MÁRCIA ITO

**UMA ANÁLISE DO FLUXO DE COMUNICAÇÃO EM
ORGANIZAÇÕES DINÂMICAS DE AGENTES**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Engenharia.

São Paulo
1999

MÁRCIA ITO

**UMA ANÁLISE DO FLUXO DE COMUNICAÇÃO EM
ORGANIZAÇÕES DINÂMICAS DE AGENTES**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Engenharia.

Área de Concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Jaime Simão Sichman

São Paulo
1999

Ito, Márcia

Uma Análise do Fluxo de Comunicação em Organizações Dinâmicas de Agentes. São Paulo, 1999.
154 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Elétrica.

1. Sistemas Multiagentes 2. Fluxo de Comunicação 3. Orientação a Objetos - UML I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Elétrica.

*Dedico este trabalho a todos aqueles que contribuem para
ao crescimento do software com código aberto.*

"I consider that the golden rule requires that if I like a program I must share it with other people who like it."

(Richard Stallman - The GNU Manifesto)

Agradecimentos

Agradeço especialmente ao meu orientador Prof. Dr. Jaime Simão Sichman, pois sem a sua intervenção este trabalho não teria sido concluído da forma como se encontra.

Aos meus amigos e professores da Escola Politécnica, que me compreenderam e ajudaram nesta longa e difícil caminhada, suprindo, por vezes, minhas deficiências na área de engenharia: Profa. Dra. Edith Ranzini, Prof. Dr. João José Neto, Profa. Dra. Liria Matsumoto e Profa. Dra. Selma S. S. Melnikoff.

Aos meus amigos e professores da Universidade Federal de São Paulo que me incentivaram e apoiaram a iniciar a minha carreira acadêmica, Prof. Dr. Daniel Sigulem e Profa. Mônica Parente Ramos.

Especial agradecimentos à Profa. Dra. Lucia Cristina Iochida que me forneceu a base de conhecimento para as simulações realizadas e todo o apoio e incentivo nas horas mais difíceis.

Ao Flávio, Jomi Fred Hübner e Mario Donato que de alguma forma me ajudaram na implementação do sistema CENINT.

Agradecimentos mais que especiais para os meus dois grandes amigos, Marcelo Fagundes Barros e Mikael R. Kiremitdjian que além de me ajudarem no desenvolvimento do meu trabalho de mestrado, suportaram todos os meus dramas, dúvidas, depressões, crises, etc., durante todo esse período.

Aos meus pais, a minha irmã e ao meu tio Tsuneo Sakai que me incentivaram e deram todo o seu apoio para a conclusão deste trabalho.

Ao CIS-EPM e a toda a sua equipe pelo apoio no início do programa de mestrado.

Ao FDTE pelo apoio financeiro e recursos de infra-estrutura fornecidos no início do programa de mestrado.

Ao CNPq pelo apoio financeiro através de suas bolsas de incentivo a pesquisa (RHAE-DTI) e de mestrado.

Enfim, a todos aqueles que ajudaram de uma maneira ou outra na conclusão deste trabalho.

Sumário

LISTA DE TABELAS

LISTA DE FIGURAS

RESUMO

ABSTRACT

1. INTRODUÇÃO	1
1.1 CONTEXTO DO TRABALHO	12
1.2 OBJETIVOS DO TRABALHO	14
1.3 MOTIVAÇÕES E RESULTADOS ESPERADOS	14
1.4 ORGANIZAÇÃO DO TRABALHO	16
2. INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA	18
2.1 CONCEITOS BÁSICOS	18
2.1.1 <i>O que é um Agente?</i>	18
2.1.2 <i>Sociedade, Organização de Agentes e Ambiente</i>	20
2.1.3 <i>Interação entre Agentes</i>	22
2.2 HISTÓRICO	23
2.3 RESOLUÇÃO DISTRIBUÍDA DE PROBLEMAS E SISTEMAS MULTIAGENTES	26
2.3.1 <i>Resolução Distribuída de Problemas</i>	26
2.3.2 <i>Sistemas Multiagentes</i>	27
2.4 APLICAÇÕES	29
3. ORGANIZAÇÃO DE SISTEMAS MULTIAGENTES	31
3.1 CONCEITOS BÁSICOS	31
3.2 ORGANIZAÇÕES ESTÁTICAS E DINÂMICAS	33
3.3 COALISÕES BASEADAS EM DEPENDÊNCIA	34
3.3.1 <i>Teoria do Poder Social</i>	35
3.3.2 <i>Mecanismo de Raciocínio Social</i>	36
3.3.3 <i>Formação de Coalisões no Modelo CBD</i>	39
3.4 REDE CONTRACTUAL	41
3.4.1 <i>O Mecanismo de Negociação e Distribuição de Tarefas</i>	42
3.4.2 <i>Formação de Coalisões no Modelo RC</i>	43
3.4.3 <i>O Modelo RC*</i>	46
4. ANÁLISE COMPARATIVA DO FLUXO DE COMUNICAÇÃO ENTRE OS MODELOS CBD E RC*	47
4.1 ANÁLISE MATEMÁTICA DOS MODELOS	47
4.1.1 <i>Coalisões Baseadas em Dependência (CBD)</i>	48
4.1.2 <i>Rede Contractual (RC*)</i>	53
4.2 SIMPLIFICAÇÕES ADOTADAS	56
4.3 ANÁLISE COMPARATIVA DOS MODELOS	59
4.3.1 <i>Estudo sob o Aspecto do g_{critico}</i>	63
4.3.2 <i>Estudo sob o Aspecto do b_{critico}</i>	65
4.3.3 <i>Estudo sob o Aspecto do k_{critico}</i>	66
4.4 CONCLUSÕES PRELIMINARES	67

5. SISTEMA CENINT	69
5.1 POR QUE ORIENTAÇÃO A OBJETOS E JAVA ?	69
5.2 ESPECIFICAÇÃO DO SOFTWARE.....	72
5.2.1 <i>Especificação do Modelo RC*</i>	72
5.2.2 <i>Restrições, Requisitos e Premissas</i>	75
5.2.3 <i>Funcionalidades do Sistema</i>	76
5.3 ARQUITETURA DO SOFTWARE.....	81
5.3.1 <i>Pacote Ambiente</i>	82
5.3.2 <i>Pacote Agente</i>	85
5.3.3 <i>Pacote Comunicação</i>	90
5.4 IMPLEMENTAÇÃO DO SOFTWARE	97
5.4.1 <i>Escalonamento dos Processos no Sistema</i>	99
6. ESTUDO DE CASOS ESPECÍFICOS.....	108
6.1 MATERIAIS E MÉTODOS	108
6.2 EXPERIMENTOS REALIZADOS.....	110
6.3 RESULTADOS OBTIDOS	110
6.3.1 <i>Resultados Obtidos na Ausência de Parceiros</i>	112
6.3.2 <i>Resultados Obtidos na Ocorrência de Autonomia</i>	122
6.3.3 <i>Resultados Obtidos na Formação de Coalisões</i>	128
6.4 ANÁLISE DOS RESULTADOS	137
7. CONCLUSÕES E DISCUSSÕES.....	139
BIBLIOGRAFIA.....	144
APÊNDICE A – UML (UNIFIED MODELING LANGUAGE)	

Lista de Figuras

Figura 2.1 – Modelo de agente [DEM90]	21
Figura 2.2 – Taxionomia da tecnologia de agentes	24
Figura 2.3 - Representação esquemática de uma resolução distribuída de problemas.....	26
Figura 2.4 - Representação esquemática de um sistema multiagente	28
Figura 4.1 – Algoritmo do cálculo do fluxo de comunicação no modelo CBD	50
Figura 4.2 – Esquema do fluxo de comunicação na apresentação do modelo CBD	51
Figura 4.3 – Esquema do fluxo de comunicação no processo de formação de coalisção do modelo CBD	52
Figura 4.4 – Algoritmo do cálculo do fluxo de comunicação no modelo RC*	54
Figura 4.5 – Esquema do fluxo de mensagens no processo de formação de coalisção do modelo RC*	56
Figura 4.6 – Gráfico da equação (5) com $k=1$ e da equação (6) com $b = 1, b = 5$ e $b = 9$, para $n=10$	61
Figura 4.7 – Gráfico da equação (5) com $k=5$ e da equação (6) com $b = 1, b = 5$ e $b = 9$, para $n = 10$	62
Figura 4.8 – Gráfico da equação (5) com $k=9$ e da equação (6) com $b = 1, b = 5$ e $b = 9$, para $n = 10$	62
Figura 4.9 – Limite superior e inferior dos modelos CBD e RC* para $n= 10$	63
Figura 4.10 – Gráficos da equação de $g_{critico}$ para $k = 1, b=1, b = \left\lceil \frac{n}{2} \right\rceil$ e $b = n-1$	64
Figura 4.11 – Gráficos da equação de $g_{critico}$ para $k = \left\lceil \frac{n}{2} \right\rceil, b = \left\lceil \frac{n}{2} \right\rceil$ e $b = n-1$	65
Figura 5.1 – Diagrama de casos de uso do sistema CENINT.....	77
Figura 5.2 – Diagrama de pacote do sistema CENINT	83
Figura 5.3 – Diagrama de classe do sistema CENINT.....	84
Figura 5.4 - Diagrama de classe do pacote Ambiente.....	87
Figura 5.5 - Diagrama de classe do pacote Agente.....	89
Figura 5.6 - Diagrama de estados do agente.....	91
Figura 5.7 – Esquema de interação do protocolo de comunicação no sistema CENINT.	94
Figura 5.8 - Diagrama de classe do pacote Comunicação.....	96
Figura 5.9 - Diagrama de pacote da fase de projeto.....	98
Figura 5.10 - Diagrama de sequência da comunicação entre agentes.....	103
Figura 5.11 - Diagrama de sequência ao encontrar um par perfeito.....	105
Figura 5.12 - Diagrama de sequência ao expirar o tempo do anúncio.	106
Figura 5.13 - Diagrama de sequência ao terminar o ciclo.....	107
Figura 6.1 – Descrição externa do agente cardio3 gerada pelo sistema CENINT.....	110
Figura 6.2 – Resultado de um ciclo das simulações realizadas com o Sistema CENINT.	111
Figura 6.3 – Resultado final de uma das simulações realizadas com o Sistema CENINT.	112
Figura 6.4 – Descrição externa dos agentes da simulação 1	115
Figura 6.5 – Descrição externa dos agentes da simulação 2	118
Figura 6.6 – Descrição externa dos agentes da simulação 3	121
Figura 6.7 – Descrição externa dos agentes da simulação 4	125
Figura 6.8 – Descrição externa dos agentes da simulação 5	128
Figura 6.9 – Descrição externa dos agentes da simulação 6	131
Figura 6.10 – Descrição externa dos agentes da simulação 7	134
Figura 6.11 – Descrição externa dos agentes da simulação 8	137

Lista de Tabelas

<i>Tabela 2.1 – Tipos de agentes.....</i>	<i>20</i>
<i>Tabela 4.1 – Notação utilizada na análise matemática dos modelos.</i>	<i>48</i>
<i>Tabela 4.2 – Características comparativas entre os modelos CBD e RC*.....</i>	<i>57</i>
<i>Tabela 4.3 – Características que influenciam no fluxo de comunicação nos modelos CBD e RC*</i>	<i>58</i>
<i>Tabela 4.4 – Equações do número de mensagens nos modelos CBD e RC*</i>	<i>60</i>
<i>Tabela 4.5 – Equações do número de mensagens nos modelos CBD e RC* para n=10</i>	<i>61</i>
<i>Tabela 4.6 – Equações do $g_{crítico}$ nas situações expostas na seção 4.2.</i>	<i>64</i>
<i>Tabela 4.7 – Valores de $b_{crítico}$.....</i>	<i>66</i>
<i>Tabela 4.8 – Valores de $k_{crítico}$.....</i>	<i>66</i>
<i>Tabela 5.1 – Descrição sumária do pacote Ambiente.....</i>	<i>82</i>
<i>Tabela 5.2 – Descrição das classes do pacote Agente.....</i>	<i>88</i>
<i>Tabela 5.3 – Notação utilizada no protocolo de interação.....</i>	<i>90</i>
<i>Tabela 5.4 – Descrição das classes do pacote Comunicação.....</i>	<i>95</i>
<i>Tabela 5.5 – Descrição dos pacotes da linguagem Java utilizados no sistema CENINT</i>	<i>97</i>
<i>Tabela 5.6 – Descrição das classes auxiliares do sistema CENINT.....</i>	<i>100</i>
<i>Tabela 6.1 – Resultado obtido na simulação 1 do modelo CBD (manual).....</i>	<i>113</i>
<i>Tabela 6.2 – Resultado obtido na simulação 1 do modelo RC* (sistema CENINT).....</i>	<i>113</i>
<i>Tabela 6.3 – Resultado obtido na simulação 2 do modelo CBD (manual).....</i>	<i>116</i>
<i>Tabela 6.4 – Resultado obtido na simulação 2 do modelo RC* (sistema CENINT).....</i>	<i>116</i>
<i>Tabela 6.5 – Resultado obtido na simulação 3 do modelo CBD (manual).....</i>	<i>119</i>
<i>Tabela 6.6 – Resultado obtido na simulação 3 do modelo RC* (sistema CENINT).....</i>	<i>119</i>
<i>Tabela 6.7 – Resultado obtido na simulação 4 do modelo CBD (manual).....</i>	<i>122</i>
<i>Tabela 6.8 – Resultado obtido na simulação 4 do modelo RC* (sistema CENINT).....</i>	<i>123</i>
<i>Tabela 6.9 – Resultado obtido na simulação 5 do modelo CBD (manual).....</i>	<i>125</i>
<i>Tabela 6.10 – Resultado obtido na simulação 5 do modelo RC* (sistema CENINT).....</i>	<i>126</i>
<i>Tabela 6.11 – Resultado obtido na simulação 6 do modelo CBD (manual).....</i>	<i>129</i>
<i>Tabela 6.12 – Resultado obtido na simulação 6 do modelo RC* (sistema CENINT).....</i>	<i>129</i>
<i>Tabela 6.13 – Resultado obtido na simulação 7 do modelo CBD (manual).....</i>	<i>132</i>
<i>Tabela 6.14 – Resultado obtido na simulação 7 do modelo RC* (sistema CENINT).....</i>	<i>132</i>
<i>Tabela 6.15 – Resultado obtido na simulação 8 do modelo CBD (manual).....</i>	<i>135</i>
<i>Tabela 6.16 – Resultado obtido na simulação 8 do modelo RC* (sistema CENINT).....</i>	<i>135</i>
<i>Tabela 6.17 – Sumário dos resultados obtidos nas simulações</i>	<i>138</i>

Resumo

Dentre as várias áreas de pesquisa em Inteligência Artificial Distribuída, priorizamos analisar a comunicação entre os agentes de uma sociedade [MAR92] [BON88]. É através da comunicação que os agentes podem trocar informações entre si e assim resolver de forma cooperativa um problema global ou local que existe na sociedade. A análise do fluxo de comunicação entre agentes é portanto de grande interesse da comunidade científica que se dedica à IAD. Neste trabalho, através do estudo teórico (análise matemática) e simulações computacionais, comparamos o fluxo de comunicação entre os agentes de dois modelos de organizações dinâmicas: uma organização em que os agentes realizam uma busca informada de um parceiro (Coalisão Baseada em Dependências - CBD) e uma organização em que os agentes realizam uma busca não informada de um parceiro (Rede Contractual - RC).

O Sistema CENINT, um sistema multiagente (SMA) baseado no modelo RC, foi desenvolvido a fim de realizar as simulações necessárias para os estudos deste trabalho. Por outro lado, sabemos que os sistemas multiagentes são utilizados para desenvolver modelos teóricos que permitem elucidar a estrutura de processos complexos e que a orientação a objetos facilita o desenvolvimento de sistemas complexos. Percebeu-se que a orientação a objetos poderia ser uma ferramenta adequada para desenvolver sistemas multiagentes. Assim neste trabalho, optou-se por desenvolver o sistema CENINT utilizando as técnicas de orientação a objetos, mais especificamente utilizar os diagramas da UML (Unified Modeling Language) para análise e projeto do sistema.

1. Introdução

Neste capítulo, é apresentada uma visão geral do trabalho realizado. Para melhor situar o leitor, o contexto de pesquisa no qual o trabalho se insere é relatado na seção 1.1. Os objetivos do trabalho são então indicados na seção 1.2. As motivações para a sua realização, bem como os resultados esperados, encontram-se na seção 1.3. Finalmente, a seção 1.4 mostra como o restante do documento está organizado.

1.1 Contexto do Trabalho

Ao contrário da Inteligência Artificial Clássica, cujos modelos de inteligência baseiam-se no *comportamento individual humano*, e cujo principal enfoque é o estudo da representação do conhecimento e métodos de inferência, a Inteligência Artificial Distribuída (IAD) baseia-se no *comportamento social*, dando ênfase às ações e interações de suas entidades [SIC92].

Na IAD, um determinado problema a ser solucionado é dividido em subproblemas. Tal divisão é realizada de modo recursivo, até que se chegue a subproblemas elementares que são executados por entidades denominadas *agentes*. A interação destes agentes tem como resultado a solução do problema original. Ao conjunto de agentes dá-se o nome de *sociedade* de agentes. O modo como a sociedade de agentes é organizada para resolver o problema original é chamado de *organização*. Uma organização de agentes pode ser concebida de forma estática ou dinâmica [SIC92] [KRA93].

Numa organização estática, os agentes já possuem previamente um problema global a ser solucionado. Neste caso, o projetista normalmente adota um procedimento *top down*: a partir do problema original, ele organiza a sociedade de agentes de tal forma que estes encontrem a solução desejada [SIC95].

Numa organização dinâmica, os agentes não possuem a priori um único problema a ser solucionado: tais agentes pré-existem a eventuais problemas que possam ser solucionados pelo sistema. Como resultado de seus esforços para resolver seus próprios problemas, **coalisões dinâmicas** são formadas entre os agentes. Chamamos de coalisão esta noção de estrutura organizacional que é formada dinamicamente pelos agentes, num procedimento *bottom-up*. Nenhum tipo de cooperação pré-concebida ou estrutura organizacional prévia está estabelecida. A organização dinâmica foi muito utilizada para modelar sistemas multiagentes (SMA) cuja perspectiva é a simulação social. Hoje, porém, tal tipo de organização também é utilizada num contexto de resolução de problemas, especificamente em *SMAs abertos*. Entende-se por SMA aberto um sistema onde os agentes entram e saem dinamicamente da sociedade, sem um controle global externo [SIC95].

Dentre os diversos modelos de organizações dinâmicas, encontramos a **Rede Contractual** (RC) e as **Coalisões Baseadas em Dependência** (CBD), cujo estudo comparativo será o objeto deste trabalho. No modelo RC, um agente não tem informações a respeito das capacidades dos outros agentes. Deste modo, quando não puder solucionar seu próprio problema, realiza uma *busca não informada* de um parceiro, ou seja, procura aleatoriamente dentro da sociedade de agentes um ou mais agentes que possam ajudá-lo a resolver o problema original [SMI80] [SMI81]. No modelo CBD, os agentes possuem informações sobre as capacidades dos outros agentes. Neste modelo, um agente realiza uma *busca informada* de um possível parceiro: primeiramente, verifica certas propriedades, tais como as suas relações e situações de dependência¹ em relação aos outros para um determinado objetivo; analisa tais propriedades e assim descobre quais agentes seriam os seus melhores possíveis parceiros. Uma proposta de coalisão é então enviada a esses agentes [SIC94] [SIC95].

¹ Tais termos são melhor definidos na seção 3.3.

Enquanto no modelo RC os agentes são essencialmente *benevolentes*, este não é o caso geral no modelo CBD. Por benevolente, entende-se um agente que aceita todas as propostas de coalisção que lhes são enviadas, independentemente de seu próprio interesse em atingir os objetivos descritos na proposta recebida.

1.2 Objetivos do trabalho

Este trabalho tem como objetivo uma análise comparativa do fluxo de comunicação global de uma sociedade de agentes que segue estes dois modelos de organização: Rede Contractual² (RC) e Coalisões Baseadas em Dependências (CBD).

Inicialmente, é realizado um estudo matemático dos modelos. Estudos de alguns casos específicos são então apresentados. O estudo consiste em analisar o número de mensagens trocadas entre os agentes, onde alguns fatores que influenciam no fluxo de comunicação global entre os agentes são modificados a cada ciclo, conforme descrito no capítulo 4.

1.3 Motivações e Resultados Esperados

Uma das vantagens da IAD é a possibilidade que oferece de modelar sistemas complexos, que necessitam do conhecimento de domínios diferentes, e que envolvem, geralmente, a distribuição física de dados. Do ponto de vista computacional, tal abordagem permite o controle da complexidade e o suporte para a replicabilidade [SIC92] [CHA81].

Assim, a IAD permite solucionar parte dos problemas de *interações* em diversos domínios de aplicação complexos, tais como telecomunicações, serviços de informação, multimídia, televisões interativas, etc.. Tais aplicações vêm se tornando cada vez mais viáveis à medida que a tecnologia se desenvolve [LEE93].

² Na realidade existem diversas variantes deste modelo, embora suas diferenças sejam mínimas. Neste trabalho, chamaremos de RC* a variante particular estudada, que está melhor especificada na seção 3.4.

As interações na IAD, por sua vez, estão intimamente relacionadas com o modo de **organização** dos agentes. Assim o estudo do comportamento dos vários modelos existentes de organizações faz-se necessário e importante. Por outro lado, é através da comunicação que os agentes realizam o intercâmbio de informações e conhecimento que são vitais para a interação e organização dos agentes de uma sociedade [SIC92] [MAR92].

Deste modo, é interessante realizar uma análise e avaliação do rendimento dos sistemas de comunicação entre os agentes pertencentes a estes dois modelos distintos de organização dinâmica:

- uma organização na qual quando um agente tem um problema a resolver, este realiza uma busca informada de um parceiro (CBD);
- uma organização na qual quando um agente tem um problema a resolver, este realiza uma busca não informada de um parceiro (RC).

Uma primeira análise empírica dos modelos leva-nos a concluir que a troca de mensagens no modelo RC será maior do que a do modelo CBD. O raciocínio neste caso é o seguinte: numa busca informada (modelo CBD), a troca de informações será menor, já que há uma escolha prévia do possível parceiro. A solicitação de auxílio é sempre direcionada e a mensagem somente é enviada ao possível parceiro previamente escolhido. No modelo RC, não havendo uma escolha antecipada do parceiro, envia-se uma mensagem a todos os agentes da comunidade, solicitando auxílio para a execução de uma determinada tarefa.

Este raciocínio, entretanto, pode ser enganoso, pois não se deve esquecer que os agentes no modelo CBD necessitam adquirir e processar o conhecimento sobre os outros, bem como enviar-lhes uma representação de suas próprias capacidades. Temos aqui, portanto, um fator que pode aumentar o número de mensagens trocadas *antes* da formação da coalisção. No modelo CBD, a escolha de um possível parceiro também não significa necessariamente que a coalisção será formada, pois os agentes podem recusar as propostas que lhes são

enviadas. Tais mensagens, assim como propostas subseqüentes, teriam o efeito de aumentar o fluxo de mensagens na sociedade.

Verifica-se então que uma análise empírica dos dados não é suficiente para concluir o comportamento do fluxo de comunicação entre os agentes, justificando uma análise objetiva do assunto em questão. Como resultado desta análise, esperam-se obter algumas métricas que possam auxiliar um projetista de SMA na escolha do modelo de organização dinâmica mais adequada para o seu problema.

1.4 Organização do Trabalho

Os principais aspectos da IAD que permitem contextualizar o trabalho proposto nesta dissertação são abordados no capítulo 2. Neste capítulo, os conceitos de agente, sociedade, organização e tipos de interação entre agentes são definidos. São também apresentados um breve histórico da área, as duas grandes sub-áreas que compõem a IAD (Resolução Distribuída de Problemas e Sistemas Multiagentes) e as possíveis aplicações com esta tecnologia.

As organizações de Sistemas Multiagentes são detalhadas no capítulo 3: as suas definições básicas, os seus tipos (estática e dinâmica) e os modelos que compõem o estudo deste trabalho (modelo RC e CDB). Em particular, apresenta-se uma variante do modelo RC, denominado RC*, que é objeto de estudo do trabalho.

A análise comparativa do fluxo de mensagens entre agentes nos modelos CBD e RC* é apresentada no capítulo 4. A partir desta análise matemática, uma interpretação crítica preliminar destes estudos é também descrita.

O sistema CENINT, um SMA baseado no modelo RC*, é apresentado no capítulo 5. O seu objetivo, as restrições adotadas, sua descrição, o modelo do agente, a especificação do

protocolo de formação de coalisões e certas decisões de projeto e implementação são abordados neste capítulo.

O estudo de casos específicos é apresentado no capítulo 6. Em particular, complementa-se a análise preliminar apresentada no capítulo 4, relaxando algumas restrições ali adotadas.

No capítulo 7, são apresentadas as principais conclusões do trabalho, bem como algumas perspectivas para trabalhos futuros.

O sistema CENINT foi totalmente modelado através da análise orientada a objetos, utilizando os diagramas da UML (Unified Modeling Language), cujo resumo encontra-se no apêndice A.

2. Inteligência Artificial Distribuída

O objetivo da IAD é resolver problemas complexos, utilizando a união das técnicas de Inteligência Artificial e da resolução de problemas de forma distribuída [DEC87]. Neste capítulo serão abordados os principais aspectos da área, de modo a contextualizar o trabalho proposto. Na seção 2.1 são abordados os conceitos básicos que compõem o universo da IAD, como a definição das noções de agente, sociedade e organização de agentes, ambiente e protocolo de interação. A seção 2.2 descreve a evolução histórica da IAD. Suas duas sub-áreas são apresentadas e suas diferenças relatadas na seção 2.3: a resolução distribuída de problemas (RDP) e os sistemas multiagentes (SMA). Finalmente, as possíveis aplicações utilizando a abordagem IAD são apresentadas na seção 2.4.

2.1 Conceitos Básicos

2.1.1 O que é um Agente?

O termo agente é derivado do conceito de agência, referindo-se a utilizar alguém para atuar em seu lugar. Um agente representa uma pessoa que interage com outras para concluir uma tarefa [TUR98]. Por exemplo, ao utilizar uma agência de viagens, delegamos a outra pessoa realizar uma reserva de voo. Existem, porém, muitas definições do termo agente, sendo que cada uma delas tenta explicá-lo de acordo com a ótica de seu autor. Aqui estão algumas destas definições:

- “Um agente é uma entidade que percebe e atua no ambiente através de sensores e atuadores, respectivamente” [RUS95].

- “Agentes autônomos são sistemas computacionais que habitam algum ambiente complexo e dinâmico, sentindo e atuando de modo autônomo no ambiente, para realizar um conjunto de objetivos ou tarefas para o qual foram desenvolvidos” [MAE95].
- “Agente é uma entidade de software persistente dedicada a um propósito específico. O fato do agente ser persistente o distingue das subrotinas, pois os agentes têm as suas próprias idéias sobre como resolver os seus problemas, ou como organizar a sua própria agenda. Propósitos especiais os diferenciam das aplicações multifuncionais, já que os agentes são tipicamente menores” [SMI94].

De acordo com [NWA96], há muitas definições do termo agente pois o mesmo não é de uso exclusivo dos pesquisadores da área, sendo utilizado de forma indistinta por várias outras. Além disso, os próprios pesquisadores não entram em consenso, apresentando outros sinônimos para aumentar ainda mais a confusão sobre o termo.

Para os propósitos deste trabalho, vamos considerar um **agente** como sendo uma entidade computacional inteligente que trabalha de forma cooperativa, pois trocas mútuas de informação são necessárias para permitir que o grupo como um todo produza uma solução para um problema proposto [DEM90].

Vamos descrever um agente genérico, proposto em [DEM90], com um mínimo de características com as quais se possa explicar os vários tipos de interações que os agentes podem realizar num universo multiagentes. Qualquer agente deve ter a representação do ambiente em que está inserido, ao qual chamaremos de *conhecimento*. Este conhecimento pode ser adquirido através de sua propriedade de *percepção* ou pela *comunicação* com outros agentes. É sempre possível também abstrair os *objetivos* pela observação do comportamento dos agentes. Esses objetivos podem estar codificados, adquiridos pelo contexto ou pela comunicação com outros agentes. Pelo conhecimento e objetivos, um agente pode gerar um conjunto de possibilidades de atuação no mundo³. Quando existem várias possibilidades, uma decisão precisa ser tomada para a escolha da melhor possível,

segundo algum critério próprio do agente. A melhor escolha para um agente entretanto, não é necessariamente a melhor escolha do ponto de vista da sociedade, pois individualmente os agentes possuem limitações, são heterogêneos, etc. (Figura 2.1) [DEM90]. Os diversos tipos de agentes, segundo algumas taxionomias, encontram-se descritos na tabela 2.1 [NWA96] [FRA96].

Tipo	Descrição
Deliberativo/ Cognitivo	O agente possui um modelo de resolução simbólica interna. Planeja e negocia com outros agentes.
Reativo	O agente reage dependendo do tipo de estímulo que o ambiente lhe envia.
Autônomo	O agente exerce controle sobre suas próprias ações.
Orientado a objetivos	O agente possui uma meta, não apenas reagindo ao meio externo.
Temporalmente contínuo	O agente permanece em processo contínuo.
Sociável	O agente interage com outros agentes, inclusive com pessoas.
Adaptativo	O agente muda o seu comportamento dependendo de experiências prévias.

