

As relações entre objetos e agentes dependem dos diferentes pontos de vistas, que essas tecnologias são vistas e podem levar a diferentes conclusões. Segundo o *OMG Agent Working Group*, nos próximos anos, ir-se-á estar mais interessados na unificação dessas tecnologias. [Bauer, 1999].

Um ponto de vista comum é que agente é diferente de objeto. Os Agentes usam uma linguagem para se comunicarem um com os outros e podem individualmente decidir se responde a mensagem de outro agente. Eles podem representar informações BDI complexas, usar inferência e mudar suas ações baseado no que ele aprendeu. Essa capacidade pode ser adicionada para objeto, mas ele se tornaria um agente. [Bauer, 1999].

De acordo com esse ponto de vista a tecnologia de objetos pode ser usada para implementar a tecnologia de agentes (dessa maneira objetos e agentes podem ser relacionados). [Bauer, 1999].

Desse ponto de vista pode-se também admitir a idéia de que agente pode chamar objeto em uma orientação a objeto, isto é, agir como objeto com referência para objetos e desse modo esperar por resposta de objetos para os quais eles enviaram uma mensagem. Da mesma forma isso pode ocorrer com objetos: eles podem enviar uma mensagem para outros objetos e aguardar uma resposta. Portanto, como podemos ver, existe bastante similaridade entre Agentes e Objetos. [Bauer, 1999].

Os simpatizantes de objetos ponderam que os objetos fazem coisas que os agentes não fazem, por exemplo: herança não é comumente vista como um aspecto de agentes, embora agentes não são impedidos de suportar alguma forma de herança. [Bauer, 1999].

Podemos assumir que agentes são objetos ou componentes, no sentido que agentes têm identidade, uma vez que se pode distinguir um agente de outro, eles têm seus próprios estados e ações diferente das de outros agentes, além deles terem interfaces através das quais se comunicam com outros agentes e com outras coisas. Por exemplo, pessoas, *hardware* ou outros componentes de *software*. [Bauer, 1999].

### 3.12 Considerações Finais

A conclusão final é que objetos são objetos se eles se comunicam com outros objetos através de mensagens e agentes são agentes se eles se comunicam via linguagem de comunicação de agentes. Algumas entidades podem fazer ambas as coisas, conseqüentemente ambos são objetos e agentes. [Bauer, 1999].

Como pode-se perceber durante todo este capítulo, é que as pesquisas de acordo com o *OMG Agent Work Group* estão apontando para uma unificação dos termos objeto e agente e essa unificação seria importante por todas as razões descritas acima.

Muitas das características requeridas pelos agentes e pelos sistemas baseados em agentes são também – independentemente - citadas como úteis para objetos e para sistemas baseados em objetos. Como tais, muitas das extensões sugeridas no próximo capítulo poderiam estar em harmonia com muitas das novas características sugeridas pela orientação a objetos (OO). [Odell, 1999].

## Capítulo 4

### PROTOCOLO DE INTERAÇÃO ENTRE AGENTES USANDO AUML

#### 4.1 UML - um breve histórico

Durante os anos 70, a programação estruturada era o método dominante para o desenvolvimento de *software*. Juntamente com ela, surge a engenharia de *software* para facilitar e formalizar o ciclo de vida de sistemas. Isto é, planejar, analisar e projetar, em síntese, construir o sistema, implementa-lo e mantê-lo. Nos anos oitenta, as linguagens de orientação a objetos (OO) ganham uma grande popularidade, iniciando com isto, novos conceitos, tais como: encapsulamento de dados, herança, mensagens e polimorfismo. Pelo final dos anos oitenta e início dos anos 90, surge no mercado uma grande quantidade de métodos para suportar OO. A quantidade de métodos orientados a objetos aumentou de 10 para mais de 50 durante o período de 1989 a 1994, dificultando para o usuário a escolha de um método que melhor atendesse as suas necessidades. Formou-se, então, uma expressiva corrente de idéias no sentido de unificar os vários métodos e em 29 de junho de 1995 foi apresentada a primeira versão do método unificado, como foi chamado pelo *Object Management Group* (OMG). E por volta de Novembro de 1997, os membros do OMG passaram a chamar o método unificado de *Unified Modeling Language* (UML). [Booch, Rumbaugh, Jacobson, 2000].

Nesse período as empresas perceberam que a UML era um recurso estratégico para seus negócios e, então, foi estabelecido um consórcio de UML com várias empresas que desejavam dedicar recursos com o propósito de trabalhar a favor de uma definição mais forte e completa da UML. Desde então ela tem evoluído bastante e ultimamente a manutenção da UML foi assumida pela RTF (Revision Task Force) do OMG, sob a responsabilidade de Cris Kobryn.

A UML é uma linguagem padrão para a elaboração da estrutura de projetos de software. Ela pode ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software. [Booch, Rumbaugh, Jacobson, 2000].

A UML é apenas uma linguagem e, portanto, é somente uma parte de um método para desenvolvimento de software. Ela tem sido empregada de maneira efetiva em domínios como os seguintes:

- sistemas de informações corporativos;
- serviços bancários e financeiros;

- telecomunicações;
- transportes;
- defesa/espaço aéreo;
- vendas a varejo;
- eletrônica médica;
- serviços distribuídos baseados na Web;

A UML unifica e formaliza os muitos métodos para software OO, incluindo Booch, Rumbaugh (OMT), Jacobson, e Odell. Ela suporta os seguintes tipos de modelos [Odell, 2000]:

- **Modelo Estático** – há dois tipos de modelo estático: o diagrama de classes e o de pacotes, ambos descrevendo a semântica de dados e mensagens. Dentro de um sistema o diagrama de classes pode ser usado para diferentes propósitos. Primeiro, ele pode modelar conceitualmente o domínio do problema. Segundo, o diagrama de classe pode modelar a implementação de classe. Em termos gerais o termo classe se refere a uma unidade encapsulada, seus conceitos e suas associações; [Odell, 2000].
- **Modelo dinâmico** – inclui diagramas de interação (isto é, diagramas de seqüência e colaboração), mapa de estado e diagramas de atividades. [Odell, 2000].
- **Caso de uso** – a especificação de ações que um sistema ou classe pode realizar por interação com outros atores. [Odell, 2000].
- **Modelo implementação** - são modelo de componentes e diagramas de equipamentos, que descrevem a distribuição de componentes por diferentes plataformas. [Odell, 2000].
- **Object constraint language (OCL)** – é uma simples linguagem formal para expressões mais semânticas dentro de uma especificação UML. Ela pode ser usada para definir *constraints* no modelo, pré e pós-condições de operações e caminhos de navegação dentro de um objeto. [Odell, 2000].

A UML está tendo uma grande aceitação na representação de *softwares* orientados a objetos. Odell, (2000) considera agente o próximo passo além de objetos e explora uma extensão da UML e idiomas dentro da UML para acomodar o requerimento de agentes, resultando em um Agente UML (AUML). [Odell, 2000].

#### 4.2 Agent Unified Modeling Language (AUML)

O formalismo da UML corrente é algumas vezes insuficiente para modelar Agente e Sistemas baseados em Agentes. Embora existam sistemas específicos baseados em

agentes desenvolvidos em UML. A programação baseada em agentes não foge à especificação técnica suportada pelo processo de engenharia de *software* – de planejar, analisar, projetar, construir, implementar e manter. Uma proposta para um completo ciclo de vida de um sistema baseado em agentes segue as técnicas da engenharia de *software*. [Odell, 2000].

Ambos FIPA e o OMG, grupos que trabalham com Agentes, estão explorando e recomendando extensões para UML para modelar agente, segundo Odell, (2000). No presente trabalho, apresentar-se-á um subconjunto de uma extensão baseada em agente para o padrão UML (AUML) para especificações de Protocolo de Interação entre Agentes (PIA) e outras noções de agentes comumente usadas.

Pretende-se ainda sugerir uma técnica de especificação para PIA com semânticas formal e intuitiva e uma notação gráfica para facilitar a visualização do modelo por parte dos usuários. A semântica segue uma precisa definição usada no processo de engenharia de *software*. A notação gráfica fornece uma linguagem comum para comunicação PIA - particularmente para pessoas não familiarizadas com métodos baseados em agentes.

### 4.3 Especificando Protocolos

Um Agente de Protocolo de Interação (PIA) descreve um padrão de comunicação como uma seqüência de mensagens entre agentes e as *constraints* (restrições) no conteúdo dessas mensagens. [Odell, 2000].

#### 4.3.1 Diagrama de Protocolo

O novo tipo de diagrama, sugerido como uma extensão UML, é chamado de Diagrama de Protocolo. Interações bem definidas entre agentes são mostrados pelos Diagramas de Protocolos. Eles são similares aos diagramas de seqüência e de colaboração, mais combinam as idéias de ambos. [Bauer, 1999].

#### 4.3.2 Semântica

Um Diagrama de Protocolo representa uma interação, a qual é um conjunto de mensagens trocadas entre diferentes papéis de agentes em colaboração para realizar um desejo de outro(s) agente(s). [Bauer, 1999].

#### 4.3.3 Notação

As dimensões de um Diagrama de Protocolo são: a dimensão vertical, representando o tempo; a dimensão horizontal, representando os diferentes agentes. Normalmente a linha de tempo procede descendo a página. Usualmente, somente a

seqüência de tempo é importante, mas, em aplicações em tempo real, o eixo do tempo poderia ser uma métrica real. A ordem dos agentes na dimensão horizontal não tem importância. [Bauer, 1999].

### 4.3.4 Opções de Representação

Os modelos dinâmicos UML são úteis para expressar interação entre agentes. O diagrama de interação captura as interações entre objetos. Diagrama de seqüência é um membro dessa família; diagrama de colaboração também. Os dois diagramas contêm as mesmas informações. O *layout* gráfico do diagrama de seqüência enfatiza a seqüência cronológica de comunicação, enquanto o diagrama de colaboração enfatiza a associação entre os agentes. O diagrama de atividade e o mapa de estado capturam o fluxo de processamento. [Bauer, 1999].

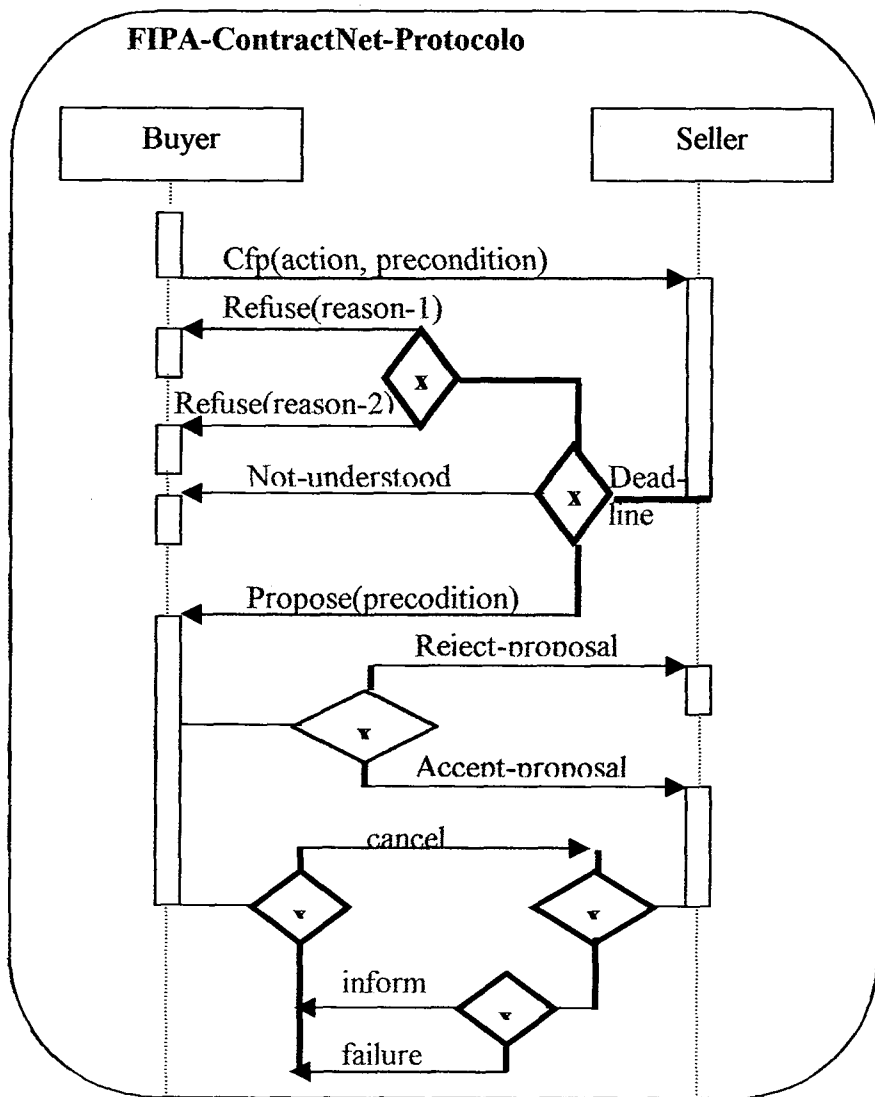


Figura 4.1 Representação do protocolo FIPA-ContractNet [Bauer,1999]

A Figura 4.1 representa um protocolo expresso como um diagrama de seqüência UML para o contrato de protocolo net. Um agente inicializador envia uma chamada a um agente participante, invocando-o a realizar uma tarefa (a seta em apenas um sentido significa comunicação assíncrona). O agente participante pode escolher uma das seguintes alternativas para responder ao inicializador: refugar a proposta, submeter à proposta ou indicar que não entendeu a proposta (o símbolo em forma de diamante indica uma decisão que pode resultar em zero ou mais comunicações, que podem ser enviadas - dependendo da condição/restrrição; o "x" no diamante indica um ou, exclusivo, ou uma decisão). Se uma proposta é oferecida, o inicializador tem uma escolha de aceitação ou rejeição da mesma. Quando o participante recebe uma proposta, ele irá informar o inicializador sobre a execução da proposta. Em contra partida, o inicializador pode cancelar a execução da proposta a qualquer momento. [Odell, 2000].

Na Figura 4.1, acima, cada caixa na dimensão horizontal representa um papel exercido pelo agente, conforme as interações e colaborações estabelecidas entre os mesmos. Para se representar adequadamente o papel de um agente em um diagrama de protocolo, deve-se seguir a semântica e as notações estabelecidas pela extensão UML. A seguir detalhar-se-á estes elementos.

#### 4.4 O Papel do Agente

Agentes podem realizar papéis diferentes dentro de um protocolo de interação. Usando o protocolo *contract-Net*, exemplificando uma transação entre um comprador e um vendedor de um produto, o inicializador do protocolo assume o papel de comprador e o participante assume o papel de vendedor, mas o vendedor pode assumir o papel de agente varejista, o qual age como vendedor em um caso e comprador no outro caso, isto é, agentes satisfazendo papéis distintos podem suportar classificação múltipla e dinâmica. Portanto, em UML essa distinção de agentes pode ser [Bauer, 1999]:

- **Classificação múltipla** – é aquela em que o agente pode assumir diversos papéis. Como exemplo tem-se o Varejista, onde o agente varejista age como um comprador, bem como um agente vendedor.
- **Classificação dinâmica** – é aquela em que um agente pode mudar sua classificação durante sua existência.

Portanto, como viu-se, um agente pode assumir papéis distintos em um diagrama de protocolo.



#### 4.4.1 Semântica

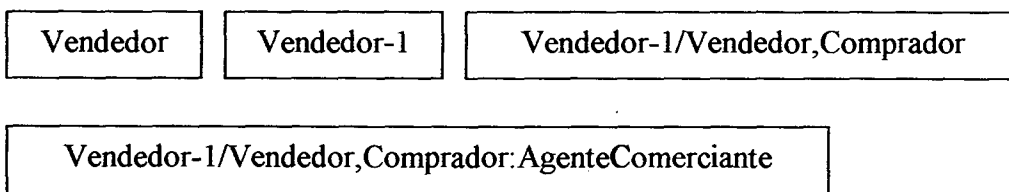
Um papel de agente descreve duas variações distintas, as quais podem ser aplicados na definição de um protocolo. Um protocolo pode ser definido entre instâncias concretas de agentes ou conjunto de agentes satisfazendo um conjunto de papéis e/ou classes.

#### 4.4.2 Notação

O papel do agente é mostrado como um retângulo. No retângulo coloca-se o nome do papel seguindo uma das formas:

- “role” denotando um arbitrário agente
- “instance / role-1 ... role-n” denotando instancias distintas de agentes  
“agentroles / role-1 ... role-n” com  $n \geq 0$
- “instance / role-1 ... role-n: class” denotando instancias distintas de agentes  
“Papel agentes / role-1 ... role-n” com  $n \geq 0$  e seguida da classe.

Exemplos:

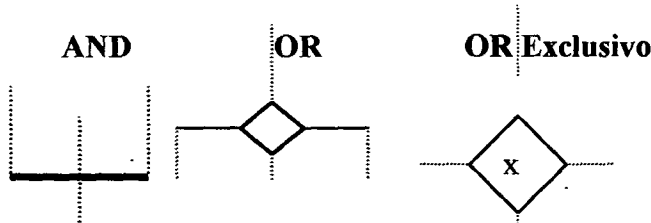


#### 4.4.3 O Ciclo de Vida do Agente

O ciclo de vida de um agente define o período de tempo enquanto o agente existe. Por exemplo: um agente usuário é criado quando um usuário conecta-se no sistema e o agente usuário é destruído quando o usuário desconecta-se do sistema. Outro exemplo é quando um agente migra de uma máquina para outra. O ciclo de vida inicia quando o agente é criado e finaliza quando o mesmo é destruído. [Bauer, 1999].

O ciclo de vida pode ser dividido para descrever decisões AND e Or paralelas e pode juntar-se em um ponto subsequente. Cada traço separado corresponde a um fluxo de mensagem. [Bauer, 1999]. AND – inicia com uma barra grossa, o OR – inicia com uma barra grossa com um símbolo de losango sem o “x”; a decisão representa o (OR exclusivo) – inicia com uma barra grossa como um símbolo de losango com o “x” dentro. Veja a seguir exemplos de representação dessas notações.

Exemplos:



**Figura 4.2 Representando o fluxo de mensagens (decisões).**  
[Bauer, 1999]

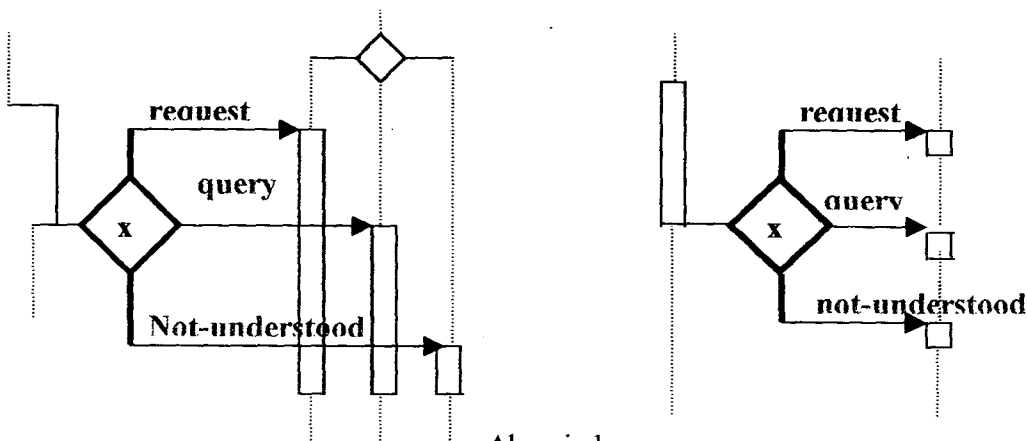
#### 4.5 Encadeamentos de Interações

O envio de mensagens pode ser feito de um modo paralelo ou como uma decisão entre procedimentos comunicados. O recebimento de diferentes procedimentos usualmente resulta em diferentes ações e diferentes respostas, isto é, a ação de um agente depende da mensagem recebida. [Bauer, 1999].

O encadeamento de interações mostra o período durante o qual o agente está realizando alguma tarefa em reação a uma mensagem recebida. Ele representa somente o tempo de duração da ação, mas não o controle relacionado entre o comunicador e o receptor da mensagem. O encadeamento de interações está sempre associado com o ciclo de vida do agente. [Bauer, 1999].

