

Lívia Lopes Azevedo

**AProSiMA – ambiente de resolução
cooperativa de problemas baseado em
simulação multiagente**

Lívia Lopes Azevedo

**AProSiMA – ambiente de resolução cooperativa de problemas
baseado em simulação multiagente**

Tese submetida ao Programa de Doutorado em Engenharia Elétrica da Universidade Federal do Estado do Espírito Santo, como requisito parcial para a obtenção do Grau de Doutor em Engenharia Elétrica – Automação.

Orientador: Prof. Dr. Crediné Silva de Menezes

Vitória, 21 de dezembro de 2007

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

	Azevedo, Livia Lopes, 1963-
A994a	AProSiMA – ambiente de resolução cooperativa de problemas baseado em simulação multiagente / Livia Lopes Azevedo. – 2007. 187 f. : il.
	Orientador: Crediné Silva de Menezes.
	Tese (doutorado) – Universidade Federal do Espírito Santo, Centro Tecnológico.
	1. Solução de problemas. 2. Agentes inteligentes (Software). 3. Simulação (Computadores). 4. Recuperação da informação. I. Menezes, Crediné Silva de. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.
	CDU: 621.3

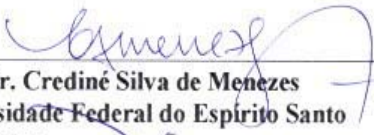
LIVIA LOPES AZEVEDO

**AProSIMA – AMBIENTE DE RESOLUÇÃO COOPERATIVA DE
PROBLEMAS BASEADO EM SIMULAÇÃO MULTIAGENTE**

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da
Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau
de Doutor em Engenharia Elétrica – Automação.

Aprovada em 21/12/2007.


COMISSÃO EXAMINADORA




Prof. Dr. Crediné Silva de Menezes
Universidade Federal do Espírito Santo
Orientador



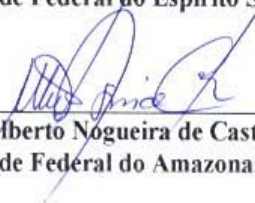
Prof. Dr. Davidson Cury
Universidade Federal do Espírito Santo



Profa. Dra. Maria Claudia Silva Boeres
Universidade Federal do Espírito Santo



Prof. Dr. Orivaldo de Lira Tavares
Universidade Federal do Espírito Santo



Prof. Dr. Alberto Nogueira de Castro Junior
Universidade Federal do Amazonas

A Felipe, meu filho precioso!

Dedicatória

A Lília, minha mãe, mulher guerreira e determinada. Sem seu apoio não teria chegado onde estou. Minha eterna admiração pela sua forma de encarar a vida.

A Felipe, meu filho amado, minha fonte de energia.

Agradecimentos

Minha eterna gratidão a Crediné, meu orientador. Que me recebeu, acreditou na minha vontade, encorajou-me a superar os desafios e me apoiou em todo o trabalho, sendo de fundamental importância para que eu pudesse alcançar os resultados tanto acadêmicos quanto pessoais.

Ao corpo docente da UFES pelos ensinamentos e pelas amizades que ganhei.

Aos amigos de curso, pelo incentivo, pela paciência e pela generosidade em compartilhar seus conhecimentos perante as minhas dificuldades, em especial: Camata, Carmem, Luciana, Paulinho, Thiago, Valéria, Zambon e ZéFram.

Os meus sinceros agradecimentos aos alunos: Alexandre, Diego, Lúcio, Fernando, Isau, Ricardo e Fellype que, determinadamente, fizeram seus projetos de final de curso baseados nas idéias propostas nesta tese.

À Lucia Bazarella, amiga dos momentos difíceis, obrigada pela paciência em ler as minhas versões e sinalizar os pontos nebulosos.

Ao amigo Pessoa, pelo incentivo e apoio nos momentos mais difíceis.

Aos meus amigos da UFMT que, de forma direta ou indireta, incentivaram a realização deste trabalho.

À Lia Jardim, pela amizade pelos ensinamentos e correção cuidadosa do texto.

A UFMT, pela minha liberação e a CAPES pelo apoio financeiro.

A Durães, pelo apoio em todas as empreitadas nas quais nos envolvemos.

Aos meus irmãos, que acompanharam o desenvolvimento de meu filho durante a minha ausência.

A minha mãe, pelo incentivo, pelo apoio e, principalmente, por cuidar do meu filhote durante este período de curso.

Ao meu pai, pela forma simples de ver a vida. Orgulho-me de ser sua filha.

Acima de tudo, agradeço a Deus, pois sem Ele nada seria possível.

Epígrafe

“Valeu a pena? Tudo vale a pena

Se a alma não é pequena.

Quem quer passar além do Bojador

Tem que passar além da dor.”

Fernando Pessoa
(Mar português)

Sumário

<i>Capítulo 1</i>	20
1 Introdução	20
1.1 Questão de investigação	24
1.2 Contribuição desta tese	26
1.3 Processo de investigação e objetivos	28
1.4 Metodologia	28
1.5 Organização deste trabalho	30
<i>Capítulo 2</i>	31
2 Sistemas complexos	31
2.1 2.1 Sistemas Multiagente (SMA)	35
2.1.1 Agentes cognitivos	36
2.1.2 Agente reativo	37
2.1.3 Aplicação de Sistemas Multiagente (SMA)	38
2.2 Simulação multiagente	40
2.2.1 Uso de simulação multiagente	41
2.2.2 Por que usar simulação multiagente?	43
2.2.3 Requisitos para a modelagem de simulação multiagente	46
<i>Capítulo 3</i>	48
3 Ambientes de simulação multiagente	48
3.1 StarLogo	49
3.2 Swarm	51
3.3 Repast	54
3.4 AgentSheets	55
3.5 NetLogo	57
3.6 Klik&Play	60