Tabela 2.1 – Tipos de agentes

2.1.2 Sociedade, Organização de Agentes e Ambiente

Em IAD, uma comunidade de agentes é denominada **sociedade**. As sociedades existem a fim de que os agentes possam se agrupar com o objetivo de realizar tarefas complexas. Para que as atividades dos agentes ocorram de forma coordenada, permitindo que exista um comportamento coerente e efetivo, tanto no nível individual quanto no nível global, esta sociedade precisa estar estruturada, ou seja, organizada [SAN97]. Por exemplo, vamos supor uma sociedade hospital onde os agentes representam os médicos. Num determinado momento, eles devem definir os plantões. Podemos dizer que todos os médicos estão com um único objetivo que é montar a planilha de plantões do hospital. Neste momento, comportamentos inteligentes devem ser apresentados para alcançar o objetivo proposto para o grupo.

³ Dependendo da arquitetura interna do agente, tais possibilidades podem ser representadas em termos de

Figura 2.1 – Modelo de agente [DEM90]

O uso de papéis pré-determinados e padrões de comunicação para os agentes é uma forma de organizar a resolução do problema. Estruturar organizacionalmente reduz a complexidade do controle da resolução de problemas dos agentes pelo controle dinâmico adequado das decisões [LES90].

O conceito de **organização** exprime as restrições aplicadas aos agentes pertencentes a uma sociedade, isto é, os meios através dos quais o projetista de tais sistemas (ou os próprios agentes) podem garantir que cada agente desejará realizar e efetivamente realizará o que deve ser feito e no momento certo [ALV97]. O fato dos agentes se organizarem em torno de um único objetivo não significa que após a tarefa ser concluída os agentes deixam de

objetivos viáveis, planos para atingí-los, ações a serem executadas, etc..

existir, e sim que o grupo formado irá se dispersar e eventualmente não se organizará mais desta forma [FRA96].

Numa sociedade, as entidades passivas são designadas pelo termo **ambiente**. Os agentes são as entidades ativas que raciocinam sobre o ambiente, sobre os outros agentes e decidem racionalmente quais objetivos devem alcançar, quais ações devem tomar, etc. [PLE92].

2.1.3 Interação entre Agentes

O termo **interação** designa as trocas de informação que podem ocorrer entre os agentes. Os agentes que possuem a habilidade de se comunicar através de uma linguagem expressiva conseguem trabalhar cooperativamente para atingir objetivos não pré-determinados e/ou complexos. Linguagens que facilitam uma comunicação de alto nível são componentes importantes na arquitetura do agente. A tecnologia de agentes pressupõe a existência de linguagens de comunicação entre agentes e padrões/protocolos de comunicação entre processos em sistemas distribuídos (Figura 2.2) [MAY96].

As linguagens de agentes compreendem todas aquelas que podem ser utilizadas para implementar a comunicação entre os agentes. De uma forma geral, qualquer linguagem de programação pode ser utilizada para este fim. Um grande destaque deve ser dado para as linguagens interpretadas (*Script languages*), como o Tcl/Tk, Java, Telescript, etc.. Estas linguagens fornecem a abstração e os recursos necessários para a implementação de agentes móveis. Porém, essas linguagens diferenciam-se das linguagens de comunicação dos agentes pela sua finalidade primária, a qual é ser uma linguagem genérica de desenvolvimento de sistemas [MAY96].

As linguagens de comunicação entre agentes são projetadas especificamente para descrever e facilitar a comunicação entre os agentes. As categorias identificadas são [MAY96]:

- Modelos da comunicação humana – Tradicionalmente modelados de acordo com a “Teoria dos Atos da Fala” (*Speech Act Theory*) [AUS62] [SEA69] [CAM90], onde se

tenta capturar o princípio e as convenções da interação entre os seres humanos para os agentes. O princípio básico desta teoria é considerar que a elocução de mensagens deve ser considerada como uma ação planejada. São executadas por agentes com a intenção de realizar uma mudança desejada na sociedade, como por exemplo, solicitar uma coalisão a um outro agente para alcançar um determinado objetivo [WOO95].

- Estruturas teóricas (*Theoretical frameworks*) – São estruturas que contêm todos os aspectos dos estados internos referentes à comunicação de agentes, baseados em capacidades humanas de comunicação [COH95][SIN96]. Particular atenção é dada à mudança de estados durante a interação entre os agentes ou entre um agente e o ambiente.
- Linguagens para comunicação de agentes – São dirigidas exclusivamente para a comunicação entre os agentes. Não se restringem apenas a protocolos que trocam dados, pois as intenções sobre os dados que estão sendo trocados também são comunicadas. Nesta categoria temos o KQML.

Os padrões e protocolos de comunicação em sistemas distribuídos, como CORBA, ILU, OpenDoc, OLE, etc., são também apontados como uma solução para a comunicação entre os agentes. São estruturas projetadas para troca de dados e métodos. Como os agentes não se reduzem a dados ou métodos, esses padrões e protocolos são melhor vistos como um substrato para a implementação de agentes.

2.2 Histórico

Historicamente, distribuição e paralelismo são temas importantes em Inteligência Artificial (IA). As bases conceituais ligando processos concorrentes à IA foram inicialmente estabelecidas no meio dos anos 50, quando a visão do cérebro por Minsky foi declarada como uma sociedade de agentes cooperativos [MIN79] [MIN80] [MIN86], e também foi estabelecido o conceito de Arbib's de que o cérebro processa as informações como uma coleção de “*schemas*” concorrentes [ARB85] [ARB88].

Figura 2.2 – Taxionomia da tecnologia de agentes

O termo IAD, porém, só foi concebida no *The 1st Workshop on Distributed AI*, ocorrido em 1980 no MIT (Boston, USA) [BON88] [MAR92]. Desde então, com uma única exceção, um pequeno grupo de pesquisadores se encontram todos os anos para o *DAI Workshop* [MAR92].

As primeiras tentativas para resolver problemas de modo cooperativo ocorreram na década de 70. Uma delas foi o projeto HEARSAY-II [ERM80], um sistema para o reconhecimento da fala que originou o modelo de resolução de problemas conhecido como quadro-negro (*blackboard*) [ENG88].

Nos anos 70, surgem alguns modelos computacionais como os *beings* [LEN75] e os atores [AGH85], desenvolvidos com o objetivo de tratar problemas relacionados com recursos compartilhados, estruturas de controle complexas e comportamentos sofisticados a partir de interações muito simples. No início da década de 80, Konolige e Nilsson [KON80] relatam a importância da representação explícita, no seio de um agente, do conhecimento sobre outros agentes num contexto de planejamento. Em 1980, Smith [SMI80] apresenta um protocolo de negociação baseado na noção de mercado econômico, denominado rede contractual.

Na mesma época, Fox [FOX81] discute a utilização de modelos organizacionais humanos em sistemas computacionais. Ao mesmo tempo, tecnologias então novas, tais como orientação a objetos, robôs reativos [BRO86] e sistemas abertos [HEW86] tornam-se mais conhecidas e alguns ambientes computacionais são desenvolvidos para testar e implementar estas tecnologias, como o DVMT [COR93], DPSK [CAR87] e o MACE [GAS87]. Trabalhos mais recentes na área são TAEMS [DEC93], DRESUN [CAR91] [CAR95], VDT [LEV94], ARCHON [WIT92] [JEN95]. Estudos e conferências sobre teorias, arquiteturas e linguagens de agentes foram também iniciados nessa última década, para atender às necessidades da IAD. Tais estudos estão resumidos em [WOO95] [WOO95a] [WOO96].

Mais recentemente, na segunda metade dos anos 90, questões micros, tais como a estrutura e o modelo interno dos agentes, também tem sido preocupação dos estudiosos da área. Isso ocorre devido ao uso indiscriminado do termo agente por outras áreas [NWA96].

2.3 Resolução Distribuída de Problemas e Sistemas Multiagentes

A IAD é dividida em duas sub-áreas: Resolução Distribuída de Problemas (RDP) e Sistemas Multiagentes (SMA) [BON88]. A diferença entre tais sub-áreas está descrita a seguir.

2.3.1 Resolução Distribuída de Problemas

Em RDP, um agente é projetado com o propósito de resolver algum problema específico (Figura 2.3) [SIC92].

$(\text{Problema}) \Rightarrow (\text{projeto de agentes}) \Rightarrow (\text{resolução do problema}) \Rightarrow (\text{solução})$

Figura 2.3 - Representação esquemática de uma resolução distribuída de problemas

Como pode se perceber, a ênfase está no problema e em como colocar múltiplos agentes trabalhando juntos para resolvê-lo de forma coerente, robusta e eficiente [DUR94].

A estratégia de resolução segue as características [ALV97][SIC92]:

- O problema é resolvido por um conjunto de agentes, fisicamente distribuídos em diversas máquinas conectadas via rede. Os agentes são desenvolvidos para resolver aquele problema específico e são executados de modo concorrente, para aumentar a velocidade de resolução.
- Os agentes não precisam conhecer os demais agentes da sociedade, já que o sistema é desenvolvido para resolver um problema específico.

- Uma organização é concebida para restringir o comportamento destes agentes. Tal organização, na maioria dos casos, é completamente definida durante a fase de análise e projeto do sistema.
- A interação entre os agentes é realizada através de trocas de mensagens ou compartilhamento de dados comuns. A estrutura é quase sempre definida durante a fase de concepção do sistema, sendo relacionada ao modelo algorítmico subjacente e ao problema que o sistema deve resolver. A comunicação, quando existe, não precisa ser complexa, pois a distribuição e divisão das tarefas já foi projetada pelo analista.
- Os agentes cooperam, dividindo entre si as diversas partes do problema original (subproblemas, tarefas), ou podem até mesmo aplicar diferentes estratégias de resolução para uma mesma tarefa.
- Existe a noção de um controle global, na maior parte dos casos implícito aos agentes, que garante um comportamento global coerente do sistema, conforme a organização prevista.
- Novos agentes não podem ser adicionados dinamicamente na sociedade. Isto significa que este sistema não pode ser considerado um sistema aberto [HEW86].

Portanto, a RDP tem o seu foco em garantir certas propriedades externas em seus sistemas, tais como robustez e performance eficiente, sob condições ambientais diversas, de agentes com propriedades estabelecidas. A RDP resolve como uma coleção particular de agentes construídos para uma determinada finalidade pode ter algum nível de performance coletiva, sendo as propriedades do ambiente dinâmicas e incontroláveis [DUR94].

2.3.2 Sistemas Multiagentes

Um SMA consiste numa sociedade de agentes **autônomos** e inteligentes em atividade. Por autônomo, entende-se os agentes que possuem uma existência própria e conhecimento sobre si e sobre o mundo que os cerca. Nos SMA, os agentes coexistem num mesmo ambiente e interagem, a fim de que alcancem um objetivo comum (Figura 2.4) [SIC92].

(agentes) $\Rightarrow$ (problema) $\Rightarrow$ (interações) $\Rightarrow$ (resolução do problema) $\Rightarrow$ (solução)

Figura 2.4 - Representação esquemática de um sistema multiagente

O foco de SMA está nos agentes, e na possibilidade destes interagirem de modo significativo uns com os outros. Por exemplo, estudou-se como agentes autônomos podem estabelecer planos comuns que beneficiem a todos [ROS94].

Os pontos importantes em SMA (utilizando um modelo de agente cognitivo) são os seguintes [ALV97] [SIC92]:

- A decomposição das tarefas é feita pelos agentes. O agente neste caso deve decidir se deve mudar seu comportamento para melhor resolver a sua tarefa (existência de vários planos para a mesma tarefa).
- Os agentes são concebidos independentemente de um problema particular a ser resolvido. Assim, os agentes são capazes de resolver outros problemas, se possuírem o perfil para resolvê-los.
- A concepção das interações também é realizada independentemente da solução de um problema em particular. Tentam-se desenvolver protocolos de interação genéricos, que possam ser reutilizados em várias aplicações similares. Essa mesma filosofia pode ser aplicada ao projeto das organizações.
- Os agentes utilizam suas representações locais dos protocolos de interação e das organizações para raciocinar e agir. Não existe, portanto, um controle global do sistema.
- O SMA é um sistema aberto, ou seja, qualquer agente pode entrar ou sair da sociedade sempre que quiser.
- O ambiente pode mudar, e o agente deve incorporar mudanças no seu modelo interno do mundo.

O fato dos agentes possuírem características internas individuais diferentes e mesmo assim interagirem e chegarem a um objetivo comum é a principal característica de SMA [DUR94].

2.4 Aplicações

Atualmente, certas decisões no mundo dos negócios são tomadas pelas máquinas e estão afetando cada vez mais nossas vidas. Essas decisões por vezes são tomadas ainda com a ajuda de outras máquinas. Por exemplo, de acordo com Rosenschein and Zlotkin [ROS94], sistemas computacionais que controlam a rede elétrica e necessitam balancear os requisitos de energia podem compartilhar a eletricidade com outros sistemas controladores de outras redes. Portanto, se há falta ou excesso de consumo de eletricidade, um sistema computacional pode vender ou comprar o excesso de energia de outros sistemas aos quais estão conectados [TUR98].

Numa perspectiva similar, já existem disponíveis ferramentas tais como assistentes digitais pessoais. Esses assistentes pessoais assumem o papel dos inúmeros sistemas computacionais envolvidos no gerenciamento de nossa vida diária, como os notebooks, máquinas de fax, telefones e escalonadores automáticos. Outra aplicação são os bancos de dados compartilhados, onde as informações estão distribuídas de forma global. Agentes são desenvolvidos com a finalidade de facilitar a obtenção das informações nestes diversos bancos de dados. Temos ainda agentes especiais denominados *softboots*, que navegam na Internet e coletam informações de bancos de dados compartilhados mundialmente [ETZ94]. Finalmente, o controle e coordenação do tráfego terrestre e aéreo ilustram situações onde os agentes estão tomando decisões baseados na comunicação e acordo prévio com outros agentes [DUR87]. O comércio eletrônico é outra área promissora para a IAD [TUR98].

De acordo com [MAR92] [DUR94], a IAD é uma tecnologia apropriada para aplicações onde:

- o conhecimento é distribuído (ex.: projetos);
- a informação é distribuída (ex.: automação de escritório);
- os dados estão distribuídos (ex.: sensores distribuídos);
- a decisão é distribuída (ex.: controle de manufatura);
- as bases de conhecimentos são desenvolvidas independentemente, mas precisam ser interconectadas ou reutilizadas (ex.: próxima geração da engenharia do conhecimento).

3. Organização de Sistemas Multiagentes

Uma sociedade composta por múltiplos agentes interagindo entre si pressupõe uma organização de agentes. Nos vários tipos de organizações propostos ou desenvolvidos, uma grande variedade de agentes têm sido considerados: desde agentes totalmente autônomos (semelhantes a pessoas) até simples entidades de processamento (como regras ou pacotes de regras) [DUR94][DEM90].

Os conceitos e características básicas de uma organização de agentes são abordados na seção 3.1. Numa sociedade de agentes, existem várias maneiras destes cooperarem uns com os outros. Essas formas de cooperação definem os vários tipos possíveis de organizações de agentes. Estas organizações podem ser divididas em estáticas e dinâmicas, e são descritas na seção 3.2.

Sendo o foco do trabalho os modelos de organizações dinâmicas, na seção 3.3 é apresentado de forma detalhada o modelo das coalisões baseadas em dependência (CBD), e na seção 3.4 é descrito o modelo da rede contractual (RC).

3.1 Conceitos Básicos

Uma organização pode definir uma série de limitações e expectativas sobre o comportamento dos agentes, isto é, um conjunto de papéis que limitam suas decisões e ações. É de consenso geral que há vários modos de implementar uma organização, e nenhuma estrutura organizacional é apropriada para todas as situações, como bem demonstram muitas teorias de organizações humanas [BON88].

Alguns autores criaram taxionomias de organizações, já que à medida que crescem, estas se tornam mais complexas e incorporam atividades diversas [FOX79][FOX81]. Os tipos de organização, de acordo com este enfoque, são listados a seguir:

- Organização **hierárquica e centralizada** – Nesta organização, temos uma hierarquia forte onde os agentes obedecem aos seus superiores e o poder de decisão e controle se concentra em um dos agentes. Os outros agentes, após aceitarem tal poder e concordarem em ser os seus subordinados, realizam as tarefas que lhes foram incumbidas.
- Organização com uma **estrutura de autoridade** – Estruturas de autoridade são uma maneira de reduzir o trabalho de coordenação. Quando um agente possui autoridade sobre os outros, estes últimos devem aceitar a tarefa de atingir o objetivo do primeiro. Neste caso, não existe a possibilidade do agente não aceitar a legitimidade de poder de seus superiores, o que é possível no caso anterior.
- Organizações **Marketlike** – Muitos autores propuseram que os sistemas fossem organizados de acordo com a teoria de mercado econômico [KUR85] [MAL88] [RAM85] [SPR78]. A rede contractual, descrita na seção 3.4, encontra-se nesta linha, pois propõe uma organização baseada em contratos.
- Organizações como uma **comunidade com regras de comportamento** – Uma organização pode ser desenvolvida como um conjunto de interpretações locais de regras de comportamento, ou melhor, deveria ter uma estrutura externa bem definida [GAS88] [GER86] [LES83]. As regras derivam da estrutura comum dos agentes. As coalisões baseadas em dependências, descritas na seção 3.3, encontram-se neste tipo de organização, já que decisões locais são realizadas para tentar encontrar um parceiro.

Uma outra forma de classificar as organizações é proposta em [SIC95], e está descrita a seguir.

3.2 Organizações Estáticas e Dinâmicas

Um sistema que pressupõe a existência de um problema global, cuja resolução exige a integração de competências individuais lógica e/ou geograficamente distribuídas, pode estar organizado de forma **estática**. Neste contexto, a sociedade existe mais como uma metáfora, sendo as interações, a organização e os agentes pré determinados [CAS90].

Não há uma verdadeira interação social, cuja origem estaria ligada ao fato de agentes autônomos buscarem atingir seus próprios objetivos. Deste modo, numa organização estática tanto a especificação dos agentes quanto a definição da organização mais adequada são tradicionalmente de responsabilidade do projetista do sistema, que visa criar um conjunto capaz de oferecer o desempenho esperado. A organização escolhida é imposta aos agentes através da especificação de uma arquitetura pré-definida, que estabelece o fluxo de controle e comunicação a ser utilizado. O fluxo de controle tem por objetivo o gerenciamento global da atividade dos agentes, envolvendo funções tais como alocação de recursos, sequenciamento de operação (*scheduling*), promoção de cooperação e controle de competição [SAN97].

Normalmente, os sistemas de RDP (resolução distribuída de problemas) utilizam tal tipo de organização.

Numa organização **dinâmica**, os agentes não possuem papéis pré-definidos. As funções que cada agente exerce dentro da sociedade podem ser alteradas dinamicamente, durante o processo de resolução do problema. Nem todos os agentes da sociedade necessariamente participam ativamente na resolução do problema. Nenhum tipo de cooperação ou estrutura organizacional prévia está estabelecida [SIC95]. Neste contexto, o agente precede e independe da existência de um problema global. Embora participe da comunidade, o agente é visto como uma entidade autônoma, direcionada por motivações e objetivos locais próprios [SAN97]. Nesta organização existe a necessidade da interação social, ou seja, os agentes devem conseguir se agrupar e interagir a fim de alcançar os seus objetivos locais,

cujas combinações eventualmente resultem na solução do objetivo global da comunidade. Neste contexto, a organização é uma entidade dinâmica, criada e mantida pela própria comunidade [SAN97]. Utilizamos o termo **coalisão** como sinônimo de organização dinâmica.

Dentre os modelos de organizações dinâmicas, encontramos as Coalisões Baseadas em Dependência (CBD), fundamentada no mecanismo de raciocínio social [SIC95], e a Rede Contractual, baseada na teoria do mercado econômico [SMI80], que serão os objetos de estudo deste trabalho e desta forma apresentadas com maior detalhe nas seções subsequentes.

Existem outros modelos de organização dinâmica, baseadas em outras teorias, como por exemplo a teoria dos jogos [ROS94], da barganha [LOU96] [KRA93a] e da análise do comportamento social [STI92], que não serão aprofundados neste trabalho.

3.3 Coalisões Baseadas em Dependência

Em [SIC95], considera-se que um mecanismo de raciocínio social é essencial para o desenvolvimento de um agente realmente autônomo. É chamado de social qualquer mecanismo de raciocínio que utiliza a informação sobre os outros com o propósito de inferir conclusões.

O modelo de Coalisões Baseado em Dependência (CBD) [SIC95] [SIC94a] é um modelo de organização dinâmica baseado na Teoria do Poder Social [CAS90], que utiliza o conceito de *relações de dependência*. Nesta organização, antes de propor uma coalisão a qualquer outro agente, um agente analisa primeiro as relações de dependência que existem entre ele e os outros componentes da sociedade. O parceiro escolhido é aquele cuja possibilidade de cooperar seja a maior. Tal possibilidade cresce se existem vantagens para ambas as partes.

3.3.1 Teoria do Poder Social

Geralmente, em sistemas multiagentes, ignoram-se os conflitos na interação entre agentes. Estes modelos pressupõem que os agentes são benevolentes e estão prontos para cooperarem uns com os outros, tomando para si o objetivo dos outros. Neste caso, o processo de negociação serve apenas para compartilhar ou delegar ações entre os agentes da sociedade [GAL88]. Por outro lado, em uma interação com bases sociais (interação social), os agentes se associam porque lhes é interessante realizar esta parceria. Para possibilitar esta interação social, os agentes da sociedade devem levar em consideração o conceito de *dependência* e de *poder* de uns sobre os outros [CAS90].

Basicamente, a noção de poder relaciona um agente com um objetivo (**poder individual**) e não é um conceito social. Na terminologia adotada, temos dois tipos de relação:

- ***poder de (power of)***: um agente Ag1 tem o poder do objetivo G1 se ele pode finalmente realizar G1;
- ***poder sobre (power over)***: um agente Ag1 tem poder sobre o agente Ag2 (no que se refere ao objetivo G2) se ele pode ajudá-lo a alcançar ou impedi-lo de alcançar G2.

Tais noções de poder são *objetivas*, ou seja, existem independentemente do fato dos agentes estarem conscientes de sua existência. Quando estas relações de poder tornam-se *subjetivas*, ou seja, representadas na mente dos agentes, estes podem utilizá-las para tentar maximizar a consecução de seus objetivos. Em outras palavras, a partir das relações de poder individual, constroem-se os princípios do **poder social**. Este normalmente significa ter o **poder de influência** nos outros. Entende-se por poder de influenciar os outros o poder de um agente Ag1 modificar os objetivos, os princípios ou as emoções de Ag2 [CAS95].

É simples entender porque a base do poder de influência é o poder sobre. É através da comunicação ou pelo exercício do poder sobre Ag2 quanto a G2 que Ag1 tem o poder de influenciar Ag2, quanto a um outro objetivo G3 (por exemplo: o médico, utilizando o poder

sobre o paciente quanto ao objetivo de curar-se, adquire o poder de influenciar o paciente a parar de fumar).

Por sua vez, a noção de **dependência** é dual à de poder: Ag1 depende de Ag2 para G1 se Ag1 não tiver o poder de alcançar G1 e Ag2 o tem; ou se Ag1 tem o poder de alcançar G1 a menos que Ag2 o impeça de alcançá-lo. As noções de poder e de dependência formam a base e a razão da sociabilidade.

O conceito de **posse** é uma relação social, base da relação de dependência, que implica na noção de recurso (X). Recurso é uma condição necessária para que Ag2 alcance G2. Portanto, Ag1 tem a posse de X se:

- ele tem o poder de impedir Ag2 de ter acesso a X;
- ele tem o poder de delegar o acesso e utilização de X.

Assim, caso o recurso X seja necessário a Ag2, este também depende de Ag1 se Ag1 tem posse de X, ou seja, Ag1 adquire o poder sobre Ag2 caso tenha posse de X.

3.3.2 Mecanismo de Raciocínio Social

O modelo CBD baseia-se na teoria do poder social e em certas propriedades definidas no chamado mecanismo de raciocínio social [SIC95]. Utiliza principalmente a noção de **relações de dependência** entre os agentes.

Entende-se que uma parte fundamental de um agente autônomo é o mecanismo de raciocínio social. É chamado de social todo mecanismo de raciocínio que utiliza a informação sobre os outros para inferir algumas conclusões. Para isto, é necessário que o agente represente internamente as informações sobre os outros agentes da sociedade. Neste modelo, chama-se de **descrição externa** [SIC95] uma estrutura de dados, interna aos agentes, onde as características dos outros agentes que pertencem à sociedade são representadas.

Cada agente tem a sua própria descrição externa, onde estão armazenadas as informações sobre todos os agentes. A descrição externa é composta de:

- **objetivos** (*goals*): os objetivos que o agente quer atingir, uma vez que um mesmo agente pode ter mais de um objetivo;
- **ações** (*actions*): as ações que o agente pode realizar;
- **recursos** (*resources*): os recursos sobre os quais o agente tem posse;
- **planos** (*plans*): os planos que um agente possui para atingir os objetivos, independentemente do fato dele poder realizá-los sozinho ou não.

Uma **situação de objetivo** é uma propriedade que relaciona um agente a um objetivo. Dada a descrição externa de um agente Ag1, pode-se ter uma das seguintes situações de objetivo relativas a G1:

- **objetivo inexistente** (NG): Ag1 não possui G1 na sua lista de objetivos;
- **plano inexistente** (NP): Ag1 possui G1, porém não possui nenhum plano para realizá-lo;
- **autônomo** (AUT): Ag1 possui G1 e tem planos para realizá-lo. Pelo menos um destes planos permite que ele alcance G1 sem necessitar da ajuda de outro agente;
- **dependente** (DEP): Ag1 possui G1 e planos para realizá-los, porém nenhum destes planos pode ser executado sem que uma coalisão com outro agente seja formada.

As **relações de dependência** ocorrem quando um agente não tem como alcançar seu objetivo sozinho, precisando portanto da ajuda de outro que saiba realizar alguma ação que ele não sabe fazer. As relações de dependência podem ser:

- **unilaterais**: quando um agente Ag1 depende de outro Ag2 para atingir um determinado objetivo G1;
- **bilaterais**: quando ambos os agentes Ag1 e Ag2 dependem um do outro para atingirem seus objetivos G1 e G2. Podemos neste caso ter duas situações distintas:
 - * **dependência mútua**: os objetivos G1 e G2 são idênticos;
 - * **dependência recíproca**: os objetivos G1 e G2 são distintos.

Desta forma, se a situação de objetivo de Ag1 quanto a G1 for DEP, ele deve escolher um possível parceiro, dentre aqueles dos quais depende, para propor uma coalisão. Para isto, Ag1 desejará concluir se um possível parceiro Ag2 possui alguma relação de dependência em relação a ele, para algum dos objetivos de Ag2. Se assim o for, Ag1 concluirá que Ag2 também depende dele quanto a G2. Desta forma, Ag1 pode enviar uma **proposta de cooperação/escambo social** (caso G2 seja igual/diferente de G1) para Ag2 com uma maior probabilidade de uma resposta afirmativa.

Para que tal evento seja possível, os parceiros devem ter a percepção de que há uma relação de dependência entre eles. Assim, foi criada uma noção denominada **situação de dependência**, que relaciona dois agentes a um objetivo. Desta forma, dado um agente Ag1 que está raciocinando socialmente, suas possíveis situações de dependência em relação a Ag2 para o objetivo G1 podem ser:

- **independente (IND)**: Utilizando seus próprios planos, Ag1 infere que não depende de Ag2 para atingir G1;
- **crença local de dependência mútua (LBMD)**: Utilizando os seus próprios planos, Ag1 conclui que há uma dependência mútua entre Ag2 e si próprio relativa a G1. Entretanto, utilizando os planos que ele acredita que Ag2 possui, Ag1 não pode afirmar que Ag2 chegará à mesma conclusão;
- **crença mútua de dependência mútua (MBMD)**: Utilizando os seus próprios planos, e os que ele crê que Ag2 possui, Ag1 pode concluir que há uma dependência mútua entre eles relativa a G1;
- **crença local de dependência recíproca (LBRD)**: Utilizando os seus próprios planos, Ag1 conclui que há uma dependência recíproca entre ele e Ag2 para G1 (objetivo de Ag1) e G2 (objetivo de Ag2). Entretanto utilizando os planos que acredita que Ag2 possui, Ag1 não pode afirmar que Ag2 chegará à mesma conclusão;
- **crença mútua de dependência recíproca (MBRD)**: Utilizando os seus próprios planos e os que ele crê que Ag2 possui, Ag1 conclui que há uma dependência recíproca entre eles, relativa a G1 e G2;

- **dependência unilateral (UD):** Utilizando os seus próprios planos, Ag1 conclui que ele depende de Ag2 para atingir G1, porém Ag2 não possui qualquer tipo de dependência em relação a ele.