##### 4.5.1 Notação

O encadeamento de interações é mostrado com um retângulo grande e delgado cujo topo está alinhado com o seu tempo inicial e cuja base está alinhada com seu tempo final. Ele representa o ciclo de vida do agente. A tarefa realizada pode ser etiquetada no encadeamento ou na margem esquerda dependendo do estilo utilizado. [Bauer, 1999]. Pode ver as opções de representação na Figura 4.3 a seguir.



Abreviado como

**Figura 4.3 Representação de encadeamentos de Interações.** [Bauer, 1999].

## 4.6 Mapeando as Mensagens

O principal propósito do protocolo é a definição dos modelos de comunicação, especialmente o envio de mensagens de um agente para outro. O envio de mensagens pode ser feito de diferentes formas, com diferentes cardinalidades, dependendo de algumas restrições ou o uso de AND/OR paralelos e decisões. [Bauer, 1999].

### 4.6.1 Semântica

Uma mensagem ou envio de procedimento de ação é uma comunicação de um agente para outro que transporta informações com a expectativa que o agente receptor poderá reagir de acordo com a semântica estabelecida no procedimento informado. A especificação do protocolo não diz nada sobre a implementação do processamento do procedimento. [Bauer, 1999].

### 4.6.2 Notação utilizada na representação do mapeamento de Mensagens

Uma mensagem enviada é exibida com uma seta horizontal sólida de um encadeamento de interação de um agente para outro encadeamento de interação de outro agente. Uma mensagem pode ser enviada de um agente para ele mesmo, nesse caso a seta inicia e termina no próprio ciclo de vida do agente. [Bauer, 1999].

Cada seta é etiquetada com a mensagem. A etiqueta de mensagem tem a seguinte sintaxe:

**Predecessor** *condição-de-guarda* **expressões de consequência** *procedimento* **lista de argumentos**

Onde

**Predecessor** – consiste de um ou mais número natural seguido por uma contra-barras (“/”) definindo a seqüência de construção ou o número de parâmetros de entrada ou de saída no contexto do encadeamento de interações. [Bauer, 1999].

**Condição-de-guarda** – é uma condição de proteção com a semântica usual da UML, na qual a mensagem somente é enviada se a guarda é verdadeira. [Bauer, 1999].

**Expressão de consequência** – a expressão de consequência é uma restrição. Especialmente “n.m” indica que a mensagem está sendo enviada n vezes para m tempos com  $n \in \mathbb{N}$ ,  $m \in \mathbb{N} \cup \{*\}$ , o asterisco indica tempo arbitrário. “*broadcast*” indica um envio de mensagem *broadcast* e “*deadline*”, um código *string* de tempo de acordo com ISO 8601, indicando o deadline até o momento em que uma mensagem tenha sido recebida. [Bauer, 1999].

**Procedimento de ação** – é o nome de uma concreta instância de um procedimento de comunicação. O nome da instância do procedimento e sua comunicação é escrita como “nome:procedimento” ou somente o nome do procedimento. Por exemplo, “*commit*”. [Bauer, 1999].

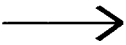
**Lista de argumentos** – é a lista de argumentos dentro de parênteses separados por vírgula. O parêntese pode ser omitido se a lista for vazia. Cada argumento é uma expressão em pseudocódigo ou em linguagem de programação apropriada ou uma expressão linguagem de comunicação de objetos (LCO) . [Bauer, 1999].

## 4.7 Opções de representação de Mensagem

### 4.7.1 Cardinalidade


A cardinalidade de uma mensagem indica que há “n” remetentes e “m” receptores de uma mensagem e é denotada escrevendo-se números naturais no início da seta e no final da seta. Isto somente é seguido se o remetente e/ou receptor não é uma instância de um agente.

#### 4.7.1.1 Mensagem assíncrona:

 é representada com uma seta delgada. Mostra como uma mensagem é passada de um objeto para outro sem descrever qualquer detalhe sobre a comunicação.

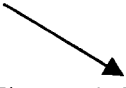
**Figura 4.4**

#### 4.7.1.2 Mensagem síncrona:

 é representada com uma seta sólida. Em um sistema concorrente, uma mensagem síncrona representa uma linha de controle com semântica de espera, ou seja, o agente remetente espera indefinidamente pela aceitação da mensagem pelo agente destinatário antes de continuar seu processamento.

**Figura 4.5**

#### 4.7.1.3 Mensagem de Intervalo:

 indica que o agente remetente esperará até que o agente destinatário esteja pronto para receber a mensagem por um período de tempo determinado antes de abortar o processo de transmissão da mensagem e continuar com seu processamento.

**Figura 4.6**

## Exemplo de representação de Mensagem

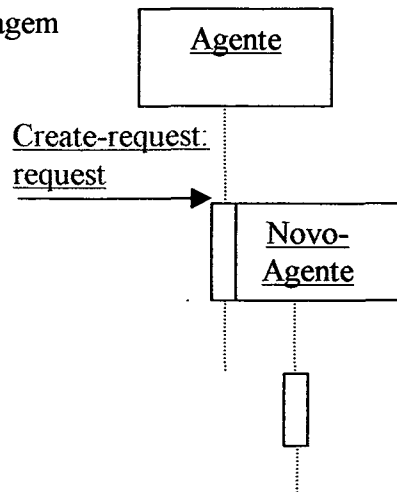


Figura 4.7 Representando uma Mensagem. [Bauer, 1999].

#### 4.8 Protocolos aninhados

Para especificar sistemas complexos de uma forma modular são introduzidos protocolos aninhados. Além disso, o uso de parte de uma especificação aumenta sua legibilidade. Adicionalmente, eles podem ser usados para definição de repetição de protocolo de acordo com as condições-de-guarda e restrições. [Bauer, 1999].

##### 4.8.1 Semântica

A semântica de um protocolo aninhado é a mesma do protocolo.

Se o protocolo aninhado é realizado com alguma condição-de-guarda então a semântica deste protocolo é a mesma semântica do protocolo inferior onde a condição-de-guarda foi avaliada como verdadeira (*true*), caso contrário, a semântica é a mesma de um protocolo vazio, isto é, nada foi especificado. [Bauer, 1999].

Se o protocolo aninhado é realizado com alguma restrição, este é repetido até enquanto a restrição seja avaliada como verdadeira (*true*). [Bauer, 1999].

Descrever-s-á a seguir cada um dos possíveis compartimentos: [Bauer, 1999].

- **Compartimento do Nome** – é o compartimento do nome do protocolo. O nome do protocolo é escrito no canto esquerdo do compartimento.
- **Compartimento de condição-de-guarda** – nele é representada a condição-de-guarda. Uma condição-de-guarda obedece a seguinte notação [condição-de-

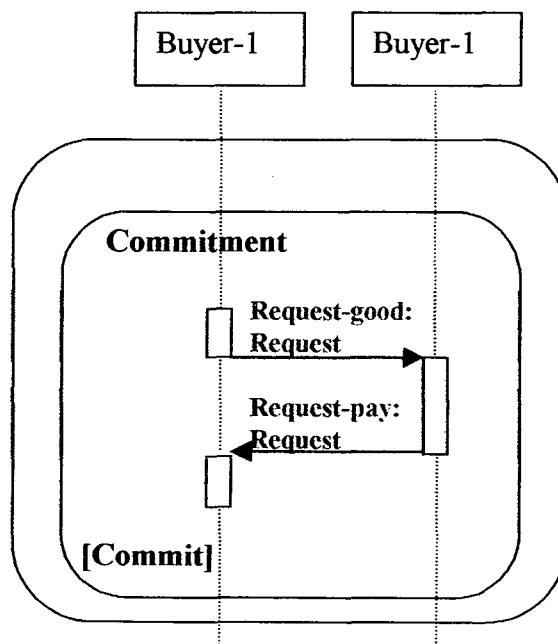
guarda]. A condição-de-guarda é escrita junto com a restrição embaixo no canto inferior esquerdo do retângulo.

- **Compartimento da restrição** – nele é representada a restrição. Uma restrição é escrita obedecendo a seguinte sintaxe: {restrição}. Em UML uma restrição pode usar a descrição n..m, para indicar que o protocolo é repetido de n a m vezes com  $n \in \mathbb{N}$ , e  $m \in \mathbb{N} \cup \{*\}$ , onde o asterisco indica tempo arbitrário, e é usado como uma restrição.

#### 4.9.3 Representação de Protocolos aninhados

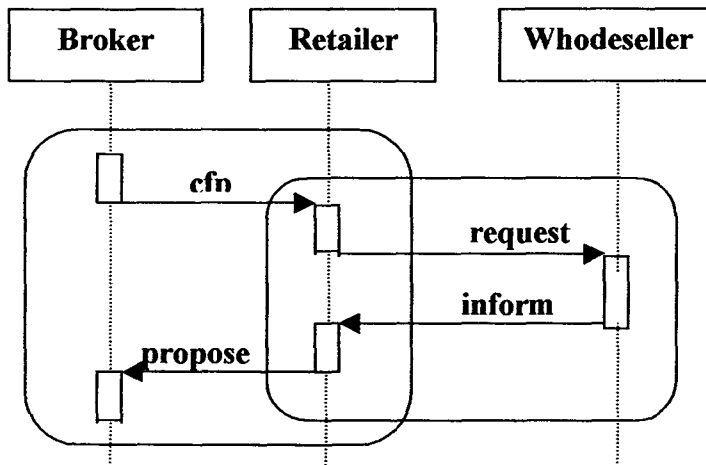
A Figura 4.8, a seguir, exemplifica a representação de um protocolo aninhado. O diagrama da Figura 4.1 pode ser usado tanto para mostrar um protocolo simples, bem como protocolos parametrizados: os protocolos parametrizados instanciados. Pode-se observar na figura 4.1 que um protocolo está totalmente dentro de outro protocolo. [Bauer,1999].

A seguir será representado um outro tipo de protocolo aninhado: o protocolo entrelaçado.



**Figura 4.8 Representação de um Protocolo Aninhado.**  
[Bauer, 1999].

A Figura 4.9, a seguir, representa um protocolo aninhado de forma entrelaçada. Num protocolo entrelaçado, enquanto um protocolo realiza algumas interações, o outro pode ser processado.

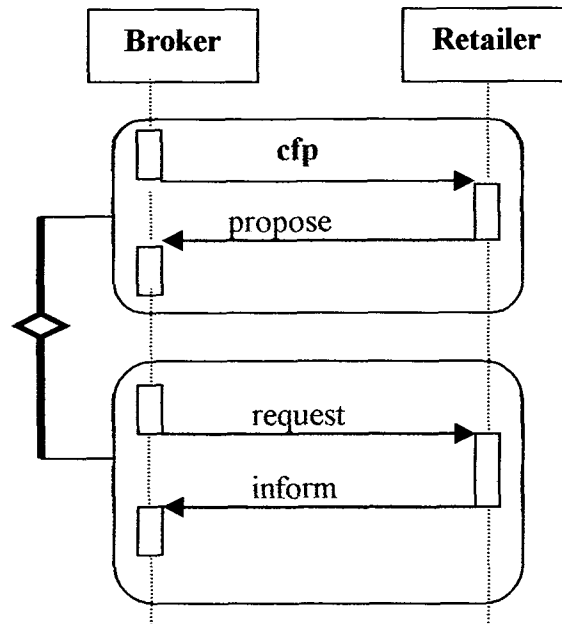


**Figura 4.9** Representação de um Protocolo aninhado entrelaçado. [Bauer, 1999].

Uma completa especificação de protocolo para uma intertransição do agente requer discussões de intra-atividade do agente e é suportado pelo Diagrama de atividade e mapa de Estado UML. [Odell, 2000].

#### 4.10 Protocolos Aninhados Complexos

Um Protocolo Aninhado Complexo define as combinações de decisão (AND, OR e OR exclusivo) do protocolo. Ele se encarrega do encadeamento das interações do início até o final do protocolo. A Figura 4.10 a seguir representa um protocolo aninhado complexo.



**Figura 4.10 Representação de um Protocolo Aninhado Complexo.**  
[Bauer, 1999].

#### 4.11 Threads de Interações e Mensagens dentro e fora de protocolo aninhado

Os encadeamentos de interação (*Thread*) e mensagens que ocorrem entre protocolos encadeados definem os parâmetros de entrada (*input*) e de saída (*output*) para esses protocolos.

Os parâmetros de entrada pertencem aos encadeamentos de interações e estão do lado de fora do protocolo e são passados para as mensagens dentro do protocolo.

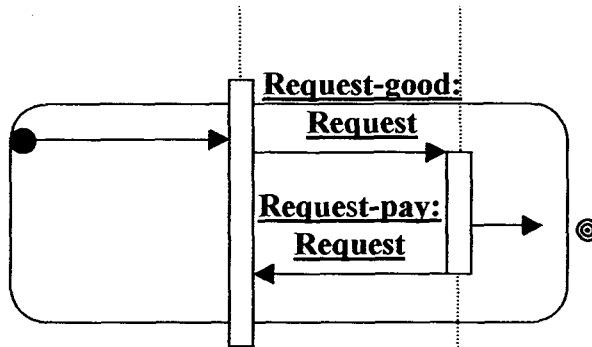
Os parâmetros de saída estão nos encadeamentos de interação dentro do protocolo e são enviados para as mensagens fora do protocolo. Uma mensagem ou um encadeamento de interação, com seus parâmetros de entrada e saída, inicia e finaliza o protocolo descrevendo uma completa conexão do diagrama de protocolo.

##### 4.11.1 Notação

Os parâmetros de entrada e saída para os encadeamentos de interação de um protocolo encadeado são mostrados como um retângulo fino e comprido o qual é desenhado da linha de topo do protocolo até a base desse protocolo.

Os parâmetros de entrada e saída das mensagens são mostrados como seta iniciando com um pequeno círculo sólido e finalizando com um pequeno círculo sólido envolto por outro círculo. Pode-se ver isso na Figura 4.11 a seguir.





**Figura 4.11 Encadeamento de Interações e Mensagens em Protocolos Encadeados. [Bauer, 1999]**

## 4.12 Protocolos Parametrizados

Um Protocolo Parametrizado é a descrição para um protocolo com um ou mais parâmetros formais. Eles, portanto, definem uma família de protocolos. Cada protocolo é especificado pela ligação dos valores atuais dos parâmetros. Tipicamente os parâmetros representam os papéis dos agentes, restrições, instâncias de procedimentos de ação e protocolos aninhados.

Um protocolo parametrizado não é um protocolo que possa ser usado diretamente. Ele só é usável quando os valores de seus parâmetros são atualizados.

### 4.12.1 Representação

Os parâmetros de um protocolo parametrizado são colocados em uma caixa retangular de linhas tracejadas no canto superior direito do protocolo. A lista de parâmetros não deve ser vazia, embora ela possa ser suprimida de sua representação.

Procedimentos de ação podem ser marcados com asterisco (\*), indicando diferentes tipos de mensagens as quais podem ser alternativamente enviadas.

A Figura 4.12, a seguir, é um exemplo de um protocolo parametrizado.

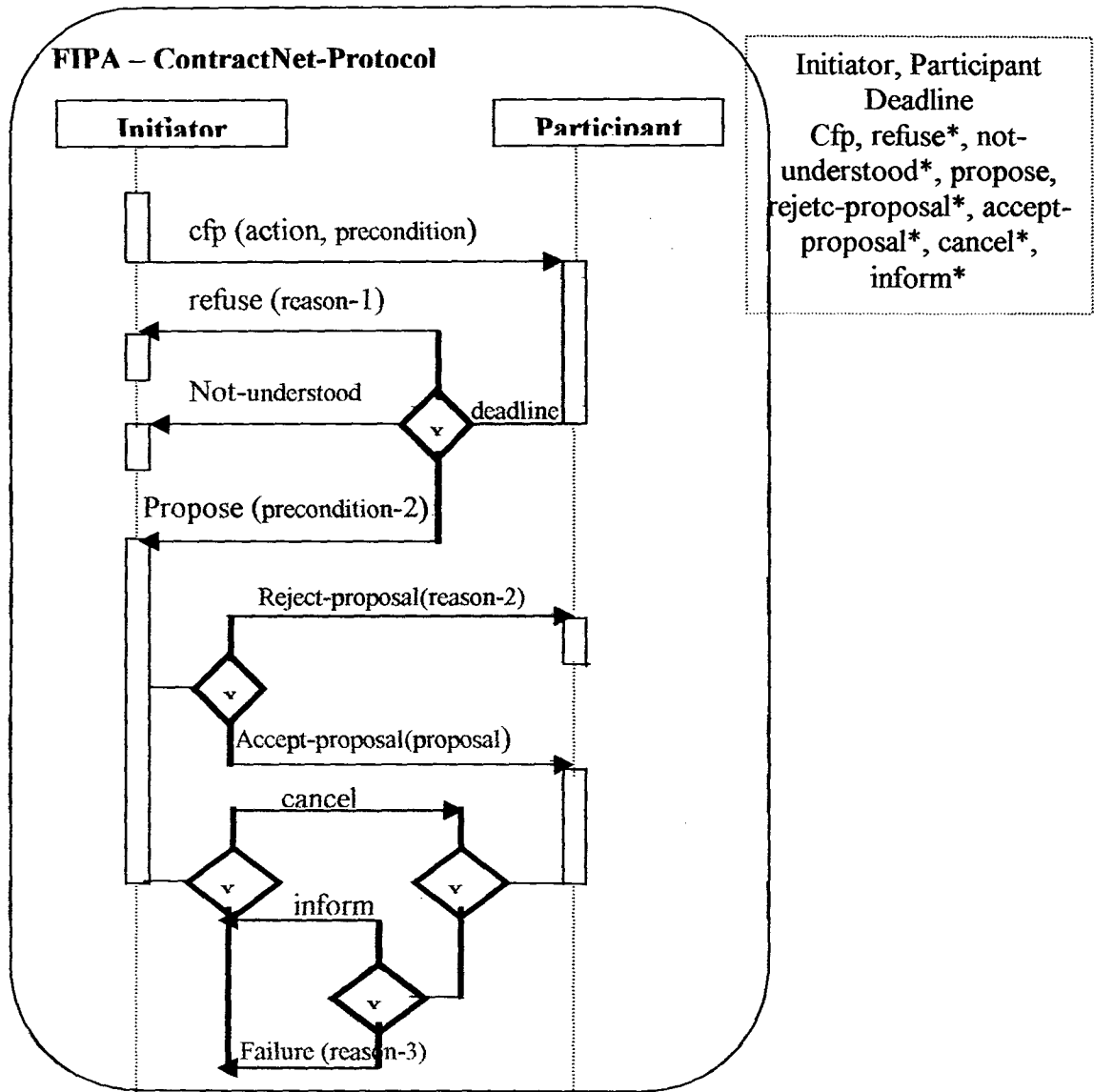


Figura 4.12 Representação de um Protocolo Parametrizado. [Bauer, 1999].

Um Diagrama de Protocolo Parametrizado não pode ser usado diretamente em uma descrição de interação normal, porque seus parâmetros não têm significado fora do escopo de declaração dos parâmetros. Para ser usado, devem ser atribuídos aos seus parâmetros os valores atualizados.

A notação para representação de um protocolo parametrizado obedece a seguinte sintaxe:

**Nome-do-protocolo-parametrizado "<" lista-de-valores ">"**

Onde:

Lista-de-valores é uma lista não vazia separados por virgula.

O número e tipos dos valores devem ser iguais com o número e tipos dos parâmetros indicados no protocolo parametrizado.