3.7	SIMULA	62
3.8	Considerações sobre as ferramentas	65
3.9	Por que desenvolver uma nova ferramenta para modelagem multiagente?.....	66
<i>Capítulo 4</i>		69
4	Resolução cooperativa de problemas	69
4.1	Situando o problema	73
4.2	Formação do grupo	75
4.2.1	Ferramenta QSabe	75
4.2.2	Recuperando informação (máquinas de busca).....	76
4.2.3	Redes sociais (<i>Social Network</i>)	77
4.2.4	Uma alternativa para a formação de grupos para o trabalho cooperativo	78
4.3	Ferramentas para resolução cooperativa de problemas na internet	80
4.4	Ferramentas de apoio ao trabalho cooperativo	83
4.4.1	Groove (atual Microsoft® Office Groove®).....	83
4.4.2	GroupSystem	84
4.4.3	Lotus Notes (atual Lotus Domino)	85
4.4.4	AmCorA	86
4.4.5	CaMILE - Collaborative and Multimedia Interactive Learning Environment... 87	
4.5	Considerações sobre os sistemas	88
<i>Capítulo 5</i>		91
5	O ambiente AProSiMA	91
5.1	Descrição do ambiente AProSiMA.....	92
5.2	Ferramentas do ambiente AProSiMA.....	93
5.2.1	Ferramenta para criação de Grupo.....	98
5.2.2	Ferramenta QSabe2	99
5.2.3	Ferramenta Esquematizador	101
5.3	Ferramenta de comunicação	104

5.3.1	Ferramenta Mural	105
5.4	Ferramenta Gerenciador.....	106
5.4.1	Sobre o funcionamento da API Lucene.....	109
5.4.2	Recuperação de informação	110
5.4.3	Atuação da ferramenta Gerenciador	112
5.5	Ferramenta de simulação – NetPlay	114
5.5.1	Linguagem de programação	115
5.5.2	Ambiente de desenvolvimento	116
5.5.3	Descrição dos Modelos.....	117
5.5.4	Definição dos agentes.....	119
5.5.5	Definição de variáveis	120
5.5.6	Regras de comportamento	120
5.5.7	Ferramentas de monitoramento	122
5.5.8	Realização dos Modelos	122
5.5.9	Ferramenta de análise gráfica	123
<i>Capítulo 6</i>		124
6	Três implementações para um mesmo modelo – comparação de ferramentas.....	124
6.1	Situando o modelo	124
6.2	Modelagem Repast.....	124
6.3	Modelo NetLogo.....	127
6.4	Modelo NetPlay	129
6.5	Considerações sobre as ferramentas	133
6.6	Características adicionais.....	135
<i>Capítulo 7</i>		137
7	Aplicação da ferramenta de simulação - NetPlay.....	137
7.1	Abordagem analítica x abordagem simulação multiagente	138
7.2	Elementos observados durante o curso	145

7.3	Atividades durante a disciplina	146
7.4	Considerações sobre a disciplina	147
7.4.1	Avaliação da ferramenta	148
<i>Capítulo 8</i>		155
8	Conclusões	155
8.1	Com relação à sensibilização para o estudo de sistemas complexos	158
8.2	Frutos desta tese	159
8.3	Trabalhos futuros	161
Referências bibliográficas		163
Apêndice A		172
Apêndice B		180
Anexo 1		183

Índice de figuras

Figura 1 - Proposta Ambiente.....	27
Figura 2 - Visão geral de um sistema complexo	34
Figura 3 - Apresenta a arquitetura para o modelo de agente cognitivo, adaptada de (Russel, et al., 2004).....	37
Figura 4 - Apresenta a arquitetura para o modelo de agente reativo simples, adaptada de (Russel, et al., 2004). O agente reativo simples age de acordo com uma regra cuja condição corresponde ao estado atual definido pela percepção.....	38
Figura 5 - Essa é uma abordagem “bottom up” chamada de Modelagem baseada em agente - Agent-based modelling (ABM).....	39
Figura 7 - Interface gráfica StarLogo(direita) / centro de controle – escrita código (esquerda)	50
Figura 8 - Montagem do modelo em blocos (dir.) e visualização em 3D e inspeção comportamento agente (esq.).....	50
Figura 9 - Peças de encaixe para montagem modelo.....	51
Figura 10 - Controle do modelo no ambiente Swarm	52
Figura 11 - Simulação desenvolvida no ambiente Swarm	53
Figura 12 - Modelo de simulação ratoeiras – Swarm (acima: gráfico de comportamento do modelo; abaixo: visualização da seqüência de interações entre os agentes no decorrer do tempo).....	53
Figura 13 - Simulação desenvolvida no ambiente Repast.....	54
Figura 14 - Galeria de definição de agentes, caixa de condições, caixa de ações – AgentSheets	55
Figura 15 - Ferramenta de composição de regras.....	56
Figura 16 - Grade de distribuição dos agentes para executar a simulação	56
Figura 17 - Interface gráfica NetLogo.....	59
Figura 18 - Fragmento de código em NetLogo	59
Figura 19 - Editor de Eventos Klik&Play	61
Figura 20 - Editor de nível / cenário execução modelo.....	61