Utilizando a descrição externa, um agente pode inferir suas relações e situações de dependência com os outros agentes. Estas situações de dependência podem ser usadas para estabelecer um critério de escolha de parceiros para o agente que queira atingir um objetivo, mas não possui condições de realizá-lo por si só. Desta forma, a seguinte ordem de preferência de situações de dependência poderia ser utilizada como critério, como explicado mais detalhadamente em [SIC95] [SIC95a]:

$$UD < LBRD < \{LBMD, MBRD\} < MBMD$$

Assim, temos que a melhor escolha é um parceiro que tenha a situação de dependência MBMD, seguida por LBMD ou MBRD, sucessivamente por LBRD e finalmente por UD.

3.3.3 Formação de Coalisões no Modelo CBD

Dadas as definições anteriores, o processo de formação de coalisões no modelo CBD se passa do seguinte modo:

A. Escolha de um objetivo

Num primeiro passo, um agente Ag1 escolhe um determinado objetivo a atingir, dentre aqueles que constam em sua descrição externa. Tal escolha é guiada pela noção de objetivos atingíveis, descrita em [SIC96a]. Caso não existam mais objetivos a atingir, o agente Ag1 não tenta mais formar coalisões.

B. Escolha de um plano

Supondo que Ag1 tenha escolhido o objetivo G1, o próximo passo é que seja escolhido um plano para que se possa atingí-lo. Como o agente pode ter mais de um plano para o mesmo objetivo, a escolha do plano baseia-se na noção de plano factível, também descrita em [SIC96a]. Caso não existam planos para atingir G1 (situação de objetivo NP), escolhe-se um novo objetivo a atingir, retornando ao passo A.

C. Análise das ações no plano

Uma vez escolhido um plano, Ag1 analisa a sua situação de objetivo quanto a G1. Tal situação, conforme o item 3.3.2, pode ser AUT ou DEP⁴. Caso a situação seja AUT, Ag1 é considerado autônomo para realizar aquele objetivo e assim não necessita da cooperação de nenhum outro agente. Neste contexto, Ag1 pode iniciar suas atividades para atingir o seu objetivo G1 sem a necessidade de formar coalisões. Na situação DEP, entretanto, o Ag1 não pode iniciar imediatamente a execução do seu plano, pois primeiro precisa encontrar um agente que realize a ação⁵ que ele não sabe executar.

D. Escolha do parceiro

Através do mecanismo de resolução social, Ag1 calcula as suas relações e situações de dependência com os outros agentes relativas a G1 e através do critério pré-estabelecido, definido na seção 3.3.2, escolhe o(s) melhor(es) possível (is) parceiro(s). Caso não existam possíveis parceiros para a ação em questão, Ag1 escolhe um novo plano para atingir G1, retornando ao passo B.

⁴ Neste caso, as situações NG e NP não ocorrem, pois garante-se nos dois passos anteriores que Ag1 tem um objetivo (G1) e ao menos um plano para atingí-lo.

⁵ Na verdade, um agente pode necessitar da ajuda de vários agentes, para realizar várias ações que ele não é capaz de executar. Entretanto, para os fins de análise do processo de formação de coalisção que nos interessa, vamos considerar o caso onde apenas uma única ação é necessária.

E. Formação de coalisão entre os agentes

Uma vez escolhido o melhor possível parceiro, aqui chamado de Ag2, Ag1 lhe enviará uma proposta de coalisão, que pode ter as seguintes respostas [SIC95]:

- Ag2 aceita a proposta e a coalisão é formada. A partir deste momento, iniciam-se os trabalhos para resolver G1. Ao final deste processo, se as ações foram realizadas corretamente, G1 é dado como concluído e Ag1 pode retornar ao passo A.
- Ag2 recusa a proposta e neste caso Ag1 tenta respectivamente encontrar outro parceiro, retornando ao passo D.

A recusa da proposta por Ag2 pode ocorrer pelos seguintes fatores:

- Ag1 inferiu uma conclusão errada a respeito de Ag2, provavelmente por ter informações incorretas ou incompletas sobre Ag2. Neste caso, Ag2 informa a Ag1 tais informações, e Ag1 pode realizar uma revisão de suas crenças sobre Ag2 [SIC96b].
- Ag2 não considerou a proposta interessante para os seus objetivos.

Este modelo foi implementado completamente no sistema DEPINT e os detalhes de implementação podem ser encontrados em [SIC95] [SIC98].

3.4 Rede Contractual

A Rede Contractual (RC) [SMI80] [SMI81] [DAV83] é um modelo de organização dinâmica, cujo objetivo principal é realizar alocações adaptativas de tarefas numa sociedade de agentes, baseada na teoria do mercado econômico e utilizando o conceito de **negociação**. A sua estrutura é baseada em anúncio de tarefas, lances e contratos fechados. Este foi o primeiro trabalho que utilizou o processo de negociação para a formação de coalisão entre agentes.

3.4.1 O Mecanismo de Negociação e Distribuição de Tarefas

No mercado econômico, negociações são necessárias para que acordos entre empresas ou nações sejam realizados. Entende-se por negociação uma discussão em que as partes interessadas trocam informações entre si, até chegarem a um consenso. De acordo com [DAV83], existem três componentes da negociação que são essenciais:

- há troca de informações por ambas as partes;
- cada parte avalia as informações trocadas pela sua ótica;
- o acordo final ocorre por concordância de ambas as partes.

Segundo [SMI80], o processo de negociação na rede contractual possui quatro componentes importantes:

- o processo de negociação é realizado pelo contratante, e em nenhum momento, o seu controle é centralizado;
- há troca de informações entre contratado e contratante durante a negociação;
- cada parte envolvida na negociação avalia as informações pelas suas perspectivas;
- o aceite final é feito por seleção mútua, ou seja, um escolheu o outro para parceiro.

O controle é distribuído e dinâmico, pois o processamento e a comunicação não são focalizados num único agente. Os agentes não são previamente designados como contratantes ou contratados, estes são apenas papéis que podem pertencer dinamicamente a qualquer um dos agentes da sociedade durante o processo de resolução do problema.

Neste modelo, quando Ag1 possui o objetivo G1, este particiona-o em ações, delegando parte delas ou todas elas para serem executados por outros agentes (Agn), em bases contratuais. Um contrato é um acordo explícito entre Ag1 (agente contratante) e Ag2 (agente contratado). O contratante monitora a execução da ação e processa os resultados da sua execução. A responsabilidade do contratado é somente a de executar a ação solicitada.

3.4.2 Formação de Coalisões no Modelo RC

De acordo com o mecanismo de negociação e distribuição de tarefas, apresentadas no item 3.4.1, o processo de formação da coalisão no modelo RC se passa do seguinte modo [SMI80]:

A. Escolha de um objetivo e do plano

Sendo Ag1 o agente contratante, este escolhe um determinado objetivo dentre aqueles que deseja alcançar. Supondo que tenha sido escolhido um objetivo G1, através de um mecanismo interno o agente Ag1 particiona-o em ações que devem ser realizadas por outros agentes da sociedade. Como este modelo é genérico, o mecanismo de particionamento do objetivo depende do conhecimento prévio do agente, não sendo escopo deste trabalho definir tal mecanismo.

B. Anúncio da ação

O processo de formação de coalisão se inicia com Ag1 anunciando as ações aos outros agentes da sociedade, através do envio de uma **mensagem de anúncio** de ações (*task announcement message*). Esta mensagem é enviada a todos os agentes da sociedade.

Em modelos extendidos, os anúncios podem ser enviados a todos os agentes disponíveis, para um pequeno grupo de agentes ou a um único agente. Cada agente armazena um histórico dos contratos realizados. Assim, a escolha é realizada através de análise do histórico, onde o agente é escolhido por já ter realizado satisfatoriamente no passado a ação que será anunciada [SMI80].

A mensagem de anúncio de ações possui quatro campos principais:

- a **especificação desejável** (*eligibility specification*): o conjunto de critérios que Ag2 deve ter para que possa dar um lance. Esta informação impede que agentes não

qualificados enviem um lance ao contratante, diminuindo assim o tráfego de mensagens e o processo de avaliação dos possíveis contratados;

- a **descrição da ação** (*task abstraction*): uma descrição sumária da ação que está sendo anunciada;
- a **especificação do lance** (*bid specification*): uma descrição das informações necessárias para a avaliação de um lance pelo contratante. Isto impede que dados irrelevantes para aquela ação sejam enviados;
- o **tempo de duração do anúncio** (*expiration time*): o período em que o contratante estará recebendo lances dos interessados. Na verdade, o tempo não é um fator limitante da negociação, ele é estipulado para que o contratante tenha um controle sobre os anúncios enviados.

C. Envio de lances

Os agentes disponíveis são aqueles que não estão executando correntemente nenhuma ação. Estes verificam se há algum anúncio do seu interesse para emitirem um lance (*bid*). Além de enviar em um lance de acordo com o formulário especificado no anúncio, podem adicionar algumas exigências necessárias para executar a ação anunciada. Estas exigências indicam que o agente necessita de maiores informações sobre a ação anunciada, para que possa realmente se candidatar.

As exigências geralmente são recursos ou necessidade de subcontratos. Quando uma ação necessita de outra ação para ser concluída, um subcontrato pode ser gerado, relacionado com o contrato que o originou.

D. Escolha do contratado

À medida que os lances chegam, o contratante seleciona os agentes que lhe interessam e que estão realmente aptos para executar a ação anunciada. É elaborada uma lista ordenada dos candidatos que mais se aproximam do perfil desejado. A seleção é realizada em forma

de negociação. Os lances que possuem exigências entram em processo de troca de informações, até que todas as dúvidas sejam esclarecidas. O contratante não precisa necessariamente esperar que o período de permanência do anúncio expire para confirmar a formação da coalisão. Se receber um *lance perfeito*, o contrato é emitido imediatamente. Um lance é dito perfeito quando as suas exigências são totalmente aceitas pelo contratado. Se o anúncio expirou e não recebeu nenhum lance adequado, podem ocorrer as seguintes situações:

- escolhe-se o agente da lista que tenha o perfil mais próximo do desejado;
- transmite-se outro anúncio de ação;
- espera-se um tempo antes da transmissão do mesmo anúncio de ação.

E. Formação de coalisão entre os agentes

Uma vez selecionado um agente Ag2, estabelece-se a coalisão com a emissão de um contrato. Neste encontram-se as informações necessárias para o controle da ação: o modo de comunicação e as regras vigentes durante a coalisão.

Enquanto a coalisão existir, as seguintes mensagens podem estar transitando entre as partes:

- **mensagem de informação** (*information message*), meio de comunicação geral, qualquer informação pertinente à contratação do agente (“acertos finais”) e durante a realização da tarefas;
- **relatório** (*report*), que pode ter duas formas:
 - * relatório parcial, que informa se a ação está parcialmente executada. O relatório parcial é utilizado para controle do contratante, pois este pode desejar que uma outra ação inicie a sua execução assim que uma determinada ação esteja parcialmente concluída;

- * relatório final, que informa se a ação foi concluída. Geralmente junto com este relatório tem-se a mensagem de **descrição do resultado** (*result description*), que relata o resultado de uma ação executada.
- **mensagem de término** (*termination message*), enviada quando termina a coalisão entre os agentes.

3.4.3 O Modelo RC*

Como já citado na introdução deste trabalho, existem diversas variações do modelo RC. Como o objetivo deste trabalho é o de realizar uma análise comparativa entre os modelos CBD e RC, adotamos uma variante que simplificasse esta tarefa. Esta variante, daqui por diante denominada RC*, tem as seguintes características:

- O modelo RC original não comporta um agente autônomo, ou seja, uma vez tendo uma tarefa para realizar, este obrigatoriamente deve procurar por um parceiro. Para melhor comparação deste modelo com o modelo CBD, iremos supor que o modelo RC* também possui agentes autônomos.
- Com a finalidade de tornar o estudo comparativo mais adequado, estamos supondo que há somente um agente contratante, mesmo que o artigo original mencione mais de um agente contratante.
- No modelo RC original, existe a possibilidade de subcontratação, ou seja, um agente contrata um agente que pode por sua vez subcontratar outro agente e assim por diante, formando assim uma rede de contratos. Este é um critério utilizado por um agente para não enviar um lance, pois pode já se encontrar contratado ou esperando um contrato de outro agente. No modelo CBD, um objetivo é dividido em ações elementares, ou seja, não existem relações de dependência aninhadas. Assim, vamos supor que no modelo RC* as tarefas são divididas em ações elementares da mesma forma que no modelo CBD.

Neste trabalho, o modelo RC* foi implementado até a fase de formação de coalisão entre os agentes, gerando o Sistema CENINT, descrito no capítulo 5.

4. Análise Comparativa do Fluxo de Comunicação entre os Modelos CBD e RC*

Neste capítulo é realizada uma análise matemática do fluxo global de comunicação entre agentes dos modelos CBD e RC*. Tal análise leva em conta o número de mensagens trocadas entre os agentes, modificando-se o número total de agentes que compõem a sociedade, o número de ciclos de interação, o número de lances no caso do modelo RC* e o número de agentes que respondem a propostas de coalisão no modelo CBD.

Na seção 4.1 é apresentado o fluxo de comunicação dos modelos, bem como a análise matemática dos mesmos. A partir desta análise matemática genérica, propõem-se algumas simplificações relatadas na seção 4.2. Na seção 4.3, a análise comparativa é efetuada, quando tabelas e gráficos são apresentados para visualizar os resultados obtidos. Um sumário conclusivo dos resultados obtidos é apresentado na seção 4.4.

4.1 Análise Matemática dos Modelos

Para a análise matemática, vamos supor uma sociedade composta por n agentes, com $n \in \mathbb{N}$ e $n > 3$. Nesta sociedade há uma quantidade m , tal que $m < n$, de agentes candidatos a parceiros, ou seja, que sabem realizar uma determinada ação requisitada. O agente ativo, responsável por encontrar parceiros, deve tentar alcançar g objetivos. O agente gasta 1 ciclo para tentar alcançar um objetivo, conseqüentemente teremos g ciclos de resolução.

A notação utilizada deste ponto em diante encontra-se na tabela 4.1.

Notação	Descrição	Modelo	
		CBD	RC*
n	total de agentes na sociedade – $n \in \mathbb{N}$	X	X
m	total de possíveis parceiros – $m < n$	X	X
g	total de ciclos	X	X
b	total de agentes que enviaram o lance – $0 \leq b \leq m$		X
k	total de agentes aos quais foram enviadas propostas de coalisção – $0 \leq k \leq m$	X	
$S_{apresentação}$	total de mensagens de apresentação	X	
S_{CBD}	total de mensagens após a execução de todos os ciclos no modelo CBD	X	
S_{ciclo}^6	total de mensagens após a execução de um ciclo	X	X
$S_{anúncio}$	total de mensagens de anúncio		X
S_{lance}	total de mensagens de lance		X
S_{RC^*}	total de mensagens após a execução de todos os ciclos no modelo RC*		X

Tabela 4.1 – Notação utilizada na análise matemática dos modelos.

4.1.1 Coalisções Baseadas em Dependência (CBD)

O modelo CBD, descrito na seção 3.3, pressupõe que agentes tentem estabelecer coalisções raciocinando socialmente a respeito de suas relações de dependência.

Ilustrando com um modelo real, podemos supor um hospital em que os médicos devem conhecer as especialidades uns dos outros. Assim, quando existe a necessidade de uma interconsulta, a escolha é baseada nas relações de dependência que existem entre os médicos. Por exemplo: um endocrinologista necessita da opinião de um cardiologista. O

endocrinologista convida então o cardiologista para trabalhar em conjunto com ele, porque sabe que este tem um caso que necessita da opinião de um endocrinologista. O cardiologista pode aceitar ou não o convite. Mediante uma recusa, o endocrinologista então procura por um outro colega.

4.1.1.1 Características

Como pudemos observar pelo modelo descrito na seção 3.3, antes de escolher o seu parceiro e iniciar a formação de coalisões um agente deve “conhecer” os demais integrantes da sociedade, realizando o que chamamos de **apresentação prévia**. A cada entrada de um agente na sociedade, ele deve se apresentar aos demais, enquanto os outros também se apresentam a ele. Da mesma forma quando um agente sai da sociedade, ele deve avisar aos demais que a está deixando. Após a apresentação de todos, os **ciclos** se iniciam. Entende-se por ciclo o período de processamento que um agente necessita para alcançar um de seus objetivos. Não há, portanto, uma nova apresentação a cada ciclo⁷. O agente ativo, aquele que procura por um parceiro, escolhe um objetivo e um plano. Faz a análise do plano e verifica se ele é autônomo, ou seja, se não necessita da ajuda de outro agente. Sendo autônomo, o agente pode executar as ações do plano sozinho; neste caso, não há processo de formação de coalisção. Caso não o seja, inicia-se o processo de formação de coalisção. Tal processo consiste em verificar inicialmente quais agentes poderiam ser seus parceiros, através do conhecimento que o agente ativo possui dos demais. Após verificar quais os prováveis parceiros questiona a cada um deles, segundo uma ordem de preferência pré-estabelecida e definida na seção 3.3, até que um deles aceite ou todos recusem a colaborar. Ao aceitar a parceria, o agente parceiro recebe uma confirmação do acordo estabelecido. Na Figura 4.1, é mostrado o algoritmo que calcula o número total de mensagens numa sociedade com n agentes, para um número total de g ciclos.

⁶ Caso seja necessário oportunamente diferenciar entre os dois modelos, utilizar-se-á um segundo subscrito, isto é, $S_{ciclo_{CBD}}$ e $S_{ciclo_{RC*}}$.

⁷ Considera-se neste trabalho que a entrada de todos os agentes se dá no início do processamento.

```

depint(n,g)
Sap = n *(n-1); # Calcula apresentação dos agentes
  Faça de i = 0 até i < g
Se Ativo autonomo então
  inicia novo ciclo;
senão
  m = Ativo.procuraParceiro(plano); # Procura e calcula o total de prováveis parceiros
  k = 0; # Inicializa a quantidade de propostas realizadas
  encontrouParceiro = false;
  Enquanto (k ≤ m) ou (não encontrouParceiro) faça
    k = k + 1;
    Se provavel parceiro aceita proposta de coalisção então
      EncontrouParceiro = true;
      k = k + 1;
  Fim
  Sc = Sc + k; # Somatória das mensagens durante a coalisção
  Fim
  Stot = Sap + Sc; # Somatória das mensagens trocadas
  Fim

```

Figura 4.1 – Algoritmo do cálculo do fluxo de comunicação no modelo CBD

4.1.1.2 Análise Matemática

A apresentação prévia é realizada através de uma troca de mensagens entre os agentes. Cada agente envia **mensagens de apresentação** aos demais integrantes da sociedade (*broadcast*). Supondo n agentes se apresentando uns aos outros tem-se que cada agente deve enviar $(n-1)$ mensagens de apresentação.

A soma das mensagens na apresentação na fase de apresentação (Figura 4.2) segue a seguinte função:

$$S_{apresentação} = n(n-1)$$

A cada ciclo, toda a comunicação entre os agentes é realizada através de troca de mensagens. O agente ativo envia **mensagens de proposta de coalisção** até que encontre um parceiro ou que não existam mais possíveis parceiros. O possível parceiro sempre responde à proposta de coalisção enviando-lhe uma **mensagem de aceite** ou **de recusa** ou **de revisão** [SIC95]. Quando o agente ativo recebe uma mensagem de aceite, ele envia uma **mensagem**

de coalisção, fechando o acordo com o agente parceiro. Se nenhum parceiro for encontrado a mensagem de coalisção não é enviada.

Figura 4.2 – Esquema do fluxo de comunicação na apresentação do modelo CBD

Vamos supor uma sociedade com n agentes, onde m agentes podem realizar a ação desejada, e que são enviadas mensagens de proposta de coalisção para k agentes⁸. Num ciclo podemos ter os seguintes casos:

- Encontra-se o parceiro em algum ponto do processo de coalisção. O fluxo de comunicação neste caso (S_{ciclo}) deve obedecer a seguinte função:

$$S_{\text{ciclo}} = 2k + 1, \text{ onde } 0 < k \leq m$$

- Nenhum parceiro foi encontrado. O fluxo de comunicação neste caso (S_{ciclo}) deve obedecer a seguinte função:

$$S_{\text{ciclo}} = 2m$$

⁸ Isto quer dizer que $k - 1$ agentes enviaram mensagens de recusa ou revisão.

- O agente é autônomo ou não há possíveis parceiros na sociedade. O fluxo de comunicação neste caso (S_{ciclo}) deve obedecer a seguinte função:

$$S_{ciclo} = 0$$

Deste modo, o total de mensagens enviadas em cada ciclo (Figura 4.3) é:

$$S_{ciclo} = \begin{cases} 2m & \text{se não encontrou parceiro} \\ 2k + 1 & \text{se encontrou parceiro} \\ 0 & \text{se o agente é autônomo} \end{cases} \quad (1)$$

Considerando agora uma ocorrência de g ciclos, o número total de mensagens trocadas entre agentes após todos os ciclos realizados (S_{CBD}) é:

$$S_{CBD} = S_{apresentacao} + \sum_{i=1}^g S_{ciclo_i} = n(n-1) + \sum_{i=1}^g S_{ciclo_i}, \text{ onde}$$

$$S_{ciclo_i} = \begin{cases} 2m_i & \text{se não encontrou parceiro} \\ 2k_i + 1 & \text{se encontrou parceiro, com } 0 < k_i \leq m \\ 0 & \text{se o agente é autônomo} \end{cases} \quad (2)$$

Figura 4.3 – Esquema do fluxo de comunicação no processo de formação de coalisção do modelo CBD

Quando um agente se retira da sociedade, este deve avisar da sua saída aos demais agentes da sociedade. Assim deve-se acrescentar à somatória final $n-1$ mensagens para cada agente que se retira da sociedade. Entretanto, não trataremos este aspecto neste trabalho.

4.1.2 Rede Contractual (RC*)

O modelo RC*, descrito na seção 3.4, pressupõe um modelo de organização da sociedade em que os agentes aceitam ou não trabalhar em conjunto de acordo com as relações de contrato realizadas.

Ilustrando com um modelo real, podemos supor um hospital no qual os médicos, sempre que necessitem de uma interconsulta, escolhem os parceiros baseados em relações de contrato. Por exemplo: um endocrinologista necessita da opinião de um cardiologista. O endocrinologista então deixa uma carta convite no escaninho de cada médico do hospital. A carta convite contém a especialidade desejada, o quanto de conhecimento se necessita para uma resposta confiável e o tempo de validade da carta. O cardiologista que se enquadra na exigência da carta convite envia seu curriculum para o endocrinologista. O endocrinologista, de posse dos currículos dos cardiologistas, analisa-os e escolhe um parceiro. O endocrinologista envia então um contrato ao cardiologista escolhido.

4.1.2.1 Características

No modelo RC*, o agente não necessita ter conhecimento sobre os componentes da sociedade. Portanto, neste modelo não há uma apresentação prévia e o procedimento de formação de coalisões pode se iniciar de imediato. O agente ativo, aquele que procura por um parceiro, escolhe um objetivo e um plano. Faz a análise do plano e verifica se ele é autônomo para aquele plano, ou seja, não necessita da ajuda de outro agente. Sendo autônomo, o agente pode executar as ações do plano sozinho; neste caso, não há processo de formação de coalisão. Caso não o seja, o agente envia a todos os agentes da sociedade uma **mensagem de anúncio** da ação que ele necessita que seja executada. Os agentes que sabem realizar a ação e se enquadram nas exigências do anúncio respondem enviando uma

mensagem de lance ao agente ativo. O agente ativo recebe os lances, faz uma análise e escolhe um parceiro. A escolha pode ocorrer após um tempo limite (*time out*) ou se um parceiro **ideal** for encontrado. Entende-se como parceiro ideal aquele que envia um lance em que as exigências são totalmente respeitadas. Se um parceiro ideal não foi encontrado e o tempo expirou (*time out*), então escolhe-se o parceiro mais próximo às exigências do anúncio. O parceiro escolhido recebe uma **mensagem de contrato**, indicando que ele foi o escolhido. Na Figura 4.4, é mostrado o algoritmo que calcula o número total de mensagens numa sociedade com n agentes, para um número total de g ciclos.

```

cenint(n,g)
  Faça de i = 0 até i < g
  Se Ativo autonomo então
    inicia novo ciclo;
  senão
    Sanuncio = n - 1;
    parPerfeito = false;
    b = 0;
    Enquanto (não time out) ou (não parPerfeito) faça
      Se Ativo recebeu lance então
        parPerfeito = analisaLance();
        b = b + 1;
    fim
    Se (parPerfeito) ou (b > 0) então
      Sciclo = Sanuncio + b + 1;
    senão
      Sciclo = Sanuncio;
  Fim
  Src = Sciclo;
Fim

```

Figura 4.4 – Algoritmo do cálculo do fluxo de comunicação no modelo RC*

4.1.2.2 Análise Matemática

Neste modelo não há apresentações, e portanto não há um fluxo inicial de comunicação entre os agentes. Portanto, a cada ciclo o agente ativo envia **mensagens de anúncio** aos integrantes da sociedade, caso não seja autônomo para o objetivo a ser alcançado. O total de mensagens enviadas após o anúncio de uma ação ($S_{anuncio}$) é:

$$S_{anuncio} = n - 1$$

Supondo que m agentes podem realizar a ação e que somente b agentes conseguiram responder (**mensagem de lance**) ao anúncio antes de expirar o prazo de envio dos lances podemos ter os seguintes casos:

- Um parceiro foi escolhido ($b > 0$).

$$S_{lance} = b + 1, \text{ onde } 0 < b \leq m$$

- Não houve lances ($b = 0$).

$$S_{lance} = 0$$

Assim numa sociedade com n agentes, m possíveis parceiros, temos que o total de mensagens enviadas após cada ciclo é (Figura 4.5):

$$S_{ciclo} = S_{anuncio} + S_{lance} = n - 1 + S_{lance}, \text{ onde } S_{lance} = \begin{cases} b + 1 & \text{se } 0 < b \leq m \\ 0 & \text{se } b = 0 \end{cases} \quad (3)$$

No modelo RC*, a troca de mensagens ocorre exclusivamente durante o processo de formação de coalisões. O número total de mensagens trocadas entre agentes após todos os ciclos realizados (S_{RC*}) é:

$$S_{RC*} = \sum_{i=1}^g S_{ciclo_i}, \text{ onde}$$

$$S_{ciclo_i} = \begin{cases} n - 1 & \text{se não encontrou parceiro} \\ n + b_i & \text{se encontrou parceiro, com } 0 < b_i \leq m \\ 0 & \text{se o agente é autônomo} \end{cases} \quad (4)$$

Figura 4.5 – Esquema do fluxo de mensagens no processo de formação de coalisção do modelo RC*

4.2 Simplificações Adotadas

Pela análise matemática apresentada na seção 4.1, podem-se encontrar características comparativas entre os dois modelos (tabela 4.2). Da mesma forma, encontram-se as variáveis que influenciam na análise dos modelos (tabela 4.3).

Situação	Consequências	
	CBD	RC*
Entrada de agentes na sociedade	Afeta a sociedade, pois há a necessidade da apresentação	Não afeta a sociedade, não há apresentação neste modelo.
Saída de agentes da sociedade	Afeta a sociedade, pois há a necessidade de avisar a todos os agentes sobre a saída.	Não afeta a sociedade, neste modelo os agentes não têm conhecimento dos agentes presentes na sociedade
Mais de um possível parceiro	Enquanto há recusa por parte dos possíveis parceiros, a comunicação continua.	A escolha é feita a partir dos lances enviados. Não há recusa por parte dos possíveis parceiros.
Encontrou parceiro ideal	Ao analisar os prováveis parceiros, encontra o parceiro que pode ajudá-lo com mais eficiência e este aceita ajudá-lo.	Assim que encontra o parceiro ideal, não analisa mais os lances.
Dentre os possíveis parceiros não encontrou algum que aceitasse a proposta	É considerado o pior caso, pois aqui todos os possíveis parceiros recusam a proposta.	Não existe tal situação desde que lances tenham sido enviados.
Não há parceiros que saibam realizar a ação	Não há troca de mensagens desde que o agente ativo tenha as informações corretas a respeito dos demais membros da sociedade.	O agente ativo não tem como saber previamente que não há possíveis parceiros. A troca de mensagens se resume às mensagens de anúncio.
O agente ativo é autônomo ou não possui objetivos	Não há procura por parceiros.	Não há procura por parceiros.
O primeiro possível parceiro da lista aceita a proposta.	É considerado o melhor caso, pois as mensagens trocadas são uma proposta, um aceite e uma coalisão.	Não afeta a sociedade, pois anúncios são enviados a todos os agentes, lances são recebidos e o contrato é enviado.

Tabela 4.2 – Características comparativas entre os modelos CBD e RC*

CBD	RC*
Total de agentes na sociedade (n)	Total de agentes na sociedade (n)
Total de objetivos a alcançar (g)	Total de objetivos a alcançar (g)
Quantidade de possíveis parceiros (m)	Quantidade de possíveis parceiros (m)
Total de agentes que respondem à proposta (k)	Total de agentes que sabem realizar as ações e enviam lances (b)
Ordenação do convite	Tempo que expira o anúncio

Tabela 4.3 – Características que influenciam no fluxo de comunicação nos modelos CBD e RC*

Tendo os fatores acima mencionados, vamos supor para efeito de análise algumas simplificações. Seja um caso particular de uma sociedade composta de n agentes, com g objetivos que necessitem da mesma ação. Vamos supor ainda que $n - 1$ agentes possam executar esta ação⁹. Nesta sociedade os agentes nunca são autônomos¹⁰ e sempre encontram pelo menos um parceiro¹¹.