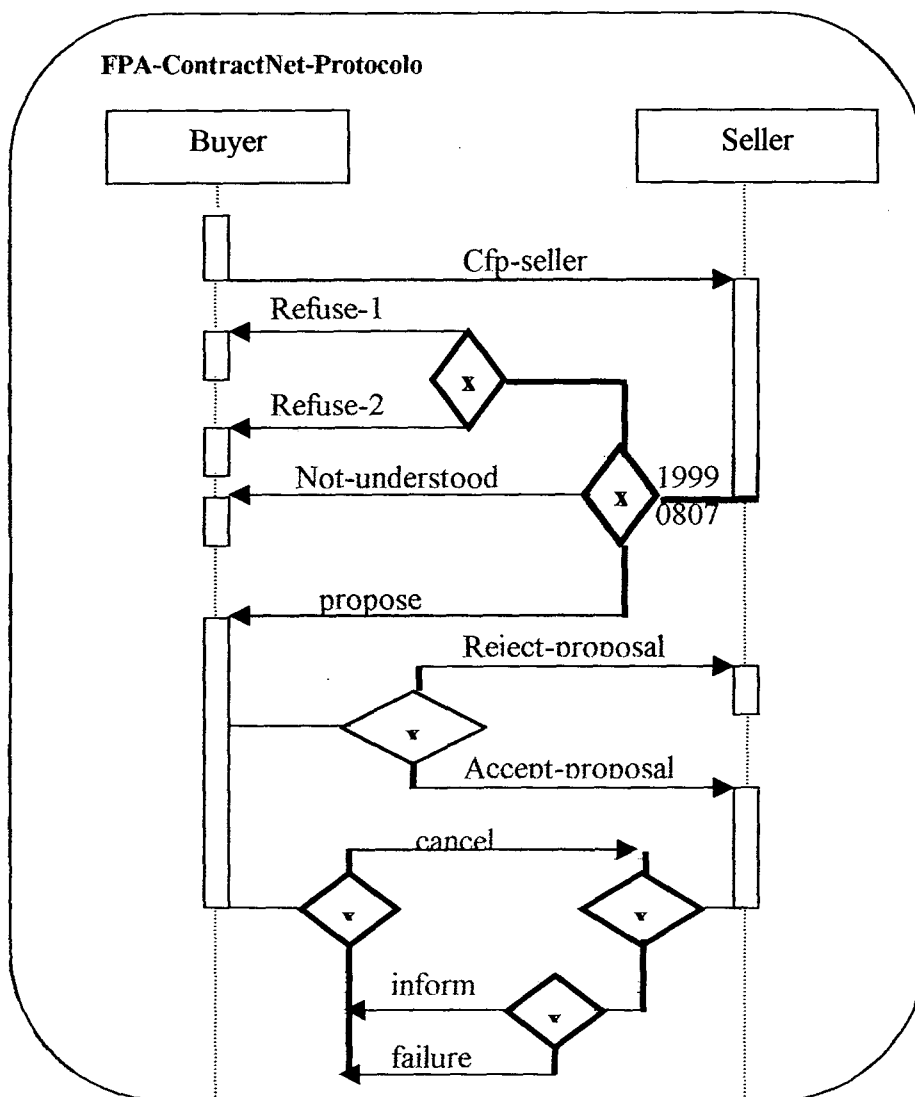
Exemplo:

```
FIPA-ContractNet-Protocolo <  
Buyer, Seller  
199990807120000  
cfp-seller, refuse-1, refuse-2, not-understood, propose, reject-  
proposal,  
accept-proposal, cancel, inform, failure  
>
```

Segundo Odell, (1999), a Figura 4.12 ilustra um tipo de ação que pode servir como solução para problemas em domínio análogo. Para ele, isso é verdadeiramente um modelo, ou seja, um gabarito (*template*), otimizando o pacote que deve ser suportado. Na Figura 4.10, a seguir, aplica-se o protocolo *contractNet* para um cenário particular envolvendo compradores (Buyers) e vendedores (Sellers). Sendo assim, pode-se notar que o agente Inicializador (Initiator) e o Participante (Participant) assumem os papéis de Buyers e Sellers, e a chamada para a proposta (call-for-proposal) torna-se o cfp (seller-cfp). Também nesse cenário, há duas formas de refugar a proposta e, por ultimo, dar uma resposta (inform) para o buyer. Observa-se também nesse diagrama do protocolo, que a qualquer momento o Buyer pode cancelar (cancel) o pedido.

Por ser um modelo, o gabarito (*template*) não pode ser usado diretamente, ele precisa ser instanciado com seus respectivos valores atualizados. Pode-se ver a seguir, a instância desse protocolo para os valores indicados no quadro acima.

Constata-se que o diagrama de protocolo, apresentado pela Figura 4.13, para representar as interações entre os agentes, é um diagrama de seqüência conforme a UML. Esse tipo de diagrama enfatiza a seqüência cronológica de comunicação. segundo Odell, (1999) pode ser utilizado outros tipos de diagramas UML para representar a comunicação entre agentes, tais como: diagrama de colaboração, que enfatiza a associação entre os agentes e, diagramas de atividade e mapas de Estado, que dão ênfase ao fluxo de processamento.

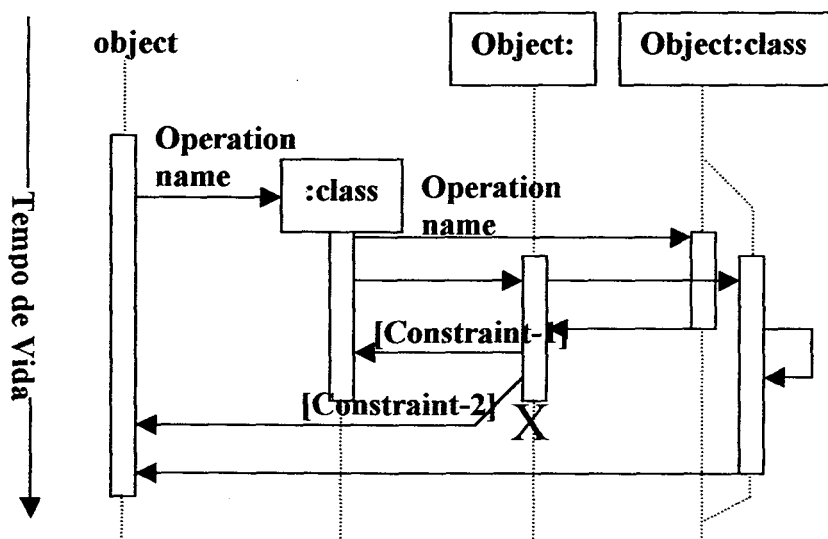


**Figura 4.13** Instância do Protocolo FIPA-ContractNet. [Bauer, 1999]

A Figura 4.13 também descreve alguns elementos básicos da comunicação entre os agentes. A caixa retangular tanto pode representar um agente individual como um conjunto de agentes, isto é, papéis ou classes de agentes. Por exemplo, um agente individual poderia ser rotulado como Bob/Cliente. Aqui Bob é uma instância de um agente assumindo o papel de Cliente. Bob poderia também assumir o papel de fornecedor, empregado, ou proprietário de acordo com a sintaxe já apresentada em seção anterior. Como pode-se observar, esta figura também mostra os procedimentos de interações concorrentes bem como as decisões a serem tomadas pelos agentes, além da seqüência do fluxo de interação, que é a principal característica do Diagrama de Seqüência.

### 4.13 Diagrama de Seqüência

Nas seções anteriores, representou-se um Agente de Protocolo de Interação (PIA) como um diagrama de seqüência. Um Diagrama de Seqüência dá ênfase à ordenação temporal das mensagens. Eles têm duas características que os diferem dos diagramas de colaboração: primeiro, existe linha de vida do objeto. Essa é uma linha tracejada verticalmente que representa a existência de um objeto em um período de tempo. A linha de vida inicia quando um objeto é estereotipado, ou seja, quando o mesmo é criado. Sua linha de vida termina quando o objeto é destruído. Nesse caso, é fornecida uma indicação visual com um grande x, marcando o fim de sua vida. Segundo, existe o foco de controle. O foco de controle é um retângulo alto e estreito, que mostra o período durante o qual um objeto está desempenhando uma ação. A parte superior é alinhada com o início da operação; a parte inferior é alinhada com sua conclusão. Na Figura 4.14, a seguir, mostrar-se-á a sintaxe básica para representar diagramas de seqüências. [Booch, Rumbaugh, Jacobson, 2000].

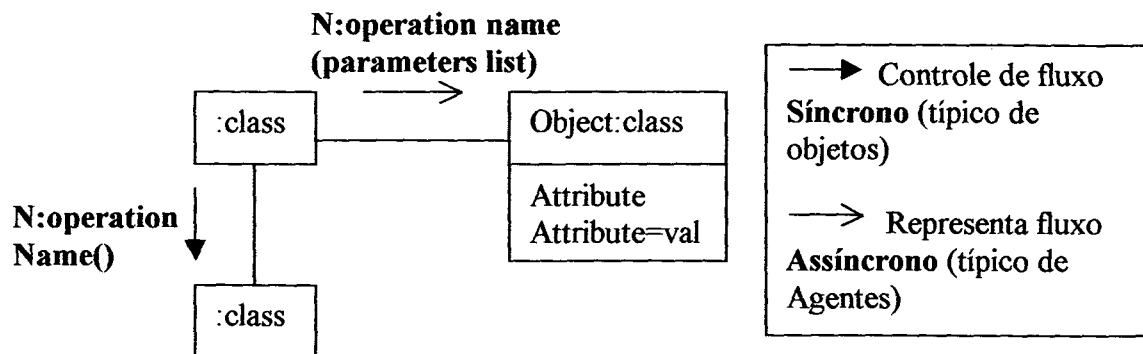


**Figura 4.14** Sintaxe básica para representação de Diagrama de Seqüência. [ERIM CEC e James Odell].

### 4.14 Diagrama de Colaboração

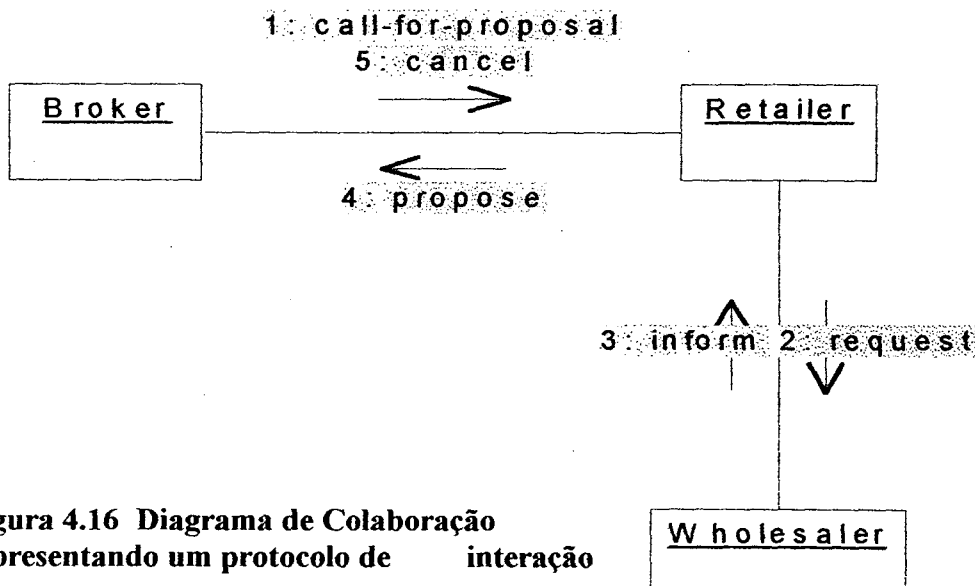
Como já citado anteriormente, o Diagrama de Colaboração também pode ser usado para representar um modelo de interação entre agentes. Uma das distinções do diagrama de colaboração é que o agente (o retângulo) pode ser colocado em qualquer lugar no diagrama; considerando o fato de que no diagrama de seqüência os agentes são dispostos na horizontal no topo do diagrama. Num diagrama de colaboração, as interações são numeradas em uma seqüência; enquanto no diagrama de seqüência elas são lidas de cima para baixo. Semanticamente eles são equivalentes; graficamente eles são similares. Pode-se utilizar os

diagramas de colaboração para visualização de processos na perspectiva de um diagrama de classes e assim, expressar ações simples, concorrentes, seqüências ambíguas, etc. A utilização de um ou do outro depende do modelo de interação descrito e do modelo que forneça o melhor entendimento. A Figura 4.15, a seguir, mostra a sintaxe básica para representar um digrama de colaboração



**Figura 4.15** Sintaxe básica para Diagrama de Colaboração. [ERIM CEC e James Odell, 1999].

A Figura 4.16 abaixo mostra um Diagrama de Colaboração para um protocolo de interação entre negociador ( Broker), um varejista (Retailer) e um vendedor (Wholesaler) para uma proposta de negociação de mercadorias.



**Figura 4.16** Diagrama de Colaboração representando um protocolo de interação

#### 4.15 Diagrama de Atividade

Protocolos de interação entre agentes muitas vezes requerem uma especificação de processamentos encadeados com semântica muito clara. O diagrama de atividade pode ser utilizado para representar processamentos concorrentes. O diagrama de atividade difere do diagrama de interação (diagrama de seqüência e colaboração), porque aquele fornece controle de *Thread* explícito. Isto é útil para representar protocolos de interações complexas que envolvam processamento concorrente. [Odell, 1999].

Segundo Odell, (1999), o diagrama de atividade é semelhante em essência às redes de Petri em várias formas. Primeiro, o diagrama de atividade é uma representação gráfica que permite, de forma simples, visualizar os processos. Segundo, o diagrama de atividade pode representar concorrência e processos assíncronos. Por último, eles podem expressar comunicação simultânea com vários correspondentes. Ainda, segundo esse autor, a principal diferença entre os dois métodos é que o diagrama de atividade é formalmente baseado em modelos de mecanismo de estado definidos na UML. A Figura 4.17, a seguir, mostra a sintaxe básica para representação de Diagramas de Atividades.

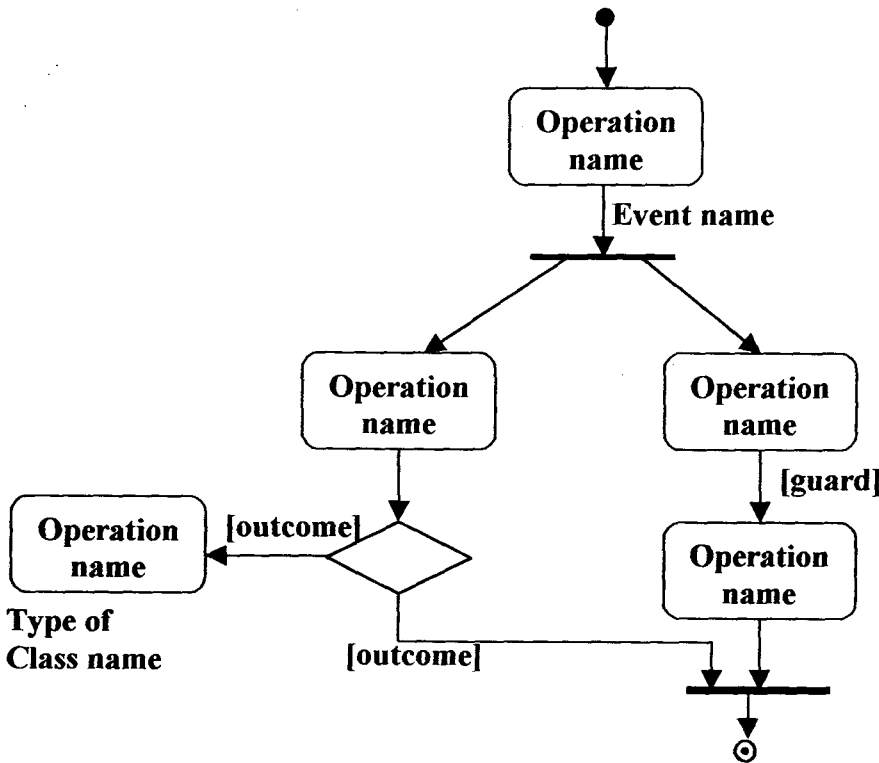
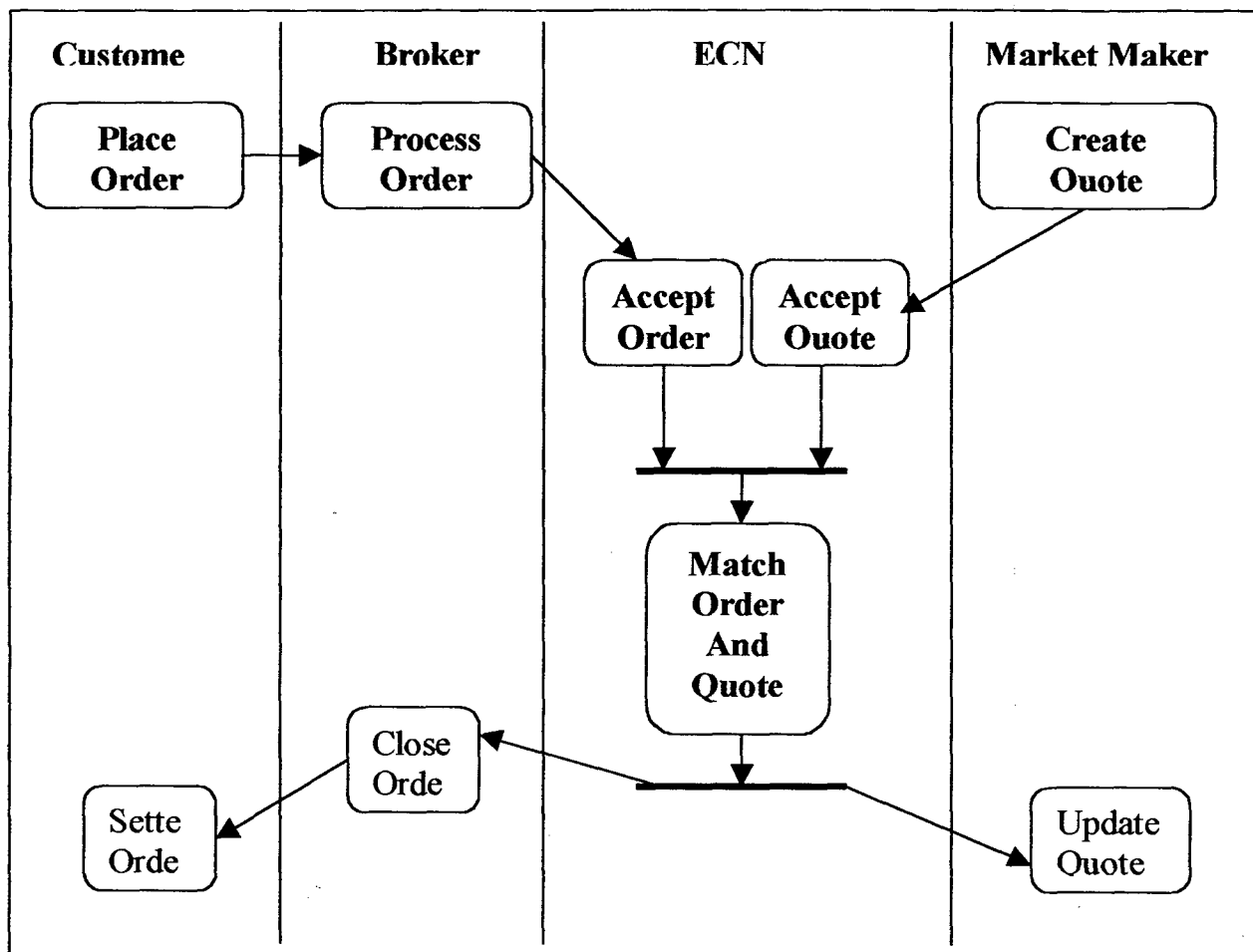


Figura 4.17 Sintaxe básica para representação de Diagrama de Atividade. [ERIM CEC e James Odell, 1999].

Na Figura 4.18, a seguir, tem-se um exemplo de um protocolo de interação entre vários agentes para processar um pedido. Nesse exemplo, um agente Cliente solicita um

produto a um negociante, que dispara um evento para processar o pedido. A proposta é encaminhada para um agente ECN, mas o ECN só pode fechar o pedido se o fabricante tiver informado a cota do produto. Observe que existem processos concorrentes como processar pedido e criar cota, e, fechar pedido e atualizar cota, etc.



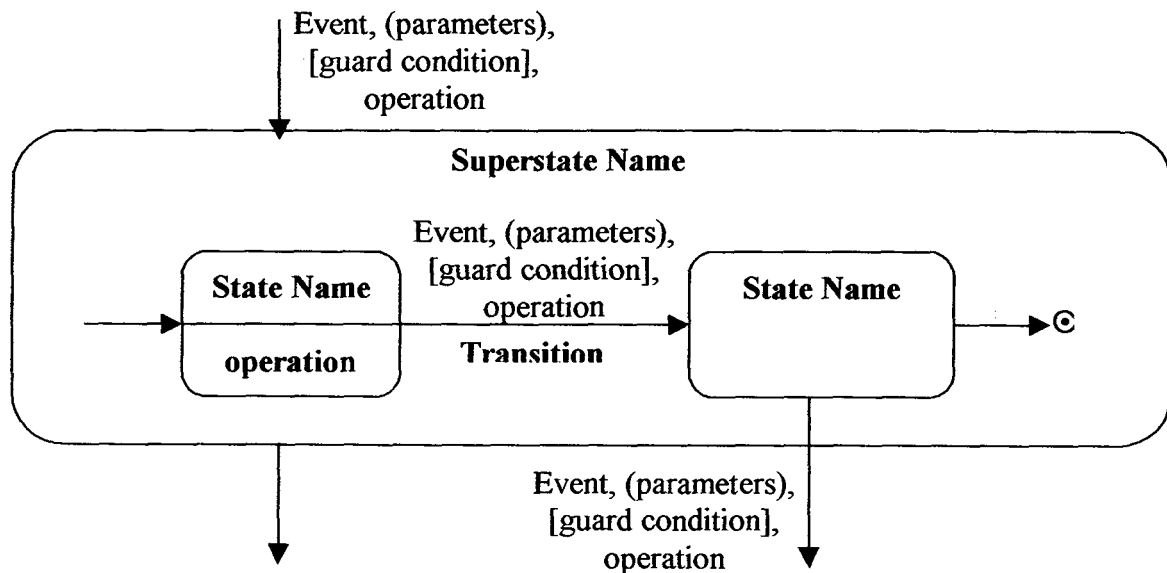
**Figura 4.18** Diagrama de Atividade para um Protocolo de venda entre vários agentes [Odell, 1999].