Figura 21 – Funcionalidades do ambiente SIMULA	62
Figura 22 - Janela de definição das regras de comportamento dos agentes	64
Figura 23 - Janela de execução do modelo de simulação - Exemplo Pets	65
Figura 24 - Mapa conceitual do processo de resolução cooperativa de problemas.....	72
Figura 25 - Modelo de interação e aquisição de conhecimento no QSabe.....	76
Figura 26 - Usuário apresenta uma questão de investigação - agente vai à busca de informação	79
Figura 27 - Documentos de interesse selecionado - agente relaciona pessoas e envia convite para formar grupo	80
Figura 28 - Pessoas aceitam convite para cooperar e forma grupo de trabalho cooperativo ...	80
Figura 29 –AProSiMA - Ambiente de resolução cooperativa de problemas baseado na simulação multiagente	92
Figura 30 – Macro funcionalidades do ambiente AProSiMA	95
Figura 31 - Diagrama de pacotes – relacionamento	96
Figura 32 – Macro funcionalidades da ferramenta para formação de Grupos	99
Figura 33 - Macro-funcionalidade da ferramenta QSabe2	100
Figura 34 - Ações do usuário da ferramenta QSabe2.....	101
Figura 35 – Sugestão de como deve ser um esquema	102
Figura 36 - Sugestão de como será a interface da ferramenta Esquematizador	103
Figura 37 – Ações do usuário para a ferramenta Esquematizador	104
Figura 38 - Ações do usuário para ferramenta comunicação	105
Figura 39 - Ações do usuário para ferramenta Mural.....	105
Figura 40 - Estrutura interna do Agente Fred.....	106
Figura 41 - Diagrama de caso de uso Gerenciador.....	107
Figura 42 - Diagrama de atividade - pesquisa Internet.....	108
Figura 43 - Utilização da API do Lucene	108
Figura 44 - Exemplo de um texto e um índice invertido construído sobre ele. As palavras são convertidas para minúscula e algumas não são indexadas (Baeza-Yates,1999).	111
Figura 45 – Estrutura interna do NetPlay	117

Figura 46- Metáfora do teatro - Diretor selecionando equipe para encenar peça.....	118
Figura 47 - Criação de agentes no NetPlay	119
Figura 48 - Composições (regras de comportamento)	121
Figura 49 - Visualização do modelo.....	122
Figura 50 - Interface gráfica do Eclipse para definição dos elementos do modelo.....	125
Figura 51 – Fragmento da criação da classe ContexCreator no Repast S para o modelo presa/predador	126
Figura 52 - Árvore de cenários e visualização gráfica	127
Figura 53 - Interface gráfica 3D e controle de parâmetros.....	127
Figura 54 - Definição de variável tipo slider.....	128
Figura 55 – Recorte código ambiente NetLogo.....	129
Figura 56 - interface gráfica NetLogo	129
Figura 57 - Definição de agente Ovelha - NetPlay.....	131
Figura 58 - Definição de variáveis (apresenta todas as variáveis criadas no modelo, operadores matemáticos e condições relacionadas à contagem dos agentes e suas características), basta o usuário selecionar quais delas usará para escrever a expressão.	131
Figura 59 - Composição para regras de comportamento	132
Figura 60 - composição “comeGrama” = alimentação ovelha	132
Figura 61 - Applet do modelo solicitado pelo NetPlay	133
Figura 62 - Controle de versões.....	135
Figura 63 - Dispositivo de análise gráfica	136
Figura 64 - Resultado da equação de Lotka - Volterra para o modelo presa x predador com populações iniciais de 50.000 e 1.250, respectivamente	141
Figura 65 – Três comportamentos diferentes para o mesmo conjunto de regras no modelo presa/predador	142

Índice de tabelas

Tabela 1 - Exemplos e relações entre modelos de sistemas complexos.....	33
Tabela 2 - Características relacionadas aos sistemas complexos na literatura	35
Tabela 3 – Requisitos para um ambiente de resolução cooperativa de problemas.....	81
Tabela 4 - Ferramentas de comunicação e interação.....	82
Tabela 5 - Avaliação entre os ambientes	135
Tabela 6 - Comportamento do sistema “presa / predador”	139
Tabela 7 - Comparação entre abordagens: analítica e de simulação	145
Tabela 8 - Questionário de avaliação da ferramenta NetPlay	150
Tabela 9 - Tabela de definição de agente – características de determinação	172
Tabela 10 - Tabela de descrição de comportamento	174
Tabela 11 - Tabela de definição de ambiente.....	175
Tabela 12 - Tabela de identidade dos agentes e definição de variáveis	175
Tabela 13 - Tabela de definição de elementos gráficos – ferramentas conceituais.....	176
Tabela 14 - Tabela de aspectos gerais – outros elementos	177
Tabela 15 - Modelos desenvolvidos durante disciplina	180

Acrônimos

ABM – Agent-based modelling

API – Application Programming Interface

AProSiMA – Ambiente de resolução cooperativa de problemas baseado em simulação multiagente

COSI - Complexity in Social Science,

CSCL – Computer Supported Collaborative Learning

CSCS – The Center for the Study of Complex Systems

CSCW – Computer Supported Cooperative Work

EBM – Equation-based Models

MIT – Massachusetts Institute of Technology

NECSI –New England Complex Systems Institute;

RI – Recuperação de informação

SMA – Sistemas Multiagente

SNC – Sistema Nervoso Central

UFES –Universidade Federal do Espírito Santo

VPL – Visual programming language

WWW - World Wide Web

WYSIWIS – What You See Is What I See

Resumo

Nos últimos anos o estudo de sistemas complexos tem atraído a atenção de pesquisadores em várias áreas do conhecimento, devido ao crescente interesse na forma de organização dos sistemas. Exemplos de sistemas complexos estão presentes numa variedade de situações que vivenciamos e têm sido amplamente estudados através da modelagem de simulação multiagente. Entretanto, existe uma lacuna entre os ambientes para estudo de sistemas complexos e os usuários que deles fazem uso. Esta tese apresenta a proposta de um ambiente, denominado AProSiMA, que possibilita a usuários leigos em programação construir seus próprios modelos de simulação multiagente. O ambiente tem ainda como proposta geral propiciar o processo de resolução cooperativa de problemas. Para tanto disponibiliza aos usuários: acesso à recuperação de informação e mecanismos para localizar especialistas que podem estar geograficamente distribuídos e convidá-los para trabalhar de modo cooperativo na busca de solução para o problema; espaço compartilhado para a discussão e formalização das estratégias de resolução de problemas; mecanismos de direcionamento de perguntas e respostas aos especialistas, formação de grupos cooperados e mecanismos de comunicação que suportem o trabalho cooperativo.