Podemos realizar um estudo comparativo com as seguinte características:

Constante (em todos os ciclos)	Variável
Ordenação dos convites	Total de agentes na sociedade (n)
Tempo em que se expira o anúncio	Total de objetivos a alcançar (g)
Total de agentes que sabem realizar as ações e enviam lances (b)	
Total de agentes que respondem à proposta (k)	
Quantidade de possíveis parceiros (m)	

⁹ Isto equivale a tomar o caso limite onde $m = n - 1$.

¹⁰ Neste caso, as terceiras definições das equações (2) e (4) nunca ocorrem.

¹¹ Neste caso, as primeiras definições das equações (2) e (4) também nunca ocorrem. Além disso, necessariamente temos $k \neq 0$ e $b \neq 0$ na tabela 4.1.

4.3 Análise Comparativa dos Modelos

Considerando as simplificações adotadas, o total de mensagens trocadas ao final de todos os ciclos é uma simplificação das equações (2) e (4), conforme temos abaixo:

1. CBD

Sendo n a quantidade de agentes na sociedade, $m=n-1$ a quantidade de possíveis parceiros, k o total de agentes que respondem à proposta e g o total de ciclos, temos a partir da equação (2):

$$S_{CBD} = n(n-1) + \sum_{i=1}^g (2k_i + 1)$$

$$S_{CBD} = n(n-1) + g(2k+1), \text{ onde } 0 < k \leq m \quad (5)$$

2. RC*

Sendo n a quantidade de agentes na sociedade, $m=n-1$ a quantidade de possíveis parceiros, b a quantidade de agentes que enviaram o lance antes do término do prazo e g o total de ciclos, temos a partir da equação (4):

$$S_{RC^*} = \sum_{i=1}^g (n + b_i)$$

$$S_{RC^*} = g(n + b), \text{ onde } 0 < b \leq m \quad (6)$$

Pelo que foi definido anteriormente, a quantidade de agentes que respondem à propostas (k) e agentes que enviam os lances (b) são fixos em todos os ciclos. Desta forma podemos fixá-los no seguintes valores:

i. Valores para k :

- $k = 1$, ou seja, o melhor caso no modelo CBD – limite inferior de k ;
- $k = \left\lceil \frac{n}{2} \right\rceil$, ou seja, caso intermediário no modelo – um valor intermediário para k ;
- $k = m$, ou seja, o pior caso no modelo CBD – limite superior de k .

ii. Valores para b :

- $b = 1$, ou seja, o agente contratante recebe somente um lance – limite inferior de b ;
- $b = \left\lceil \frac{n}{2} \right\rceil$, ou seja, o agente contratante recebe lance de metade dos agentes compõem a sociedade – um valor intermediário para b ;
- $b = m$, ou seja, o agente contratante recebe lance de todos os possíveis parceiros – limite superior de b .

Substituindo os valores acima mencionados nas equações (5) e (6), obtemos as equações referentes a estes casos, representadas na tabela 4.4.

	$k = 1$	$k = \left\lceil \frac{n}{2} \right\rceil$	$k = n-1$
$b = 1$	$S_{CBD} = n(n-1) + 3g$ $S_{RC^*} = g(n+1)$	$S_{CBD} = n(n-1) + g(2\left\lceil \frac{n}{2} \right\rceil + 1)$ $S_{RC^*} = g(n+1)$	$S_{CBD} = n(n-1) + g(2n-1)$ $S_{RC^*} = g(n+1)$
$b = \left\lceil \frac{n}{2} \right\rceil$	$S_{CBD} = n(n-1) + 3g$ $S_{RC^*} = g\left\lceil \frac{3n}{2} \right\rceil$	$S_{CBD} = n(n-1) + g(2\left\lceil \frac{n}{2} \right\rceil + 1)$ $S_{RC^*} = g\left\lceil \frac{3n}{2} \right\rceil$	$S_{CBD} = n(n-1) + g(2n-1)$ $S_{RC^*} = g\left\lceil \frac{3n}{2} \right\rceil$
$b = n-1$	$S_{CBD} = n(n-1) + 3g$ $S_{RC^*} = g(2n-1)$	$S_{CBD} = n(n-1) + g(2\left\lceil \frac{n}{2} \right\rceil + 1)$ $S_{RC^*} = g(2n-1)$	$S_{CBD} = n(n-1) + g(2n-1)$ $S_{RC^*} = g(2n-1)$

Tabela 4.4 – Equações do número de mensagens nos modelos CBD e RC*

Com a finalidade de podermos analisar graficamente essas situações, vamos supor uma sociedade com 10 agentes ($n = 10$) e substituir este valor nas equações da tabela 4.4. As equações resultantes encontram-se na tabela 4.5 e os respectivos gráficos nas Figuras 4.6, 4.7 e 4.8.

	$k = 1$	$k = 5$	$k = 9$
$b = 1$	$S_{CBD} = 90 + 3g$ $S_{RC^*} = 11g$	$S_{CBD} = 90 + 11g$ $S_{RC^*} = 11g$	$S_{CBD} = 90 + 19g$ $S_{RC^*} = 11g$
$b = 5$	$S_{CBD} = 90 + 3g$ $S_{RC^*} = 15g$	$S_{CBD} = 90 + 11g$ $S_{RC^*} = 15g$	$S_{CBD} = 90 + 19g$ $S_{RC^*} = 15g$
$b = 9$	$S_{CBD} = 90 + 3g$ $S_{RC^*} = 19g$	$S_{CBD} = 90 + 11g$ $S_{RC^*} = 19g$	$S_{CBD} = 90 + 19g$ $S_{RC^*} = 19g$

Tabela 4.5 – Equações do número de mensagens nos modelos CBD e RC* para $n=10$

Figura 4.6 – Gráfico da equação (5) com $k=1$ e da equação (6) com $b = 1, b = 5$ e $b = 9$, para $n=10$

Figura 4.7 – Gráfico da equação (5) com $k=5$ e da equação (6) com $b=1$, $b=5$ e $b=9$, para $n=10$

Figura 4.8 – Gráfico da equação (5) com $k=9$ e da equação (6) com $b=1$, $b=5$ e $b=9$, para $n=10$

Podemos colocar num mesmo gráfico os limites superiores e inferiores para ambos os modelos, numa sociedade com 10 agentes (Figura 4.9). A área hachurada na Figura indica todos os possíveis números de mensagens trocadas no modelo RC^* , para $n=10$, onde

$$S_{CBD} \leq S_{RC^*}.$$

Figura 4.9 – Limite superior e inferior dos modelos CBD e RC* para $n=10$

Através deste estudo inicial, pode-se verificar que um ponto de análise interessante é a intersecção entre as retas obtidas pelas equações (5) e (6), pois percebe-se que a partir deste ponto utilizar o modelo CBD se torna mais vantajoso ($S_{CBD} < S_{RC^*}$). Isto ocorre a partir de um ciclo específico que vamos chamar de $g_{critico}$.

4.3.1 Estudo sob o Aspecto do $g_{critico}$

O $g_{critico}$ é obtido igualando-se as equações (5) e (6), assim temos:

$$n(n-1) + g_{critico}(2k+1) = g_{critico}(n+b)$$

$$g_{critico} = \left\lceil \frac{n(n-1)}{n+b-2k-1} \right\rceil, \text{ onde } n+b-2k-1 > 0 \quad (7)$$

As equações do $g_{critico}$, referentes aos valores típicos de k e b a partir da equação (7), estão representadas na tabela 4.6. Os gráficos respectivos encontram-se nas Figuras 4.10 e 4.11. Nesta tabela, estão representados como impossível as situações onde $g_{critico}$ é negativo ou tende a infinito.

	$k = 1$	$k = \left\lceil \frac{n}{2} \right\rceil$	$k = n-1$
$b = 1$	$g_{\text{critico}} = \left\lceil n \left(\frac{n-1}{n-2} \right) \right\rceil$	Impossível	Impossível
$b = \left\lceil \frac{n}{2} \right\rceil$	$g_{\text{critico}} = \left\lceil \frac{2n}{3} \left(\frac{n-1}{n-2} \right) \right\rceil$	$g_{\text{critico}} = \left\lceil 2n \left(\frac{n-1}{n-2} \right) \right\rceil$	Impossível
$b = n-1$	$g_{\text{critico}} = \left\lceil \frac{n}{2} \left(\frac{n-1}{n-2} \right) \right\rceil$	$g_{\text{critico}} = \left\lceil n \left(\frac{n-1}{n-2} \right) \right\rceil$	Impossível

Tabela 4.6 – Equações do $g_{\text{crítico}}$ nas situações expostas na seção 4.2.

Figura 4.10 – Gráficos da equação de $g_{\text{crítico}}$ para $k=1$, $b=1$, $b = \left\lceil \frac{n}{2} \right\rceil$ e $b=n-1$

Figura 4.11 – Gráficos da equação de g_{critico} para $k = \left\lceil \frac{n}{2} \right\rceil$, $b = \left\lceil \frac{n}{2} \right\rceil$ e $b = n-1$

4.3.2 Estudo sob o Aspecto do b_{critico}

Pelo estudo anterior, vimos que em certas condições, fixados os valores de k e b , o modelo CBD nunca terá um número de mensagens menor do que o modelo RC*.

Suponhamos que a seguinte questão fosse levantada: Dado um valor fixo de k , qual o valor mínimo de b (lances) a partir do qual é possível obter-se $S_{\text{CBD}} < S_{\text{RC}^*}$? Podemos chamar esse número mínimo de lances de b_{critico} . O b_{critico} é obtido então igualando-se as expressões de $S_{\text{ciclo}_{\text{CBD}}}$ e $S_{\text{ciclo}_{\text{RC}^*}}$ nas equações (5) e (6):

$$2k + 1 = n + b_{\text{critico}}$$

$$b_{\text{critico}} = 2k - n + 1 \quad (8)$$

Substituindo os mesmos valores utilizados na seção 4.2.1 na equação (8), temos os valores de b_{critico} representados na tabela 4.7.

Valor de k	$b_{crítico}$	Observações
1	$3 - n$	Sempre é possível obter um certo $g_{crítico}$ tal que $S_{CBD} < S_{RC^*}$
$\left\lceil \frac{n}{2} \right\rceil$	1	
$n - 1$	$n - 1$	Nunca é possível obter $S_{CBD} < S_{RC^*}$

Tabela 4.7 – Valores de $b_{crítico}$

4.3.3 Estudo sob o Aspecto do $k_{crítico}$

De modo análogo, podemos fixar o valor de b e variar o valor de k , obtendo uma métrica similar denominada $k_{crítico}$. Neste caso, a questão a ser levantada seria a seguinte: Dado um valor fixo de b , qual o valor máximo de k abaixo do qual é possível obter-se $S_{CBD} < S_{RC^*}$? O $k_{crítico}$ também é obtido então igualando-se as expressões de $S_{ciclo_{CBD}}$ e $S_{ciclo_{RC^*}}$ nas equações (5) e (6):

$$2k_{crítico} + 1 = n + b$$

$$k_{crítico} = \left\lceil \frac{n + b - 1}{2} \right\rceil \quad (9)$$

Substituindo novamente os valores utilizados na seção 4.2.1 na equação (9), temos os valores de $k_{crítico}$ representados na tabela 4.8.

Valor de b	$k_{crítico}$	Observações
1	$\left\lceil \frac{n}{2} \right\rceil$	
$\left\lceil \frac{n}{2} \right\rceil$	$\left\lceil \frac{3n - 2}{4} \right\rceil$	
$n - 1$	$n - 1$	Sempre é possível um certo $g_{crítico}$ tal que $S_{CBD} < S_{RC^*}$

Tabela 4.8 – Valores de $k_{crítico}$

4.4 Conclusões Preliminares

Pelos gráficos da Figura 4.6, 4.7 e 4.8, observamos que para determinados valores de k e b há um ciclo ($g_{critico}$) a partir do qual o total de mensagens trocadas entre os agentes do modelo CBD é sempre menor que no modelo RC*. Percebe-se ainda que a medida que o número de agentes que respondem à proposta (k) aumenta é necessário um número cada vez maior de ciclos para que este possa ter um número de mensagens inferior ao modelo RC*. Este fato independe do número de lances (b) enviados no modelo RC*. Assim, num determinado momento ($k = n-1$) torna-se impossível para o modelo CBD ter um número de mensagens menor que o modelo RC*.

Tal situação pode ser entendida lembrando-se das equações de formação de coalisção dos modelos:

$$S_{RC^*} = \sum_{i=1}^g S_{ciclo_{RC^*_i}}$$

e

$$S_{CBD} = S_{apresentacao} + \sum_{i=1}^g S_{ciclo_{CBD_i}}$$

Como ambas as funções $S_{ciclo_{CBD}}$ e $S_{ciclo_{RC^*}}$ são lineares em relação a g , se o coeficiente linear do modelo RC* ($n + b$) for menor do que o do modelo CBD ($2k + 1$), então sempre teremos $S_{ciclo_{RC^*}} < S_{ciclo_{CBD}}$ e sempre teremos $S_{RC^*} < S_{CBD}$.

De uma forma geral, podemos concluir que dado um certo número total de agentes n sempre existirá um intervalo para os valores k e b (área hachurada na Figura 4.9) no qual sempre o modelo CBD conseguirá em algum momento ter um número menor de mensagens que o modelo RC*.

Pela Figura 4.9 podemos notar que sociedades onde os agentes sabem escolher um parceiro que responda de forma afirmativa numa primeira tentativa ($k = 1$) no modelo CBD sempre

terão em algum momento o número de mensagens inferior ao modelo RC*, não importando o número de agentes na sociedade. Por outro lado, à medida que o agente ativo no modelo CBD receber um número maior de recusas, o número de ciclos deve ser cada vez mais alto para que o modelo RC* possa superar o modelo CBD em número total de mensagens.

Ao fixar o número de agentes que receberam propostas de coalisão (k) em todos os ciclos e estudando o fluxo de comunicação de tal forma a encontrar a partir de que lance tem-se um número maior de mensagens no modelo RC* em relação ao modelo CBD ($g_{crítico}$), podemos notar que:

- para $k=1$, **sempre** existe um momento que $S_{ciclo_{CBD}} < S_{ciclo_{RC^*}}$;
- para $k = n-1$, **nunca** existe um momento que $S_{ciclo_{CBD}} < S_{ciclo_{RC^*}}$.

Com isto, podemos concluir que à medida que um agente no modelo CBD tem dificuldade em encontrar um parceiro (o número de recusas é maior), torna-se mais difícil conseguir neste modelo um fluxo de comunicação menor que o do modelo RC*.

Assim, podemos concluir inicialmente que o modelo CBD demonstra-se melhor nos casos em que o número de recusas for baixo e que se torna inviável à medida que as recusas aumentam. Dito de outra forma, caso o mecanismo de raciocínio social for eficaz (valor de k baixo e próximo de 1), o modelo CBD terá, a partir de um certo número de ciclos de interação, um fluxo de comunicação global menor do que o do modelo RC*.

5. Sistema CENINT

Este capítulo apresenta o sistema CENINT, um sistema multiagentes desenvolvido a partir do modelo RC*, descrito na seção 3.4.

Na seção 5.1, descrevem-se algumas escolhas feitas durante o desenvolvimento do sistema. A sua especificação detalhada, as restrições adotadas e as funcionalidades do sistema são apresentadas na seção 5.2. A arquitetura do sistema, que compreende o mapeamento das classes, dos módulos (pacotes) e o protocolo de comunicação, encontra-se descrita na seção 5.3. Alguns detalhes relevantes da implementação, tal como a estratégia de sincronização utilizada, são descritos na seção 5.4.

Por concisão, no decorrer deste capítulo apresentamos apenas as principais características do sistema. Uma descrição completa pode ser consultada em [ITO99].

5.1 Por que Orientação a Objetos e Java ?

Aplicações baseadas em agentes consistem em uma nova abordagem, de diferente nível de abstração, sobre um determinado problema e do projeto de sua solução. Todavia, implementar tais soluções requer uma infraestrutura de serviços [LEV97]:

- Criação do agente: Como cada agente necessita ser executado de forma independente, este requer uma infraestrutura para disparar novos processos ou *threads* e mecanismos de escalonamento de tarefas.
- Meio de comunicação entre agentes: Um agente deve ser capaz de compor mensagens e enviá-las para os outros agentes. Necessita, portanto, de um protocolo de interação, formato do conteúdo da mensagem, mecanismos de resolução de nomes para

determinar os destinatários, bem como um suporte de comunicação e rede para envio destas mensagens.

- Meio de locomoção de agentes: Agentes podem ser capazes de se moverem de uma máquina a outra. Neste trabalho, os agentes não são capazes de realizar tal ato, porém a arquitetura proposta permite uma implementação deste porte no futuro.

As plataformas de *hardware* e ferramentas de desenvolvimento de software existentes até o momento foram projetadas para construir aplicações centralizadas e tradicionais. Torna-se interessante que tal infraestrutura seja desenvolvida para facilitar o projeto de aplicações baseadas em agentes [LEV97].

Escolheu-se para o desenvolvimento do sistema a abordagem orientada a objetos, pois dentre suas características temos [TAY91][BOO94]:

- A orientação a objetos é uma tecnologia que produz modelos que refletem um domínio, assim como negócios e máquinas, de uma forma natural, utilizando o próprio vocabulário do domínio.
- Os modelos de orientação a objetos, quando construídos corretamente, são fáceis e simples de se entender, modificar, expandir e validar o domínio.
- Os sistemas desenvolvidos utilizando a tecnologia de orientação a objetos são flexíveis, têm uma arquitetura bem definida e fornecem a oportunidade para criar e implementar componentes reutilizáveis.
- Os modelos gerados com o paradigma da orientação a objetos são facilmente convertidos em código ao se utilizar uma linguagem de programação orientada a objetos. Ressalta-se, entretanto, que a orientação a objetos não se limita a simples mecanismos para a programação.

Além dessas características gerais, de acordo com [LEV97], no nível de implementação agentes podem ser vistos como uma extensão de objetos. Os objetos são entidades encapsuladas que modelam um sistema de uma forma modular onde estas entidades devem interagir entre si. Por outro lado, objetos são entidades passivas que somente mudam o seu

estado em resposta a uma função neles invocadas, e não têm controle sobre estas funções. Ao contrário, agentes são autônomos e decidem quando realizar as funções invocadas por outros agentes.

Assim preliminarmente, pode-se concluir que um agente é uma entidade computacional mais complexa do que um objeto. O agente não responde apenas a um serviço nele invocado, mas raciocina e decide se deve ou não realizar o serviço requisitado através de seus próprios critérios. Os princípios da orientação a objetos se mostram mais adequados para a implementação de agentes e sistemas multiagentes, uma vez que estes conseguem encapsular os componentes existentes nos sistemas deste porte, permitindo agregar novos modelos de agentes, de comunicação e de organização, como serão explicados ao longo deste capítulo.

Para a implementação do sistema escolheu-se a linguagem de programação Java. A decisão de utilizar tal linguagem deve-se às suas características [FLA97] [LIN96] [GOS96]:

- Trata-se de uma linguagem de programação orientada a objetos, ou seja, não é possível programar de forma estruturada com o Java. Uma classe em Java contém atributos e métodos.
- Java é uma linguagem semi-interpretada, o que a torna extremamente portátil. Isto é possível pois possui uma arquitetura neutra, ou seja, o código binário (*byte code*) originado da compilação do código fonte pode ser executado independente da plataforma e do sistema operacional existente, utilizando-se JVM (*Java Virtual Machine*).
- Java possui um extenso conjunto de classes, organizadas em pacotes que podem ser utilizados diretamente no programa. Isso facilita a programação e promove a reutilização de código. É possível ainda implementar pacotes específicos.
- É uma linguagem dinâmica e distribuída. Qualquer classe implementada em Java pode ser dinamicamente instanciada e distribuída, pois existe um alto nível de operações de rede. Outra vantagem é a utilização de RMI (*Remote Method Invocation*), através da qual é possível invocar um método em objetos remotos. A combinação dessas duas

características permitem que sistemas desenvolvidos em Java possam “caminhar” pela rede e ser executados não somente em suas máquinas nativas.

- Possui suporte para implementação de *threads* para criar aplicações *multithreads*. Utiliza-se de monitoração individual de objetos e classes.
- Com o *Java Native Method Interface*, todos os programas executados pelo JVM podem ativar programas escritos em C e C++.

Portanto, dadas estas características, a linguagem Java demonstra ser adequada para a implementação de sistema multiagentes, pois possui todas as funcionalidades necessárias para projetar e implementar sistemas complexos. O fato de ser portátil e apresentado em pacotes permite que os agentes possam ser executados em várias plataformas e da mesma forma interagirem entre si. Além disso, sendo uma linguagem dinâmica e distribuída, os pacotes gerados podem tornar o agente móvel, como comprovado no artigo de [WON99].

A técnica escolhida para a análise e projeto do sistema foi a orientação a objetos. Os diagramas usados na metodologia são da UML version 1.1 (*Unified Modeling Language*) [BOO97] (ver Apêndice A). O sistema foi documentado numa ferramenta CASE (especificamente, a versão demo do Rational Rose® 4.0 da Rational Software Corporation). A linguagem de programação escolhida foi o Java™ (JDK versão 1.1.6 da Sun Microsystems, Inc.) e a sua codificação obedece o documento Java™ Code Convention (<http://java.sun.com/docs/codeconv/index.html>).

5.2 Especificação do Software

5.2.1 Especificação do Modelo RC*

Da mesma forma que no sistema DEPINT (sistema que implementa o modelo CBD) [SIC95], cada objetivo (tarefa) é processado num período que chamaremos de **ciclo**. Dentro deste ciclo, um agente será **ativo**, ou seja, irá raciocinar sobre os outros e descobrir quais os

seus melhores possíveis parceiros. Os demais agentes serão **passivos**, ou seja, ficarão esperando propostas de coalisão para serem analisadas.

No sistema CENINT, o agente ativo (Ag1) é denominado **contratante**. Os agentes passivos (Ag2) podem ser de dois tipos:

- **possível contratado**, agentes aptos a dar um lance ao anúncio de uma ação;
- **contratado**, agente que foi escolhido para a coalisão com o contratante.

Todos os agentes têm uma descrição externa, contendo uma descrição sobre si próprios e sobre os outros agentes, a fim de que possam escolher um objetivo, um plano e ter meios para encontrar o parceiro adequado.

Durante um ciclo, inicialmente é escolhido o agente ativo na sociedade. Os outros agentes são considerados passivos. O agente ativo escolhe um determinado objetivo, baseado num critério próprio¹². Caso o agente não tenha objetivos o ciclo termina. Tendo escolhido um objetivo, o agente ativo escolhe então um plano. O critério de escolha do plano é baseado no valor total das ações associadas¹³. Caso o agente ativo não tenha planos para o objetivo corrente, ele procura por um outro objetivo.

Após escolher o plano, o agente ativo verifica se ele é autônomo, ou seja, se ele consegue realizar todas as ações do plano escolhido. Sendo autônomo o ciclo termina. Não sendo, inicia-se a fase de procura de um parceiro.

O agente ativo para encontrar seus prováveis parceiros anuncia a ação através de uma mensagem (*task announcement message*) emitida para todos os agentes passivos (*broadcast*). A estrutura da mensagem de anúncio é composta de:

- remetente;
- destinatário;

¹² O sistema permite modelar critérios de minimização/maximização de valor, ficando a critério do projetista escolher uma ou outra função para modelar o agente.

- período em que a mensagem terá validade;
- assunto da mensagem:
 - ✓ nome da ação anunciada;
 - ✓ nome de objetivo escolhido;
 - ✓ qual critério é utilizado para a escolha (ex: preço da ação);
 - ✓ valor mínimo/máximo da ação a ser aceita.

Os agentes que recebem a mensagem de anúncio emitem um lance (*bid*), desde que se enquadrem nos requisitos exigidos, ou seja, que o valor da ação seja igual ou menor/maior do que a da ação desejada, conforme a função de avaliação escolhida. Para emitir um lance, devem enviar ao contratante a mensagem com o seguinte conteúdo:

- remetente;
- destinatário;
- assunto da mensagem:
 - ✓ nome da ação (ação oferecida) ao qual o agente está dando um lance;
 - ✓ valor da ação oferecida.

À medida que os lances chegam, o contratante verifica se o lance estudado pertence a um parceiro ideal, ou seja, o valor da ação do candidato é exatamente igual ao valor do anúncio. Se encontrar tal candidato, a coalisão é formalizada através do envio de um contrato (*award message*) ao candidato, mesmo que o prazo de validade do anúncio não tenha terminado. Há muitas maneiras para se tratar os lances que chegam após o término do prazo do anúncio. Neste caso, o agente passivo ao perceber que o tempo expirou não enviará mais o seu lance.

Ao expirar o tempo, caso um parceiro ideal não tenha sido encontrado, realiza-se a análise dos lances enviados. Podem ocorrer as seguintes situações:

- havendo mais de um lance, o candidato que melhor se enquadra ao anúncio é escolhido, ou seja, aquele de menor/maior valor;

¹³ Novamente, o sistema permite modelar critérios de minimização/maximização de valor, ficando a critério

- não havendo lance algum, busca-se outro plano.

Ao encontrar um parceiro, uma mensagem (*award message*) é enviada, formando a coalisção entre o contratante e o contratado. O conteúdo desta mensagem é:

- contratante;
- contratado;
- nome da ação.

No caso do anúncio não receber nenhum lance adequado até o momento em que o tempo de validade expira e que, portanto, a escolha do parceiro não tenha sido realizada, procura-se por outro plano que contemple o objetivo corrente e um novo ciclo se inicia. Caso não exista outro plano, um novo objetivo é então escolhido e outro ciclo se inicia. Não havendo mais objetivos, o ciclo termina.

Como não é escopo do trabalho analisar a comunicação após estabelecida a coalisção, a rede contractual foi desenvolvida até este ponto.

Apesar do software oferecer várias opções de critério de escolha para os objetivos, planos e parceiros, o projetista deve escolher um deles para o desenvolvimento dos agentes que fazem parte do sistema CENINT. A escolha adotada no estudo de casos específicos é detalhada no capítulo 6.

5.2.2 Restrições, Requisitos e Premissas

Algumas limitações foram impostas no desenvolvimento do Sistema CENINT, de modo a simplificar o seu desenvolvimento. Tais aspectos que não fazem parte do escopo deste trabalho são as seguintes:

- Para eliminar problemas de concorrência entre os agentes, há somente um agente contratante (ativo) e os demais são os possíveis contratados (passivos).

do projetista escolher uma ou outra função para modelar o agente.

- Todos os lances recebidos após terminar o tempo de validade são descartados, porém para efeito de fluxo de comunicação são contados como mensagens enviadas.
- Afim de simplificar o controle dos lances e melhor avaliar o desempenho da comunicação entre anúncio, lances e contrato, não há mais de uma ação sendo anunciada por ciclo.
- O contrato nesta versão não foi implementado, pois o estudo se dá até o nível de coalisão entre os agentes. A comunicação após a coalisão não é avaliada.
- O protocolo não foi implementado na sua totalidade como proposto no modelo original pois pelos limites acima propostos alguns eventos não ocorrem.
- O ciclo não pode ser interrompido uma vez que se iniciou.
- O sistema estabelece uma única sociedade que é regida por um único modelo de organização dinâmica. Com esta sociedade é possível realizar vários ciclos, onde é permitido adicionar e retirar agentes.
- Nesta versão, é possível armazenar a sociedade estabelecida em meio físico, porém não existem ainda métodos para recuperar essas informações.

5.2.3 Funcionalidades do Sistema

O diagrama de casos de uso [JAC92] é uma técnica de modelagem utilizada para especificar todas as funcionalidades do sistema. Este diagrama forma a base dos processos de desenvolvimento e testes do sistema. O diagrama de casos de uso é composto por atores e casos de uso. Os atores são entidades externas que interagem com o sistema de alguma forma. O caso de uso especifica uma funcionalidade completa envolvendo os atores interessados. O diagrama de casos de uso inclui todos os atores e todos os casos de uso identificados para cada ator (mais detalhes encontram-se no Apêndice A). O diagrama de casos de uso do sistema encontra-se representado na Figura 5.1.

Figura 5.1 – Diagrama de casos de uso do sistema CENINT

A descrição de cada caso de uso está listada a seguir:

a. Controla o Ambiente

- O usuário cria a sociedade de agentes: escolhe os agentes que farão parte desta sociedade e o modelo organizacional desejado¹⁴.
- O usuário acrescenta agentes na sociedade.
- O usuário retira agentes da sociedade
- O usuário cria novos agentes.
- O usuário modifica a descrição externa dos agentes, ou seja, modifica o seu comportamento individual.

b. Inicia Simulação

- O usuário escolhe agente ativo, os demais agentes são considerados passivos.
- Prepara-se o ambiente para que a comunicação se estabeleça entre os agentes.
- Comunica-se ao agente ativo que ele deve começar a busca por um objetivo.
- Os demais agentes são comunicados que serão passivos neste ciclo e permanecem esperando uma proposta.

c. Escolhe Objetivo

- O agente ativo escolhe um objetivo a alcançar neste ciclo, segundo um critério próprio.
- Não possuindo objetivos, o agente termina o ciclo.
- Ao escolher um objetivo, passa a escolher um plano para realizá-lo.

d. Escolhe Plano

- O agente ativo verifica se possui planos para o objetivo escolhido.
- Não possuindo planos, o agente procura por um novo objetivo.
- Caso o agente tenha somente um plano para executar o objetivo, ele o analisa.