#### 4.16 Gráficos de Estado

Um outro processo relacionado com os diagramas da UML é o Gráfico de Estado. Um Gráfico de Estado é um gráfico que representa uma máquina de estado. As máquinas de Estado são empregadas para a modelagem dos aspectos dinâmicos do sistema e, isso envolve a especificação do tempo de vida das instâncias de um objeto, um caso de uso ou um sistema inteiro. As instâncias dos objetos poderão responder a eventos. Quando um evento ocorre, alguma atividade acontecerá, dependendo do estado do objeto. Uma atividade é uma execução não-atômica em andamento em uma máquina de estado. O estado



de um objeto é uma condição ou situação da vida de um objeto durante a qual ele satisfaz alguma condição, realiza alguma atividade ou aguarda algum evento. Um evento é uma especificação de uma ocorrência significativa que tem localização no tempo e no espaço. Em relação a uma máquina de estado, um evento é uma ocorrência de um estímulo capaz de ativar uma transição de estado. Por sua vez, uma transição é um relacionamento entre dois estados, indicando que um objeto no primeiro estado realizará certas ações e entrará no segundo estado quando um evento especificado ocorrer e quando condições especificadas sejam satisfeitas. Estados são representados como retângulos com cantos arredondados, enquanto as transições são geralmente conduzidas pelos arcos que conectam os estados. A Figura 4.19, a seguir, mostra a sintaxe básica para se representar um gráfico de estado. [Booch, Rumbaugh, Jacobson, 2000].



**Figura 4.19 Sintaxe Básica para Representação Gráfico de Estado.**  
[ERIM CEC e James Odell]

Pode-se visualizar uma máquina de estado de duas formas: dando-se ênfase ao fluxo de controle de uma atividade para outra ou dando-se ênfase aos estados potenciais dos objetos e às transições entre esses estados. No primeiro caso, utilizam-se os diagramas de atividades para representar a máquina de estado e, no segundo, utiliza-se os gráficos de estado.

A Figura 4.20 mostra um exemplo de um Gráfico de Estado para um protocolo de Pedido. Neste, se um pedido está em um estado *Request* (requisitado), um agente fornecedor (*Supplier*) pode submeter a requisição negociada – resultando em uma transação para um estado de negociação (Committed). Além disso, este diagrama indica que a ação de submeter o pedido pelo agente somente pode ocorrer, se o pedido está em um estado

requisitado (*Requested*). O estado *Requested* tem duas outras possibilidades de ação: o fornecedor pode refugar o pedido e o cliente pode voltar atrás e cancelar o pedido.

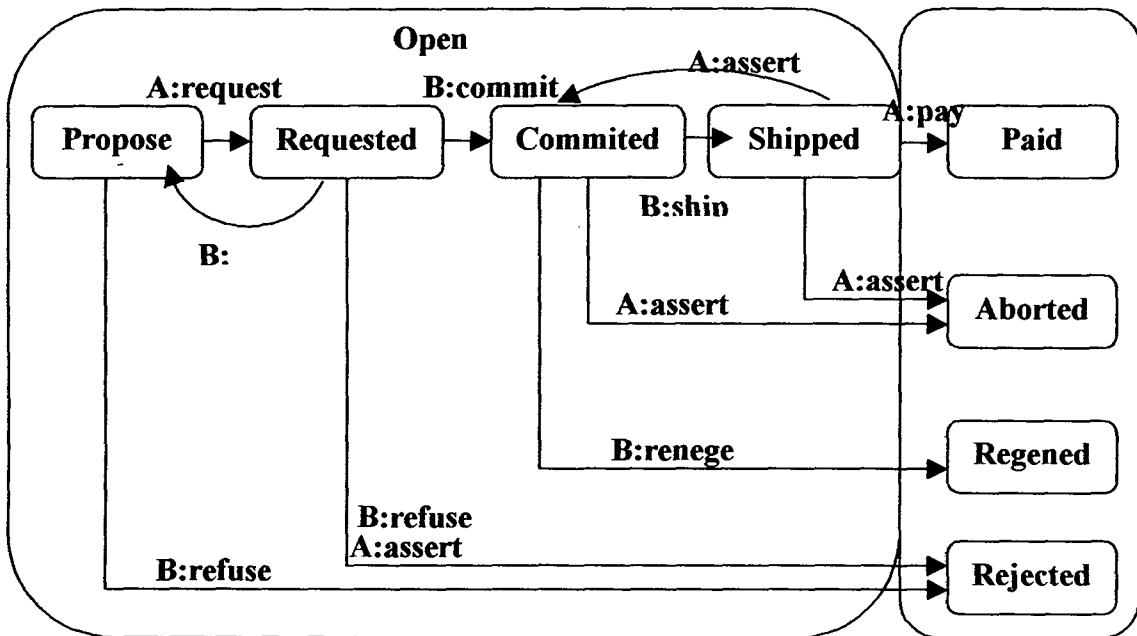


Figura 4.20 Gráfico de Estado para um Protocolo de Pedido. [Odell, 1999]

#### 4.17 Outras Considerações AUML

Nos itens anteriores examinou-se algumas extensões UML que podem ser usadas para expressar protocolos de interação entre agentes. A seguir, serão apresentadas algumas outras extensões AUML, que também podem ser empregadas para especificação de sistemas baseados em agentes.

##### 4.17.1 Especificações de Papéis

A especificação do papel que um agente pode executar no curso de suas interações com outros agentes é uma técnica vital na modelagem de sistemas baseados em agentes. A UML já dispõe de algumas técnicas que facilitam a representação de papéis e de mudanças de papéis de agentes no curso de suas interações. Isso pode ser feito, através dos diagramas de seqüência, onde se pode exibir cada agente com uma simples linha de vida, onde cada ativação (retângulo) é rotulada com um apropriado nome de papel, através dos diagramas de atividades, associando cada atividade com um nome de um apropriado papel, dentre outras técnicas. [Odell, Paranuka, Bauer, 1999].

### 4.17.2 Representando Papéis

A seguir será apresentada a sintaxe básica usada na UML para representar papéis e mudanças de papéis para agentes. [Odell, Paranuka, Bauer, 1999].

#### Sintaxe básica:

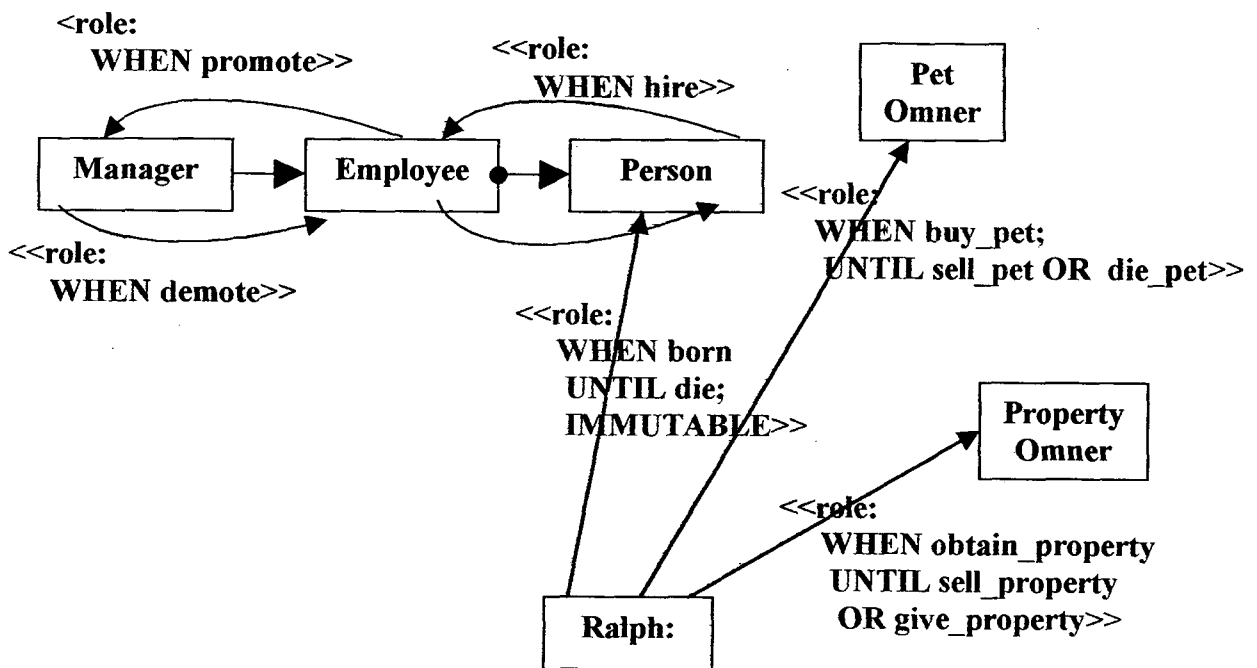
```

<<role:
  [role-name]
  WHEN <events>
  UNTIL <events>
  [;IMMUTABLE;
  MANDATORY]>>
<<role change (classify):
  FROM <type(s)>
  TO <type(s)> >>
  
```

**MANDATORY** - toda instância de Agente deve ter este papel.

**IMMUTABLE** - uma vez um agente ter recebido este papel, ele nunca pode perdê-lo.

A Figura 4.21, a seguir, exemplifica a mudança de papéis de agentes usando um Diagrama de Classes.



**Figura 4.21 Representando mudança de Papel usando Diagrama de Classe.**  
[Odell, Paranuka, Bauer, 1999].

#### 4.18 Pacotes

Os pacotes são empregados para apresentar diferentes visões da arquitetura de um sistema, bem como eles também podem ser usados para organizar os elementos de modelagem em conjuntos maiores que possam ser manipulados como grupo. Os pacotes bem estruturados agrupam elementos que estão próximos semanticamente e que tendem a se modificar em conjunto. [Booch, Rumbaugh, Jacobson, 2000].

Um pacote é um mecanismo de propósito geral para a organização de elementos em grupo. A figura 4.22, a seguir, mostra um exemplo de um modelo de uma extensão de pacotes UML para Agentes.

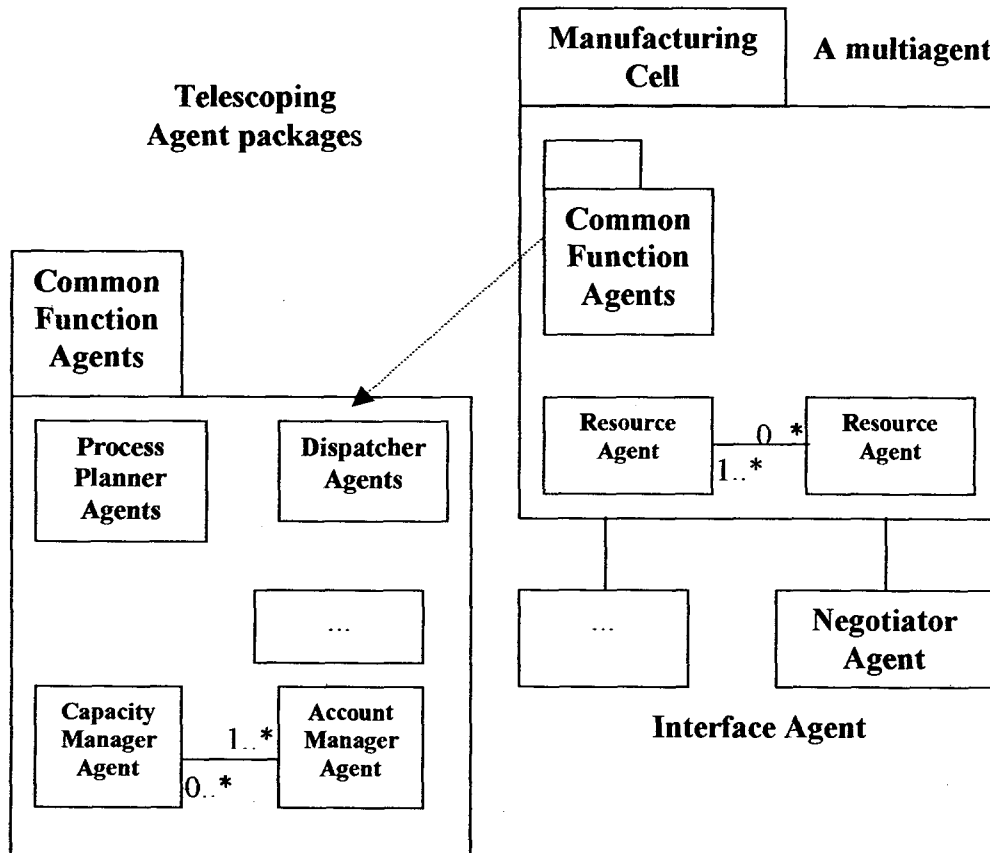


Figura 4.22 Extensão de Pacotes para Agentes.  
[Odell, 1999].

#### 4.19 Diagrama de Implementação

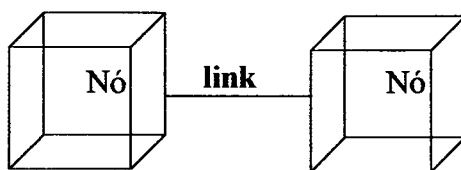
Quando se define a arquitetura de um sistema complexo de *software*, precisa-se considerar as dimensões lógicas e físicas do sistema. Na parte lógica, são definidos itens tais como classes, interações, colaborações, máquinas de estados, etc. Na dimensão física, encontram-se os componentes que representam o pacote físico desses itens lógicos e os nós

que representam o *hardware* onde esses componentes são instalados e executados. Em UML, um nó é um elemento físico que existe em tempo de execução e representa um recurso computacional geralmente com capacidade de processamento. [Booch, Rumbauch, Jacobsom, 2000].

Os diagramas de implementação descrevem os relacionamentos entre os componentes físicos de *software* e de *hardware*. Eles servem também para se ter uma visualização da configuração do sistema e pode ser usado também com diagramas de componentes. [Odell, 2000].

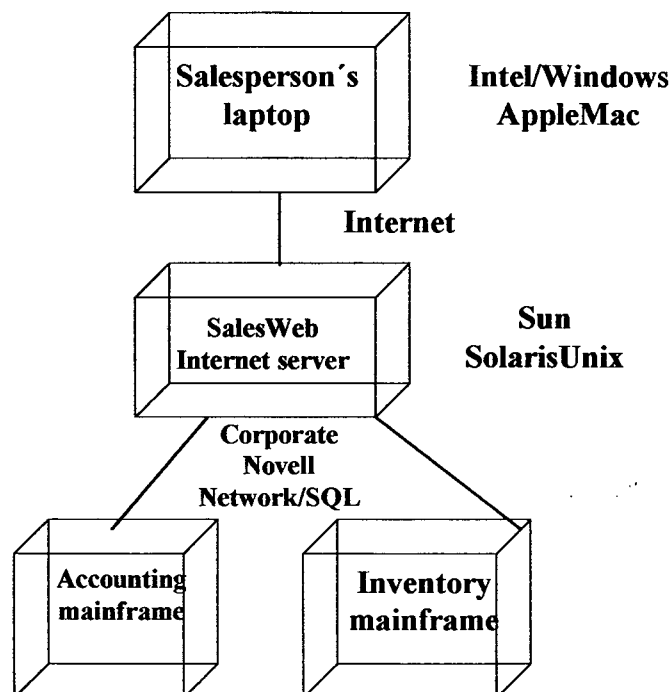
A Figura 4.23, a seguir, mostra a sintaxe básica para representação de diagrama de implementação.

#### Sintaxe Basica



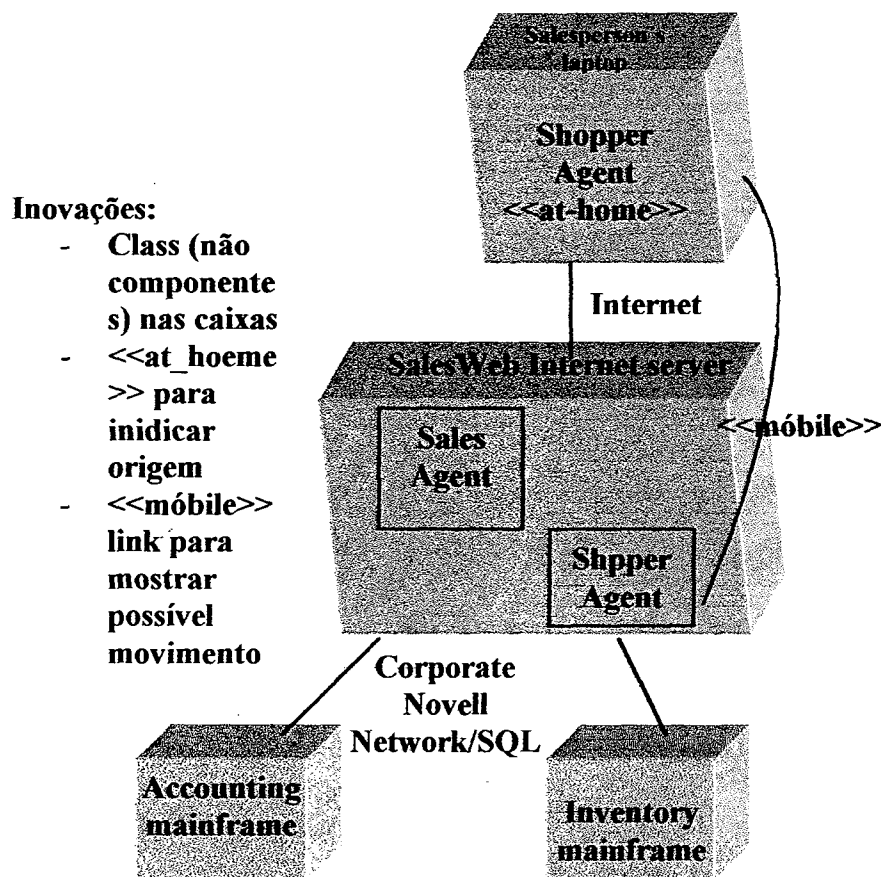
**Figura 4.23 Diagrama de Implementação.**  
[Odell, 2000].

Um exemplo de diagrama de implementação em UML é mostrado na Figura 4.24 abaixo.



**Figura 4.24 Exemplo de um Diagrama de Implementação usando UML.**

Segundo Odell, (1999), pode-se usar uma extensão UML de Diagrama de Implementação para representar Agentes. A Figura 4.25, a seguir, é um exemplo de um Diagrama de Implementação para um Agente Móvel, segundo esse autor.



**Figura 4.25 Extensão de Diagrama de Implementação UML para Agentes Móveis. [Odell, 1999].**

#### 4.20 Considerações Finais

Pode-se concluir que a UML fornece ferramenta para especificar protocolos de interação entre agentes, expressar o modelo de interação entre agentes dentro do protocolo, representar as ações internas de um agente, representar outras extensões de UML relacionadas com agentes que já são comumente usadas, tais como: especificação e mudanças de papéis, pacotes como interfaces de agentes, diagramas de implementação indicando a mobilidade, dentre outras citadas neste trabalho.

Apesar de tudo isso, a UML ainda apresenta algumas deficiências em seus diagramas para representar Agentes, ou seja, a UML não tem bloco de construção para

mobilidade; mudanças de papéis; funções gerativas; tais como clonagem, nascimento, reprodução; nem para parasitismos e simbiose; bem como para representar fenômenos emergentes. [Odell, 1999].