Palavras chave: *Resolução de problemas, sistemas multiagente, simulação multiagente, recuperação informação.*

Abstract

The complex systems studies has caught researchers' attention in several knowledge areas lately, because of increasing interest in the way of systems' organization. Examples of complex systems are presented in a variety of situation which we live and they have been largely studied through model of multiagent simulation. However, there is a gap between environments for complex systems studies and their users. This thesis shows a proposal to an environment named AProSiMA, which allows no experts users in program planning, build up their own model of multiagent simulation. The environment still presents like a general proposal to propose a process of cooperative problem solving. By the way it's available to the users: access to information recover and mechanisms to locate experts who may be geographically distributed and invite them to work in a cooperative system looking for a solution to the problem, shared place to discuss and formalize the problems' resolution strategies; mechanisms to direct questions and answers to the experts, cooperated groups formation and communication mechanisms which may support up the cooperative work.

Key-words: Problem solving, multiagent systems, multiagent simulation, information retrieval

Capítulo 1

1 Introdução

Nos últimos anos, tem sido crescente o interesse no estudo sobre a forma de organização dos sistemas. Observa-se que a forma como as formigas procuram alimento é determinada não pelas ordens de uma “formiga rainha”, mas pelas interações locais entre milhares de formigas trabalhadoras; um congestionamento de trânsito surge das interações locais entre vários veículos individuais; as relações entre milhares de compradores e vendedores dão origem ao mercado financeiro; no sistema imunológico, um exército de anticorpos procura por bactérias num ataque sistematicamente coordenado, sem a participação de qualquer “general” determinando o plano de guerra.

Seguindo as idéias anteriores, cientistas (representados por vários centros de pesquisas como: Santa Fe Institute; NECSI –New England Complex Systems Institute; COSI - Complexity in Social Science, CSCS – The Center for the Study of Complex Systems) têm constatado que sistemas como: colônia de insetos sociais, mercado financeiro, situações de trânsito de veículos e pessoas, sistema imunológico, entre outros, são determinados não por alguma autoridade centralizada, mas por interações locais dos componentes descentralizados. Segundo (Resnick, 1994), idéias sobre descentralização e auto-organização estão se espalhando pela sociedade e cada vez mais as pessoas estão escolhendo modelos descentralizados para descrever as organizações que elas constroem no mundo e para explicar as teorias que elas constroem sobre o mundo. Em todas as esferas do conhecimento há evidência de idéias descentralizadas.

Da mesma forma que houve uma transformação da era industrial para a era da informação, nosso modo de pensar sobre a sociedade e nosso papel nessa sociedade também sofre mudanças. Essas mudanças estão diretamente relacionadas à estrutura e dinâmica sociais, a crescente interdependência da economia global, dos sistemas sociais e das formas de governo. Mas, mesmo com a crescente influência da descentralização, há uma profunda resistência a tais idéias, pois as pessoas parecem ter um forte apego à forma centralizada de pensar (Resnick, 1994).

Quando as pessoas procuram modelos no mundo, elas geralmente assumem que há algum tipo de controle centralizado sobre ele. Esse modo de pensar está atrelado à visão

Newtoniana do universo que tem dominado o mundo da ciência (Resnick, 1994) (Jacobson, 2000). Newton descreveu o universo como uma máquina. Sendo essa regida por causas e efeitos lineares – uma engrenagem gira, faz uma segunda engrenagem girar, essa por sua vez faz uma terceira engrenagem girar, e assim por diante. Entretanto, em meados do século passado, a pesquisa científica em várias áreas, incluindo a física, biologia, dinâmica não-linear, antropologia, neurociências, sociologia tem desafiado a visão “Newtoniana” do universo (Kauffman, 1995; Pagels, 1988; Prigogine & Stengers, 1984 citado por (Jacobson, 2000)) (Gell-Mann, 1996) (Resnick, 1994). O modo de pensar a ciência em termos de modelo baseado em engrenagem e controle centralizado tem sido substituído por novas perspectivas que são inspiradas em modelos biológicos, ações descentralizadas, eventos probabilísticas e que mudam dinamicamente no decorrer do tempo.

Esse modo de pensar tem sido caracterizado como *complexidade* e os fenômenos aos quais essas idéias se aplicam denominam-se *sistemas complexos* (Bar-Yam, 1997) (Gell-Mann, 1996) (Resnick, 1994) (Jacobson, 2000). Por enquanto, definiremos como sistemas complexos aqueles que podem ser caracterizados pela interação de vários elementos ou agentes individuais (relativamente simples), que se auto-organizam num nível mais elevado de sistema mostrando propriedades emergentes e complexas, que não são passíveis de serem alcançadas pelos elementos individuais.

As abordagens de sistemas complexos em conjunção com o rápido avanço das tecnologias computacionais têm possibilitado aos pesquisadores estudarem aspectos do mundo real por meio de eventos e ações que têm múltiplas causas e conseqüências, e onde a ordem e a estrutura coexistem em diferentes escalas de tempo, espaço e organização. Segundo (Bar-Yam, 1997) (Gell-Mann, 1996) (Jacobson, 2000) (Jacobson, et al., 2006), dentro desse *framework* o comportamento crítico dos sistemas que foram sistematicamente ignorados ou simplificados pela ciência clássica podem agora ser incluídos como elementos básicos que respondem por muitos aspectos observados em nosso mundo.