¹⁴ Atualmente, só encontra-se disponível o modelo RC*. Como explicada mais adiante, outras opções podem ser disponibilizadas no futuro.

- Tendo mais de um plano que atinja o objetivo corrente, o agente escolhe o melhor plano segundo um critério próprio baseado no valor total das ações de cada plano. Após escolher o plano, analisa-o.

e. Analisa Plano

- O agente ativo analisa o plano.
- O agente ativo descobre que é autônomo para este plano, ou seja, sabe realizar todas as ações necessárias. Neste caso, não há necessidade de buscar um parceiro e assim o ciclo termina.
- O agente ativo descobre que não é autônomo, ou seja, necessita da ajuda de um outro agente para realizar uma ação. Neste caso, inicia o processo de busca do parceiro enviando um anúncio aos agentes passivos.

f. Anuncia a Ação

- O agente ativo calcula o tempo de validade do anúncio (*time out*).
- O agente ativo elabora o anúncio.
- O agente ativo envia o anúncio (mensagem) pelo meio de comunicação.
- Tendo êxito ao enviar os anúncios, o agente ativo aguarda a chegada de lances.

g. Recebe/Envia Mensagens

- O meio de comunicação, tendo que enviar um *broadcast*, coloca a mensagem em cada caixa de correio dos agentes que pertencem à sociedade, com exceção da caixa de correio do remetente.
- O meio de comunicação, tendo que enviar uma mensagem a um agente específico, coloca a mensagem na caixa de correio do agente destinatário.
- O meio de comunicação avisa o(s) agente(s) que tem (têm) novas mensagens a serem lidas.
- O(s) agente(s) que tem (têm) mensagens novas lê(em) as suas mensagens.

h. Analisa Mensagem

- O agente verifica se expirou o tempo do anúncio.
- Caso o tempo tenha expirado, o agente ativo escolhe um parceiro, caso contrário prossegue com a análise da mensagem.
- Caso o tempo não tenha expirado, o agente verifica o tipo de mensagem.
- Caso o tipo de mensagem seja um anúncio, analisa o anúncio.
- Caso o tipo de mensagem seja um lance, analisa o lance.
- Caso o tipo de mensagem seja um contrato, termina o ciclo.

i. Analisa Anúncio

- O agente passivo lê a validade do anúncio.
- Se o agente passivo decide que pode executar a ação dentro das exigências do agente ativo, elabora e envia uma mensagem de lance, caso o tempo de validade da proposta ainda não tenha expirado.
- Tendo enviado a mensagem com êxito, o agente fica esperando por uma mensagem de contrato.
- Se o agente passivo decide que não está apto para executar a ação anunciada, fica esperando por outras propostas.

j. Analisa Lances

- Se o agente ativo descobre que encontrou um par perfeito, ou seja, a ação oferecida tem o mesmo valor que a ação necessária, elabora e envia uma mensagem de contrato e fica esperando que o parceiro receba o contrato.
- Se o agente ativo descobre que não encontrou um par perfeito, fica então esperando por novos lances.

k. Escolhe parceiro

- Se o tempo expirou e não foi encontrado um parceiro ideal, o agente ativo analisa os lances armazenados.

- Se o agente ativo recebeu lances, escolhe aquele que dentre eles tem o valor mais adequado, elabora e envia uma mensagem de contrato.
- Tendo enviado um contrato, o agente ativo fica esperando que o parceiro receba o contrato.
- Caso o agente ativo não receba nenhum lance, conclui que ninguém pode ajudá-lo e assim escolhe um outro plano.

5.3 Arquitetura do Software

Pelo capítulo 2, podemos concluir que os componentes de um sistema multiagentes são a sociedade, organização, interação e os agentes. Estes componentes por sua vez podem ser sintetizados nos seguintes pacotes¹⁵ :

- Agente: contém a estrutura do agente genérico e do agente que comporta o modelo RC*.
- Ambiente: gerencia e controla a sociedade de acordo com o modelo de organização especificado. Por enquanto, comporta somente o modelo RC*.
- Comunicação: gerencia toda a comunicação que existe entre os agentes da sociedade.

Esses pacotes constituem a arquitetura do sistema que está representada pelo diagrama de pacotes (Figura 5.2). O diagrama de pacotes representa os módulos (pacotes) e os seus relacionamentos que pertencem ao sistema.

O diagrama de classes representa o modelo estático do sistema, ou seja, descreve a visão estática do sistema em termos de classes e seus relacionamentos (herança, agregação e associação). Uma classe no diagrama de classe pode ser implementada diretamente em uma linguagem de programação orientada a objetos (outros detalhes ver Apêndice A). O diagrama de classe do sistema completo encontra-se na figura 5.3, que exhibe o conteúdo

¹⁵ É um mecanismo geral para organizar elementos em grupos semânticamente relacionados (ver apêndice A).

dos pacotes. Cada classe e o pacote ao qual pertence a classe são descritos e apresentados nos próximos itens.

5.3.1 Pacote Ambiente

O pacote Ambiente é responsável pela entrada e saídas dos agentes na sociedade e pelo escalonamento das tarefas que os agentes realizam durante uma coalisção. O diagrama de classe deste pacote encontra-se na figura 5.4. A descrição sumária destas classes encontra-se na tabela 5.1 (para uma documentação completa ver [ITO99]).

Nome da classe	Descrição sumária
Society	É responsável por formar a comunidade de agentes, desde o aliciamento e remoção dos agentes até o controle da organização e o estabelecimento da comunicação.
Organization	É responsável pelo escalonamento das tarefas, organizando os agentes que estão trabalhando em conjunto. Esta classe tem a função principal de manter em ordem todas as tarefas realizadas pelos agentes, estando envolvida com a sincronização destas atividades. Em resumo, trata-se de um monitor dos agentes.
SocietyRandom	É um tipo específico de sociedade, onde o usuário apenas escolhe a quantidade agentes que farão parte da sociedade e a própria sociedade escolhe os agentes que farão parte da sociedade. É uma subclasse da classe Society.
OrganizationRandom	É um tipo específico de organização, onde a própria organização define os papéis que cada agente irá desempenhar a cada coalisção, ou seja, dentre os agentes que fazem parte da sociedade designa o agente ativo, sendo os demais passivos naquele ciclo. É uma subclasse da classe Organization.

Tabela 5.1 – Descrição sumária do pacote Ambiente.

Figura 5.2 – Diagrama de pacote do sistema CENINT

Figura 5.3 – Diagrama de classe do sistema CENINT

A decisão de projetar uma classe para representar o conjunto de agentes e outra para representar uma coalisão surgiu da idéia de que no futuro poderíamos ter numa mesma sociedade várias organizações de agentes coexistentes. Deste modo, tal decisão facilita uma implementação futura mais completa do modelo RC, onde poderíamos ter subcontratações, ou até mesmo organizações com modelos diferentes numa mesma sociedade.

Por outro lado, como gostaríamos de ter a oportunidade de ter simulações ocorrendo de modo aleatório, para que uma análise do fluxo de comunicação pudesse ser genérica optou-se por acrescentar funcionalidades que permitissem tal atividade. Para implementar tais atividades, projetaram-se duas novas classes: `SocietyRandom` e `OrganizationRandom`. Como tais classes apenas acrescentam funcionalidades que permitem uma escolha aleatória tanto dos agentes que compõem a sociedade como daqueles que serão ativos em cada ciclo, estas foram projetadas respectivamente como subclasses das classes `Society` e `Organization`.

5.3.2 Pacote Agente

5.3.2.1 Modelo de Agente

Os agentes no sistema CENINT são baseados no modelo ASIC [BOI93] e têm as seguintes capacidades [SIC92] [CAR93]:

- comunicam-se com os outros agentes, incluindo a troca de informações, a troca de partes do conhecimento e planos;
- raciocinam sobre as atividades dos outros agentes;
- decidem quando comunicar ou planejar.

5.3.2.2 Descrição Externa

Neste trabalho, a estrutura interna que informa aos agentes as suas próprias características se compõe de:

- objetivos: estados do mundo que o agente quer atingir. Um mesmo agente pode ter mais de um objetivo, sendo ele ativo ou não naquele ciclo;
- ações: as ações que o agente pode realizar e o seu valor;
- recursos: os recursos sobre os quais o agente tem controle e o seu valor;
- planos: os planos que um agente possui para atingir um objetivo. Um agente pode ter mais de um plano para atingir um mesmo objetivo.

5.3.2.3 Estrutura do Pacote Agente

Pelo capítulo 2, podemos concluir que um agente deve ter a capacidade de:

- conhecer-se a si mesmo e saber manipular este auto-conhecimento;
- conseguir se comunicar com outros agentes, ou seja, enviar e receber mensagens;
- raciocinar sobre si mesmo e sobre os outros, de acordo com o papel que desempenha na sociedade.

O diagrama de classe do pacote Agente com as características acima expostas e que contemplam o modelo RC* estão representadas na Figura 5.5. A descrição sumária destas classes encontra-se na tabela 5.2 (para uma documentação completa ver [ITO99]).

Figura 5.4 - Diagrama de classe do pacote Ambiente

Nome da classe	Descrição sumária
AgentStream	Representa o auto-conhecimento (descrição externa) do agente e permite que o agente possa guardar, modificar e resgatar esse conhecimento.
AgentThread	Classe controle do agente, é através dela que o agente consegue se comunicar com os outro agentes (envio e recebimento de mensagens).
CnetAgent	O agente deve ter um determinado tipo de comportamento dependendo do modelo de organização e seu papel na sociedade. Esta classe que deriva da classe AgentThread implementa o comportamento nos possíveis papéis a serem desempenhados pelo agente no modelo RC*.
ContractNetModel	A lógica de raciocínio do agente depende do modelo de organização que rege a sociedade e estas regras encontram-se nesta classe.
Goal	Classe que possui todas as informações de um objetivo do agente, assim como todos os métodos associados ao objetivo. (Por exemplo: fornecer todos os planos que pertencem ao objetivo).
Plan	Classe que possui todas as informações de um plano de um determinado objetivo, assim como todos os métodos relacionados com o plano. (Por exemplo: Calcular o valor total do plano)
Action	Classe que possui todas as informações de uma ação a ser realizada por um agente, assim como todos os métodos de acesso associados. Esta ação pode eventualmente fazer parte de um plano.

Tabela 5.2 – Descrição das classes do pacote Agente

Como previamente descritos no item 5.2.1, os estados, e por consequência o comportamento, que o agente pode ter durante a sua permanência na sociedade são: ativo ou passivo.

Figura 5.5 - Diagrama de classe do pacote Agente

O diagrama de estados (outros detalhes ver apêndice A) é uma das representações do modelo dinâmico do sistema. O diagrama de estados é uma representação gráfica dos diferentes estados que uma classe pode assumir, incluindo as condições para mudança de estado e as ações realizadas na transição.

Assim, o diagrama de estados do agente (Figura 5.6) apresenta os estados que o agente pode ter enquanto permanecer na sociedade. Ao se estabelecer uma organização, são definidos os papéis que cada agente desempenhará. Cada papel possui um comportamento que se encontra descrito no diagrama de estados do agente.

Pelo modo que o agente foi modelado, percebe-se que através de substituições de suas classes podemos ter outros tipos de agentes. Por exemplo, pode-se modificar o modelo de organização e substituir a classe *ContractNetModel* pela classe que representa a nova organização. O agente que contém o comportamento desta outra organização será uma subclasse da classe *AgentThread*. Esse agente inclusive poderá ter outros papéis além de ativo e passivo.

5.3.3 Pacote Comunicação

5.3.3.1 Especificação do Protocolo de Formação de Coalisão

A linguagem de interação utilizada entre os agentes neste sistema é uma adaptação da que foi proposta em [DEM95]. Esta linguagem define o vocabulário e a sintaxe comum para todos os agentes do sistema. A notação utilizada para definir a linguagem e o protocolo de interação encontra-se na tabela 5.3.

Símbolo	Descrição
< >	elemento não terminal, é composto por outros elementos
[]	elemento terminal
{ }	{ conteúdo do elemento }

Tabela 5.3 – Notação utilizada no protocolo de interação

Figura 5.6 - Diagrama de estados do agente

A sintaxe da linguagem é:

$\langle \text{interação} \rangle ::= \langle \text{comunicação} \rangle \langle \text{multiagentes} \rangle \langle \text{aplicação} \rangle$ onde,

$\langle \text{comunicação} \rangle \rightarrow$ define os aspectos da implementação do sistema distribuído em questão. É composto por: identificação do remetente (*origem*) e identificação do destinatário (*destino*):

$\langle \text{comunicação} \rangle ::= \langle \text{origem} \rangle \langle \text{destino} \rangle$

$\langle \text{multiagentes} \rangle \rightarrow$ define a dimensão multiagentes do sistema. Compõe-se dos seguintes campos:

- * Os **Tipos de Mensagens** (*tipo*), que podem transitar entre os agentes. As primitivas utilizadas são as propostas em [GAS91]: solicitação (*request*), resposta (*answer*) ou informação (*inform*).
- * A **Natureza da Mensagem** (*natureza*) [BOI93], especifica o tipo de informação que foi enviada. Seus custos possíveis são: anúncio (*announcement*), lance (*bid*) ou contrato (*award*).

$\langle \text{multiagentes} \rangle ::= [\text{tipo}] [\text{natureza}]$

$\langle \text{aplicação} \rangle \rightarrow$ contém os termos do domínio da aplicação e deve ser customizada para cada aplicação. É composta basicamente dos termos propostos no domínio.

Uma vez definida a linguagem, é necessário estabelecer a maneira pela qual os agentes interagem durante a execução do sistema. Para representar o protocolo de interação foi adotada uma adaptação do modelo proposto no protocolo Contract Net [SMI80]. Ressalta-se que a especificação do protocolo de execução e de seu gerenciamento não fazem parte

do escopo deste projeto. O protocolo de formação de coalisão é formado por anúncio, lance e contrato. Um interação típica do sistema encontra-se esquematizada na Figura 5.7.

O conteúdo de cada mensagem é:

1. Anúncio:

<destino>:= {“Broadcast”}¹⁶
<origem>:= [nome do agente contratante] [host do agente contratante]
[tipo] = {“request”}
[natureza] = {“annoucement”}
<aplicação>:= [objetivo a ser alcançado] [ação anunciada] [especificação de
eligibilidade¹⁷] [especificação do lance¹⁸]

No caso particular deste trabalho temos que:

[especificação de eligibilidade] = [custo do preço mínimo para executar a ação]
[especificação do lance] = {“preço da ação”}

2. Lance:

<destino>:= [nome do agente contratante] [host do agente contratante]
<origem>:= [nome do agente candidato] [host do agente candidato]
[tipo] = {“answer”}
[natureza] = {“bid”}
<aplicação>:= [nome da ação oferecida] [abstração do nó¹⁹]

¹⁶ Com este parâmetro, sabe-se que a mensagem deve ser enviada a todos os agentes da sociedade.

¹⁷ Especificações necessárias para que o lance seja válido.

¹⁸ Especificação necessária para análise do lance.

¹⁹ Informações do agente candidato em relação a especificações exigidas pelo contratante.

No caso particular deste trabalho temos:

[abstração do nó] = [preço da ação do agente candidato]

3. Contrato:

<destino>:= [nome do agente contratado] [host do agente contratado]

<origem>:= [nome do agente contratante] [host do agente contratante]

[tipo] = {"inform"}

[natureza] = {"award"}

<aplicação>:= [ação contratada]

Figura 5.7 – Esquema de interação do protocolo de comunicação no sistema CENINT.

5.3.3.2 Estrutura do Pacote Comunicação

A fim de que os agentes não necessitem saber da existência uns dos outros, foi desenvolvido um servidor de comunicação que atua como se tivéssemos um correio responsável pela entrega da correspondência entre os agentes. O diagrama de classe do pacote Comunicação encontra-se na Figura 5.8. A descrição sumária destas classes encontra-se na tabela 5.4 (para uma documentação completa, ver [ITO99]).

Nome da classe	Descrição sumária
CommunicationServer	É o servidor de comunicação entre os agentes, responsável por receber e entregar as mensagens aos agentes destinatários.
CommunicationThread	Classe auxiliar de CommunicationServer. Evita manipulação de filas para a entrega de mensagens. Cada objeto é encarregado de entregar uma mensagem aos agentes destinatários.
CommunicationLog	É onde se encontra o histórico de todas as atividades do CommunicationServer. É responsável pela estatística das mensagens trocadas pelos agentes.
Mbox	Armazena e controla as mensagens recebidas pelo agente ao qual ele pertence.
Message	É a mensagem propriamente dita, é composta pelo conteúdo já mencionado no item 5.3, e pela especificação do protocolo de comunicação.

Tabela 5.4 – Descrição das classes do pacote Comunicação.

Figura 5.8 - Diagrama de classe do pacote Comunicação

5.4 Implementação do Software

O ambiente utilizado para o desenvolvimento deste sistema foi o JDK 1.1.6 (Linux) e o JDK 1.1.2 (MS Windows 95).

Para a implementação do sistema, utilizaram-se classes das bibliotecas nativas do Java, além de algumas por nós implementadas. Os pacotes utilizados foram (tabela 5.5): `java.lang.*`, `java.io.*`, `java.util.*`, `java.net.*`, `java.awt.*` (detalhes sobre os pacotes acessar documentação *on line* em: <http://java.sun.com>). A interação desses pacotes com a arquitetura do sistema encontra-se no diagrama de pacotes da fase de projeto (Figura 5.9).

Pacote	Descrição sumária
<code>java.lang.*</code>	Contém as classes que definem somente o comportamento essencial das classes na linguagem Java.
<code>java.io.*</code>	Contém classes que definem métodos para leitura e armazenamento de dados de diversas formas.
<code>java.util.*</code>	Contém estruturas de dados relacionados com os programas de aplicações, como por exemplo, manipulação de data e hora.
<code>java.net.*</code>	Contém um conjunto de classes para a comunicação através da rede.
<code>java.awt.*</code>	É o <i>Abstract Windowing Toolkit</i> (awt), um conjunto de ferramentas para criar interfaces gráfica com o usuário (GUI).

Tabela 5.5 – Descrição dos pacotes da linguagem Java utilizados no sistema CENINT

Figura 5.9 - Diagrama de pacote da fase de projeto

As seguintes classes auxiliares foram necessárias na implementação de multivaloração entre as classes, armazenamento dos dados da simulação, troca de mensagens e sincronização entre os agentes. Uma descrição sumária dessas classes encontra-se na tabela 5.5 (para uma documentação completa ver [ITO99]). Outras classes da biblioteca nativa do Java foram utilizadas mas não afetam diretamente o projeto.

5.4.1 Escalonamento dos Processos no Sistema

Sistemas distribuídos são compostos por processos autônomos interligados através de uma rede de comunicação, trabalhando em determinado grau de independência mútua. Em geral, um planejamento de acesso a recursos compartilhados (arquivos, dispositivos e segmentos de memória) faz-se necessário, de modo que o acesso a tais recursos não ocorra simultaneamente (para escrita) e também não haja invalidação de recursos compartilhados (para leitura) [COU94].

A sincronização é o processo de coordenação entre processos (*threads*) que estão sendo executados de forma concorrente, de tal forma que um não interfira na execução do outro, ou seja, que eles não estejam concorrentemente tentando modificar ou acessar recursos compartilhados.

Na UML, a visão²⁰ concorrente é aquela que demonstra a divisão do sistema em processadores e processos, trabalhando com a concorrência entre estes últimos. Tal visão é melhor representada pelo diagrama de sequência. Este diagrama demonstra como os objetos interagem numa determinada situação. Demonstra ainda a colaboração dinâmica entre os objetos. A sequência das mensagens²¹ é o ponto forte deste diagrama.

²⁰ A UML é composta de modelos e visões. Visões mostram os diferentes aspectos do sistema que será modelado. A visão não é um gráfico, mas uma abstração composta por vários diagramas.

²¹ Na orientação a objetos, enviar uma mensagem a um objeto significa ativar a execução de uma de suas operações.

Classe	Descrição sumária
Vector ²²	Implementa uma matriz crescente de objetos. Como uma matriz, contém o componente que pode acessado utilizando um indexador inteiro. Entretanto o tamanho do Vector pode aumentar ou diminuir quando necessário, acrescentando-se ou retirando-se elementos depois do vetor ter sido criado.
Thread ²³	Implementa as principais funcionalidades de um <i>thread</i> .
SocketServer ²⁴	Implementa um servidor de <i>sockets</i> . Um servidor de <i>sockets</i> espera por um pedido que venha da rede. Realiza algumas operações básicas nesta requisição, e então permite o retorno de uma resposta ao requisitante.
Socket ²⁵	Esta classe implementa o socket cliente, também chamado simplesmente de <i>socket</i> . O <i>socket</i> é o ponto final da comunicação entre duas máquinas.
DataSimulation	Contém as informações geradas durante toda a simulação.
DataCycle	Contém as informações geradas durante um ciclo.
Barrier	Implementa a barreira utilizada pela classe Organization, a fim de monitorar, sincronizar e escalonar as tarefas dos agentes dentro da organização.

Tabela 5.6 – Descrição das classes auxiliares do sistema CENINT

O principal problema de sincronização do sistema CENINT ocorre devido ao fato de ter sido este originalmente modelado para ser executado em um ambiente verdadeiramente distribuído. Devido a limitações de prazo e recursos, tal modelo não pôde ser implementado completamente, limitando-se a implementação atual a um único *host*. Obviamente, isto trouxe problemas de acomodação do número necessário de processos sendo executados simultaneamente. A solução utilizada para contornar os problemas de

²² Classe pertencente ao pacote do Java (java.util.*), outros detalhes em <http://java.sun.com>.

²³ Classe pertencente ao pacote do Java (java.lang.*), outros detalhes em <http://java.sun.com>.

²⁴ Classe pertencente ao pacote do Java (java.net.*), outros detalhes em <http://java.sun.com>.

²⁵ Classe pertencente ao pacote do Java (java.net.*), outros detalhes em <http://java.sun.com>.

acesso a recursos compartilhados e *deadlocks* foi a implementação de duas classes: Barrier e Organization, descritas respectivamente nas tabelas 5.6 e 5.3. Assim, quando ocorrer a migração deste sistema para um ambiente realmente distribuído, estas classes deverão ser reimplementadas.

No Sistema CENINT, existem três situações no qual observamos necessidade de sincronização, ou seja, as ações precisam acontecer numa determinada ordem a fim de que a simulação ocorra de modo adequado. São elas:

- comunicação entre agentes (envio e recebimento de mensagens);
- escolha de um parceiro pelo agente ativo;
- término do ciclo (especificamente na simulação de forma randômica ou na que envolve mais de um ciclo por simulação).

A seguir, temos a exposição de cada um destes problemas e a sua resolução.

5.4.1.1 Comunicação entre agentes

Para que os agentes possam receber/enviar mensagens entre si é necessário que todos eles tenham um tempo de execução garantida. A fim de que isto ocorra, implementou-se um sistema de barreiras que cuida da sincronização entre o recebimento e o envio de mensagens. Quando há o envio de mensagens, a barreira é avisada e inicia-se o processo de sincronização. Num primeiro momento, a barreira verifica se não há algum agente em processamento. Caso haja, espera até que todos fiquem em estado de espera. Assim que todos estiverem em estado de espera, a barreira permite então que os agentes que desejam enviar mensagens o façam um de cada vez. Após o envio de todas as mensagens, a barreira coordena o procedimento de leitura, ou seja, cada agente tem a sua vez para ler e processar a mensagem que lhe foi enviada. Ao final deste processo, a barreira verifica se algum agente deseja enviar uma nova mensagem. Em caso positivo, o processo recomeça, caso contrário todos ficam em estado de espera até que algum evento ocorra para acordar a todos

ou a algum agente em particular. O diagrama de sequência da comunicação entre os agentes encontra-se representado na Figura 5.10.

5.4.1.2 Escolha de um parceiro

Há dois momentos em que o agente ativo pode optar por escolher o seu parceiro:

- ao receber um lance, quando deve verificar se este lance é de um par perfeito;
- ao expirar o tempo do anúncio, quando deve escolher um parceiro dentre os lances recebidos.

No primeiro caso, o agente ativo deve parar de processar o restante dos lances e enviar uma mensagem de contrato. Deve ainda ignorar os lances recebidos, preparar o contrato e avisar a barreira que deseja enviar uma mensagem, quando ocorre o processo de sincronização descrito no item anterior. O diagrama de sequência encontra-se representado na Figura 5.11.

No segundo caso, os agentes podem estar ainda em processamento quando o tempo do anúncio expira. Optamos por sincronizar tal evento através da classe Organization. Ao expirar o tempo, os seguintes eventos podem estar em curso:

- todos os agentes se encontram em estado de espera, assim o objeto Organization somente acorda-os para que o processamento prossiga;
- há agentes processando, assim o objeto Organization aguarda até que todos fiquem em estado de espera para depois acordá-los e prosseguir com o processamento.

O diagrama de sequência que evidencia estes fatos encontra-se representado na Figura 5.12.

Figura 5.10 - Diagrama de sequência da comunicação entre agentes.

5.4.1.3 Término de ciclo

A decisão de que o ciclo terminou pode se dar tanto pelo agente ativo como pelo agente passivo. O agente ativo termina o ciclo quando é autônomo ou não possui objetivos. Por sua vez, o agente passivo termina o ciclo quando é contratado, pois é ele quem avisa que recebeu a mensagem de contrato e portanto o ciclo pode ser terminado.

O agente que termina o ciclo envia uma mensagem de final de ciclo²⁶ aos demais agentes. Cada agente termina as suas tarefas e fica então em um estado de espera por um próximo papel a lhe ser atribuído. O diagrama de sequência para término de ciclo encontra-se representado na Figura 5.13.

²⁶ Tal mensagem é consequência exclusiva do modelo de implementação adotado, não sendo portanto contabilizada no fluxo de comunicação global que está sendo estudado.

Figura 5.11 - Diagrama de sequência ao encontrar um par perfeito.

Figura 5.12 - Diagrama de sequência ao expirar o tempo do anúncio.

Figura 5.13 - Diagrama de sequência ao terminar o ciclo.

6. Estudo de Casos Específicos

Este capítulo apresenta o estudo de casos específicos, com as respectivas análises dos resultados obtidos. Na seção 6.1, descrevem-se os materiais e métodos utilizados para o estudo. Um breve relato dos experimentos é apresentado na seção 6.2. Os resultados dos experimentos estão tabulados na seção 6.3. Uma análise prévia dos resultados obtidos nos experimentos é descrita então na seção 6.4.

6.1 Materiais e Métodos

A sociedade de agentes que é simulada no estudo de casos específicos é a mesma dos exemplos apresentados na seção 4.2. O ambiente é um hospital, mais especificamente um ambulatório de cardíacos e hipertensos, e os agentes são profissionais da saúde que se ajudam mutuamente²⁷.

Os agentes podem ser médicos (clínico geral, endocrinologista, cardiologista, nefrologista, ortopedista e oftalmologista), um professor de educação física, um técnico de laboratório, um nutricionista e um fisioterapeuta. Todos pretendem conseguir realizar a prevenção de um infarto do miocárdio no paciente. Estes pacientes podem ter diagnósticos prévios (objetivos) fazendo com que as condutas (planos) sejam diferentes nos vários casos. A conduta é um conjunto de procedimentos (ações) que devem ser realizadas a fim de impedir, neste caso, o infarto do miocárdio no paciente.

Por exemplo, seja o caso de um clínico geral (*clinico1*) que tendo um paciente obeso e hipertenso, pretende realizar procedimentos preventivos para o infarto do miocárdio. Assim

²⁷ Devido à complexidade dos diagnósticos e seus respectivos tratamentos simplificamos o nosso escopo tornando algumas decisões não tão exatas como verdadeiras. Desta forma, este experimento não pode ser considerado uma realidade no meio médico.

este diagnóstico passa a ser um objetivo a ser alcançado (*obeso_hipertenso*) e os planos a ele associados visam prevenir um infarto do miocárdio. O plano que o clínico escolhe se constitui de orientar uma dieta e solicitar uma interconsulta com um especialista em hipertensão (*obeso_hipertenso:=orientacao_dietetica;interconsulta_hipertensao*). Este plano contém duas ações (*orientacao_dietetica* e *interconsulta_hipertensao*), uma das quais (*interconsulta_hipertensao*) o clínico geral não é capaz de executar e que, portanto, o levará a tentar formar uma coalisão. Supondo que na sociedade estudada esteja presente um nefrologista (*nefro1*), este provavelmente irá enviar um lance para executar esta ação.

O valor da ação é o nível de conhecimento do médico sobre o assunto (ação) requisitada. Assim tem-se que quanto *mais* o agente conhece o assunto, *menor* é o valor da ação. O agente tem conhecimento pleno do assunto (é um especialista) quando o valor da ação é 0 (zero). Desta forma o critério de escolha adotado é:

- objetivo: o objetivo de maior valor (é o mais importante para o agente);
- plano: o plano de menor valor (o agente deseja que todos conheçam muito bem sobre o assunto);
- parceiro: o valor da ação oferecida deve ser a menor, ou seja, conhece mais sobre o assunto dentre os concorrentes a parceiro.