No entanto, algumas dessas ferramentas podem ser aplicadas diretamente para modelagem de sistemas baseados em agentes, adotando conversões simples. Em outros casos, pode-se adotar outras extensões UML que suportem as funções exigidas pelos agentes. Muitas dessas extensões propostas já são consideradas pela comunidade OO (Orientação a Objetos) como extensões úteis para desenvolvimento OO na UML 2.0.

No próximo capítulo tentar-se-á utilizar algumas das técnicas apresentadas aqui, para modelar e especificar um protótipo de um sistema, que utiliza Agentes Inteligentes de Software para selecionar serviços turísticos para clientes, de acordo com o perfil informado.

## Capítulo 5

### PROPOSTA DE UM PORTAL DE TURISMO UTILIZANDO AGENTES

#### 5.1. Introdução

O Protótipo proposto é um Portal de Turismo do Rio Grande do Norte. Ele tem como meta, oferecer aos visitantes um catálogo de serviços turísticos do Estado.

Os serviços oferecidos pelo portal são: Relação de hospedagens disponíveis nos principais pólos turísticos do Estado, os melhores passeios oferecidos por esses pólos turísticos, bem como, uma relação dos restaurantes, informações sobre a vida noturna na cidade do Natal e outras informações que poderão ser úteis aos possíveis visitantes.

Além dessas informações, o visitante do *site*, tem a opção de obter sugestões dos principais serviços turísticos, tais como: hospedagens, passeios, vida noturnas e restaurantes, que melhor adequar-se ao seu perfil. Para isso, existem em quase todas as telas do *site*, *links* para “Sugestões”. Ao clicar no *link* sugestões, será exibido para o visitante um formulário vazio para o mesmo informar o seu perfil. No capítulo 6 ter-se-á um exemplo demonstrativo desse formulário.

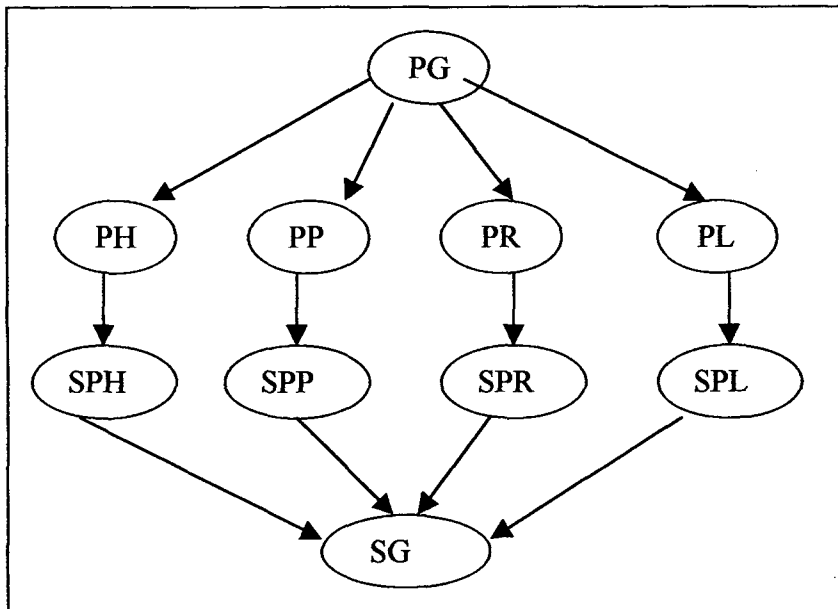
Quando o visitante do *site* confirmar o perfil, será então, iniciado o processo de seleção das sugestões para o solicitante. O problema inicial, encontrar as sugestões dos serviços turísticos é dividido em módulos menores para facilitar a resolução do problema. Portanto, o problema geral (PG), é dividido nos módulos encontrar hospedagens (SPH), passeios (SPP), restaurantes (SPR) e lazer (SPL). Estes subproblemas individuais são resolvidos por Agentes que se comunicam e cooperam entre si quando necessário. Finalmente, os resultados dos subproblemas são integrados para montar a solução global (SG), como mostra a Figura 5.1.

Portanto, é apresentada uma solução para o problema real da prestação de serviços turísticos sugerido neste trabalho, utilizando-se Agentes, baseada na Resolução Distribuída de Problemas (RDP), apresentada no capítulo 3, onde o problema inicial é dividido em problemas menores e, distribuídos entre os agentes. A interação entre os agentes será realizada ou pela troca de mensagens, ou por meio do compartilhamento de dados comuns.

Para que exista cooperação entre os Agentes na busca da solução do problema, é essencial que exista um processo de comunicação entre eles. Este processo de comunicação é guiado por Protocolos visando identificar, simplificar e normalizar as tarefas dentro do



contexto, ou seja, o protocolo de interação entre os agentes vai determinar o comportamento da interação entre os mesmos.



**Figura 5.1** Resolução distribuída para o protótipo turismo

Onde

- PG – Problema geral
- SPH – Subproblema Encontrar Hospedagens
- SPP – Subproblema Encontrar Passeios
- SPR – Subproblema Encontrar Restaurantes
- SPL – Subproblema Encontrar Lazer

Como mencionado no capítulo 3, a troca de mensagens entre os Agentes pode ser apresentada de uma das três maneiras: *Unicast*, *Broadcast* e *Multicast*. Para a solução do protótipo proposto, utilizaremos a troca de mensagens *Unicast*, ou seja, a troca de mensagens baseadas em objetos. As mensagens são para um endereço específico, onde o transmissor conhece o receptor e vice-versa. Para especificação do protocolo de comunicação proposto, serão utilizados os diagramas de máquinas de estado da UML, onde as trocas de informações entre os Agentes vão promover mudanças de estados. Esta troca é efetuada através de mensagens que possuem o formato mostrado na tabela 5.1.

## 5.2 O Protocolo de Comunicação

O Protocolo de Comunicação especificado para o modelo do problema proposto (Portal de Turismo), que fará a comunicação entre os Agentes do sistema, é formado por cinco tipos de Agentes: O Agente Coordenador que gerência e monitora as transações do sistema e instância quando devido os demais agentes do sistema, o Agente Hospedagem responsável pela seleção das hospedagens que sejam mais ideais para o solicitante de acordo com o perfil informado, o Agente Passeios, responsáveis pela seleção dos passeios, o Agente Restaurante, responsável pela seleção dos restaurantes e, o Agente Lazer, que é o responsável pela seleção dos divertimentos noturnos que melhor se enquadrem no perfil informado.

**Tabela 5.1 – Formato da Mensagem do Protocolo de Comunicação**

Identificador	Descrição
Origem	Identifica o Agente que envia a mensagem
Estado	Identifica o estado do Agente.
Mensagem	Conteúdo que será trocado entre os Agentes
Destino	Identifica o Agente receptor da mensagem
Pré-condição	Fatos que devem ser verdade no ambiente do Agente para este envie ou receba uma mensagem.
Pós-condição	Fatos que passam a ser verdade no ambiente do Agente após este ter enviado ou recebido uma mensagem.

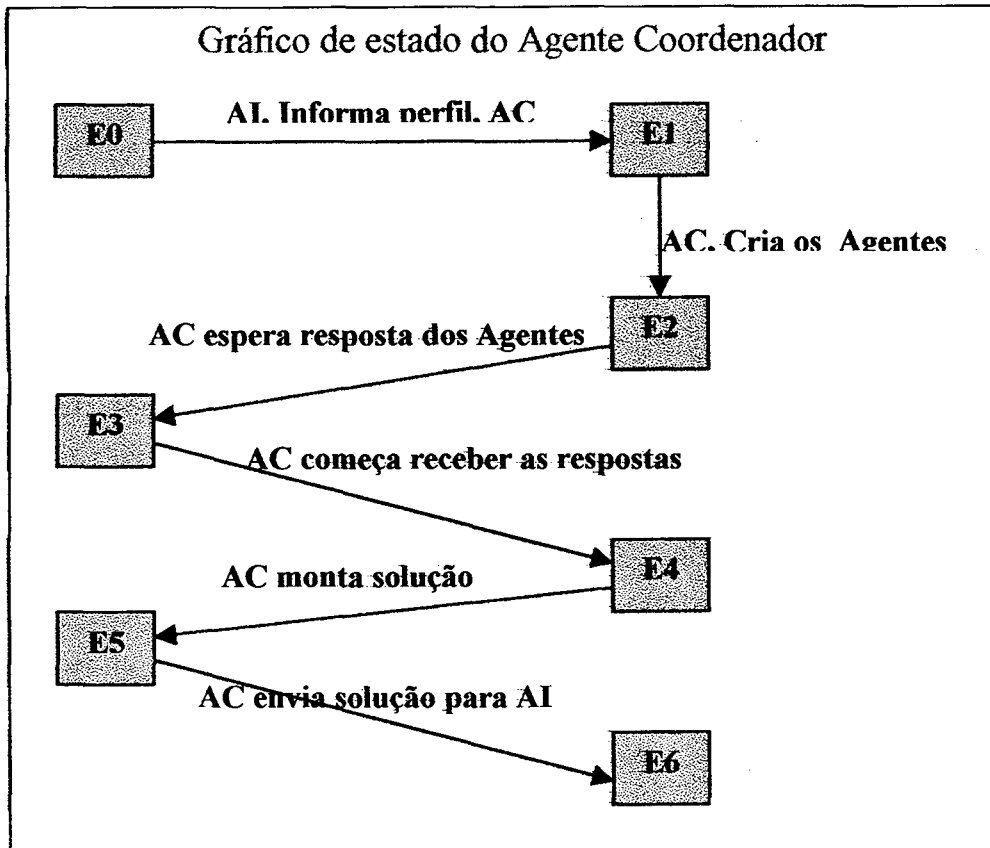
Para facilitar o entendimento do Protocolo, vamos utilizar uma notação para cada elemento do protocolo, como mostra a tabela 4.2 a seguir:

**Tabela 4.2 – Notação dos elementos do Protocolo (Agentes)**

Notação	Abreviatura
Agente Coordenador	AC
Agente Hospedagem	AH
Agente Passeio	AP
Agente Restaurante	AR
Agente Lazer	AL

O Protocolo é representado através de estados, identificando uma ação junto aos Agentes e a transição de um estado para outro ocorre com o recebimento ou envio de mensagem ou sinal. A Figura 5.2 mostra o protocolo ao nível do Agente Coordenador, através de um diagrama de gráfico de estado UML. O estado também identifica o papel do

Agente, ou seja, hora ele está informando, hora ele está criando, em outro momento ele está destruindo, e assim por diante.



**Figura 5.2 – Protocolo ao nível do Agente Coordenador (AC)**

**E0:** Estado inicial E0: representa o estado de espera onde o Agente Coordenador (AC), fica monitorando todas as requisições dos Agentes Interfaces (AI).

**E1:** O AC passa para este estado quando recebe uma mensagem do AI informando os dados do perfil do solicitante, o AC então devolve uma mensagem de confirmação de recebimento da mensagem e retorna ao estado de espera.

**E2:** O Agente AC cria os Agentes AH, AP, AR e AL, passa os dados do perfil para estes agentes e retorna para o estado de espera.

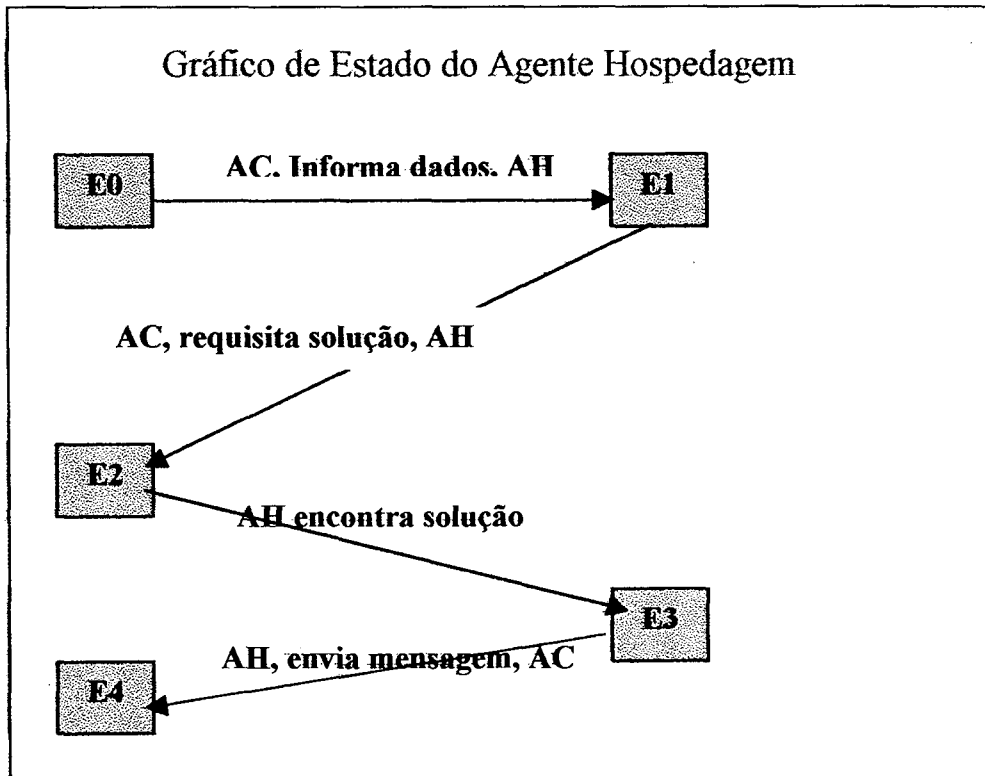
**E3:** Neste estado o agente AC espera pelas respostas dos agentes.

**E4:** O AC passa para este estado quando começa receber as mensagens dos agentes AH, AP, AR, AL com as soluções encontradas por eles, e volta ao estado de espera.

**E5:** Neste estado, após ter recebido o conjunto de soluções dos outros agentes, o AC monta a solução para o problema solicitado e, volta para o estado de espera.

**E6:** Neste estado o AC identifica o AI que fez a requisição das sugestões e devolve uma mensagem com a solução, para que o AI apresente ao cliente e, volta para o estado de espera.

A Figura 5.3 a seguir, mostra o Protocolo do Agente Hospedagem, sendo apresentado através de um diagrama de gráfico de estado da UML.



**Figura 5.3 – Protocolo de comunicação do Agente Hospedagem**

**Estado 0 (E0):** O Agente Hospedagem (AH), passa para este estado quando ele é instanciado pelo Agente Coordenador (AC), e começa monitorar as informações vindas do AC. Este é o estado de espera do AH.

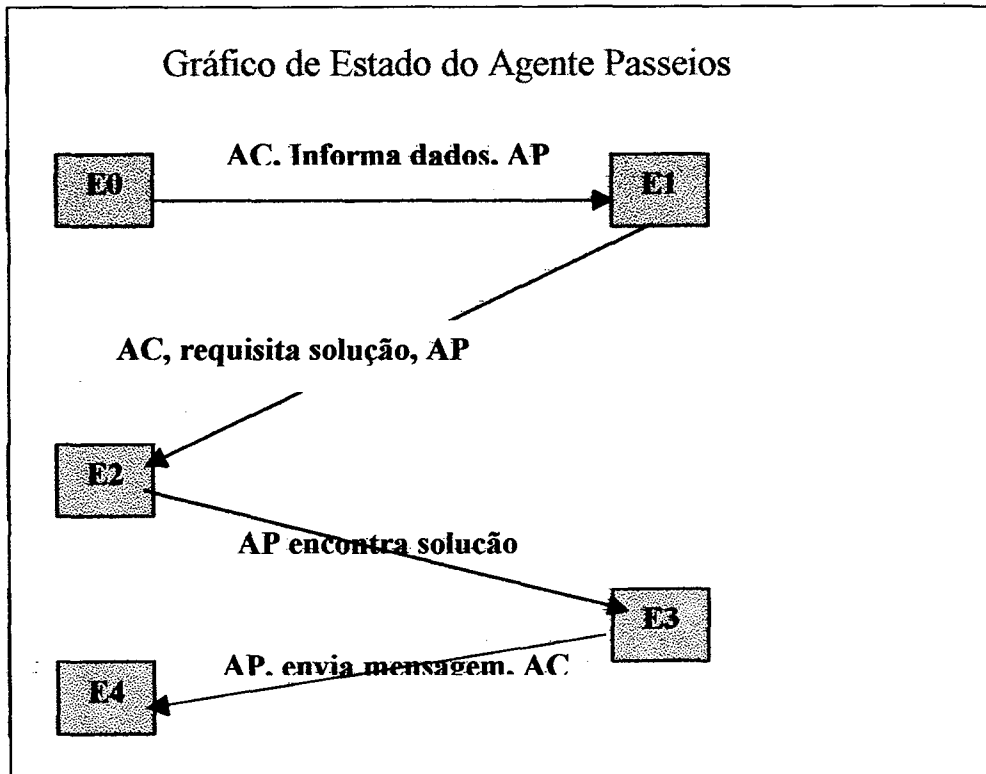
**E1:** O AH passa para este estado quando começa receber as informações do AC. Estes dados são oriundos do perfil do visitante do *site*.

**E2:** O AH procura solução que se enquadrem nos dados informados.

**E3:** Neste estado o AH compõe a solução para o problema solicitado.

**E4:** Após ter montado a solução para o problema, o AH envia uma mensagem para o AC, com a solução requisitada. Neste momento ele destruído pelo AC.

A Figura 5.4 a seguir, mostra o Protocolo do Agente Passeios, sendo apresentado através de um diagrama de gráfico de estado da UML.



**Figura 5.4 – Protocolo de comunicação do Agente Passeio**

**Estado 0 (E0):** O Agente Passeio (AP), passa para este estado quando ele é instanciado pelo Agente Coordenador (AC), e começa monitorar as informações vindas do AC. Este é o estado de espera do AP.

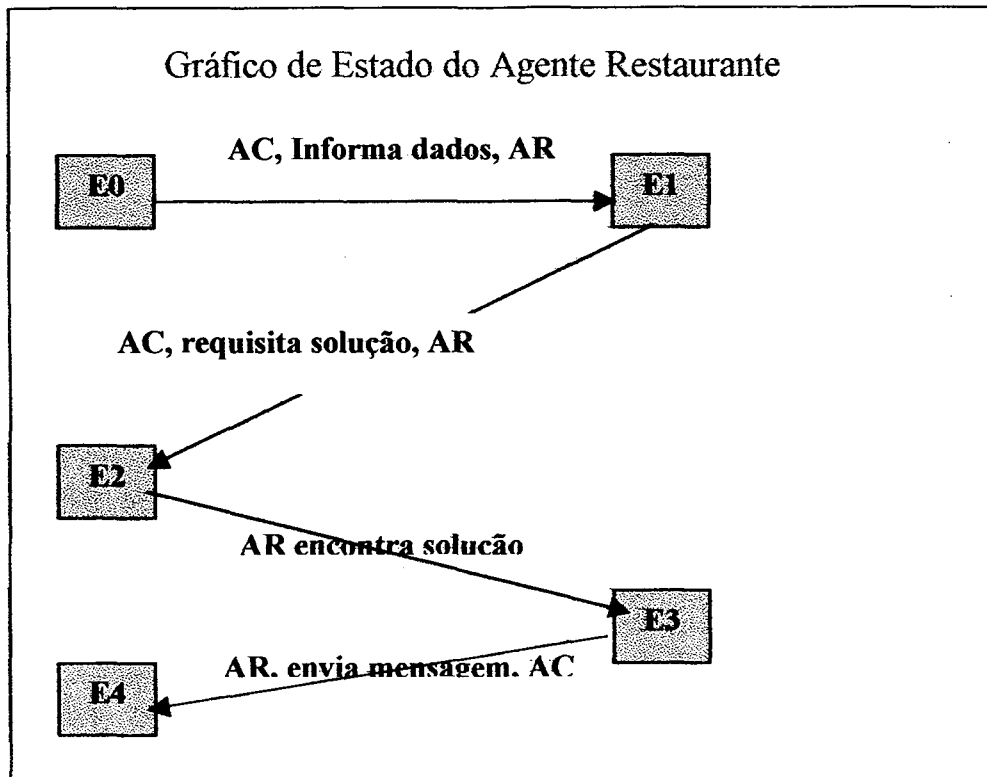
**E1:** O AP passa para este estado quando começa receber as informações do AC. Estes dados são oriundos do perfil do visitante do *site*.

**E2:** O AP procura solução que se enquadrem nos dados informados.

**E3:** Neste estado o AP compõe a solução para o problema solicitado.

**E4:** Após ter montado a solução para o problema, o AP envia uma mensagem para o AC, com a solução requisitada. Neste momento ele destruído pelo AC.