Muitas áreas de pesquisa estudam fenômenos descentralizados e sistemas complexos, pois os conceitos subjacentes a esses fornecem uma nova abordagem para muitos problemas. Esses conceitos originaram-se dos esforços para compreender sistemas físicos, biológicos e sociais. Sistemas complexos estão presentes numa variedade de situações que vivenciamos e que, portanto, precisam ser estudados e compreendidos utilizando abordagens adequadas. Estamos motivados nesta pesquisa para iniciar um processo de sensibilização para o estudo de

sistemas complexos, mostrando que é possível e simples modelar situações nas quais os indivíduos têm comportamentos diferentes e complexos.

Muitos estudos têm mostrado que, tanto no público em geral quanto no âmbito escolar, não é fácil encontrar bons pensadores sobre sistemas cujo comportamento global emerge de interações entre partes compostas por regras simples – modelos descentralizados (Resnick, 1994) (Resnick, et al., 1993). Possivelmente pelo seu caráter não-linear, raramente estes sistemas são encontrados nas disciplinas do ensino médio (Jacobson, et al., 2006).

O comportamento não linear dos sistemas complexos significa que aproximações matemáticas, estatísticas ou probabilísticas são problemáticas, isto é, não há um conjunto de equações que podem relacionar e resolver as características do sistema. Entretanto, duas abordagens têm sido amplamente utilizadas no estudo de sistemas complexos: Redes Neurais e Modelagem Baseada em Agentes (simulação multiagente). Redes Neurais devido às ligações entre os componentes do sistema e a forma de interação entre eles (Bonabeau, 2002). Modelagem multiagente pelo fato de esses sistemas serem compostos por muitos agentes interagindo entre si e com o ambiente por meio de regras simples (Axelrod, 1997), (Bonabeau, 2002), (Drogoul, et al., 1992), (Railsback, et al., 2006), (Resnick, 1994), (Repenning, 1993). A simulação no computador tem sido bastante apropriada para descrever esses sistemas, pois de modo geral, a simulação consiste em reproduzir artificialmente uma situação ou fenômeno natural.

A construção de modelos de simulação multiagente conduz seu criador a identificar relações entre as partes componentes do sistema, estabelecer seus controles, avaliar as suas conexões, e perceber que algumas coisas podem ser facilmente entendidas e outras não. Embora existam muitas ferramentas para descrever modelos de simulação baseados em agentes, para muitos usuários, o desenvolvimento de tais modelos ainda representa um obstáculo. Segundo (Ropella, et al., 2002), (Minar, et al., 1996), (Railsback, et al., 2006), este problema ocorre, em grande parte, pela ausência de treinamento e habilidade no uso de software por pesquisadores (usuários) de muitos campos que usam a modelagem baseada em agente (por exemplo: biólogos, ecólogos, economistas, cientistas políticos, sociólogos, entre outros).

Além disso, pesquisas têm mostrado que o processo de criação de modelos, oposto ao simples uso de um modelo criado por outros, não apenas nutre a capacidade de construção do modelador, mas também possibilita desenvolver um maior entendimento dos conceitos que

estão embutidos no próprio modelo (diSessa, 1987) e (Confrey, Doerr, 1994; Talsma, 2000 apud (Colella, et al., 2001)). O processo de criação do modelo gera questionamentos e esses motivam a busca por soluções além de sugerirem a participação de outras pessoas num processo de construção cooperativa.

Diante desses fatos um dos aspectos que ainda requerem estudos e desenvolvimentos é a de construção dos modelos, segundo a ótica de um usuário (observador de sistemas) que seja leigo em programação. Em geral, a definição de agentes e de famílias¹ de agentes, suas propriedades relevantes, as interações interfamiliares e intrafamiliares, requerem a escrita de código em alguma linguagem de programação, fator que impossibilita que muitos usuários utilizem os ambientes existentes.

Outro aspecto que merece destaque na construção de modelos de simulação de sistemas complexos é a quantidade de variáveis e de relações que definem o modelo, geralmente percebidas por meio da observação e análise do fenômeno real. Capturar todos os componentes do modelo de simulação é uma tarefa difícil, pois quanto maior o número de elementos e mais detalhado for o modelo a ser simulado, maior será a quantidade de variáveis que co-existem simultaneamente, fazendo com que um fator adicional cresça em importância: *a cooperação*. Além disso, o desenvolvimento de modelos de simulação multiagente no âmbito de sistemas complexos, que retratem o problema de forma mais próxima possível do real, exige da equipe de modeladores conhecimento de domínio específico. Um obstáculo para a realização desse trabalho é o fato de que geralmente os especialistas no assunto estão geograficamente distribuídos.

Com o advento e a atual expansão da *World Wide Web (www)* ao oferecer um suporte de comunicação e tratamento à distância da informação, aproxima os especialistas facilitando o estabelecimento de equipes de trabalho para a resolução cooperativa de problemas. Estudos têm mostrado que, particularmente, quando as pessoas que têm condições de resolver os problemas estão dispersos, o sucesso depende criticamente da sua habilidade de usar efetivamente ferramentas *groupware*² (Nunamaker, 1999).

¹ Usamos o termo famílias de agentes para especificar grupos de agentes com as mesmas características.

² Usaremos a definição de (Ellis, et al., 1991) *groupware* são, “sistemas baseados em computador que apóiam os grupos de pessoas engajadas numa tarefa comum (ou meta) e que fornece um ambiente de interface compartilhada.”; (tradução nossa); e de (Zwass, 1998 apud (Tommarello, 2002)) que usa o termo *groupware* para referir o tipo de ambiente de software necessário para apoiar o trabalho da equipe, cujos membros colaboram sobre uma network.