Para o estudo de casos foi utilizado um domínio constituído por 46 agentes. Vários arquivos texto²⁸ contendo as descrições externas dos agentes que participaram das simulações foram gerados pelo sistema (um exemplo encontra-se na figura 6.1). Em relação aos seus objetivos, os agentes podem ser autônomos ou podem necessitar de um único parceiro²⁹.

²⁸ Para não tornar as figuras repetitivas, optamos por colocar os objetivos à esquerda do seu plano. Objetivos sem planos aparecem somente com o seu nome e valor no início da linha.

²⁹ Para simplificar o controle dos agentes, não há mais de uma ação sendo anunciada por ciclo (ver item 5.2.2).

```
Name: Cardio3
Host: localhost
Action(s): nao_faz_nada(1)
           orientacao_exercicio(8)
           tto_medicamentoso(1)
           interconsulta_hipertensao(5)
           interconsulta_cardiologia(1)
           orientacao_dietetica(8)
Goal(s)/Plan(s):
hipertenso_sem_causa_melhorou(20):=nao_faz_nada(0);
```

Figura 6.1 – Descrição externa do agente cardio3 gerada pelo sistema CENINT.

6.2 Experimentos Realizados

Vários experimentos foram realizados com uma sociedade composta por 10 agentes. Cada simulação é composta de 10 ciclos, onde a cada ciclo um agente é escolhido para ser o ativo. A escolha dos agentes que fazem parte da sociedade e o agente ativo a cada ciclo ocorrem de forma aleatória. Foram realizadas 10 simulações que foram documentadas pelo sistema, uma vez que são gerados arquivos com o resultado de cada ciclo (Figura 6.2) e do fluxo de comunicação global da simulação (Figura 6.3).

6.3 Resultados Obtidos

Dentre os experimentos realizados identificamos situações que merecem uma análise especial:

- simulações onde em alguns de seus ciclos ocorria ausência de parceiros, ou seja, não houve lances;
- simulações onde em alguns de seus ciclos existiam agentes autônomos;
- simulações em que houve formação de coalisões.

```
Cycle number: 5
Simulation: soc10g5
Society: soc10g
Organization model: ContractNet
Active: Clinico7
Passive(s):
Clinico6
Ortop7
Nefro7
Ortop2
Cardio3
Cardio4
endocrino2
Cardio6
Ortop8
Choosed Goal: hipertenso_assintomatico_hipercolerestemia
Choosed Plan: orientacao_dietetica; orientacao_exercicio;
tto_medicamentoso;
Action needed: orientacao_exercicio
Partner: Ortop8
Total of Announce: 9
Total of Bid: 9
Total of Award: 1
Total of messages: 19
```

Figura 6.2 – Resultado de um ciclo das simulações realizadas com o Sistema CENINT.

Deste modo, dentre os resultados obtidos escolheram-se algumas simulações significativas que representavam as situações acima expostas para serem apresentadas nas próximas seções. Por outro lado, para uma melhor análise, simulamos manualmente os mesmos casos para o modelo CBD. No modelo CBD, supomos que os agentes não modificam sua descrição externa entre os ciclos. Desta forma, o provável parceiro recusa a parceria quando o agente ativo não faz parte das suas opções de parceria [SIC95]. Os diversos resultados são apresentados a seguir.

```

Society: soc10g
Organization model: ContractNet
Society members:
Clinico6
Ortop7
Nefro7
Ortop2
Cardio3
Cardio4
endocrino2
Cardio6
Clinico7
Ortop8
cycle|msg in the cicle|msg accumulate|active
0 |9|9|Ortop2
1 |0|9|Cardio3
2 |0|9|Nefro7
3 |9|18|Ortop8
4 |0|18|Cardio3
5 |19|37|Clinico7
6 |0|37|Cardio3
7 |19|56|Clinico7
8 |0|56|Cardio3
9 |11|67|Cardio4

```

Figura 6.3 – Resultado final de uma das simulações realizadas com o Sistema CENINT.

6.3.1 Resultados Obtidos na Ausência de Parceiros

A principal característica das sociedades retratadas nestas simulações é que os seus integrantes pouco se ajudam, ou seja, um agente ativo dificilmente encontra um parceiro quando necessita de uma certa ação.

Simulação 1:

Os integrantes da sociedade nesta simulação são: cardio6 (cardiologista), cardio7 (cardiologista), cardio9 (cardiologista), clinico4 (clínico geral), clinico5 (clínico geral), clinico6 (clínico geral), nefro4 (nefrologista), nutri1 (nutricionista), oftalmo1 (oftalmologista) e ortop10 (ortopedista). A descrição externa de cada agente da sociedade é apresentada na figura 6.4. Os resultados para o modelo CBD e RC* encontram-se respectivamente nas tabelas 6.1 e 6.2.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	0	90	cardio9	-
2	0	90	cardio9	-
3	2	92	nefro4	-
4	6	98	nutri1	-
5	2	100	oftalmo1	-
6	2	102	oftalmo1	-
7	0	102	cardio9	-
8	6	108	nutri1	-
9	6	114	cardio7	-
10	16	130	clinico6	-

Tabela 6.1 – Resultado obtido na simulação 1 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	9	9	cardio9	-
2	9	18	cardio9	-
3	11	29	nefro4	oftalmo1
4	13	42	nutri1	cardio9
5	11	53	oftalmo1	clinico4
6	11	64	oftalmo1	clinico4
7	9	73	cardio9	-
8	13	86	nutri1	cardio9
9	13	99	cardio7	nefro4
10	18	117	clinico6	nutri1

Tabela 6.2 – Resultado obtido na simulação 1 do modelo RC* (sistema CENINT)

```

Name: Cardio6
Host: localhost
Action(s): nao_faz_nada(1)
orientacao_exercicio(8)
tto_medicamentoso(1)
interconsulta_hipertensao(5)
interconsulta_cardiologia(1)
orientacao_dietetica(8)
Goal(s)/Plan(s):
hipertenso_causa_definida(40):=interconsulta_nefrologista(0);

```

Name: Cardio7 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(30) tto_medicamentoso(1) interconsulta_cardiologia(1) orientacao_dietetica(30) Goal(s)/Plan(s): obeso_hipertenso(40):=interconsulta_hipertensao(0);
Name: Cardio9 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(70) tto_medicamentoso(1) interconsulta_hipertensao(35) interconsulta_cardiologia(1) orientacao_dietetica(8) Goal(s)/Plan(s): assintomatico_obeso2_sem_comorbidade(40):=orientacao_dietetica(0); interconsulta_endocrinologia(0);orientacao_exercicio(0);
Name: Clinico4 Host: localhost Action(s): verificar_sintomas(1) verificar_peso(1) orientacao_dietetica(10) Goal(s)/Plan(s): obeso_hipertensao(20):=interconsulta_hipertensao(0);orientacao_dietetica(0);
Name: Clinico5 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso_causa_definida(100):=interconsulta_nefrologista(0);
Name: Clinico6 Host: localhost Action(s): orientacao_exercicio(40) tto_medicamentoso(1) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestemia(100):=orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);

Name: Nefro4 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso(50):=fundo_de_olho(0);orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);
Name: Nutril Host: localhost Action(s): orientacao_dietetica(1) interconsulta_nutricionista(1) Goal(s)/Plan(s): preparar_cardapio(100):=orientacao_dietetica(0);nao_faz_nada(0);
Name: Oftalmol Host: localhost Action(s): interconsulta_oftamologista(1) fundo_de_olho(1) Goal(s)/Plan(s): consulta(100):=fundo_de_olho(0);verificar_sintomas(0);
Name: Ortop10 Host: localhost Action(s): orientacao_exercicio(5) orientacao_dietetica(20) interconsulta_ortopedista(1) Goal(s)/Plan(s): assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedista(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);

Figura 6.4 – Descrição externa dos agentes da simulação 1

Simulação 2:

Os integrantes da sociedade nesta simulação são: endocrino8 (endocrinologista), fisio1 (fisioterapeuta), clinico3 (clínico geral), clinico4 (clínico geral), clinico5 (clínico geral), cardio3 (cardiologista), clinico2 (clínico geral), endocrino5 (endocrinologista), clinico9 (clínico geral) e ortop10 (ortopedista). A descrição externa de cada agente da sociedade é apresentada na figura 6.5. Os resultados para o modelo CBD e RC* encontram-se nas tabelas 6.3 e 6.4, respectivamente.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	0	90	cardio3	-
2	0	90	clinico3	-
3	6	96	clinico4	-
4	0	96	clinico2	-
5	0	96	clinico2	-
6	0	96	clinico3	-
7	0	96	clinico2	-
8	3	99	fisio1	ortop10
9	3	102	ortop10	fisio1
10	6	108	clinico4	-

Tabela 6.3 – Resultado obtido na simulação 2 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	0	0	cardio3	-
2	9	9	clinico3	-
3	13	22	clinico4	cardio3
4	9	31	clinico2	-
5	9	40	clinico2	-
6	9	49	clinico3	-
7	9	58	clinico2	-
8	11	69	fisio1	ortop10
9	11	80	ortop10	fisio1
10	13	93	clinico4	cardio3

Tabela 6.4 – Resultado obtido na simulação 2 do modelo RC* (sistema CENINT)

```

Name: endocrino8
Host: localhost
Action(s): interconsulta_endocrinologia(5)
           orientacao_exercicio(10)
           interconsulta_hipertensao(10)
           orientacao_dietetica(10)
Goal(s)/Plan(s):
obeso_hipertensao(50):=interconsulta_hipertensao(0);

```

Name: Fisio1 Host: localhost Action(s): orientacao_exercicio(1) interconsulta_fisioterapeuta(1) Goal(s)/Plan(s): fisioterapia(100):=orientacao_exercicio(0);interconsulta_ortopedista(0);
Name: Clinico3 Host: localhost Action(s): tto_medicamentoso(1) orientacao_dietetica(10) Goal(s)/Plan(s): assintomatico_com_sobrepeso(10):=interconsulta_nutricionista(0);
Name: Clinico4 Host: localhost Action(s): verificar_sintomas(1) verificar_peso(1) orientacao_dietetica(10) Goal(s)/Plan(s): obeso_hipertensao(20):=interconsulta_hipertensao(0);orientacao_dietetica(0);
Name: Clinico5 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso_causa_definida(100):=interconsulta_nefrologista(0);
Name: Cardio3 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) interconsulta_cardiologia(1) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso_sem_causa_melhorou(20):=nao_faz_nada(0);
Name: Clinico2 Host: localhost Action(s): verificar_sintomas(1) verificar_peso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso(10):=realizar_exames(0);orientacao_dietetica(0);

Name: endocrino5 Host: localhost Action(s): interconsulta_endocrinologia(1) orientacao_exercicio(10) interconsulta_hipertensao(5) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestemia_nao_respondeu_tto(100): =interconsulta_endocrinologia(0);
Name: Clinico9 Host: localhost Action(s): orientacao_exercicio(30) tto_medicamentoso(1) Goal(s)/Plan(s): hipertenso_dor_no_peito(100):=interconsulta_cardiologia(0);
Name: Ortop10 Host: localhost Action(s): orientacao_exercicio(5) orientacao_dietetica(20) interconsulta_ortopedista(1) Goal(s)/Plan(s): assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedi sta(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);

Figura 6.5 – Descrição externa dos agentes da simulação 2

Simulação 3:

Os integrantes da sociedade nesta simulação são: ortop7 (ortopedista), cardio5 (cardiologista), ortop1 (ortopedista), clinico8 (clínico geral), nefro10 (nefrologista), cardio3 (cardiologista), cardio1 (cardiologista), ortop2 (ortopedista), cardio8 (cardiologista) e ortop5 (ortopedista). A descrição externa de cada agente da sociedade é apresentada na figura 6.6. Os resultados para o modelo CBD e RC* encontram-se respectivamente nas tabelas 6.5 e 6.6.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	0	90	cardio3	-
2	0	90	ortop1	-
3	0	90	ortop7	-
4	0	90	ortop5	-
5	0	90	ortop1	-
6	0	90	cardio3	-
7	0	90	ortop2	-
8	0	90	cardio3	-
9	0	90	ortop1	-
10	0	90	ortop1	-

Tabela 6.5 – Resultado obtido na simulação 3 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	0	0	cardio3	-
2	9	9	ortop1	-
3	9	18	ortop7	-
4	9	27	ortop5	-
5	9	36	ortop1	-
6	0	36	cardio3	-
7	9	45	ortop2	-
8	0	45	cardio3	-
9	9	54	ortop1	-
10	9	63	ortop1	-

Tabela 6.6 – Resultado obtido na simulação 3 do modelo RC* (sistema CENINT)

```

Name: Ortop7
Host: localhost
Action(s): orientacao_exercicio(5)
           orientacao_dietetica(20)
           interconsulta_ortopedista(1)
Goal(s)/Plan(s):
assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedi
sta(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);

```

Name: Cardio5
Host: localhost
Action(s): nao_faz_nada(1)
orientacao_exercicio(8)
tto_medicamentoso(1)
interconsulta_hipertensao(5)
interconsulta_cardiologia(1)
orientacao_dietetica(8)
Goal(s)/Plan(s):
hipertenso_assintomatico_hipercolerestmia_caso_familiar(30):=inter
consulta_nutricionista(0);orientacao_exercicio(0);tto_medicamentos
o(0);interconsulta_hipertensao(0);

Name: Ortopl
Host: localhost
Action(s): orientacao_exercicio(5)
orientacao_dietetica(20)
interconsulta_ortopedista(1)
Goal(s)/Plan(s):
assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedi
sta(0);interconsulta_fisioterapia(0);orientacao_dietetica(0);

Name: Clinico8
Host: localhost
Action(s): orientacao_exercicio(40)
tto_medicamentoso(1)
Goal(s)/Plan(s):
hipertenso_assintomatico_hipercolerestemia_nao_respondeu_tto(100):
=interconsulta_endocrinologia(0);

Name: Nefro10
Host: localhost
Action(s): Goal(s)/Plan(s):
hipertenso_causa_definida(100):=interconsulta_nefrologista(0);

Name: Cardio3
Host: localhost
Action(s): nao_faz_nada(1)
orientacao_exercicio(8)
tto_medicamentoso(1)
interconsulta_hipertensao(5)
interconsulta_cardiologia(1)
orientacao_dietetica(8)
Goal(s)/Plan(s):
hipertenso_sem_causa_melhorou(20):=nao_faz_nada(0);

Name: Cardio1 Host: localhost Action(s): nao_faz_nada(1) interconsulta_cardiologista(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso(50):=fundo_de_olho(0);orientacao_dietetica(0);orientacao_exercicio(1);
Name: Ortop2 Host: localhost Action(s): orientacao_exercicio(5) orientacao_dietetica(20) interconsulta_ortopedista(1) Goal(s)/Plan(s): assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedista(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);
Name: Cardio8 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(30) interconsulta_cardiologia(1) Goal(s)/Plan(s): assintomatico_com_sobrepeso(40):=orientacao_exercicio(0);
Name: Ortop5 Host: localhost Action(s): orientacao_exercicio(5) orientacao_dietetica(20) interconsulta_ortopedista(1) Goal(s)/Plan(s): assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedista(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);

Figura 6.6 – Descrição externa dos agentes da simulação 3

6.3.2 Resultados Obtidos na Ocorrência de Autonomia

A principal característica das sociedades utilizadas nestas simulações é que os seus integrantes são na maioria autônomos, ou seja, sabem resolver os seus próprios problemas.

Simulação 4:

Os integrantes da sociedade nesta simulação são: cardio5 (cardiologista), ortop8 (ortopedista), endocrino2 (endocrinologista), clinico4 (clínica geral), clinico5 (clínico geral), endocrino3 (endocrinologista), nefro7 (nefrologista), nefro6 (nefrologista), clinico6 (clínico geral) e cardio7 (cardiologista). A descrição externa de cada agente da sociedade é apresentada na figura 6.7. Os resultados para o modelo CBD e RC* encontram-se nas tabelas 6.7 e 6.8, respectivamente.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	0	90	ortop8	-
2	0	90	endocrino2	-
3	8	98	clinico4	-
4	0	98	nefro7	-
5	8	106	clinico4	-
6	3	109	nefro6	clinico6
7	8	117	clinico4	-
8	0	117	endocrino2	-
9	0	117	nefro7	-
10	14	131	endocrino3	-

Tabela 6.7 – Resultado obtido na simulação 4 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	9	9	ortop8	-
2	9	18	endocrino2	-
3	14	32	clinico4	nefro7
4	0	32	nefro7	-
5	14	46	clinico4	nefro7
6	17	63	nefro6	ortop8
7	14	77	clinico4	nefro7
8	9	86	endocrino2	-
9	0	86	nefro7	-
10	17	103	endocrino3	cardio5

Tabela 6.8 – Resultado obtido na simulação 4 do modelo RC* (sistema CENINT)

Name: Cardio5
Host: localhost
Action(s): nao_faz_nada(1)
orientacao_exercicio(8)
tto_medicamentoso(1)
interconsulta_hipertensao(5)
interconsulta_cardiologia(1)
orientacao_dietetica(8)
Goal(s)/Plan(s):
hipertenso_assintomatico_hipercolerestmia_caso_familiar(30):=inter
consulta_nutricionista(0);orientacao_exercicio(0);tto_medicamentos
o(0);interconsulta_hipertensao(0);

Name: Ortop8
Host: localhost
Action(s): orientacao_exercicio(5)
orientacao_dietetica(20)
interconsulta_ortopedista(1)
Goal(s)/Plan(s):
assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedi
sta(0);interconsulta_fisioterapia(0);orientacao_dietetica(0);

Name: endocrino2
Host: localhost
Action(s): interconsulta_endocrinologia(5)
orientacao_exercicio(10)
interconsulta_hipertensao(10)
orientacao_dietetica(10)
Goal(s)/Plan(s):
hipertenso(20):=realizar_exames(0);orientacao_dietetica(0);orienta
cao_exercicio(0);

Name: Clinico4 Host: localhost Action(s): verificar_sintomas(1) verificar_peso(1) orientacao_dietetica(10) Goal(s)/Plan(s): obeso_hipertensao(20):=interconsulta_hipertensao(0);orientacao_dietetica(0);
Name: Clinico5 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso_causa_definida(100):=interconsulta_nefrologista(0);
Name: endocrino3 Host: localhost Action(s): verificar_colesterol(2) interconsulta_endocrinologia(1) verificar_pressao(5) Goal(s)/Plan(s): obeso_hipertenso(50):=interconsulta_endocrinologia(0);orientacao_dietetica(0);
Name: Nefro7 Host: localhost Action(s): orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) Goal(s)/Plan(s): obeso_hipertenso(100):=interconsulta_hipertensao(0);
Name: Nefro6 Host: localhost Action(s): tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) orientacao_dietetica(30) Goal(s)/Plan(s): hipertenso(50):=orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);
Name: Clinico6 Host: localhost Action(s): orientacao_exercicio(40) tto_medicamentoso(1) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestemia(100):=orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);

```

Name: Cardio7
Host: localhost
Action(s): nao_faz_nada(1)
           orientacao_exercicio(30)
           ttto_medicamentoso(1)
           interconsulta_cardiologia(1)
           orientacao_dietetica(30)
Goal(s)/Plan(s):
obeso_hipertenso(40):=interconsulta_hipertensao(0);

```

Figura 6.7 – Descrição externa dos agentes da simulação 4

Simulação 5:

Os integrantes da sociedade nesta simulação são: endocrino4 (endocrinologista), ortop1 (ortopedista), clinico10 (clínico geral), fisio1 (fisioterapeuta), lab1 (laboratório), nefro2 (nefrologista), nefro7 (nefrologista), endocrino7 (endocrinologista), oftalmo1 (oftalmologista) e cardio3 (cardiologista). A descrição externa de cada agente da sociedade é apresentada na figura 6.8. Os resultados para o modelo CBD e RC* encontram-se respectivamente nas tabelas 6.9 e 6.10.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	0	0	endocrino7	-
2	0	0	endocrino4	-
3	0	0	cardio3	-
4	0	0	endocrino7	-
5	3	93	ortop1	fisio1
6	0	93	nefro7	-
7	0	93	nefro2	-
8	0	93	endocrino7	-
9	3	96	ortop1	fisio1
10	10	106	clinico10	-

Tabela 6.9 – Resultado obtido na simulação 5 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	0	0	endocrino7	-
2	0	0	endocrino4	-
3	0	0	cardio3	-
4	0	0	endocrino7	-
5	11	11	ortop1	fisio1
6	0	11	nefro7	-
7	0	11	nefro2	-
8	0	11	endocrino7	-
9	11	22	ortop1	fisio1
10	15	37	clinico10	nefro7

Tabela 6.10 – Resultado obtido na simulação 5 do modelo RC* (sistema CENINT)

Name: endocrino4 Host: localhost Action(s): interconsulta_endocrinologia(1) orientacao_exercicio(10) interconsulta_hipertensao(5) Goal(s)/Plan(s): obeso_hipertensao(50):=interconsulta_hipertensao(0);
Name: Ortop1 Host: localhost Action(s): orientacao_exercicio(5) orientacao_dietetica(20) interconsulta_ortopedista(1) Goal(s)/Plan(s): assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedista(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);
Name: clinico10 Host: localhost Action(s): orientacao_exercicio(30) tto_medicamentoso(1) orientacao_dietetica(20) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestemia_caso_familiar(100):=interconsulta_hipertensao(0);orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);
Name: Fisio1 Host: localhost Action(s): orientacao_exercicio(1) interconsulta_fisioterapeuta(1) Goal(s)/Plan(s): fisioterapia(100):=orientacao_exercicio(0);interconsulta_ortopedista(0);

Name: Lab1 Host: localhost Action(s): realizar_exame(1) realizar_examenes(1) realiza_examenes(1) realiza_exame(1) Goal(s)/Plan(s): testes(100):=realizar_exame(0);nao_faz_nada(0);
Name: Nefro2 Host: localhost Action(s): tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso_causa_definida(50):=interconsulta_nefrologista(0);
Name: Nefro7 Host: localhost Action(s): orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) Goal(s)/Plan(s): obeso_hipertenso(100):=interconsulta_hipertensao(0);
Name: endocrino7 Host: localhost Action(s): interconsulta_endocrinologia(1) orientacao_exercicio(10) interconsulta_hipertensao(5) Goal(s)/Plan(s): assintomatico_obeso2_sem_comorbidade(80):=orientacao_exercicio(0); interconsulta_endocrinologia(0);
Name: Oftalmol Host: localhost Action(s): interconsulta Oftamologista(1) fundo_de_olho(1) Goal(s)/Plan(s): consulta(100):=fundo_de_olho(0);verificar_sintomas(0);

```
Name: Cardio3
Host: localhost
Action(s): nao_faz_nada(1)
           orientacao_exercicio(8)
           ttto_medicamentoso(1)
           interconsulta_hipertensao(5)
           interconsulta_cardiologia(1)
           orientacao_dietetica(8)
Goal(s)/Plan(s):
hipertenso_sem_causa_melhorou(20):=nao_faz_nada(0);
```

Figura 6.8 – Descrição externa dos agentes da simulação 5

6.3.3 Resultados Obtidos na Formação de Coalisões

Nestas simulações, as sociedades são compostas por agentes interdependentes que geralmente conseguem formar uma equipe e se ajudam mutuamente.

Simulação 6:

Os integrantes da sociedade nesta simulação são: endocrino3 (endocrinologista), endocrino1 (endocrinologista), clinico5 (clínico geral), nefro9 (nefrologista), nefro5 (nefrologista), clinico2 (clínico geral), cardio1 (cardiologista), cardio6 (cardiologista) oftalmo1 (oftalmologista) e cardio5 (cardiologista). A descrição externa de cada agente da sociedade é apresentada na figura 6.9. Os resultados para o modelo CBD e RC* encontram-se respectivamente nas tabelas 6.11 e 6.12.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	3	93	nefro5	clinico5
2	3	96	nefro9	cardio6
3	10	106	endocrino1	-
4	3	109	nefro9	cardio6
5	10	119	endocrino3	-
6	3	122	oftamo1	clinico2
7	10	132	endocrino1	-
8	3	135	cardio6	nefro9
9	2	137	cardio1	-
10	3	140	nefro9	cardio6

Tabela 6.11 – Resultado obtido na simulação 6 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	15	15	nefro5	cardio1
2	13	28	nefro9	cardio1
3	15	43	endocrino1	cardio1
4	13	56	nefro9	cardio1
5	15	71	endocrino3	cardio1
6	11	82	oftamo1	clinico2
7	15	97	endocrino1	cardio1
8	12	109	cardio6	nefro9
9	11	120	cardio1	oftalmo1
10	13	133	nefro9	cardio1

Tabela 6.12 – Resultado obtido na simulação 6 do modelo RC* (sistema CENINT)

```

Name: endocrino3
Host: localhost
Action(s): verificar_colesterol(2)
           interconsulta_endocrinologia(1)
           verificar_pressao(5)
Goal(s)/Plan(s):
obeso_hipertenso(50):=interconsulta_endocrinologia(0);orientacao_d
ietetica(0);

```

Name: endocrinol Host: localhost Action(s): interconsulta_endocrinologia(1) orientacao_exercicio(10) interconsulta_hipertensao(5) Goal(s)/Plan(s): assintomatico_com_sobrepeso(20):=orientacao_dietetica(0);
Name: Clinico5 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso_causa_definida(100):=interconsulta_nefrologista(0);
Name: Nefro9 Host: localhost Action(s): orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_nefrologista(6) Goal(s)/Plan(s): hipertenso_dor_no_peito(100):=interconsulta_cardiologia(0);
Name: Nefro5 Host: localhost Action(s): orientacao_exercicio(56) tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) Goal(s)/Plan(s): hipertenso(50):=orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);
Name: Clinico2 Host: localhost Action(s): verificar_sintomas(1) verificar_peso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso(10):=realizar_exames(0);orientacao_dietetica(0);
Name: Cardiol1 Host: localhost Action(s): nao_faz_nada(1) interconsulta_cardiologista(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso(50):=fundo_de_olho(0);orientacao_dietetica(0);orientacao_exercicio(1);

Name: Cardio6 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) interconsulta_cardiologia(1) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso_causa_definida(40):=interconsulta_nefrologista(0);
Name: Oftalmo1 Host: localhost Action(s): interconsulta Oftamologista(1) fundo_de_olho(1) Goal(s)/Plan(s): consulta(100):=fundo_de_olho(0);verificar_sintomas(0);
Name: Cardio5 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) interconsulta_cardiologia(1) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestmia_caso_familiar(30):=interconsulta_nutricionista(0);orientacao_exercicio(0);tto_medicamentoso(0);interconsulta_hipertensao(0);

Figura 6.9 – Descrição externa dos agentes da simulação 6

Simulação 7:

Os integrantes da sociedade nesta simulação são: Lab1 (laboratório), endocrino9 (endocrinologista), Clinico5 (clínico geral), Nefro9 (nefrologista), Nefro5 (nefrologista), Clinico2 (clínico geral), Nutri1 (nutricionista), Cardio6 (cardiologista), Oftalmo1 (oftamologista) e Cardio5 (cardiologista). A descrição externa de cada agente da sociedade é apresentada na figura 6.10. Os resultados para o modelo CBD e RC* encontram-se respectivamente nas tabelas 6.13 e 6.14.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	3	93	oftalmo1	clinico2
2	3	96	clinico5	nefro5
3	3	99	nefro5	clinico5
4	3	102	cardio6	nefro9
5	2	104	clinico2	-
6	3	107	nefro5	clinico5
7	3	110	nefro9	cardio6
8	2	112	endocrino9	-
9	3	115	nefro9	cardio6
10	3	119	nefro5	clinico5

Tabela 6.13 – Resultado obtido na simulação 7 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	11	11	oftalmo1	clinico2
2	12	23	clinico5	nefro9
3	16	39	nefro5	nutri1
4	12	51	cardio6	nefro9
5	11	62	clinico2	lab1
6	16	78	nefro5	nutri1
7	12	90	nefro9	cardio6
8	11	101	endocrino9	oftalmo1
9	12	113	nefro9	cardio6
10	16	129	nefro5	nutri1

Tabela 6.14 – Resultado obtido na simulação 7 do modelo RC* (sistema CENINT)

```

Name: Lab1
Host: localhost
Action(s): realizar_exame(1)
  realizar_exam(1)
  realiza_exam(1)
  realiza_exam(1)
Goal(s)/Plan(s):
testes(100):=realizar_exame(0);nao_faz_nada(0);

```

Name: endocrino9 Host: localhost Action(s): interconsulta_endocrinologia(5) orientacao_exercicio(10) interconsulta_hipertensao(10) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso(20):=fundo_de_olho(0);orientacao_dietetica(0);orientacao_exercicio(1);
Name: Clinico5 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso_causa_definida(100):=interconsulta_nefrologista(0);
Name: Nefro9 Host: localhost Action(s): orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_nefrologista(6) Goal(s)/Plan(s): hipertenso_dor_no_peito(100):=interconsulta_cardiologia(0);
Name: Nefro5 Host: localhost Action(s): orientacao_exercicio(56) tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) Goal(s)/Plan(s): hipertenso(50):=orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);
Name: Clinico2 Host: localhost Action(s): verificar_sintomas(1) verificar_peso(1) orientacao_dietetica(10) Goal(s)/Plan(s): hipertenso(10):=realizar_exames(0);orientacao_dietetica(0);
Name: Nutril Host: localhost Action(s): orientacao_dietetica(1) interconsulta_nutricionista(1) Goal(s)/Plan(s): preparar_cardapio(100):=orientacao_dietetica(0);nao_faz_nada(0);

Name: Cardio6 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) interconsulta_cardiologia(1) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso_causa_definida(40):=interconsulta_nefrologista(0);
Name: Oftalmol Host: localhost Action(s): interconsulta Oftamologista(1) fundo_de_olho(1) Goal(s)/Plan(s): consulta(100):=fundo_de_olho(0);verificar_sintomas(0);
Name: Cardio5 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) interconsulta_cardiologia(1) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestmia_caso_familiar(30):=interconsulta_nutricionista(0);orientacao_exercicio(0);tto_medicamentoso(0);interconsulta_hipertensao(0);

Figura 6.10 – Descrição externa dos agentes da simulação 7

Simulação 8:

Os integrantes da sociedade nesta simulação são: lab1 (laboratório), ortop10 (ortopedista), cardio7 (cardiologista), nefro4 (nefrologista), endocrino1 (endocrinologista), clinico8 (clínico geral), endocrino3 (endocrinologista), nefro10 (nefrologista), cardio1 (cardiologista) e fisio1 (fisioterapeuta). A descrição externa de cada agente da sociedade é apresentada na figura 6.11. Os resultados para o modelo CBD e RC* encontram-se respectivamente nas tabelas 6.15 e 6.16.