A Figura 5.5 a seguir, mostra o Protocolo do Agente Restaurante, sendo apresentado através de um diagrama de gráfico de estado da UML.



**Figura 5.5 – Protocolo de comunicação do Agente Restaurante**

**E0:** O Agente Restaurante (AR), passa para este estado quando ele é instanciado pelo Agente Coordenador (AC), e começa monitorar as informações vindas do AC. Este é o estado de espera do AR.

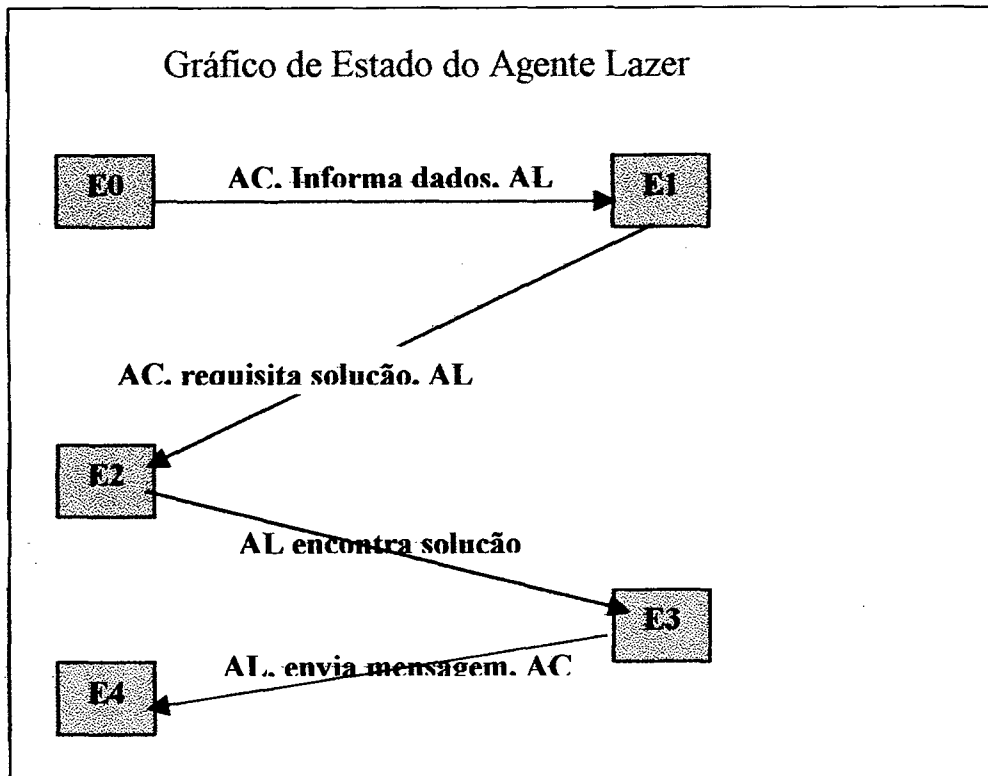
**E1:** O AR passa para este estado quando começa receber as informações do AC. Estes dados são oriundos do perfil do visitante do *site*.

**E2:** O AR procura solução que se enquadrem nos dados informados.

**E3:** Neste estado o AR compõe a solução para o problema solicitado.

**E4:** Após ter montado a solução para o problema, o AR envia uma mensagem para o AC, com a solução requisitada. Neste momento ele é destruído pelo AC.

A Figura 5.6 a seguir, mostra o Protocolo do Agente Lazer, sendo apresentado através de um diagrama de gráfico de estado da UML.



**Figura 5.6 – Protocolo de comunicação do Agente Lazer**

**E0:** Estado 0 (E0): O Agente Lazer (AL), passa para este estado quando ele é instanciado pelo Agente Coordenador (AC), e começa monitorar as informações vindas do AC. Este é o estado de espera do AL.

**E1:** O AL passa para este estado quando começa receber as informações do AC. Estes dados são oriundos do perfil do visitante do *site*.

**E2:** O AL procura solução que se enquadrem nos dados informados.

**E3:** Neste estado o AL compõe a solução para o problema solicitado.

**E4:** Após ter montado a solução para o problema, o AL envia uma mensagem para o AC, com a solução requisitada. Neste momento ele é destruído pelo AC.

### 5.3 Estrutura do Sistema

A estrutura do sistema de Agentes é apresentada na Figura 5.7 a seguir, onde podemos ver que ele é composto por Agentes Hospedagens, Passeios, Restaurantes e Lazer, sendo criados e coordenados por um Agente Coordenador.

Estes agentes vão trocar mensagens entre eles se necessário e entre o agente coordenador e acessar a base de dados para encontrar a solução para o problema requisitado, ou seja, encontrar os serviços turísticos que melhor se adequem ao perfil do visitante do *site*, que vez a solicitação.

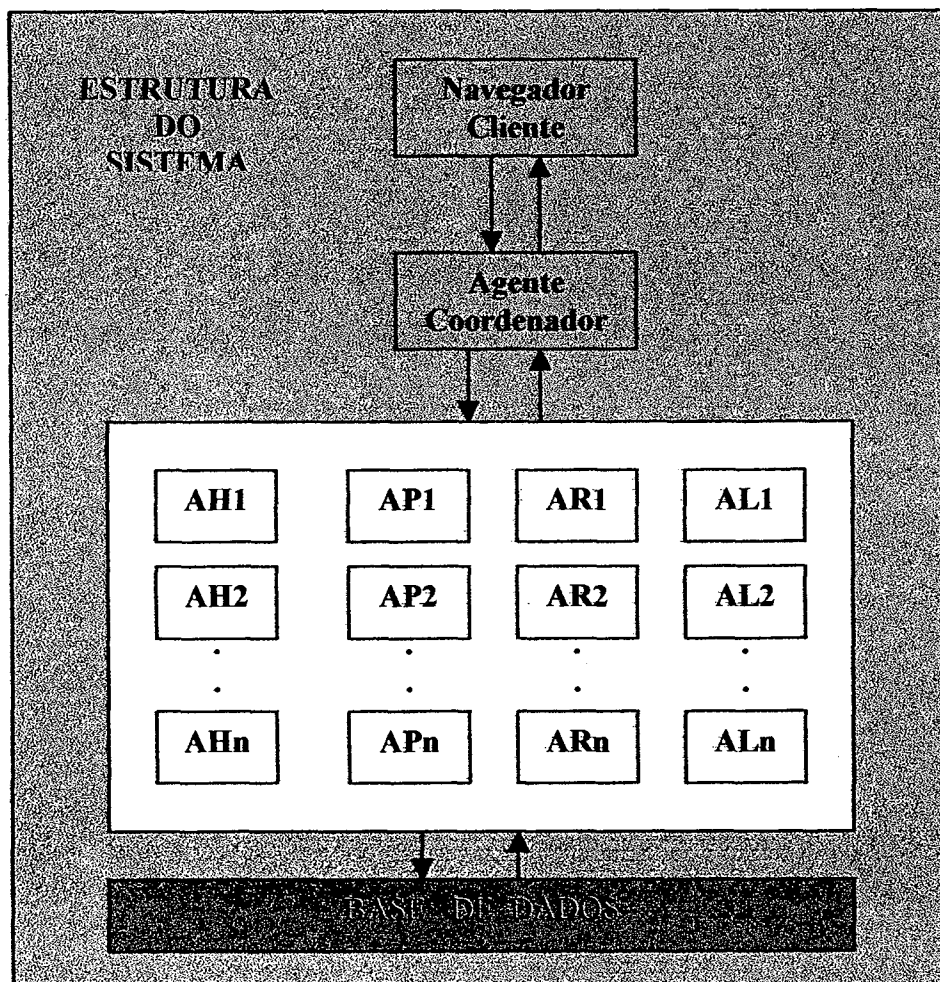


Figura 5.7 – Estrutura do Sistema de Agentes



## 5.4 Planejamento do Protótipo Portal de Turismo do Rio Grande do Norte

O planejamento do projeto é uma fase que antecede a implementação do projeto e tem como meta identificar os artefatos necessários para a implementação do projeto. Como exemplo de artefato pode-se citar: atores, caso de uso, lista de eventos, diagrama de classe, diagrama de seqüência, de colaboração, de gráfico de estado, de equipamentos e assim por diante. Nessa seção ir-se-á identificar alguns desses artefatos necessários para a implementação do modelo no capítulo 6.

### 5.4.1 Encontrando os Atores para o Protótipo

Os atores são fundamentais para obter vários artefatos no projeto. A identificação deles nos oferece uma melhor visão dos eventos que o sistema tem de suportar. A tabela 1 abaixo, lista todos os atores identificados para esse protótipo.

Os atores usualmente são considerados como sendo seres humanos, mas também podem ser outros sistemas, temporizadores e relógios, ou dispositivos de *hardware*. Os atores estimulam o sistema e são os iniciadores de ventos; eles também podem receber estímulos do sistema e nesse caso são considerados passivos.

**Tabela 1** Atores Propostos para o Portal de Turismo RN

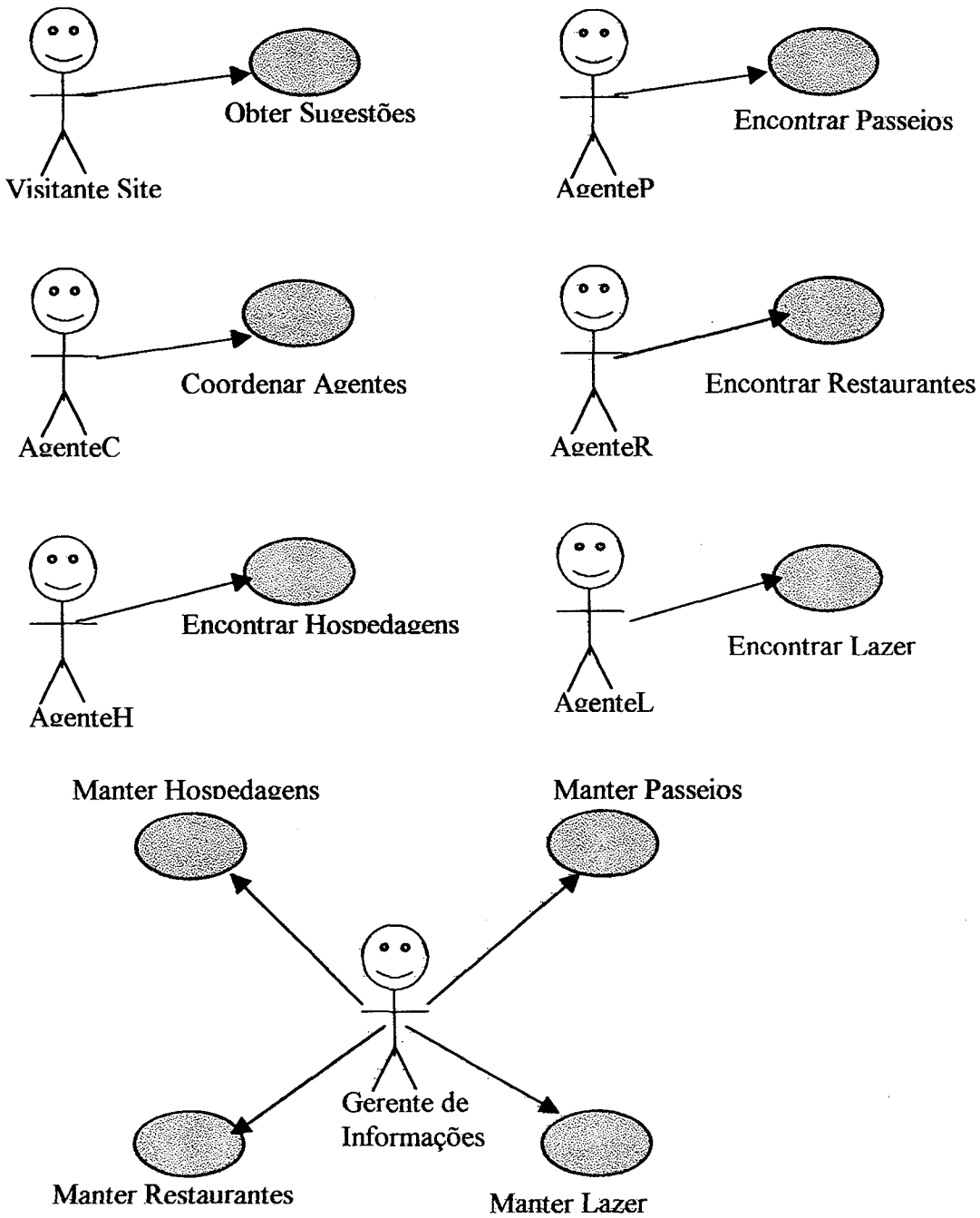
Ator	Descrição
Visitante do <i>site</i>	Uma pessoa que acessa o <i>site</i>
Agente Coordenador	Coordena os outros agentes
Agente Hospedagem	Responsável pela tarefa encontrar hospedagens
Agente Passeio	Responsável pela tarefa encontrar passeios
Agente Restaurante	responsável pela tarefa encontrar restaurantes
Agente Lazer	responsável pela tarefa encontrar lazer
Gerente de informações	responsável em manter atualizada a base de dados do sistema.

### 5.4.2 Identificando os casos de Uso para o Protótipo

Tendo-se definido os atores e os eventos de interesse no sistema, passa-se em seguida a atribuir os eventos a casos de uso. O caso de uso é a ferramenta UML que focaliza o quê o sistema deverá fazer, não como ele o fará. Os casos de uso são o centro a partir do qual são derivados todos os requisitos.

Na página a seguir será mostrado o diagrama de casos de uso identificados para o protótipo proposto em relação ao módulo dos agentes. Além disso, faz-se um relacionamento de cada ator com o respectivo caso de uso.

**Diagrama de Casos de Uso do Protótipo Portal de Turismo do Rio Grande do Norte**



**Figura 5.8 Diagrama de Casos de Uso do protótipo Proposto.**

Tendo-se definido os casos de uso, o passo seguinte é fazer um detalhamento de cada caso de uso.

---

Nome do Caso de Uso: Obter Sugestões

Descrição do caso de Uso: Este caso de uso começa quando o visitante do *site* tecla na opção “Sugestões”. Então, é exibido um formulário em branco para o mesmo informar o seu perfil. O caso de uso termina quando o visitante recebe as informações sugeridas.

Atore(s): Visitante do *site*.

Localização: em qualquer lugar, onde houver *Internet*.

Pré-condições: o internauta entrou com o URL correto do *site*.

Pós-condições: - as sugestões requisitadas foram retornadas.

Caminho Primário: o internauta solicita sugestões dos serviços turísticos.

Caminho(s) Alternativo(s): o internauta cancela solicitação.

Caminhos de Exceção: - o internauta informa CPF incorreto;  
- o internauta não informa nome;  
- o internauta não informa a data de nascimento.

Detalhamento do caminho primário – o internauta solicita sugestões dos serviços turísticos.

1. o visitante do *site* fornece seu CPF.
2. o CPF é conhecido como já cadastrado.
3. CPF não cadastrado.
4. são solicitados os demais dados do visitante do *site*.
5. as informações são confirmadas.
6. neste momento é então ativado o agente coordenador que recebe as informações do perfil digitado.

7. o agente coordenador, ativa os agentes: Hospedagens, Passeios, Restaurantes e Lazer e atribui aos mesmo os dados do perfil informado, referente as suas responsabilidades respectivamente.
8. os agentes: Hospedagens, Passeios, Restaurantes e Lazer, vão atrás das informações solicitadas.
9. os agentes: Hospedagens, Passeios, Restaurantes e Lazer retornam as soluções encontradas para o agente coordenador.
10. o agente coordenador retorna em forma de HTML a solução encontrada para o solicitante das sugestões.
11. a solicitação pode ser cancelada a qualquer momento.

Nome do Caso de Uso: Encontrar Hospedagens

Descrição do caso de Uso: Este caso de uso começa quando o visitante do *site* confirma Dados do seu perfil. O caso de uso termina quando o visitante recebe as informações sugeridas.

Atore(s): Agente Hospedagem.

Localização: em qualquer lugar, onde houver *Internet*.

Pré-condições: o internauta informou perfil.

Pós-condições: - as sugestões requisitadas foram retornadas.

Caminho Primário: o Agente Coordenador envia mensagem para o agente hospedagem.

Caminho(s) Alternativo(s): Nenhum

Caminhos de Exceção: Nenhum

Detalhamento do caminho primário – o Agente Coordenador envia mensagem para o agente hospedagem.

1. o agente coordenador cria o agente hospedagem.
2. o agente coordenador envia uma mensagem com os dados do perfil.
3. o agente hospedagem procura pelas informações solicitadas.

4. o agente hospedagem envia para o agente coordenador as informações encontradas.

Nome do Caso de Uso: Encontrar Passeios

Descrição do caso de Uso: Este caso de uso começa quando o visitante do *site* confirma Dados do seu perfil. O caso de uso termina quando o visitante recebe as informações sugeridas.

Atore(s): Agente Passeio.

Localização: em qualquer lugar, onde houver *Internet*.

Pré-condições: o internauta informou perfil.

Pós-condições: - as sugestões requisitadas foram retornadas.

Caminho Primário: o Agente Coordenador envia mensagem para o agente passeio.

Caminho(s) Alternativo(s): Nenhum

Caminhos de Exceção: Nenhum

Detalhamento do caminho primário – o Agente Coordenador envia mensagem para o agente passeio.

1. o agente coordenador cria o agente passeio.
2. o agente coordenador envia uma mensagem com os dados do perfil.
3. o agente passeio procura pelas informações solicitadas.
4. o agente passeio envia para o agente coordenador as informações encontradas.

Nome do Caso de Uso: Encontrar Restaurantes

Descrição do caso de Uso: Este caso de uso começa quando o visitante do *site* confirma Dados do seu perfil. O caso de uso termina quando o visitante recebe as informações sugeridas.

Atore(s): Agente Restaurante.

Localização: em qualquer lugar, onde houver *Internet*.

Pré-condições: o internauta informou perfil.

Pós-condições: - as sugestões requisitadas foram retornadas.

Caminho Primário: o Agente Coordenador envia mensagem para o agente restaurante.

Caminho(s) Alternativo(s): Nenhum

Caminhos de Exceção: Nenhum

Detalhamento do caminho primário – o Agente Coordenador envia mensagem para o agente restaurante.

1. o agente coordenador cria o agente restaurante.
2. o agente coordenador envia uma mensagem com os dados do perfil.
3. o agente restaurante procura pelas informações solicitadas.
4. o agente restaurante envia para o agente coordenador as informações encontradas.

Nome do Caso de Uso: Encontrar Lazer

Descrição do caso de Uso: Este caso de uso começa quando o visitante do *site* confirma Dados do seu perfil. O caso de uso termina quando o visitante recebe as informações sugeridas.

Atore(s): Agente Lazer.

Localização: em qualquer lugar, onde houver *Internet*.

Pré-condições: o internauta informou perfil.

Pós-condições: - as sugestões requisitadas foram retornadas.

Caminho Primário: o Agente Coordenador envia mensagem para o agente lazer.

Caminho(s) Alternativo(s): Nenhum

Caminhos de Exceção: Nenhum

Detalhamento do caminho primário – o Agente Coordenador envia mensagem para o agente lazer.

1. o agente coordenador cria o agente lazer.

2. o agente coordenador envia uma mensagem com os dados do perfil.
  3. o agente lazer procura pelas informações solicitadas.
  4. o agente lazer envia para o agente coordenador as informações encontradas.
- 

#### **5.4.3 Identificando as Classes para o Caso de Uso “Obter Sugestões”.**

Durante a fase inicial de modelagem do protótipo foi feito um passeio pelo aplicativo e isso produziu uma lista de casos de uso em escopo e todos os seus caminhos individuais (primário, alternativo e de exceção), bem como o detalhamento para seus *happy pathways* (caminhos primários). O passo a seguir é identificar as classes que farão parte do protótipo. Estas classes serão usadas para construir o esqueleto do desenho do protótipo e o diagrama de seqüência como artéria do protótipo representando o fluxo de mensagens entre os componentes do esqueleto ou classe. Na página seguinte é mostrado um diagrama de classe para o protótipo proposto.