É sabido que muitas pessoas em vários setores da sociedade estão empenhadas na busca de soluções para os problemas de sistemas complexos. Muitas dessas pessoas objetivam a solução como um benefício comum para toda a sociedade. Entretanto, pode ocorrer que a busca pela solução de um problema comum esteja acontecendo de forma isolada, em locais geograficamente distribuídos e com os interessados desconhecidos entre si. Entendemos que seria importante que essas pessoas tomassem conhecimento do que cada um está realizando, que descobertas têm feito, que fatos novos têm levantado, etc., de modo a contribuir com seus resultados em busca da solução do problema. Pessoas com esse propósito certamente se interessariam na criação de ambientes cooperativos de apoio à resolução de problemas complexos. Torna-se necessário, então, disponibilizar um local na web onde esses pesquisadores possam compartilhar seus conhecimentos.

Apesar de já existirem muitas iniciativas que buscam apoiar a resolução cooperativa de problemas, a questão ainda está em aberto, principalmente quando se trata de problemas que mobilizam recursos humanos de diferentes partes do país ou do mundo. Sabemos que o uso integrado de técnicas de Recuperação de Informação e de Inteligência Artificial ainda é pouco explorado para recuperar casos, reunir pessoas e distribuir atribuições em um ambiente cooperativo de apoio à resolução de problemas complexos usando simulação multiagente. É nesse sentido que pretendemos contribuir com esta tese.

1.1 Questão de investigação

Conforme relatado, o uso de modelos de simulação multiagente para o estudo de sistemas complexos tem crescido rapidamente em muitas áreas. Esse crescimento pode ser atribuído, primeiramente, à capacidade de esses modelos enfocarem o problema, capacidade que os sistemas convencionais (modo analítico) não alcançam. Segundo, devido ao crescimento, em número e qualidade, dos ambientes de software projetados para esta finalidade.

Apesar da existência e disponibilidade de vários ambientes para modelagem de simulação multiagente, o desenvolvimento de modelos ainda representa um obstáculo para muitos pesquisadores, professores e estudantes. Pois a exigência de conhecimento em linguagem de programação imposta por esses ambientes (a maioria deles é projetada para uso profissional) impede tanto a exploração mais ampla do estudo de sistemas complexos, quanto a sua inserção no contexto educacional.

Assim, um dos aspectos que ainda requer estudos e desenvolvimentos é a de construção dos modelos, segundo a ótica de um usuário (observador de sistemas) que seja leigo em programação. Ainda carece de melhoras a forma de observação do funcionamento desses sistemas, por meio de ferramentas de acompanhamento e análise. Essas possibilitarão aos usuários avaliar as alterações do comportamento do modelo.

Além, disso, trazer a abordagem de sistemas complexos para o contexto mais próximo de nossa realidade educacional possibilitaria aos estudantes discutir questões referentes aos fenômenos, coletar dados relacionados, elaborar estratégias de solução fundamentadas nos modelos representantes do sistema, analisar as conseqüências advindas das tomadas de decisão e, principalmente, desenvolver seu próprio conhecimento num ambiente cooperativo de resolução de problemas.

Vale a pena ressaltar que a modelagem de um fenômeno complexo apresenta um grande número de variáveis, envolve abordagem interdisciplinar e, geralmente, necessita de participação cooperativa. Como os especialistas em um domínio específico encontram-se geograficamente distribuídos, faz-se necessário um mecanismo para agregar esses elementos, do mesmo modo, são necessárias ferramentas de comunicação que apóiem a interação entre os cooperadores no processo de resolução de problemas.

Se dispuséssemos de um sistema capaz de congrega pesquisadores de diferentes partes do mundo que pudessem compartilhar informações e trabalhar cooperativamente num ambiente onde não houvesse impedimento geográfico e temporal, certamente o alcance dos resultados seria consideravelmente aumentado e, conseqüentemente, todo o grupo de “resolvedores” usufruiria desse mérito.

Diante do exposto levantamos as seguintes questões de investigação:

- 1. Como sensibilizar as pessoas sobre a importância do estudo de sistemas complexos e como ajudar a ampliar a quantidade de adeptos da modelagem de simulação multiagente para a construção de modelos de sistemas complexos?**
- 2. Que componentes seriam necessários num ambiente web para apoiar a resolução cooperativa de problemas de sistemas complexos usando a simulação multiagente?**
- 3. Como estabelecer a interação entre pesquisadores, ou seja, como localizar e envolver esses pesquisadores num único local, de modo a trabalharem cooperativamente?**

1.2 Contribuição desta tese

A contribuição desta tese é, então, agregar estas duas grandes áreas em um só local: **resolução cooperativa de problemas e modelos de simulação multiagente no âmbito de sistemas complexos**, tornando-os acessíveis através da web. Disponibilizar um ambiente que:

- utilize ferramenta *groupware* para comunicação e interação entre os usuários e que seja capaz de selecionar e montar equipes de especialistas oferecendo todas as informações relevantes sobre o problema, suportando o gerenciamento da resolução do problema, compartilhando um local de trabalho – fórum de discussão - que não dependa da disposição geográfica ou temporal, ainda mais, armazenando as estratégias e as resoluções do problema (sucessos e fracassos) para consultas prévias, análises e viabilidade da resolução.
- utilize uma ferramenta para construção de modelos de simulação que tenha uma interface amigável e seja acessível aos usuários leigos em programação. Isto é, uma ferramenta que seja orientada para o domínio do problema e não para o entendimento de uma linguagem de programação (que utiliza bibliotecas e *frameworks* entendíveis aos programas). Assim, é importante disponibilizar uma ferramenta que faça uso de artefatos como: visualização, animação, áudio, diálogo, e experiências táteis³. Ao incorporar essas modalidades num novo modo de utilizar as ferramentas computacionais, estabelecem-se novas metáforas de programação que possibilitam aos leigos em programação conceituar os problemas em termos de noções concretas, fazendo analogias com conceitos familiares.