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	90	-	-
1	8	98	endocrino3	-
2	8	106	endocrino3	-
3	2	108	nefro10	-
4	4	112	clinico8	-
5	4	116	clinico8	-
6	0	116	cardio1	-
7	0	116	nefro4	-
8	8	124	endocrino3	-
9	4	128	clinico8	-
10	4	132	lab1	-

Tabela 6.15 – Resultado obtido na simulação 8 do modelo CBD (manual)

ciclo	msg no ciclo	acumulado	ativo	parceiro
Início	0	0	-	-
1	14	14	endocrino3	cardio1
2	14	28	endocrino3	cardio1
3	11	39	nefro10	nefro4
4	12	51	clinico8	endocrino3
5	12	63	clinico8	endocrino3
6	9	72	cardio1	-
7	9	81	nefro4	-
8	14	95	endocrino3	cardio1
9	12	107	clinico8	endocrino3
10	12	119	lab1	cardio

Tabela 6.16 – Resultado obtido na simulação 8 do modelo RC* (sistema CENINT)

```

Name: Lab1
Host: localhost
Action(s): realizar_exame(1)
    realizar_examenes(1)
    realiza_examenes(1)
    realiza_exame(1)
Goal(s)/Plan(s):
testes(100):=realizar_exame(0);nao_faz_nada(0);

```

Name: Ortop10 Host: localhost Action(s): orientacao_exercicio(5) orientacao_dietetica(20) interconsulta_ortopedista(1) Goal(s)/Plan(s): assintomatico_obeso1_com_hernia_disco(100):=interconsulta_ortopedista(0);interconsulta_fisioterapeuta(0);orientacao_dietetica(0);
Name: Cardio7 Host: localhost Action(s): nao_faz_nada(1) orientacao_exercicio(30) tto_medicamentoso(1) interconsulta_cardiologia(1) orientacao_dietetica(30) Goal(s)/Plan(s): obeso_hipertenso(40):=interconsulta_hipertensao(0);
Name: Nefro4 Host: localhost Action(s): orientacao_exercicio(10) tto_medicamentoso(1) interconsulta_hipertensao(3) interconsulta_nefrologista(6) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso(50):=fundo_de_olho(0);orientacao_dietetica(0);orientacao_exercicio(0);tto_medicamentoso(0);
Name: endocrinol Host: localhost Action(s): interconsulta_endocrinologia(1) orientacao_exercicio(10) interconsulta_hipertensao(5) Goal(s)/Plan(s): assintomatico_com_sobrepeso(20):=orientacao_dietetica(0);
Name: Clinico8 Host: localhost Action(s): orientacao_exercicio(40) tto_medicamentoso(1) Goal(s)/Plan(s): hipertenso_assintomatico_hipercolerestemia_nao_respondeu_tto(100):=interconsulta_endocrinologia(0);
Name: Nefro10 Host: localhost Action(s): Goal(s)/Plan(s): hipertenso_causa_definida(100):=interconsulta_nefrologista(0);

Name: Cardio1 Host: localhost Action(s): nao_faz_nada(1) interconsulta_cardiologista(1) orientacao_exercicio(8) tto_medicamentoso(1) interconsulta_hipertensao(5) orientacao_dietetica(8) Goal(s)/Plan(s): hipertenso(50):=fundo_de_olho(0);orientacao_dietetica(0);orientacao_exercicio(1);
Name: Fisiol Host: localhost Action(s): orientacao_exercicio(1) interconsulta_fisioterapeuta(1) Goal(s)/Plan(s): fisioterapia(100):=orientacao_exercicio(0);interconsulta_ortopedista(0);

Figura 6.11 – Descrição externa dos agentes da simulação 8

6.4 Análise dos Resultados

Em todos os resultados, observamos no modelo CBD um número de mensagens acumuladas antes que os ciclos se iniciem. Estas mensagens são as apresentações que os agentes fazem de si para os outros. O modelo RC*, pelo contrário, não possui um fluxo de comunicação inicial, pois os agentes não necessitam de conhecimento mútuo.

Notamos ainda que na ausência de parceiros, ou seja, quando na comunidade não há agentes aptos a realizar a ação necessária, não há fluxo de comunicação no modelo CBD, enquanto que no modelo RC* temos mensagens de anúncio sendo enviados.

O fato de existir prováveis parceiros para serem escolhidos não garantem uma formação de coalisção no modelo CBD, pois neste modelo o agente escolhido não aceita necessariamente a parceria. Por outro lado, sempre que há prováveis parceiros a coalisção é formada no modelo RC*, uma vez que o agente que respondeu ao lance é candidato para a ação. Desta forma, nos experimentos reais nem sempre tivemos sucesso ao tentar formar uma coalisção

no modelo CBD, o que no nosso trabalho afeta no fluxo de comunicação. Muitas vezes não tivemos a mensagem de confirmação da formação de coalisção pelo agente ativo.

Como o critério de escolha de parceiros é diferente nos dois modelos, nem sempre os parceiros escolhidos em ambos os modelos coincidem. Cabe lembrar que no modelo CBD o agente escolhe seus parceiros baseado nas suas relações de dependência, enquanto que no modelo RC* a escolha é baseada nos valores das ações anunciadas. Tal característica pode ser bem observada nas tabelas 6.11 a 6.14.

Pelos resultados obtidos podemos confirmar, como inicialmente previsto na seção 4.3, que o fluxo inicial de comunicação no modelo CBD faz com que sejam necessários muitos ciclos para que o mesmo tenha um número de mensagens menor que o modelo RC*. Notamos também que no modelo CBD, o fato de existir um ciclo em que $k = m$ pode fazer com que o fluxo de comunicação global do modelo CBD seja mais alto do que o modelo RC* por mais ciclos do que se esperaria, conforme os estudos apresentados no capítulo 4, quando supusemos o valor de k constante em todos os ciclos.

Um resumo dos resultados obtidos encontra-se apresentado na tabela 6.17.

Situação	Simulação	modelo CBD	modelo RC*
ausência de parceiros	1	130	117
	2	108	93
	3	90	63
autônomos	4	131	103
	5	106	37
formação de coalisção	6	140	133
	7	119	129
	8	132	119

Tabela 6.17 – Sumário dos resultados obtidos nas simulações

7. Conclusões e Discussões

Neste capítulo, apresentamos as conclusões finais e uma discussão sobre trabalhos futuros.

Pela análise matemática apresentada no capítulo 4, ao fixar o número de agentes que enviam o lance (b) no modelo RC* e os que respondem à proposta (k) no modelo CBD em todos os ciclos do estudo, observamos que para **determinados** valores de k e b há um ciclo ($g_{crítico}$) a partir do qual o total de mensagens trocadas entre os agentes torna-se **maior no modelo RC* do que no modelo CBD**. A tendência é imaginar que sempre existirá este ciclo, pois dadas as equações (2) e (4) do capítulo 4, parece que em algum momento tais equações se igualam. Porém, não é isto que ocorre, como demonstra o gráfico da figura 4.8. Esta situação inclusive independe do valor de b . Podemos entender tal fato, analisando as equações de formação de coalisção dos modelos RC* e CBD:

$$S_{RC^*} = \sum_{i=1}^g S_{ciclo_{RC^*_i}}$$

e

$$S_{CBD} = S_{apresentacao} + \sum_{i=1}^g S_{ciclo_{CBD_i}}$$

Assim nota-se que quando $(n+b)$ é menor do que $(2k+1)$, temos $S_{ciclo_{RC^*}} < S_{ciclo_{CBD}}$ e portanto sempre teremos $S_{RC^*} < S_{CBD}$.

Por sua vez, nos estudos de caso realizados notou-se que obter este $g_{crítico}$ é mais difícil na prática, pois não se considerou nesta análise preliminar o fato de que existem agentes autônomos, ausência de parceiros e coalisões não formadas. Observa-se esta tendência na tabela 6.17.

Observamos ainda que basta haver alguns ciclos em que encontrar um parceiro é difícil no modelo CBD para que o número de ciclos tenha que ser muito alto a fim de se obter um

número de mensagens menor do que o do modelo RC* (tabela 6.11 e 6.12). Por outro lado, quando o modelo CBD tem êxito em encontrar o seu parceiro na primeira proposta, obtém-se o $g_{\text{crítico}}$ com mais facilidade como notamos nas tabelas 6.13 e 6.14. Cabe ressaltar que tal situação denota um raciocínio social apurado (k próximo de 1).

Posteriormente, optamos por fixar somente um dos parâmetros (k e b) e analisar o comportamento do outro modelo. Assim obtivemos os valores $k_{\text{crítico}}$ e $b_{\text{crítico}}$, que fazem com que o coeficiente linear dos modelos se iguale. A análise demonstrou que ao fixar k em 1 (raciocínio social apurado), **sempre** existe um momento em que o número de mensagens do RC* se torna maior do que o modelo CBD, independente do valor de b . Ao contrário, quando o valor de b é constante e igual a 1, para que o modelo CBD continue tendo um número menor de mensagens o número máximo de propostas recusadas tem que ser aproximadamente a metade do número de agentes existente na sociedade. Isso somente é possível se os agentes ativos no modelo CBD tiverem uma idéia errônea sobre os outros agentes na sociedade e muitas revisões de crenças devam ser feitas.

Nos estudos de caso realizados, notamos que em determinados tipos de sociedade podemos ter uma noção de quanto é mais fácil ou difícil o modelo CBD ter um número menor de mensagens que o modelo RC*. Observamos que:

- numa sociedade em que os seus integrantes pouco se ajudam, ou seja, há ausência de obtenção de parceiros, fica relativamente mais fácil o modelo CBD se tornar mais eficaz que o modelo RC*, como demonstram as tabelas 6.5 e 6.6. Em particular, pode-se observar que em tais situações o número de mensagens enviadas no ciclo é 0 para o modelo CBD e no mínimo igual a $(n - 1)$ para o modelo RC*;
- numa sociedade em que os seus integrantes são na sua maioria independentes (autônomos) fica mais difícil para o modelo CBD ter o número de mensagens menor que o modelo RC*, como se verifica na tabela 6.9 e 6.10. Tal conclusão se explica pelo fato dos dois modelos não enviarem mensagens nos ciclos, o que acarreta um maior peso final nas mensagens de apresentação no modelo CBD;

- numa sociedade em que seus integrantes são agentes que geralmente conseguem formar coalisões, o modelo CBD se torna mais eficiente em termos de fluxo de comunicação quando obtém respostas afirmativas na primeira proposta (tabelas 6.13 e 6.14). Por outro lado, quando temos recusas no modelo CBD, este se torna menos eficiente em termos de fluxo de comunicação (tabelas 6.15 e 6.16).

Podemos concluir então que caso o mecanismo de raciocínio social seja eficaz (valor de k baixo e próximo de 1), o modelo CBD terá, a partir de um certo número de ciclos de interação, um fluxo de comunicação global menor do que o do modelo RC*.

Por outro lado, sabemos que os sistemas multiagentes são utilizados para desenvolver modelos teóricos que permitem elucidar a estrutura de processos complexos e que a orientação a objetos facilita o desenvolvimento de sistemas complexos. Percebeu-se que a orientação a objetos poderia ser uma ferramenta adequada para desenvolver sistemas multiagentes. Assim neste trabalho, optou-se por desenvolver o sistema CENINT utilizando as técnicas de orientação a objetos, mais especificamente utilizar os diagramas da UML (Unified Modeling Language) para análise e projeto do sistema.

No presente trabalho, observou-se que os diagramas da UML e as técnicas de orientação a objetos se mostraram eficazes na análise e projeto do sistema CENINT. A utilização de tais técnicas permite adicionar com mais facilidade novos modelos de organização, agentes e comunicação no sistema CENINT, como documentados no capítulo 5.

Além disso, o sistema foi desenvolvido em 3 camadas como é recomendado pelas metodologias de arquitetura de sistemas distribuídos e cliente-servidor, o que facilita a interface com o usuário e o acesso aos diferentes tipos de agentes (figura 5.9).

O uso da linguagem Java na implementação do sistema CENINT permite a portabilidade do mesmo para as mais variadas plataformas existentes. Uma prova disto é que as

simulações foram realizadas em máquinas com sistema operacional Linux e máquinas com sistema operacional Windows NT/95.

Antes de realizar as simulações, optou-se por realizar um estudo matemático sobre o provável comportamento do fluxo de comunicação dos modelos CBD e RC*, a fim de direcionar da melhor forma possível estudos posteriores. Portanto, para eliminar fatores externos, optou-se por fixar certos parâmetros e deixar flexível somente aqueles que poderiam realmente influenciar no fluxo de comunicação. Estas opções estão detalhadas na seção 4.2. Através desta análise matemática, confirmou-se que o comportamento do fluxo de comunicação não é probabilístico, não justificando uma análise estatística. Entretanto, o estudo do comportamento determinístico dos modelos foi realizado através de simulações, cujo resultado final é um conjunto de dados que nos permite analisar as diferenças comportamentais de ambos os modelos propostos.

O objetivo deste trabalho é o estudo do fluxo global de comunicação entre os agentes dos modelos RC* e CBD. Para tanto desconsideramos alguns aspectos, como por exemplo o tempo de processamento interno dos agentes. Isolamos o nosso estudo a fim de que a análise fosse realizada sem as imperfeições do ambiente e desta forma ter um estudo enfocando somente o comportamento desejado. Uma análise de performance mais completa deveria obviamente levar em conta tal aspecto.

Neste trabalho, não analisamos uma sociedade com características dinâmicas, ou seja, uma comunidade onde os agentes têm a liberdade de permanecer nela ou não e ainda modificar a sua descrição externa. Pelos estudos realizados sem considerar tais características, demonstra-se que pode existir uma variação nos resultados nestes casos, que poderiam ser questões interessantes a serem abordadas em trabalhos futuros.

Após analisar o fluxo de comunicação entre os agentes de modo isolado, acreditamos que uma análise considerando as imperfeições do ambiente e características dinâmicas em ambos os modelos deve ser uma continuidade interessante ao trabalho, a fim de complementar os estudos iniciados.

Para que tal estudo seja viável, seria interessante que o sistema CENINT fosse implementado realmente como um sistema distribuído. Em trabalhos futuros, prevê-se tornar o sistema CENINT distribuído. Para isto, devemos executar os pacotes Ambiente e Comunicação num servidor, enquanto que o pacote Agente poderia ser executado em diversas máquinas cliente. Ao formar a sociedade, através do RMI (Remote Method Invocation), os agentes são instanciados nas máquinas clientes. Após formar a sociedade, o servidor de comunicação cria a caixa de correio de cada agente e o envia ao seus donos, através do *object serialization*. Alterações também devem ser realizadas nas classes Organization e Barrier.

Bibliografia

- [AGH85] AGHA, G.; HEWITT, C. E. **Concurrent Programming Using Actors: Exploiting Large Scale Parallelism**. AI Memo 865, Massachusetts Institute of Technology, 1985. *apud* [BON88]
- [ALV97] ALVARES, L.O.; SICHMAN, J.S. **Introdução aos Sistemas Multiagentes**. XVI Jornada de Atualização em Informática. Brasília, DF. 2-8 agosto 1997.
- [ARB85] ARBIB, M.A. Brain Theory and cooperative Computation. **Humam Neurobiology**, v.4, p.201-218. 1995 *apud* [BON88]
- [ARB88] ARBIB, M. A.; **The Metaphorical Brain 2: An Introduction to Schema Theory and Neural Networks**. Willey-Interscience, 1988. *apud* [BON88]
- [AUS62] AUSTIN, J.L. **How to Do Things with Words**. Oxford, Oxford University Press, 1962. *apud* [WOO94]
- [BEL98] BELLOQUIM, A.; ITO, M. **Análise, Projeto e Programação Orientada a Objetos**. São Paulo, 1998 100p. Apostila. /Apresentado nos cursos de orientação a objetos da Choose Technologies/.
- [BOI93] BOISSIER, O. **Problème du Contrôle dans un Système Intégré de Vision. Utilisation d'un Système Multi-Agents**. Grenoble, 1993. n.p. Tese (Doutorado) - Institut National Polytechnique de Grenoble. *apud* [SIC95]
- [BON88] BOND, A.H.; GASSER, L. **Readings in Distributed Artificial Intelligence**. Palo Alto, Morgan Kaufmann Publishers, Inc., 1988.
- [BOO94] BOOCH, G. **Object-Oriented Analysis and Design with Applications**. Redwood City, CA: Benjamin Cummings. 1994.
- [BOO97] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **Unified Modeling Language Semantics and Notation Guide 1.0** San Jose, CA, 1997. (Relatório Técnico da Rational Software Corporation).
- [BRO86] BROOKS, R. A Robust Layered Control System for a Mobile Robot. **IEEE Journal of Robotics and Automation**, v.2, n.1, p.14-23, March, 1986.
- [CAM90] CAMPBELL, J. A.; D'INVERNO, M. P. Knowledge interchange protocols. In: Demazeau, Y; Müller, J.P. editors, **Decentralized A.I. - First European Workshop on Modelling Autonomous Agents in a Multi-Agent World**, England, 1990. **Proceedings**. Netherlands, Elsevier Science Publishers B.V., 1990, v.1, p. 63-80.

- [CAR87] Cardozo, E. **DPSK: a kernel for distributed problem solving**. Pittsburgh, 1987. n.p. Tese (Doutorado) - Carnegie-Mellon University.
- [CAR91] CARVER, N.; CVETANOVIC, Z.; LESSER, V. Sophisticated Cooperation in Distributed Problem Solving. Proceedings of the 9th National Conference on Artificial Intelligence 1. **Proceedings**. Anaheim, pp. 191-198. *apud* [NWA96]
- [CAR93] CARDOZO, E.; SICHMAN, J. S.; DEMAZEAU, Y. Using the Active Object Model to Implement Multi-Agent Systems. In: 5th. IEEE Int. Conf. on Tools with Artificial Intelligence. Boston, 1993. **Proceedings**. Los Alamitos, IEEE Computer Soc. Press., 1993. p. 70-77. v.2.
- [CAR95] CARVER, N.; LESSER, V. The DRESUN Testbed for Research in FA/C Distribution Situation Assessment: Extensions to the Model of External Evidence. Proceedings of the 1st. International Conference on Multi-Agent Systems (ICMAS-95). **Proceedings**. San Francisco, June, pp. 33-30, 1995. *apud* [NWA96]
- [CAS90] CASTELFRANCHI, C. Social Power. A point Missed in Multi-Agent, DAI and HCI. In: Demazeau, Y; Müller, J.P. editors, Decentralized A.I. - First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, England, 1990. **Proceedings**. Netherlands, Elsevier Science Publishers B.V., 1990, v.1, p. 49-62.
- [CAS95] CASTELFRANCHI, C. Guarantees for autonomy in cognitive agent architecture. . In: Wooldridge, M.J.; Jennings, N.R. editors, **Intelligent Agents**, pp. 56-70 – Berlin, DE, Springer-Verlag, 1995 *apud* [SIC95].
- [CHA81] CHANDRASEKARAN, B. Natural and Social System Metaphors for Distributed Problem Solving: Introduction to the Issue. **IEEE Transactions on Systems, Man and Cybernetics**, v. 11, n. 1, p.1-5, 1981.
- [COH95] COHEN, P. R.; LEVESQUE, H.J. Communicative actions for artificial agents. Proceedings of the International Conference on Multi-Agent Systems, June 1995. **Proceedings**. AAAI Press. *apud* [MAY96]
- [COR93] CORKILL, D.D.; LESSER, V.R. The distributed vehicle monitoring testbed: a tool for investigating distributed problem solving networks. **AI Magazines**. V.4, n.3, p. 15-33, Fall 1993. *apud* [ALV97]
- [COU94] COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Distributed Systems - Concepts and Designs**. 2a. edição. Addison Wesley Publishing Company, 1994.
- [DAV83] DAVIS, R.; SMITH, R. G. Negotiation as a metaphor for distributed problem solving. **Artificial Intelligence**, v.20, p.63-109, 1983. *apud* [KRA93]

- [DEC87] DECKER, K. Distributed problem-solving techniques: a survey. **IEEE Transactions on System, Man and Cybernetics**, v.17, p.729-740, 1987. *apud* [MAR92]
- [DEC93] DECKER, K.S.; LESSER, V.R. Designing a Family of Coordination Algorithms. Proceedings of the 11th National Conference on Artificial Intelligence. **Proceedings**. Washington, pp. 217-224. *apud* [NWA96]
- [DEM90] DEMAIZEAU, Y.; MÜLLER, J. P. **Decentralized A.I.** - Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, England, 1989. Netherlands, Elsevier Science Publishers, B.V. 1990. v.1.
- [DEM94] DEMAIZEAU, Y.; BOISSIER, O.; KONING, J. Interaction protocols in distributed artificial intelligence and their use to control robot vision systems. In: 6th. International Conference on Artificial Intelligence and Information-Control Systems of Robot's 94, Slovaquia, 1994. **Proceedings**. Singapore, World Scientific, 1994. p. 17-30. *apud* [SIC95]
- [DEM94a] DEMAIZEAU, Y.; BOISSIER, O.; KONING, J. Using interaction protocols to control vision systems. IEEE International Conference on Systems, Man and Cybernetics, San Antonio, 1994. **Proceedings**. New York, IEEE, 1994. p.1616-1621. v.2
- [DEM95] DEMAIZEAU, Y. From Interactions to collective behaviour in agent-based systems. 1st European Conference on Cognitive Science. St. Malo, 1995. **Pre-proceedings**. St. Malo, 1995. n.p.
- [DUR87] DURFEE, E. H.; LESSER, V. R.; CORKILL, D. D. Coherent Cooperation Through Communication Problems Solvers. **IEEE Transactions on Computers**, v.36, p. 1275-1291, 1987.
- [DUR94] DURFEE, E. H.; ROSENSCHEIN, J. S. Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples. In: 13th International Workshop on Distributed Artificial Intelligence. Seattle, 1994. **Proceedings**. Seattle, AAIL, 1994. p. 46-55.
- [ENG88] ENGELMORE, R.; MORGAN, T. eds. **Blackboard systems**. Reading, Addison-Wesley, 1988. *apud* [ALV97]
- [ERI98] ERIKSSON, H. E.; PENKER, M. **UML Toolkit**. New York. John Wiley & Sons, Inc. 1998.
- [ERM80] ERMAN, L.D. et al. The HEARSAY-II speech-understanding system: integrating knowledge to resolve uncertainty. **Computing Surveys**, v.12, n.2, p. 213-253, June 1980. *apud* [ALV97]

- [ETZ94] ETZIONI, O.; WELD D.S. A Softbot-based Interface to the Internet. **Communications of the ACM**, vol.37 n.7 July 1994.
- [FLA97] FLANAGAN, D. **Programming in Java**. Salus, P. H. (eds.) In: Handbook of Programming Languages. vol I. Object-Oriented Programming Languages. USA. Macmillan Technical Publishing. 1a. edição, 1998.
- [FOX79] FOX, M.S. **Organization Structuring: Designing Large Complex Software**. Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, December, 1979. (Technical Report CMU-CS-79-155) *apud* [BON88]
- [FOX81] FOX, M.S. An Organizational view of distributed systems. **IEEE Transactions on Systems, Man and Cybernetics**, v.11, n. 1, p. 70-80. Jan 81. *apud* [ALV97]
- [FRA96] FRANKLIN, S.; GRAESSER A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. Proceedings of the 3rd. International Workshop on Agent Theories, Architectures, and Languages. **Proceedings**. Springer-Verlag, 1996.
- [GAL88] GALLIERS, J.R. **A theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict**. London, UK, Phd Thesis, Department of Psychology, Open University, September 1988 *apud* [SIC95].
- [GAS87] GASSER, L. et al. MACE: a flexible testbed for distributed AI research. In: **Hunhs, M.N. ed. Distributed Artificial Intelligence** Los Altos, Morgan Kaufmann, 1987.
- [GAS88] GASSER, L. Distribution and Coordination of tasks among intelligent agents. Proceedings of the first Scadinaviam Conference on Artificial Intelligence, p. 177-192. **Proceedings**. International Organisations Services B.V., Van Diemenstraat 94, 1013CN, Amsterdam, Netherlands, March 1988. *apud* [BON88]
- [GAS91] GASPAR, G. **Communication and belief changes in a society of agents: Towards a formal modelo of an auntonomous agent**. In: Demazeau, Y.; Müller, J. P., Decentralized A.I. 2 ed., pp. 245-255. Amsterdam, NL, Elsevier Science Publishers B.V., 1991 *apud* [SIC95].
- [GER86] GERSON, E.M.; STAR, S.L. Analyzing DueProcess in the Workplace. **ACM Tansactions on Office Informations Systems**, v.4, n.3, p.257-270, July 1986. *apud* [BON88]
- [GOS96] GOSLING J.; JOY, B.; STEELE, G. **What is Java?** Salus, P. H. (eds.) In: Handbook of Programming Languages. vol I. Object-Oriented Programming Languages. USA. Macmillan Technical Publishing. 1a. edição, 1998.
- [HEW77] HEWITT, C. Viewing Control Structures as Patterns of Passing Messages. **Artificial Intelligence**. v.8, n.3, p.323-364. *apud* [NWA96]

- [HEW86] HEWITT, C. **Offices are open systems**. In BOND, A.H.; GASSER, L., ed. Readings in distributed artificial intelligence. San Mateo, Morgan Kaufmann Publishers Inc., 1988, p. 321-329. *apud* [SIC92]
- [ITO99] ITO, M. **Manual do Sistema CENINT**: Manual. São Paulo, LTI/EPUSP, 1999.
- [JAC92] JACOBSON, I.; CHRISTERSON, M.; JONSSON, P.; ÖVERGAARD, G. **Object-Oriented Software Engineering**. New York. 1a. edição. Addison-Wesley Publishing Company. 1992.
- [JEN95] JENNINGS, N.; CORERA, J. M.; LARESGOITI, L.; MAMDANI, E., PERRIOLLAT, F.; SKAREK, P.; VARGA, L. Using ARCHON to Develop Real World DAI Applications for Electricity Transportation and Particle Accelerator Control. **IEEE Expert**, Special Issue on Real World Applications of DAI Systems. *apud* [NWA96]
- [KON80] KONOLIGE, K.; NILSSON, N. J. Multiple-agent planning systems. Proceedings of the 1st National Conference on Artificial Intelligence, pp. 138-141. **Proceedings**. Stanford, CA, August 1980 *apud* [SIC95].
- [KRA93] KRAUS, S. Agents Contracting Tasks in Non-Collaborative Environments. 11th National Conference on Artificial Intelligence. Washington, 1993. **Proceedings**. Menlo Park, AAAI Press, 1993. p. 243-248.
- [KRA93a] KRAUS, S.; WILKENFELD, J. A Strategic Negotiations Model with Applications to an International Crisis. **IEEE Transactions on Systems, Man and Cybernetics**, v.23, n.1, p.313-323, 1993.
- [KUR85] KUROSE, J.F.; SCHWARTZ, M.; YEMINI, Y. A Microeconomic Approach to Optimization of Channel access policies in multiaccess networks. Proceeding of the Fifth International Symposium on Distributed Computing Systems, p. 70-80. **Proceedings**. May, 1985. *apud* [BON88]
- [LEE93] LEE, K. C.; MABSFUEKD JR., W.H.; BELLCORE, A. P. S. A Framework for Controlling Cooperative Agents. **IEEE Computer**, v.26, n.7, p.8-15, 1993
- [LEN75] LENAT, D. B. BEINGS: Knowledge as Interacting Experts. Proceedings of the 1975 International Joint Conference on Artificial Intelligence. **Proceedings**. pp. 126-133, 1975. *apud* [BON88]
- [LES83] LESSER, V.R.; CORKILL, D.D. The Distributed Vehicle Monitoring Testbed: A tool for investigating distributed problem solving networks. **AI Magazine**. 15-33, Fall 1983. *apud* [BON88]