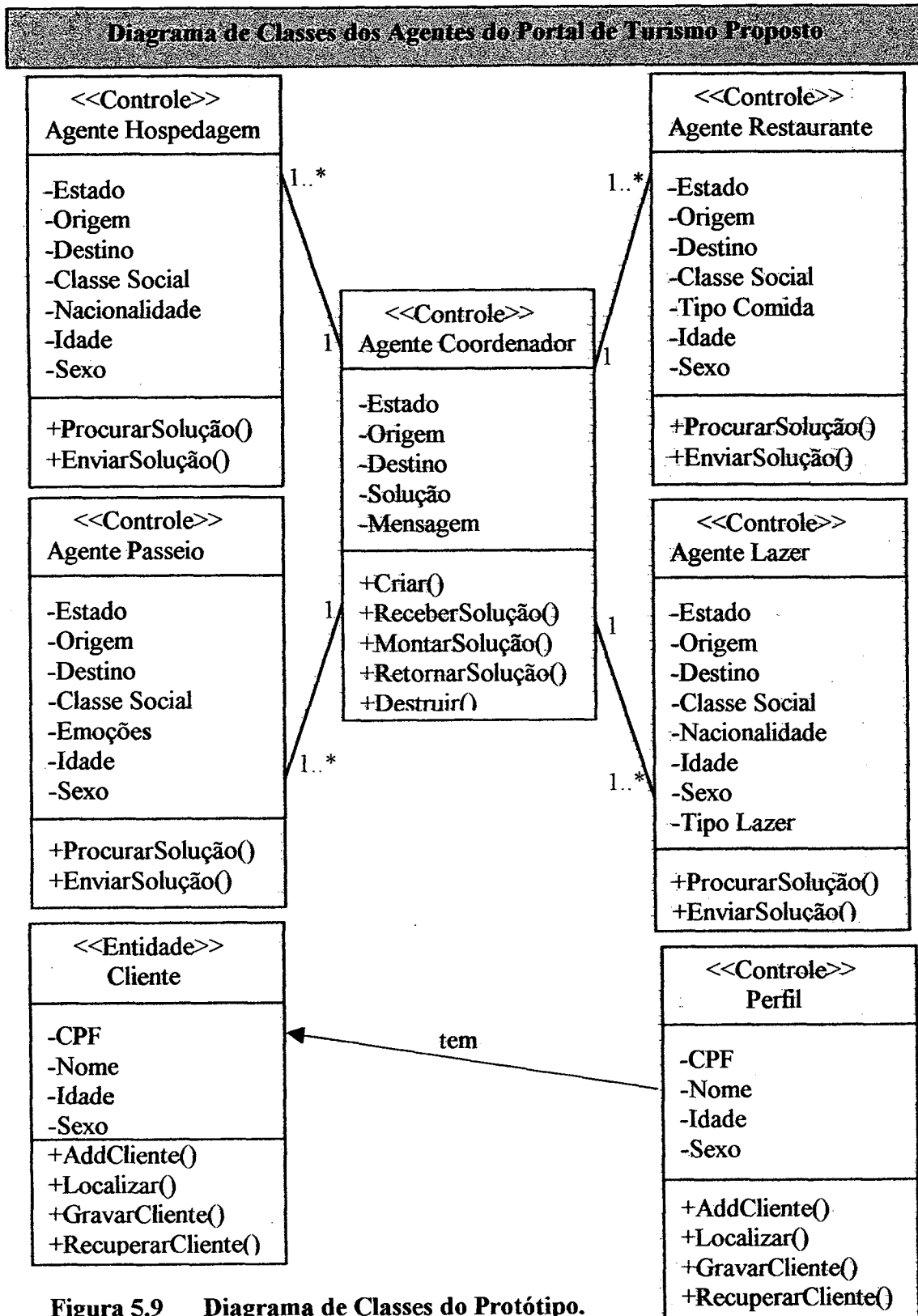
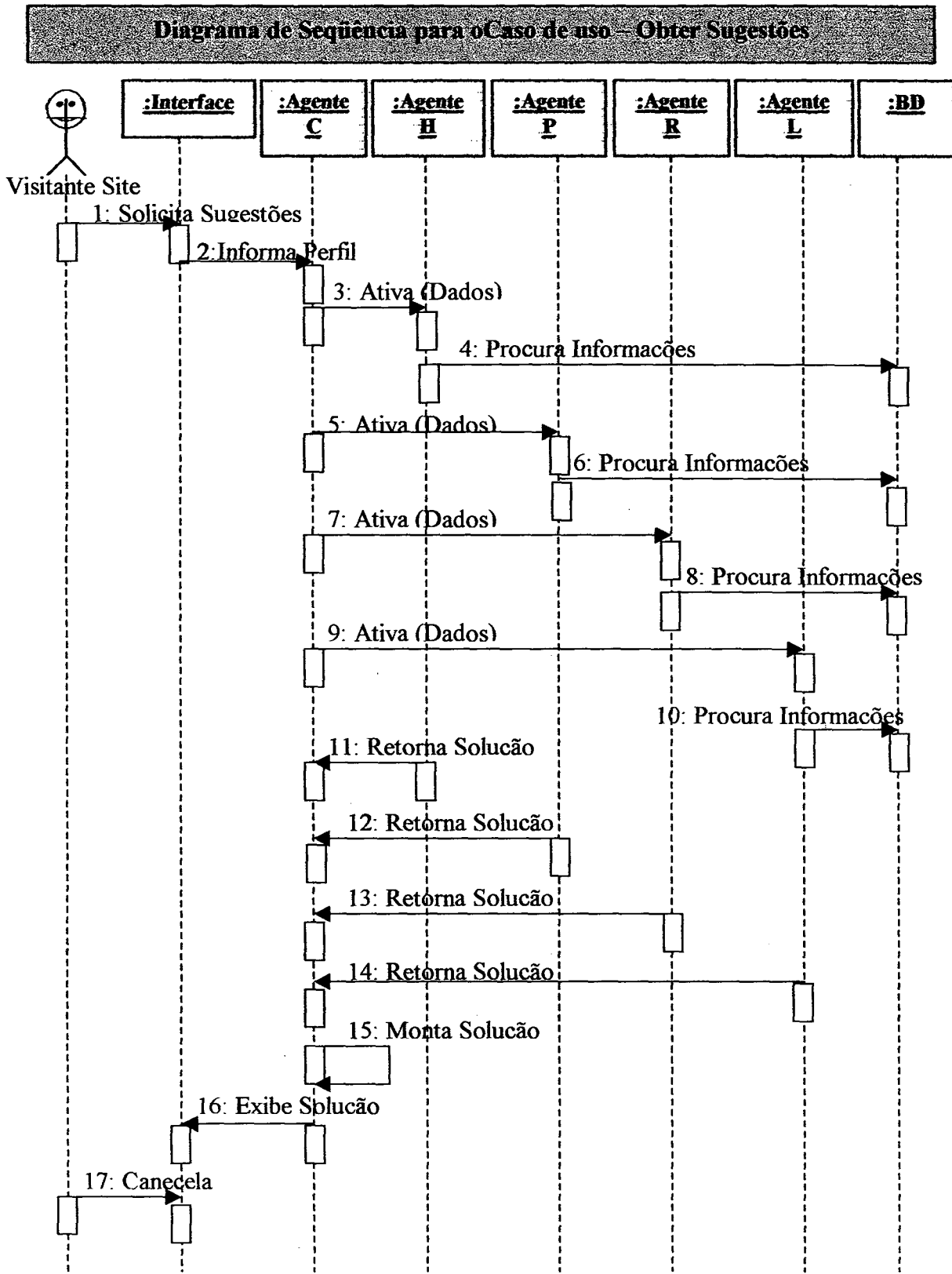


Figura 5.9 Diagrama de Classes do Protótipo.



#### **5.4.4 Diagrama de Seqüência para o Caminho Primário do Caso de Uso “Obter Sugestões”.**

Os diagramas de seqüência fazem parte dos modelos dinâmicos dos artefatos UML. Esses modelos focalizam as interações entre objetos – instâncias de classes identificadas no modelo de caso de uso que funcionam em conjunto para implementar os caminhos detalhados ao longo desses casos de uso. Portanto, na página seguinte é mostrado o diagrama de seqüência que representa o caminho primário do caso de uso obter sugestões.



**Figura 5.10 Diagrama de Seqüência do protótipo Pronosto.**

### 5.4.5 Infra-estrutura em nível de Componente para o Protótipo

A infra-estrutura para o protótipo agora pode ser reproduzida como um diagrama de componente UML. Nesse diagrama mostra-se o mapeamento de todos os componentes para o portal de turismo. Como pode-se ver nesse diagrama o componente Agente Coordenador (Acoordenador) é um EXE que roda fora de processo e os demais são DLL que rodam no mesmo processo do componente chamador (O EXE) sob o MTS – Microsoft Transaction Server. Não pode-se esquecer de que essa é uma aplicação *Internet* e que precisamos também dos componentes *Web*. No protótipo proposto o componente de entrada é uma página HTML que pode tomar como entrada um número de CPF e um *Script ASP* - Active Server Page que recebe esse CPF e processa as informações solicitadas no servidor. A Figura 5.11, a seguinte é mostra um diagrama de componentes para o protótipo portal de turismo.

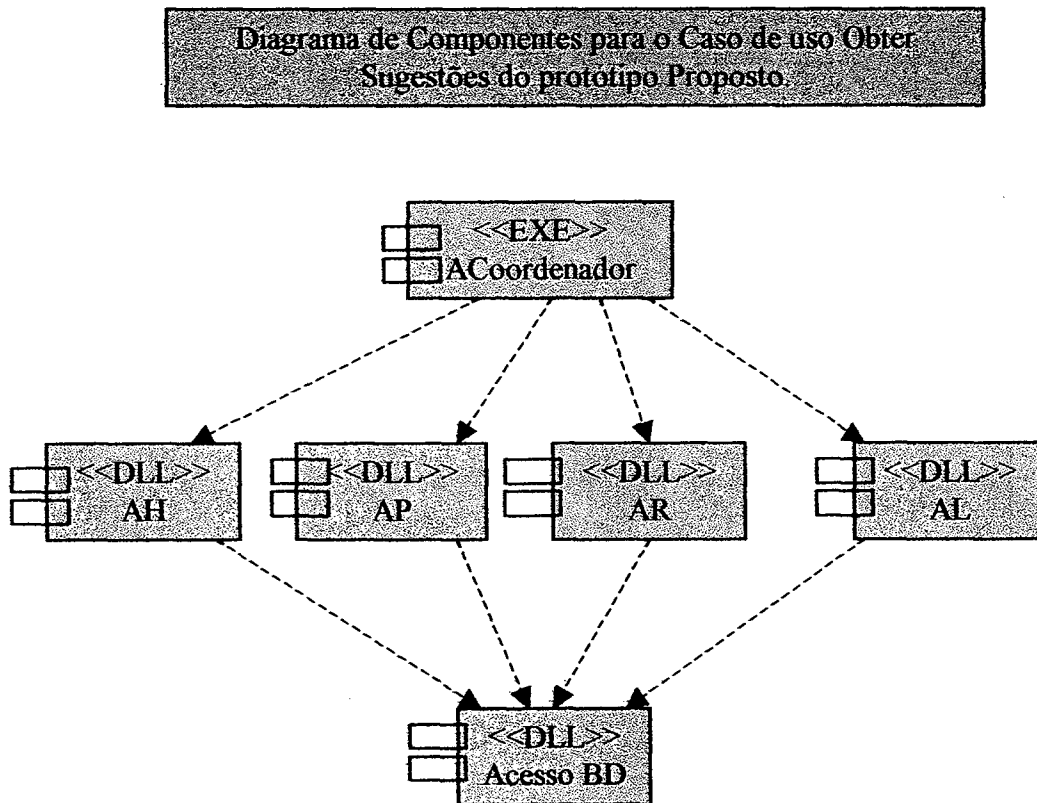


Figura 5.11 Diagrama de Componentes para o Protótipo Proposto.

## 5.5 – Considerações Finais

Neste capítulo procuro-se mostrar um modelo para implementação do Protótipo do Portal de turismo, apresentando o protocolo de comunicação a ser adotada pelos Agentes, utilizando os diagramas de gráficos de estado da UML, onde a comunicação entre os agentes é realizada pela troca de mensagens sendo que, os Agentes mudam de estado sob determinadas condições. A mudança de estado também identifica um novo papel a ser assumido pelo Agente.

O estudo de caso apresentado constitui-se em um modelo que consiste na solicitação de sugestões de serviços turísticos por parte do visitante do *site*, onde o mesmo informa o seu perfil e aguarda o retorno das sugestões oferecidas como a melhor solução de acordo com seu perfil. Diante dos dados do perfil informado, o Agente Coordenador cria os agentes Hospedagens, Passeios, Restaurantes e lazer e fornece os dados do perfil para os mesmo e, fica aguardando a solução. Contudo, para encontrar a solução para o problema requisitado, os Agentes Hospedagens, Passeios, Restaurantes e lazer, efetuam apenas procedimentos simples e não negociação.

## Capítulo 6

### IMPLEMENTANDO O MÓDELO

#### 6.1 - Introdução

Este capítulo procura mostrar aspectos referentes à implementação do sistema, levando-se em conta aspectos tais como: Ambiente de Desenvolvimento, Tecnologias e Ferramentas utilizadas. Observando ainda aspectos como Escalabilidade, Multiencadeamento, empacotamento, sincronização entre as comunicações, simplicidade e acesso aos dados.

#### 6.2 – Ambiente de Desenvolvimento

Levando-se em conta de que o protótipo proposto é um *Site*, ele tem uma parte que roda no Cliente e uma outra que roda no Servidor. No Cliente vão rodar páginas HTML com Scripts ASP (*ActiveX Server Page*). Nos Scripts ASP estão sendo usadas as linguagens de Script: *VBScript* e *JavaScript*. No Servidor está o sistema de Agentes que são Servidores *ActiveX EXE*. As principais razões para a escolha de Servidores EXE, foram:

- O Servidor EXE roda em um encadeamento separado, de modo que o aplicativo que fez a solicitação não fica bloqueado durante o processo de obtenção da solução;
- Um Servidor EXE permite que múltiplos Clientes compartilhem o mesmo enlace de comunicação;
- As Interfaces (objetos) podem ser criados e usados por aplicativos clientes e rodam no servidor como um aplicativo independente;
- Como o Servidor EXE roda em seu próprio encadeamento (*Thread*), ele pode receber solicitações em segundo plano sem interferir com o desempenho do aplicativo que está usando os objetos (Interfaces).
- Simplicidade de desenvolvimento e depuração;
- Facilidade de empacotamento, utilizando a tecnologia MTS (*Microsoft Transaction Server*), que permite a integração do Servidor Web e páginas ASP;

### 6.2.1 – Banco de Dados

Para o gerenciamento de sistema de banco de dados relacionais, está sendo utilizada a tecnologia ADO (*ActiveX Data Object*). Com o ADO pode-se recuperar e mudar dados de várias origens. Estas não estão limitadas aos bancos de dados tradicionais, podem incluir arquivos de sistemas, armazenar e-mails, e mesmo dados na Internet. Para se fazer à conexão a fonte de dados pode ser usada tanto OLE DB – é um conjunto de interfaces de COM (*Component Object Model*), para construir componentes de banco de dados, quanto ODBC – *Open DataBase Connectivity*. Com o ODBC pode-se acessar diversas fontes de dados, tais como *SQL Server*, *Oracle*, *Microsoft Access*, arquivos DBFs, arquivos TXT, dentre muitos outros.

O ADO faz a conexão e criam-se os objetos de dados como *Recordsets*, cursores e etc, mas para se fazer à manipulação dos dados está sendo utilizado SQL – *Structured Query Language*. Neste protótipo estão sendo utilizados ADO para se fazer à conexão ao banco e criar os objetos de dados, ODBC para fazer a ligação com a fonte de dados e SQL para fazer a manipulação dos dados.

### 6.3 – Estrutura de Implementação dos Agentes

A princípio, queremos que qualquer cliente possa solicitar sugestões e receber uma notificação com a solução proposta. Portanto, devem ser feitas múltiplas solicitações por vários clientes, por isso, o Servidor EXE deverá suportar múltiplos clientes

Queremos que o Servidor EXE possa enfileirar as solicitações dos diferentes clientes e, em seguida, tratá-las uma de cada vez.

Os objetos devem se comunicar um com o outro, através de suas propriedades, métodos e estrutura de dados no módulo global. Portanto, temos um objeto que pode ser criado ou ativado por qualquer aplicativo.

Como já mencionado em capítulos anteriores, o sistema é composto de cinco Agentes. Um Agente Coordenador (AC), um Agente Hospedagem (AH), um Agente Passeio (AP), um Agente Restaurante (AR) e um outro Agente Lazer (AL).

#### 6.3.1 – O Agente Coordenador

O Agente Coordenador é ativado quando o visitante do *site* clica na opção “Sugestões” e informa o seu perfil. Ele é formado por uma estrutura de métodos e propriedades. Os métodos identificam os diferentes papéis que o agente exerce durante o seu ciclo de vida. O agente muda de papel, quando ocorre a mudança de seu estado pelo

envio de mensagem ou de um sinal. O estado do Agente é configurado como uma propriedade do mesmo.

O Agente Coordenado possui os seguintes métodos e propriedades:

#### **Métodos:**

- **ReceberPeril** – recebe uma solicitação de sugestões e os dados do perfil do visitante;
- **CriarAgentes** – cria os Agentes AH, AP, AR e AL e passa os dados do perfil para eles;
- **ReceberRespostas** – receber as informações da solução encontrada pelos agentes;
- **MontarSolução** – montar a solução encontrada no formato HTML para ser enviada para o visitante;
- **EnviaSolução** – envia a solução para o visitante;
- **DestruirAgentes** – destruir os agentes AH, AP, AR, e AL, com base nos seus estados.

#### **Propriedades:**

- **Estado** – identifica o estado do agente;
- **Matriz \_Solução** – uma matriz parametrizada para receber a solução enviada pelos outros agentes;
- **Origem** – identifica a origem da mensagem;
- **Destino** – identifica para quem vai a mensagem;
- **Mensagem** – o corpo da informação.

### **6.3.2 – Agentes Hospedagem, Passeio, Restaurante e Lazer**

Todos estes Agentes são criados pelo Agente Coordenador, eles têm a função de localizar no banco de dados todas as informações referentes a hospedagens, passeios, restaurantes e lazer respectivamente, que estiverem de acordo com o perfil informado e retornar estas informações para o Agente Coordenador. Estes Agentes também possuem

métodos e propriedades. Os métodos são `EncontrarSolução` e `EnviarSolução`. Uma propriedade comum a cada a deles é o seu `Estado`, ou seja, cada `Agente` tem uma propriedade, que representa o seu `Estado`.

#### 6.4 – A Arquitetura do Sistema

A Figura 6.1 a seguir ilustra a arquitetura geral do sistema de agentes. Quando o cliente ou visitante do *site* solicita sugestões através de um link em uma das páginas do *site*, é automaticamente ativado o `Agente Coordenador`, que instantaneamente cria os outros agentes e passa os dados do perfil para os mesmos. As sugestões são retornadas ao visitante no formato HTML. Para auxiliar os agente na localização das informações no banco de dados, foi desenvolvido um componente de código, ou seja, uma DLL, que contém todos os métodos necessários para manipular os dados no banco de dados.

Toda as páginas do *site* são montadas dinamicamente, ou seja, todas as informações para montagem das páginas estão armazenadas em tabelas no Banco de dados e, são carregadas para as páginas quando estas são solicitadas. Para que isto seja possível, está sendo utilizado ASP (*Active Server Page*). O ASP atuará na realidade como interface e controlador do processo. Ele irá intermediar a solicitação do *browser* e então instanciar e enviar mensagens para o objeto apropriado e assim retornar informações adequadas para criar uma resposta HTML de retorno.