A Figura 1 apresenta um esboço geral da contribuição desta tese.

³ Disponibilizando meios que possibilitem a passagem de experimentos do virtual para o real, por exemplo: mecanismos de sensores e atuadores. Elementos que permitam aos usuários manipular objetos e definir ações a serem realizadas.

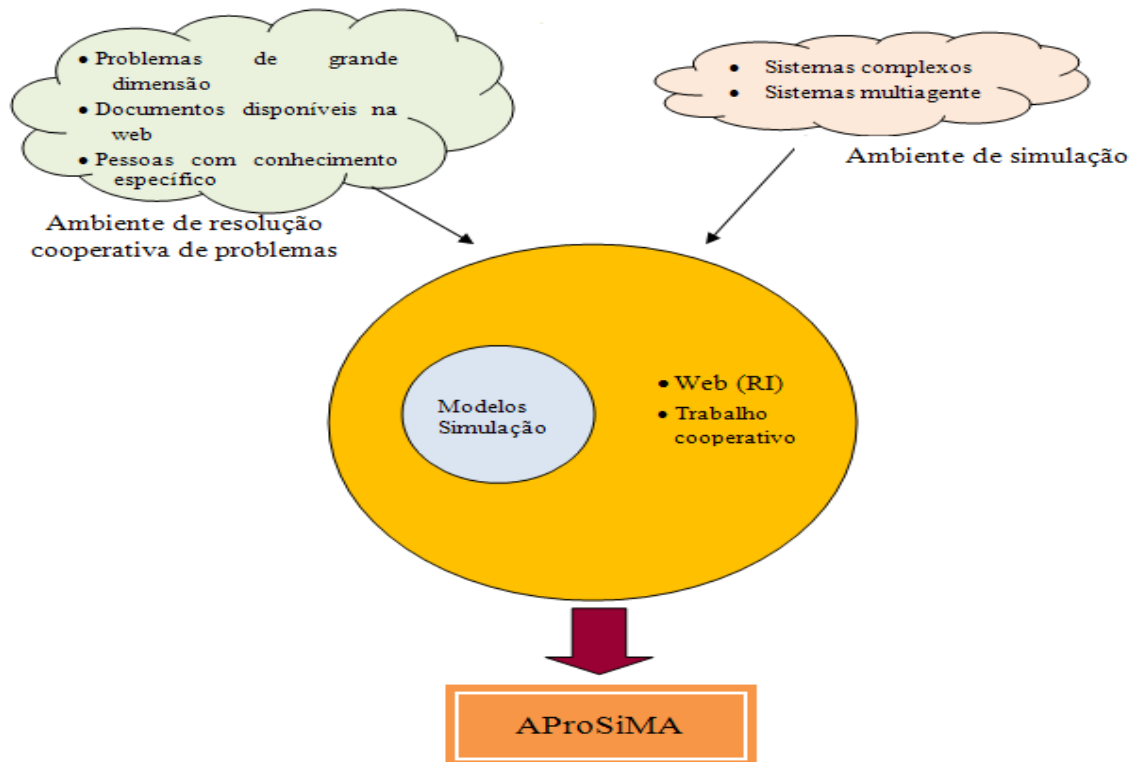


Figura 1 - Proposta Ambiente

A implementação do ambiente com características para web possibilita, em primeiro lugar, facilidade de uso e acesso por meio de navegadores disponíveis nas máquinas dos usuários, dispensando a instalação de novos programas. Sendo um ambiente web, a troca de informações entre os usuários e a participação cooperativa torna-se muito mais dinâmica. Todos os trabalhos realizados ficam disponíveis para que outros usuários possam consultar e reaproveitar para o desenvolvimento de novos modelos ou traçar estratégias diferentes – reúso. A interação entre os participantes dos grupos de resolução de problemas torna-se mais dinâmica e independente de plataforma, pois não há necessidade de compartilhar um mesmo software de apoio ao trabalho cooperativo.

A resolução cooperativa se justifica pelo fato de um modelo de simulação multiagente envolver abordagem interdisciplinar. Além disso, num processo cooperativo, cada indivíduo contribui com seu entendimento acerca do problema, e a troca de experiências entre os pares descreve melhor o modelo. Assim o ambiente deve dispor de um mecanismo para agregar os cooperadores geograficamente distribuídos

A ferramenta de simulação multiagente que disponha de uma linguagem acessível possibilita aos usuários criarem seus modelos independentemente de conhecimento de linguagem de programação e simulação multiagente. Um estilo de programação fácil de ser

entendido por pessoas leigas em computação é a programação visual (Visual programming language – VPL). Vários pesquisadores têm defendido a idéia de se trabalhar com linguagens de programação visual como uma forma de despertar o pensamento criativo, a programação dinâmica orientada ao domínio, bem como o amadurecimento em programação textual (Smith, 1997), (Smith, et al., 2001), (Repenning, 1993), (Glinert, et al., 1990), (Lieberman, 2001), (Repenning, et al., 2000), (Koelma, et al., 1992), (Resnick, 2007). Eles têm defendido o uso da programação visual em seus trabalhos descrevendo as muitas vantagens em relação às linguagens de programação textual. Segundo eles, com a programação visual o usuário vai se concentrar mais na resolução de problemas e menos na escrita dos programas.

1.3 Processo de investigação e objetivos

As questões propostas nos conduziram aos seguintes objetivos: 1) análise dos ambientes de resolução de problemas existentes; 2) análise dos ambientes de simulação multiagente existentes; 3) definição dos requisitos para composição do ambiente com relação à configuração, arquitetura, alcance, abrangência e tecnologia a ser utilizada.