- [LES90] LESSER, V.R. An Overview of DAI: Viewing Distributed AI as Distributed Search. **Journal of Japanese Society for Artificial Intelligence Special issue on Distributed Artificial Intelligence** v.5, n. 4, 1990.
- [LEV94] LEVITT, R.; COHEN, P.; KUNZ, J.; NASS, C.; CHRISTIANSEN, T.; JIN, Y. **The Virtual Design Team: Simulating how Organisational Structure and Communications Tools affect Team Performance.** In: Carley, K.; Prietural, M. (eds.) Computational Organisation Theory, San Francisco: Lawrence Erlbaum. 1994. *apud* [NWA96]
- [LEV97] LEVY, R.; EROL, K.; MITCHELL JR., J.H. **A Study of InfraStructure Requirements and Software Platforms for Autonomous Agent.** In: <http://www.aaria.uc.edu/platforms.html>. 1997.
- [LIN96] LINDEN, P.; **Just Java.** Java Series. California. SunSoft Press A Prentice-Hall Title. 1996.
- [LOU96] LOUÇÃ, J. A.; COELHO, H.; PINSON, S. Forming Coalition in Task Oriented Multi-Agent Systems. In: 1st Ibero American Workshop on Distributed Artificial Intelligence and Multiagent Systems. Nalapa, 1996. **Proceedings.** Nalapa, LAMIA&MIA Universidad VeraCruzana, 1996. p.109-125.
- [MAE95] MAES, P. Artificial Life Meets Entertainment: Life-like Autonomous Agents. **Communications of ACM**, v. 38, n. 11, 1995. *apud* [TUR98]
- [MAL88] MALONE, T.W.; FIKES, R.E.; GRANT, K. R.; HOWARD, M.T. **Enterprise: A Market-Like Task Scheduler for Distributed Computing environments.** In: Bernardo A.Huberman, editor, The Ecology of Computing. Elsevier Science Publishers/North Holland 1988. *apud* [BON88]
- [MAR92] MARTIAL, F.V. **Coordinating Plans of Autonomous Agents.** Berlin, Springer-Verlag Berlin Heidelberg, 1992. (Lecture Notes in Artificial Intelligence - Subseries of Lecture Notes in Computer Science)
- [MAY96] MAYFIELD, J.; LABROU, Y.; FININ, T. Evaluation of KQML as an Agent Communication Language. In: Wooldridge, M.; Müller, J. P.; Tambe M. Intelligent Agents Volume II – Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages. **Proceedings.** Lecture Notes in Artificial Intelligence, Springer-Verlag, 1996.
- [MIN79] MINSKY, M. **The Society Theory of Thinking.** In: Artificial Intelligence: an MIT Perspective, p. 423-450, Mit Press, Cambridge, MA, 1979. *apud* [BON88]
- [MIN80] MINSKY, M. **K-Lines: A Theory of Memory. Cognitive Science.** v. 4, n. 2, p.117-133. 1980, *apud* [BON88]

- [MIN86] MINSKY, M. **The Society of the Mind**. Simon and Schuster. New York. 1986. *apud* [BON88]
- [NWA96] NWANA, H. Software Agents: An Overview. **Knowledge Engineering Review**. v. 11, n 3, pp. 205-244, October/November 1996.
- [PLE92] **Vers une taxinomie du vocabulaire pour les systèmes multi-agents**. In: Actes de la lère. Journée Systèmes Multi-Agents du PRC-GDR Intelligence Artificielle, Nancy, France, 1992. *apud* [ALV97]
- [RAM85] RAMAMRITHAM, K. STANKOVIC, J.A. Distributed task scheduling in Hard Real-Time Distributed System. **IEEE Software**. v.7, p.65-75, 1985.
- [ROS94] ROSENSCHEIN, J. S.; ZLOTKIN, G. Designing Conventions for Automated Negotiation. **AI Magazine**, v.15, n.3, p.29-46, 1994.
- [ROS94] ROSENSCHEIN, J.S.; ZLOTKIN, G. Designing Conventions for Automated Negotiation. **AI Magazine**, v. 15, n. 3. Fall, 1994. *apud* [TUR98]
- [RUS95] RUSSELL, J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Englewood Cliffs, New Jersey, Prentice-Hall Inc., 1995. *apud* [TUR98]
- [SAN97] SANDHOLM, T. W.; LESSER, V. R. Coalitions among Computationally Bounded Agents. **Artificial Intelligence**. v. 94, n.1, p.99-137, Special issue on Economic Principles of Multiagent Systems.
- [SEA69] SEARLE, J. R. **Speech Acts; An Essay in the Philosophy of Language**. Cambridge. Cambridge University Press. 1969. *apud* [WOO94]
- [SIC92] SICHMAN, J. S.; DEMAZEAU, Y. When can knowledge-based systems be called agents?. 9th Brazilian Symposium on Artificial Intelligence. Rio de Janeiro, 1992. **Proceedings**. Rio de Janeiro, SBIA, 1992. p. 172-185.
- [SIC94] SICHMAN, J. S.; DEMAZEAU, Y. Using Class Hierarchies to Implement Social Reasoning in Multi-Agent Systems. In: 11th Brazilian Symposium on Artificial Intelligence. Fortaleza, 1994. **Proceedings**. Fortaleza, LIA-UFCE, 1994. p. 27-41.
- [SIC94a] SICHMAN, J.S.; DEMAZEAU, Y.; CONTE, R.; CASTELFRANCHI, C. A Social Reasoning Mechanism Based on Dependence Network. In: 11th European Conference on Artificial Intelligence, Amsterdam, 1994. **Proceedings**. Chichester, John Wileys & Sons, 1994. p.188-192.
- [SIC95] SICHMAN, J. S. **Du Raisonnnement Social Chez les Agents: Une Approche Fondée sur la Théorie de la Dépendance**. Grenoble, 1995. 282 p. Tese (Doutorado) - Institut National Polytechnique de Grenoble.

- [SIC95a] SICHMAN, J.S.; DEMAZEAU, Y. Exploiting social reasoning to enhance adaptation in open multi-agent systems. In: Jacques Wainer and Ariadne Carvalho editors, *Advances in AI - 12th Brazilian Symposium on Artificial Intelligence*, Campinas, 1994. **Proceedings**. Germany, Springer-Verlag, 1995. p.253-263. v. 991. (Lecture Notes in Artificial Intelligence - Subseries of Lecture Notes in Computer Science)
- [SIC96a] SICHMAN, J. S. "On Achievable Goals and Feasible Plans in Open Multi-Agent Systems" *Proceedings 1st Ibero-American Workshop on DAI/MAS*. **Proceedings**. Xalapa, Mexico, October 1996.
- [SIC96b] SICHMAN, J. S.; DEMAZEAU, Y. "A Model for the Decision Phase of Autonomous Belief Revision in Open Multi Agent Systems" **Journal of the Brazilian Computer Society**, v.3, n.1, p.40-50, 1996.
- [SIC98] SICHMAN, J. S. "DEPINT: Dependence-Based Coalition Formation in an Open Multi-Agent Scenario" **Journal of Artificial Societies and Social Simulation**, v.1, n.2, 1998 /<http://www.soc.surrey.ac.uk/JASSS/1/2/3.html/>.
- [SIN96] SINGH, M.P. Semantical considerations on some primitives for agent specification. In: Wooldridge, M.; Müller, J. P.; Tambe, M. *Intelligent Agents Volume II – Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*. **Proceedings**. Lecture Notes in Artificial Intelligence. Springer-Verlag, 1996. *apud* [MAY96]
- [SMI80] SMITH, R.G. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. **IEEE Transactions on Computers**, v.29, n.12, p. 1104-1113, 1980.
- [SMI81] SMITH, R.G.; DAVIS, R. Frameworks for Cooperation in Distributed Problem Solving. **IEEE Transactions on Systems, Man, and Cybernetics**, v.11, n.1, p. 61-70, 1981.
- [SMI94] SMITH, D.C., et al. KidSim: Programming Agents Without a Programming Language. **Communications of ACM**. v. 37, n. 7. 1994. *apud* [TUR98]
- [SPR78] SPROULL, R.F.; COHEN, D. High-level Protocols. *Proceedings of the IEEE*. **Proceedings**. v. 66, p.1371-1386, 1978. *apud* [BON88]
- [STI92] STIRLING, W. Multi Agent Coordinated Decision-Making Using Epistemic Utility Theory. In: Castelfranchi, C.; Werner, E. editors, *Artificial Social Systems - 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, S. Martino al Cimino, 1992. **Proceedings**. Germany, Springer-Verlag, 1995. p.164-183. v. 830. (Lecture Notes in Artificial Intelligence - Subseries of Lecture Notes in Computer Science)

- [TAY91] TAYLOR, D. **Object-Oriented Technology: A Manager's Guide**. Reading, MA: Addison Wesley, 1991. *apud* [ERI98]
- [TUR98] TURBAN, E.; ARONSON, J.E. **Decision Support Systems and Intelligent Systems**. 5a. edição. New Jersey, Prentice-Hall Inc., 1998. Cap 19, p 720 - 754: Intelligent Agents and Creativity.
- [WIT92] WITTIG, T. **ARCHON: An Architecture for Multi-Agent Systems**. London: Ellis Horwood. 1992. *apud* [NWA96]
- [WON99] WONG, D.; PACIOREK, N.; MOORE, D. Java-based Mobile Agents. *Communications of ACM*. v. 42, n. 3, 1999.
- [WOO95] WOOLDRIDGE M.; JENNINGS, N. R. Intelligent Agents: Theory and Practice. **Knowledge Engineering Review**, v.10. n.2, p.115-152. 1995.
- [WOO95a] WOOLDRIDGE M.; JENNINGS, N. R. Intelligent Agents. **Lecture Notes in Artificial Intelligence 890**, Heidelberg: Springer-Verlag, 1995. *apud* [NWA96]
- [WOO96] WOOLDRIDGE M.; MUELLER, J. P.; TAMBE, M. Intelligent Agents II. **Lecture Notes in Artificial Intelligence 1037**, Heidelberg: Springer-Verlag, 1996. *apud* [NWA96]

Apêndice A – UML (Unified Modeling Language)

A.1 Introdução

Em todas as disciplinas da engenharia, modelos são fundamentais. Quando se deseja construir algo, tais modelos (desenhos, esquemas, equações) são projetados para descrever a aparência e o comportamento do produto. O produto pode ser uma casa, uma máquina, um novo departamento de uma companhia, ou um *software*.

Há tempos, discute-se sobre a “crise do *software*” que é baseada no fracasso dos projetos de *software*, pois estes não atendem às necessidades e os requisitos dos usuários, além de exceder prazos e custos. Atualmente, além destes fatores, outros prejudicam a produtividade, como as novas tecnologias (programação orientada a objetos, programação visual, ambientes cliente/servidor, etc.) e as exigências no desenvolvimento de software (alinhamento estratégico, conectividade, tecnologia de informação, flexibilidade, etc.).

Esses problemas no desenvolvimento de sistemas ocorrem porque os projetos iniciam a sua programação muito cedo, concentrando enormes esforços e tempo na codificação. Isto ocorre principalmente porque a maioria dos gerentes sente-se mais segura ao ver alguém programando (“sensação” de produção) do que modelando de modo abstrato o sistema.

Porém, à medida que a indústria de software tem se deparado com sistemas complexos em ambientes distribuídos, a necessidade de se modelar o software antes de implementá-lo tem se tornado um ponto importante no desenvolvimento de sistemas.

A UML (Unified Modeling Language) [BOO97] foi desenvolvida para tentar solucionar alguns dos problemas acima mencionados. O seu objetivo, portanto, é criar uma linguagem única de modelagem que possa ser utilizada por homens e máquinas de tal forma que seja

possível modelar sistemas e não somente o software com os conceitos de orientação a objetos.

A UML está baseado no conceito de modelo³⁰. Assim, transportando a concepção de modelo para o desenvolvimento de sistemas, o analista deve criar modelos que descrevam todos os diferentes aspectos do sistema a ser desenvolvido. O modelo possui visões, onde cada uma descreve um aspecto específico do sistema. Assim a cada fase do desenvolvimento de software são adicionados detalhes aos modelos. Com a UML pode-se modelar o funcionamento estático e dinâmico do sistema, denominados respectivamente de estrutura estática e comportamento dinâmico.

A UML não é uma metodologia, pois o seu objetivo não é dizer como deve ser o processo de desenvolvimento de software. Os modelos são os produtos gerados quando se segue um método. Um modelo é expresso através de uma linguagem de modelagem. Uma linguagem de modelagem possui uma notação (símbolos utilizados nos modelos) e um conjunto de regras. As regras são sintáticas, semânticas e pragmáticas.

A sintaxe define os símbolos e como eles se combinam entre si na linguagem. Podemos comparar a sintaxe com as palavras na linguagem natural, é importante saber a ortografia correta das palavras e como podemos combiná-las para formar uma frase.

A semântica define o significado de cada símbolo e como ele pode ser interpretado de modo isolado e quando combinado com os demais símbolos. No nosso exemplo, trata-se do significado das palavras dentro do contexto.

As regras pragmáticas definem a intenção dos símbolos dentro do modelo, tornando o modelo compreensível por todos os que o analisam. Este corresponde na linguagem natural a um bom texto. Livros de estilos de redação ditam esse tipo de pragmatismo.

³⁰ Modelo é a descrição de alguma coisa. O projetista precisa criar modelos que descrevam os vários aspectos do produto.

Tendo estas definições, as partes que compõem a UML são as seguintes:

- **Elementos do modelo** são os conceitos utilizados nos diagramas.
- **Diagramas** são gráficos que mostram os elementos do modelo dispostos de forma a ilustrar um aspecto do sistema. Um diagrama é parte específica de uma visão.
- **Visões** mostram os diferentes aspectos do sistema que é modelado. A visão não é um gráfico, mas uma abstração composta por vários diagramas.

A.2 Elementos do Modelo

Os elementos dos modelos são os conceitos utilizados nos vários diagramas da UML. Esses diagramas compõem os vários aspectos do sistema (visões) e o conjunto desses aspectos constitui o modelo do sistema. Os elementos dos modelos são definidos pela sua semântica, definição formal do elemento e um símbolo que o represente. Um elemento pode existir em tipos diferentes de diagramas, mas há regras e representações diferentes para os elementos em cada tipo de diagrama. A representação gráfica dos principais elementos se encontra na Figura A.1, enquanto a sua semântica encontra-se na Tabela A1.

Elemento	Semântica
Objeto	É uma entidade que combina o comportamento com atributos e estados relevantes. Uma coleção de dados com instruções relevantes para manipular estes dados.
Classe	A declaração e definição geral de uma coleção idêntica de objetos. É a especificação dos dados necessários para algum instante de sua vida. Representa a interface, ou seja, contém uma lista e a implementação das operações (métodos) disponíveis.
Mensagem	Um comando ao objeto. Enviar uma mensagem a um objeto significa ativar a execução de uma de suas operações.
Atributos	São os dados que queremos manter sobre o objeto ou a classe como um todo.
Métodos	São operações que permitem que os objetos possam exibir o comportamento esperado. Os métodos numa classe descrevem o que a classe pode fazer, ou seja, quais os serviços que ela oferece.
Herança	É a capacidade que uma classe tem de herdar atributos, métodos, interfaces e/ou implementações de outra classe.

Agregação	É um relacionamento entre as classes, significa que uma classe é composta por outras classes.
Associação	A capacidade de uma classe poder referir a outra classe. Pode ser bidirecional, define o relacionamento saber sobre.
Pacote	É um mecanismo geral para organizar elementos em grupos semânticamente relacionados. A arquitetura do sistema tradicionalmente é constituída de partes. Na UML cada parte é chamada de pacote.
Ator	São entidades externas que interagem com o sistema de alguma forma.
Caso de Uso	Especificação de uma funcionalidade completa do sistema, envolvendo os atores interessados. Deve sempre terminar com o resultado que deve ser dado ao ator.
Estado	É o resultado de atividades prévias realizadas pelo objeto, e é geralmente determinada pelos valores dos atributos e relacionamentos com outros objetos.
Processador	Representação da máquina que é utilizada para o processamento do sistema. Por exemplo: Servidor Risc.
Dispositivo	Representação da máquina que é utilizada para auxiliar no processamento do sistema. Por exemplo: impressora.
Componente	Trata-se da implementação numa arquitetura física dos conceitos e funcionalidades definidos na arquitetura lógica. São os arquivos que contém as implementações no ambiente de desenvolvimento.

Tabela A.1 – Semântica dos elementos básicos da UML.

A.3 Diagramas

Diagramas são gráficos que mostram os elementos do modelo dispostos de tal forma a fim de ilustrar um aspecto do sistema. O modelo do sistema típico tem uma série de diagramas de cada tipo. O diagrama é uma parte específica de uma visão. Alguns diagramas podem fazer parte de mais de um tipo de visão. A seguir temos os diagramas que compõe a UML.

- Diagrama de casos de uso

O diagrama de casos de uso [JAC92] descreve as entidades externas (ator) e suas interações com as funcionalidades (casos de uso) que o sistema oferece. Neste caso, interagir com o sistema, significa que o ator troca informações com o sistema, assim atores definem papéis que os usuários do sistema podem desempenhar, sejam eles humanos ou não. Como exemplo de um usuário não humano, considere-se uma máquina de laboratório de análises

clínicas que realiza certos exames e que pode alimentar um outro sistema responsável pela divulgação dos resultado dos exames. A máquina é considerada um usuário porque interage com o sistema de alguma forma.

Figura A1 - Principais elementos do modelo na UML

As funcionalidades do sistema são representados pelos casos de uso. O caso de uso sempre é iniciado por um ator e deve retornar um resultado (valor) a algum dos atores que interagem com o caso de uso.

O diagrama de casos de uso inclui todos os atores e todos os casos de uso indentificados para cada ator (Figura A2).

Sendo o caso de uso uma sequência de transações relacionadas, como ocorrem no diálogo de um ator com o sistema, uma descrição dos casos de usos representados no diagrama de casos de uso complementa o entendimento do modelo. Portanto a descrição do caso de uso é o detalhamento de cada caso de uso identificado no diagrama de casos de uso. Ele pode ser descrito em linguagem natural, numa lista de eventos (figura A3), ou utilizando o diagrama de atividades³¹.

Figura A2 - Exemplo de um diagrama de caso de uso de um sistema para um restaurante.

- O cliente pede o prato escolhido.
- Caso o prato escolhido não está sendo servido, o cliente escolhe outro prato.
- O cliente pede a bebida escolhida.
- Caso a bebida escolhida não está sendo servida, o cliente escolhe outra bebida.
- Garçon anota o pedido.
- Garçon leva o pedido para a cozinha.
- Cliente fica esperando pelo pedido.

³¹ O diagrama de atividades é explicado com mais detalhes nos próximos itens.

Figura A3 - Exemplo da descrição do caso de uso - Cliente realiza pedido.

- Diagrama de classe

O diagrama de classe [BOO94] descreve a estrutura estática do sistema em termos de classes e seus relacionamentos. Uma classe no diagrama de classe pode ser implementada diretamente numa linguagem orientada a objetos. O diagrama de classe pode ser utilizado para visualizar os papéis e responsabilidades das entidades que fornecem o comportamento do sistema e capturar a estrutura das classes que formam a arquitetura do sistema. Num diagrama de classe (Figura A4) são possíveis os seguintes relacionamentos³²: herança, agregação e associação.

Figura A4 - Exemplo de um diagrama de classe de um sistema para um restaurante.

³² As definições dos principais relacionamentos encontram-se no item A2, para maiores detalhes recomenda-se a bibliografia [BOO94] e [ERI98].

- Diagrama de objetos

O diagrama de objetos [BOO97] é uma variante do diagrama de classe e demonstra as classes instanciadas num determinado instante. É a representação gráfica dos objetos durante sua execução num determinado momento no tempo (Figura A5). É considerado um diagrama de suporte, ajuda a entender e exemplificar um diagrama de classe complexo, demonstrando como as instâncias e os seus relacionamentos se comportam num determinado cenário.

Figura A5 - Exemplo de um diagrama de objeto do cenário cliente realiza o pedido.

- Diagrama de estados

O diagrama de estados complementa a descrição das classes e representa o modelo dinâmico do sistema. O diagrama de estados é uma representação gráfica dos diferentes

estados que uma classe pode assumir, incluindo as condições para mudança de estados e as opções realizadas na transição.

Estado é o resultado de atividades prévias realizadas pelo objeto, e é geralmente determinado pelos valores dos atributos e relacionamentos com os outros objetos. O objeto muda de estado quando um evento acontece.

O diagrama de estados representa dois momentos dinâmicos do objeto: a interação e a mudança do estado interno do objeto. A interação relata qual evento deve ocorrer para que haja a mudança do estado interno do objeto. A mudança do estado interno do objeto pode ser a alteração de um atributo interno do objeto. Por exemplo, quando uma mesa está disponível (evento) está no estado de liberada (valor do atributo ocupado é não) e quando está ocupada (evento), passa para o estado de ocupada (valor do atributo ocupado é sim) (Figura A6).

Figura A6 – Diagrama de estados da classe mesa do sistema do restaurante.

- Diagrama de colaboração

O diagrama de colaboração demonstra a colaboração dinâmica entre os objetos. É utilizado para representar a execução de um cenário, onde o objetivo principal é estudar as trocas de mensagens entre os objetos, ou seja, qual objeto colabora com outro objeto em termos de oferecer os seus serviços. O diagrama de colaboração (Figura A7) é semelhante ao diagrama de objetos, a exceção é que temos setas que indicam a comunicação entre os objetos.

Figura A7 – Diagrama de colaboração no cenário do cliente realizando um pedido.

- Diagrama de sequência

O diagrama de sequência representa a colaboração dinâmica entre os objetos. O aspecto mais importante neste diagrama é a sequência das mensagens. Diferente do diagrama de colaboração que está interessado em estudar os objetos que interagem entre si, este diagrama estuda a sequência da comunicação entre os objetos e a sua sincronização.

Neste diagrama, os objetos são dispostos em linhas e as mensagens são representadas como setas horizontais e dispostas na sequência em que são trocadas entre os objetos. O tempo de processamento em cada objeto é representado por uma coluna não cheia, representando quando uma mensagem é síncrona ou assíncrona (Figura A8).

Figura A8 – Diagrama de sequência do cenário cliente realizando pedido.

- Diagrama de atividades

O diagrama de atividades demonstra o fluxo sequencial das atividades. O diagrama de atividades é tipicamente utilizado para descrever as atividades realizadas durante uma tarefa. Este diagrama pode ser utilizado para descrever uma interação entre os objetos como um caso de uso. É como se fosse desenvolvido um fluxograma macro de uma tarefa (Figura A9).

Figura A9 – Parte do diagrama de atividades do caso de uso cliente realizando pedido.

- Diagrama de componentes

O diagrama de componentes representa a estrutura física da implementação em termos de código. Este diagrama descreve os componentes de *software* e as suas dependências (Figura A10). Os componentes são a implementação numa arquitetura física dos conceitos e funcionalidades definidos na arquitetura lógica (classes, objetos, seus relacionamentos e colaborações), ou seja, são os arquivos que contém as implementações no ambiente de

desenvolvimento. Os tipos de componentes são: fonte (código fonte, o arquivo que contém a implementação de uma ou mais classes), binário (código objeto ou uma biblioteca, é o resultado da compilação) e executável (código executável, resultado da linkedição, é executado pelo processador).

Figura A10 – Diagrama de componentes do sistema do restaurante.

- Diagrama de distribuição

O diagrama de distribuição descreve a arquitetura física do hardware e do software no sistema. Representa os nós, os dispositivos e as conexões entre eles, onde sistema será implantado (Figura A11). É mais utilizado para representar a arquitetura de rede para um sistema distribuído e/ou cliente-servidor.

Figura A11 – Diagrama de distribuição do sistema do restaurante.

- Diagrama de pacotes

O Diagrama de pacotes representa a interação entre os módulos de um sistema (figura A12). Não é propriamente um diagrama da UML, mas tem sido utilizada pelos analistas de sistemas orientados a objetos. Este diagrama está presente em ferramentas CASE que possuem a notação da UML, inclusive na própria Rational Rose Corporation® o qual pertence aos criadores da UML.

Figura A12 – Diagrama de pacotes do sistema do restaurante.

A.4 Visões

Nem sempre um único gráfico obtém êxito em definir um sistema inteiro. Um único gráfico não captura todas as informações necessárias para descrever um sistema. Um sistema é definido pelos seus aspectos funcionais (estrutura estática e interações dinâmicas), não funcionais (requisitos, recursos, etc.) e organizacionais (cronograma, organograma, etc.). Portanto, um sistema deve ser descrito de forma que cada aspecto seja representado separadamente. Na UML, tal aspecto é capturado através das várias visões que podemos ter do sistema. A seguir temos as visões existentes na UML.

- Visão de casos de uso

A visão de casos de uso demonstra as funcionalidades do sistema pela percepção dos atores externos. É responsável pelo modelo de casos de uso, o qual tem a finalidade de:

- * decidir e descrever os requisitos funcionais do sistema;
- * fornecer uma descrição clara e consistente do que o sistema irá realizar;
- * fornecer a base para os demais modelos e visões;
- * fornecer a base para os testes e validação do sistema (ver o item teste em A5);
- * fornecer subsídios para encontrar as classes e as operações do sistema.

A visão de casos de uso se utiliza do diagrama e descrição de casos de uso. É construído através de processos iterativos de discussões entre os desenvolvedores do sistema e os usuários. Essas reuniões devem ter o objetivo de que todos cheguem a um denominador comum em relação aos requisitos do sistema.

- Visão lógica

A visão lógica demonstra como a funcionalidade é projetada dentro do sistema, na sua forma estática e dinâmica. Ao contrário da visão de casos de uso, a visão lógica estuda as estruturas internas que compõem o sistema, seus relacionamentos e interações. A estrutura estática é descrita pelos diagramas de classes e objetos. A estrutura dinâmica é representada pelos diagramas de colaboração, sequência, estados e atividades.

- Visão de componente

A visão de componente demonstra a organização dos componentes codificados. É composta pelo diagrama de componentes. Os componentes, que são os diferentes tipos de módulos codificados, são descritos na sua estrutura e dependências. Informações adicionais, como alocação de recursos ou outras informações administrativas, podem ser incorporadas nesta visão, porém não existem diagramas específicos para este fim. O diagrama de pacote adiciona valor ao prover um mecanismo geral para organizar os componentes em grupos semanticamente relacionados.

- Visão concorrente

A visão concorrente demonstra a divisão do sistema em processadores e processos, trabalhando com a sincronização destes últimos. Este aspecto, que é uma propriedade não funcional do sistema, assegura um uso adequado dos recursos, execução paralela e prevê eventos assíncronos no ambiente. É representado pelo diagrama de sequência e de estados.

- Visão dos recursos

A visão de recursos demonstra os recursos do sistema na arquitetura física. É representado pelo diagrama de distribuição. Esta visão inclui um mapa que descreve como os componentes são alocados na arquitetura física, por exemplo, qual programa ou objeto é executado em que máquina.

A.5 Metodologia e a UML

Os diagramas da UML podem ser utilizados nas cinco fases do desenvolvimento do sistema, como proposto por [ERIC98] [BEL98]. Esta metodologia foi utilizada no desenvolvimento do sistema CENINT. A seguir apresentamos a metodologia sugerida:

1. Análise de Requisitos - Utiliza a visão de casos de uso para modelar os requisitos do usuário. Os produtos nesta fase são: o diagrama de casos de uso e a descrição do caso de uso.
2. Análise - Responsável pela primeira abstração (classes e objetos) e mecanismos que estão presentes no domínio do problema. Na análise somente as classes que representam o domínio do problema (conceitos do mundo real), são modelados. Não devem ser modelados as classes de implementação que definem detalhes e soluções do software do sistema. Inicialmente desenvolvem-se os componentes do domínio do problema (classes e objetos) e depois modela-se a interação entre eles. Os produtos

nesta fase são: diagrama de classes da análise, diagrama de estados dos objetos da análise e o diagrama de sequência/colaboração da análise.

3. Projeto - O resultado da análise é expandido para a solução técnica. Na fase de projeto, novas classes são adicionadas ao modelo para prover a infraestrutura técnica: interface do usuário, acesso a dados, interface para outros sistemas, etc.. Refinamentos sucessivos no diagrama de sequência ajudam a encontrar as novas classes que devem ser adicionadas no diagrama de classe. Os produtos desta fase são: diagrama de classe (final), diagrama de sequência (final), diagrama de pacotes, diagrama de distribuição e a especificação detalhada das classes.
4. Implementação - As classes da fase de projeto são codificadas. Na implementação, recomenda-se utilizar uma linguagem orientada a objetos, pois dependendo da capacidade da linguagem será mais fácil ou não codificar as classes. A programação é uma fase separada durante a qual os modelos são convertidos em código. O produto nesta fase são: diagrama de componentes e o código fonte do sistema.
5. Teste - Os diferentes testes são realizados com a ajuda dos diferentes diagramas construídos ao longo do desenvolvimento de software. Os testes são compostos de:
 - * Teste de unidade - classe individual ou grupo de classes (diagrama de classe)
 - * Teste de integração - verificar se as classes cooperam entre si como o especificado (diagrama de classe, diagrama de sequência e/ou colaboração).
 - * Teste do sistema - o sistema deve ser visto como uma caixa preta, quando se verifica se ele possui as funcionalidades finais esperadas pelo usuário (diagrama de uso de casos)
 - * Teste de aceitação - semelhante ao teste de sistema, porém é realizado pelo usuário. O usuário verifica se o sistema satisfaz os seus requisitos. Deve ser apoiado no diagrama de uso de casos a fim de que a aderência aos requisitos seja respeitada.