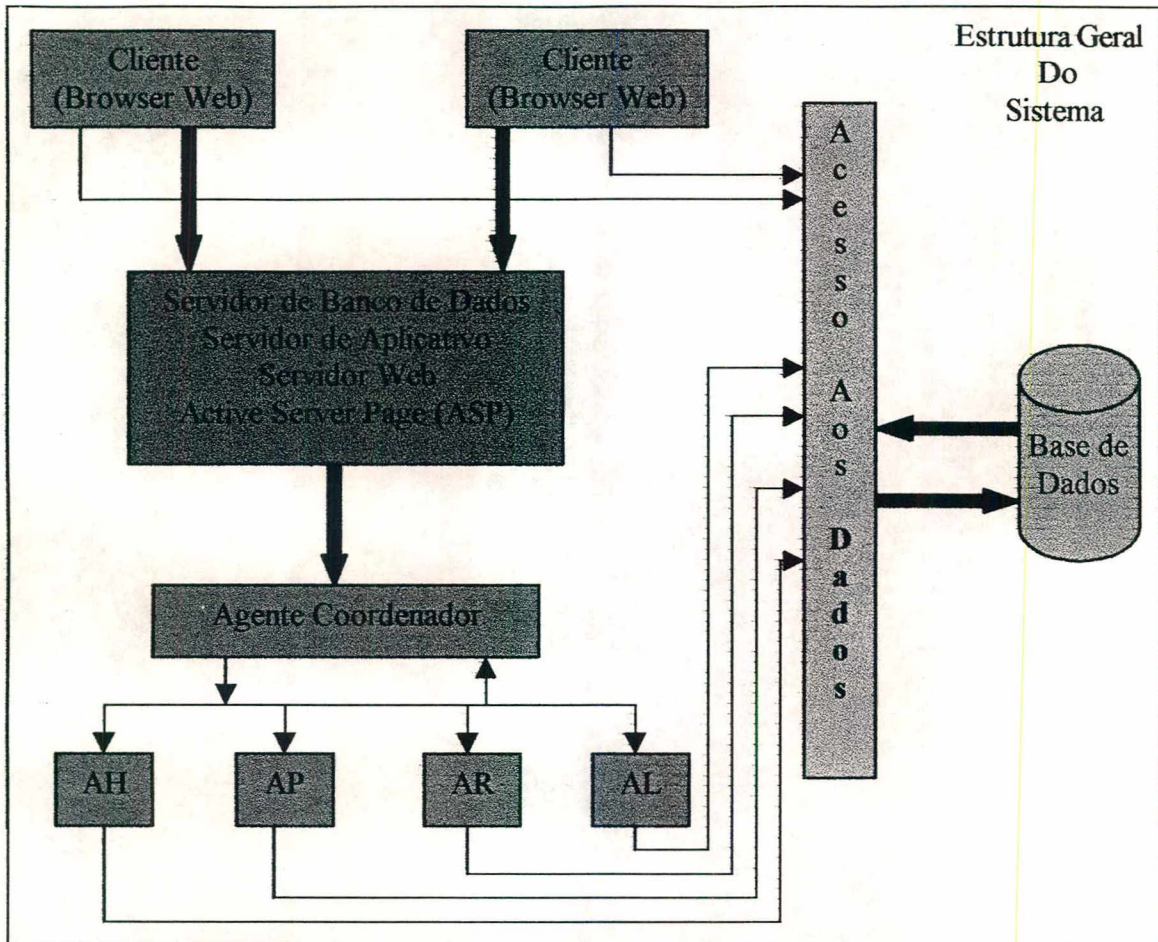


Figura 6.1 – Arquitetura Geral do Sistema para o protótipo

## 6.5 – Implementação

Para implementação do protótipo, o sistema foi dividido em dois módulos, sendo o primeiro o módulo das páginas ASP, por onde serão feitas as solicitações das sugestões e exibição das soluções encontradas respectivamente. O segundo é do Agente Coordenador, onde o mesmo é uma aplicação de execução automática e, multiencadeado, usando comunicação assíncrona. Este é o módulo principal do sistema uma vez que é o responsável pelo recebimento do perfil informado, criação e distribuição dos outros agentes e pelo retorno da solução encontrada.

### 6.5.1 Interface do Sistema

São duas páginas ASP que rodam do lado do servidor da aplicação, onde uma é a responsável pela informação do perfil e a outra é a responsável pela exibição da solução encontrada ao visitante do *site* solicitante.

A tela do formulário que representa a entrada do perfil é mostrada na Figura 6.2 a seguir.

The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads 'SUGESTÕES - MICROSOFT INTERNET EXPLORER'. The address bar shows 'HTTP://LOCALHOST/SUGESTOES.ASP'. The page content includes a header with 'Bem vindo! ao Portal de Turismo de... Natal Rio Grande do Norte'. Below the header is a navigation menu with buttons for 'PAGES', 'Comissão', 'Download', 'EMAIL', and 'Sugestões'. The main heading is 'Sugestões'. The form is titled 'Por favor, informe o seu perfil' and contains the following fields and controls:

- CPF... [input field] [Confirmar]
- Nome... [input field with value 'Maria']
- Classe Social... [dropdown menu with value 'Alta']
- Idade... [input field with value '43']
- Sexo...  Masculino  Feminino
- Nacionalidade... [dropdown menu with value 'Brasileira']
- Gosta de emoções...  Sim  Não
- Comidas de sua preferência... [dropdown menu with value 'Coisas do mar']
- Tipo de Lazer... [dropdown menu with value 'Passeios']
- Badalação... [dropdown menu with value 'Boates']

At the bottom of the form are two buttons: 'Confirmar' and 'Limpar'.

Figura 6.2 Formulário de entrada do perfil.

A solução retornada ao solicitante das informações turísticas é apresentada no formato apresentado pela Figura 6.3 abaixo.

<b>Hospedagens</b>						
Hospedagem	Preço Diária	Categoria	Localização	Telefone	Site	E-Mail
Ocean Palace	Não disponível	*****	Via costeira	848004144	<a href="http://www.oceanpalace.com.br">www.oceanpalace.com.br</a>	Ocean Palace

<b>Passeios</b>					
Passeio	Descrição	Frequência	Ideal Para	Emoção (S/N)?	Telefone

<b>Restaurantes</b>				
Restuarante	Especialidade	Localização	Telefone	Tipo
Guinza	Sushi, Sashimi	Via costeira	84	Alacarte
Tábua de Carne	Carnes	Av. Roberto Freire	842194469	Alacarte
Raro Sabor	Culinária francesa	Tirol	842021857	Alacarte
Mama Itália	Típica Italiana	Tirol	842021622	Alacarte
Xique-Xique	Típica regional	Tirol	842115545	Alacarte

<b>Badalacão Noturna</b>				
Estabelecimento	Descrição	Localização	Telefone	Estilo
Taverna Pub	bar em estilo inglês numa construção que reproduz um castelo Medieval.	Ponta Negra	842363696	Bar

Figura 6.3 – Solução retornada ao visitante do site.

Na implementação do protótipo foram utilizadas diversas ferramentas. Por exemplo, para construção das páginas de menu foi utilizada a ferramenta da Macromedia Flash 4.0. Para o desenvolvimento dos botões, dos títulos, e de algumas faixas animadas foi utilizada a ferramenta da Macromedia Fireworks e, para a estruturação e desenvolvimento do site

foi utilizada a ferramenta da Macromedia Dreamweaver. Para desenvolvimento e depuração das páginas ASP, foi utilizado o Visual InterDev da Microsoft. O componente de acesso aos dados, é um ActiveX DLL e foi desenvolvido em Visual Basic da Microsoft. Os agentes são componentes de código ActiveX EXE, configurados com multiencadeamento e de segmentação segura, também desenvolvido em Visual basic da Microsoft.

## 6.6 – Simulação da Utilização do Protótipo

A simulação foi feita no laboratório de informática do CEFET-RN. Este laboratório está ligado na Intranet do CEFET-RN, que está ligado há um provedor Internet. Através do *browser Internet Explorer*, foi feito acesso ao endereço WWW do *site* proposto. Assim que a página de abertura do *site* é carregada, o Agente Coordenador é ativado sem intervenção do usuário, e fica em estado de espera. Se o visitante clicar na opção “Sugestões”, então será passada uma mensagem para o Agente Coordenador, e este automaticamente criar os outros agentes e delega aos mesmos a tarefa de encontrar a solução para o problema requisitado.

## 6.7 – Considerações Finais

A partir de testes realizados verificou-se que, para o problema apresentado, um Sistema Multiagente gera uma boa solução em um tempo de resposta excelente, levando-se em conta que, o problema maior é subdividido em problemas menores e, estes problemas menores, são distribuídos entre quatro componentes do sistema, ou seja, ao invés do problema ser resolvido por apenas um componente, ele é resolvido por quatro componentes simultaneamente. Além do mais, deve-se observar que, os subproblemas têm uma complexidade inferior ao problema geral. Além disso, a principal vantagem desta abordagem de especificação é a semelhança do funcionamento do modelo implementado com a realidade.

## Capítulo 7

### CONCLUSÕES E RECOMENDAÇÕES

Ao longo deste trabalho falou-se muito sobre termos como Internet, comércio eletrônico, ferramentas de implantação de sistemas e metodologias de modelagens de dados. No capítulo 1 e 2, foi abordado, o crescimento da Internet no mundo e, principalmente a sua evolução no Brasil e, dentro desta perspectiva o comércio eletrônico surge com a maior perspectiva de crescimento na rede mundial. Ainda, neste contexto, aparece o Turismo virtual como uma nova força de crescimento da área. Não poderia deixar de ser, uma vez que a área de prestação de serviço é uma das mais promissoras em nosso país, devido o seu grande potencial turístico. Além disso, este setor precisava melhorar a forma de prestação de serviço, e as tecnologias de comércio eletrônico, supriram estas necessidades. Dentro deste contexto, uma tecnologia que vem se mostrando bastante útil ao comércio eletrônico, é a tecnologia de agentes inteligentes de software, em vários aspectos como no processo de negociação, no auxílio na busca de informações, na localização de produtos e também na prestação de um modo geral.

O modelo de agente proposto para o protótipo de turismo que tem como ênfase, sugerir serviços turísticos de acordo com um perfil previamente informado pelo visitante do site, está baseada na Resolução Distribuída de Problemas (RDP), onde o problema inicial é dividido em problemas menores e, distribuídos entre agentes, onde a interação entre esses agentes é realizada pela troca de mensagens, obedecendo, um certo protocolo de comunicação.

Um fator importante para o sistema é o Protocolo de Comunicação entre os agentes. Neste trabalho foi apresentado um modelo utilizando-se de diagramas de gráficos de estado da UML, onde a comunicação entre os agentes é realizada pela troca de mensagens sendo que, os Agentes mudam de estado sob determinadas condições. A mudança de estado determina a mudança de papel a ser assumida pelo agente.

Durante o presente trabalho também foi feita uma abordagem sobre os aspectos teóricos e sobre as pesquisas que estão sendo realizadas em relação à tecnologia de agentes. No capítulo quatro foi abordado com maior ênfase a metodologia de modelagem a objetos de maior utilização no momento, a UML, destacando-se a extensão UML para modelar Agentes, a AUML.

Para a implementação do protótipo foi feito uso de diversas tecnologias e ferramentas, devido ao próprio aspecto da aplicação. Foram utilizadas tecnologias como ASP, ADO e ferramentas como Flash 4.0, *Dreamweaver*, *Fireworks* da macromedia, para desenvolvimento das páginas do site e, para o desenvolvimento dos agentes foi utilizada a

ferramenta Visual basic da Microsoft, observando-se os aspectos de escalabilidade, multiencadeamento, segmentação segura e principalmente a comunicação assíncrona como uma forma de melhorar o desempenho.

Após testes realizados com o protótipo, pode-se dizer que os resultados foram satisfatórios, levando-se em conta a rapidez com que o problema foi solucionado e, dessa forma mostrando que, a divisão do problema geral em problemas menores, realmente simplifica a tarefa a ser realizada, tendo como principal ponto o desempenho da aplicação, que é um ponto fundamental em aplicações que rodam na Internet.

Não é pretensão deste trabalho esgotar a discussão sobre o assunto, pelo contrário, e como recomendações de pesquisas futuras deve-se ressaltar:

- Desenvolvimento de agentes móveis para localizar remotamente nos *sites* dos Hotéis os preços das diárias;
- À medida que for implementado, os agentes móveis, seja desenvolvido um processo de negociação para determinar o melhor preço da diária;
- Fazer um estudo de outros paradigmas de linguagens de programação que possam ser utilizados na implementação dos agentes.

## Anexo I

### Glossário

Ação	Uma computação atômica executável que resulta em alteração do estado do sistema ou no retorno de um valor.
Ação assíncrona	Uma solicitação em que o objeto emissor não faz uma pausa para aguardar os resultados.
Ação síncrona	Uma solicitação em que o objeto emissor faz uma pausa para aguardar os resultados.
Artefato	Um conjunto de informações utilizado ou produzido por um processo de desenvolvimento de software.
Ator	Um conjunto coerente de papéis que os usuários de caso de uso desempenho ao interagir com os casos de uso.
Atributo	Uma propriedade nomeada de um classificador, descrevendo uma faixa de valores que as instâncias da propriedade poderão manter.
Característica	Uma propriedade, como uma operação ou um atributo, que é encapsulada em outra propriedade, como uma interface, uma classe ou um tipo de dados.
Caso de uso	A descrição de um conjunto de seqüências de ações, incluindo variantes, que um sistema realiza, fornecendo o resultado observável do valor de um ator.
Componente	Uma parte física e substituível de um sistema com o qual está em conformidade e proporciona a realização de um conjunto de interfaces.
Diagrama	A apresentação gráfica de um conjunto de elementos, em geral representada como um gráfico conexão de vértices (itens) e arcos (relacionamentos).

Diagrama de atividade	Um diagrama que mostra o fluxo de uma atividade para outra; os diagramas de atividade abrangem a visão dinâmica do sistema.
Diagrama de caso de uso	Um diagrama que mostra um conjunto de casos de uso e atores e seus relacionamentos; o diagrama de caso de uso abrange a visão estática de caso de uso de um sistema.
Diagrama de classes	O diagrama que mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Estes abrangem a visão estática de projeto de um sistema.
Diagrama de colaboração	Um diagrama de interação que dá ênfase à organização estrutural de objetos que enviam e recebem mensagens; um diagrama que mostra as interações organizadas ao redor de instâncias e os vínculos entre elas.
Diagrama de componentes	Um diagrama que mostra a organização e as dependências existentes em um conjunto de componentes; os diagramas de componentes abrangem a visão estática de implementação de um sistema.
Diagrama de gráfico de estados	Um diagrama que mostra uma máquina de estados; os diagramas de gráficos de estados abrangem a visão dinâmica de um sistema.
Diagrama de implantação	Um diagrama que mostra a configuração dos nós de processamento em tempo de execução e os componentes que neles existem; um diagrama de implantação abrange a visão estática de funcionamento de um sistema.
Diagrama de interação	Um diagrama que mostra uma interação, composta por um conjunto de objetos e seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles; os diagramas de interação abrangem a visão dinâmica de um sistema.
Diagrama de seqüência	Um diagrama de interação que dá ênfase à ordenação temporal de mensagens.



Estado	Uma condição ou situação durante a vida de um objeto, durante a qual ele satisfaz alguma condição, realiza uma atividade ou aguarda algum evento.
Estímulo	Uma operação ou um sinal.
Evento	A especificação de uma ocorrência significativa, que tem uma localização no tempo e no espaço; no contexto de uma máquina de estados, o evento é a ocorrência de um estímulo capaz de ativar a transição de um estado.
Herança	O mecanismo pelo qual, elementos mais específicos incorporam a estrutura e o comportamento de elementos mais gerais.
Instância	Uma manifestação concreta de uma abstração; uma entidade à qual, um conjunto de operações podem ser aplicadas, e que tem um estado para armazenar os efeitos das operações.
Interação	Um comportamento que abrange um conjunto de mensagens trocadas entre um conjunto de objetos em um determinado contexto para a realização de um propósito.
Interface	Uma coleção de operações utilizadas para especificar o serviço de uma classe ou de um componente.
Máquina de estado	Um comportamento que especifica as seqüências de estados pelas quais um objeto passa durante seu tempo de vida como resposta a eventos, juntamente com suas respostas a esses eventos.
Mensagem	A especificação de uma comunicação entre objetos que contém informações à espera de que a atividade acontecerá; o destinatário da instância de uma mensagem costuma ser considerado a instância de um evento.
Método	A implementação de uma operação.
Modelo	Uma simplificação da realidade, criada com a finalidade de proporcionar uma melhor compreensão

	do sistema que está sendo gerado; uma abstração semântica próxima de um sistema.
Nó	Um elemento físico existente em tempo de execução que representa um recurso computacional, geralmente dispendo de pelo menos alguma memória e, na maioria das vezes, capacidade de processamento.
Objeto	Uma manifestação concreta de uma abstração; uma entidade com uma fronteira bem-definida e uma identidade que encapsula estado e comportamento; a instância de uma classe.
Pacote	Um mecanismo de propósito geral para a organização de elementos em grupos.
Papel	O comportamento de uma entidade que participa de um determinado contexto.
Parâmetro	A especificação de uma variável que pode ser alterada, passada ou retornada.
Pós-condição	Uma restrição que precisa ser verdadeira na conclusão de uma operação.
Pré-condição	Uma restrição que precisa ser verdadeira quando uma operação é chamada.
Propriedade	Um valor nomeado, denotando uma característica de um elemento.
Template	Um elemento parametrizado.
Thread	Um fluxo de controle que pode ser executado concorrentemente com outros <i>threads</i> no mesmo processo.
UML ( <i>Unified Modeling Language</i> )	Linguagem de Modelagem Unificada, uma linguagem para a visualização, especificação, construção e documentação de artefatos de um sistema complexo de software.



**BIBLIOGRAFIA**

BOGGS, Wendy; BOGGS, Michael. Mastering UML with Rational Rose. San Francisco SYBEX, 1999.

BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. UML Guia do Usuário. Rio de Janeiro: Campus, 2000.

BUCZEK, Greg. ASP guia do programador. São Paulo: Market Books, 2000.

BUER, Bernhard. Extending UML for the Specification of Agent Interaction Protocols. Disponível em: <<http://drogo.cseit.stet.it/fipa/>>. Acessado em: 10 junho 2000.

CHEONG, Fah-Chun. Internet Agents: Spiders, Wanderers, Brokers, and Bots. UK: New Riders Publishing, 1996.

COCKAYNE, William R. Mobile Agentes. Greenwich: Manning Publications Co, 1997.

COUGO, Paulo Sérgio. Modelagem Conceitual e Projeto de Banco de Dados. Rio de Janeiro: Campus, 1997.

DENNETT, Searle. The Intentional Stance. Disponível em: <<http://www.cs.umbc.edu/>>. Acessado em: 08 julho 2000.

FURLAN, José Davi. Modelagem de Objetos através da UML – the Unified Modeling Language. São Paulo: Makron Books, 1998.

GARFINKEL, Simson; SPAFFORD, Gene. Comércio & Segurança na Rede: Riscos, Tecnologias e Estratégias. São Paulo: Market Books Brasil, 1999.

GUNDERLOY, Mike. ADO Guia do Programador. São Paulo: Market Books Brasil Ltda, 2000.

JONES, A. Russell. Visual Basic guia do programador para ASP e IIS. São Paulo: Market Books, 2000.

REED, Paulo R. Jr.. Desenvolvendo Aplicativos com Visual Basic e UML. Projeto e Implementação de Aplicativos Orientados a Objetos. São Paulo: Markron Books do Brasil, 2000.

KIRTLAND, Mary. Projetando soluções baseadas em componentes. Rio de Janeiro: Campus, 1999.

MARCONDES, Christian Alfim. Programando HTML 4.0. São Paulo: Érica, 1998.

MARTIN, James. Princípios de análise e projeto baseados em objetos. Rio de Janeiro: Campus, 1994.

OBJECT MANAGEMENT GROUP. Agent Technology. Disponível em: <<http://www.omg.org/>>. Acessado em: 02 junho 2000.

ODELL, James. Agent Technology – Green Paper. USA. Object Management Group – OMG, 2000.

ODELL, James. Agents. Disponível em: <<http://www.cs.umbc.edu/>>. Acessado em: 01 agosto 2000.

ODELL, James; PARANUK, H. Van Dyke; BAUER, Bernhard. Extending UML for Agents. Disponível em: <<http://www.cs.umbc.edu/>>. Acessado em: 12 junho 2000.

RICH, Elaine; KNIGHT, Kevin. Inteligência artificial. São Paulo: Makron Books, 1993.

SEYBOLD, Patrícia; MARSHK, Ronni T.. Clientes.com. São Paulo: Markron Books, 2000.

STURM, Jake. VB6 UML Design and Development: Using Frameworks and Patterns. Canada: Wrox Press Ltd, 1999.

TECUCI, Gheorghe. Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies. California: Academic Press, 1998.

WANTSON, Ian. Applying Case-Based Reasoning: Techniques for Enterprise Systems. San Francisco: Morgan Kaufmann Publishers, 1997.

WILKINSON, Nancy M.. Using CRC Cards: An Informal Approach to Object-Oriented Development. AT&T Bell Laboratories. New York: Cambridge University Press, 1995