Para alcançar o objetivo proposto, as seguintes questões serviram de guia para as formulações apresentadas nesta tese:

- ✓ Quais os requisitos necessários no ambiente para dar suporte ao desenvolvimento do trabalho cooperativo?
- ✓ Como estabelecer a interação entre pesquisadores. Como localizar e envolver esses pesquisadores num único local?
- ✓ Quais os requisitos necessários para uma ferramenta de construção de modelos de simulação que seja acessível para usuários com pouca experiência em linguagens de programação?
- ✓ O que pode ser aproveitado dos ambientes existentes, tanto para o trabalho cooperativo quanto para a construção de modelos?
- ✓ Qual deve ser a arquitetura desse ambiente?

1.4 Metodologia

Estabelecidas as questões de investigação as etapas subsequentes ocorreram da seguinte forma:

- Identificação dos contornos do problema, descrição dos objetivos da pesquisa e das questões de investigação;
- Exploração da literatura na área de ferramentas *groupware*, tomando como referência os ambientes do tipo CSCW/CSCL (*Computer Supported Cooperative Work / Computer Supported Collaborative Learning*, traduzido para o português como: “Trabalho Cooperativo Suportado por Computador” e sua variante “Aprendizagem Colaborativa Apoiada por Computador”.);
- Exploração da literatura na área de resolução de problemas, em particular nos métodos de resolução, resolução cooperativa e estratégias adotadas;
- Exploração da literatura na área de sistemas complexos para levantar os requisitos que compõem esses sistemas;
- Exploração da literatura na área de ambientes de simulação, destacando sistema multiagente;
- Identificação de um *framework* conceitual para construção de modelos de simulação baseado em multiagente;
- Formulação de uma proposta de ambiente de resolução cooperativa de problemas numa abordagem de simulação multiagente, com foco na Web e baseada no reúso de modelos;
- Desenvolvimento de um protótipo utilizando o *framework* definido, considerando as ferramentas *groupware* para o trabalho cooperativo e os elementos essenciais para a ferramenta de construção de modelos de simulação usando a tecnologia multiagente;
- Construção de modelos de simulação multiagente, no âmbito de sistemas complexos, e comparação com as versões desses modelos, construídas em ambientes de simulação já estabelecidos;
- Oferecimento de uma disciplina no curso de graduação em ciências da computação da Universidade Federal do Espírito Santo – UFES, como mecanismo de pré-avaliação do *framework* desenvolvido e também para avaliar o entendimento dos estudantes com relação a sistemas complexos por meio de simulação multiagente;

Após uma primeira avaliação do framework, alguns pontos em aberto foram verificados, o que contribuiu bastante para ajustes na própria implementação da ferramenta. Entretanto, novos testes devem ser realizados para validar as possibilidades e alcance do ambiente como um todo. Da mesma forma, um trabalho mais intensificado sobre a importância do ensino de sistemas complexos deve ser elaborado junto com estudantes e professores do nível fundamental, médio e superior. .

1.5 Organização deste trabalho

Dando seqüência ao trabalho, apresentamos no Capítulo 2 a contextualização sobre sistemas complexos e descrevemos o porquê da abordagem multiagente para seu estudo. No Capítulo 3 fazemos um levantamento, não exaustivo, dos ambientes de simulação multiagente, seguido dos requisitos para a modelagem de simulação multiagente. Apresentamos as considerações sobre os ambientes visitados e justificamos a necessidade de criação de um novo ambiente de simulação multiagente. A resolução de problemas é abordada no Capítulo 4. Neste capítulo situamos o problema do trabalho cooperativo, em particular a formação de grupos de resolução cooperativa. Procuramos fundamentá-lo com um levantamento de algumas ferramentas que objetivam a formação de grupos e apresentamos uma alternativa para o problema. E, ainda, o complementamos com um levantamento de ambientes de apoio ao trabalho cooperativo suportado por uma rede de computadores. No Capítulo 5 apresentamos o ambiente AProSiMA, proposta desta tese, com a descrição detalhada de cada uma das suas ferramentas e seus propósitos. Destaque dado às ferramentas Gerenciador e Simulador – NetPlay. No Capítulo 6, fazemos um comparativo da ferramenta Simulador – NetPlay com dois outros ambientes disponíveis. A aplicação da ferramenta simulador – NetPlay em uma disciplina do curso de graduação em Ciências da Computação da Universidade Federal do Espírito Santo – UFES. A avaliação deste curso é apresentada no Capítulo 7. Por fim, são apresentadas as conclusões e perspectivas futuras, seguidas das referências bibliográficas.

Capítulo 2

2 Sistemas complexos

Sistemas complexos é uma nova abordagem para a ciência que estuda como a relação entre partes faz surgir um comportamento coletivo do sistema, e como o sistema interage e forma relações com seu ambiente (Bar-Yam, 2004). Por exemplo, se observarmos os sistemas sociais, perceberemos que eles surgem (em parte) devido às relações entre as pessoas; o comportamento do cérebro é resultado das relações entre neurônios; a organização em uma colônia de formigas é o resultado da interação entre milhares de formigas trabalhadoras; da mesma forma, um congestionamento de trânsito ocorre (e dissolve) devido ao comportamento descrito pelos veículos que o compõem. Esses são alguns exemplos de sistemas complexos.

Sussman (Sussman, 2000) catalogou vinte idéias de pesquisadores⁴, em diferentes áreas, sobre sistemas complexos. Apesar das diferentes áreas, existe um sentido que permeia todas elas. Os Sistemas complexos podem ser identificados pelo que fazem – ao exibir organização sem um princípio organizacional centralizado, e também pela forma com que podem ou não ser analisados. Ao decompor um sistema e analisar as sub-partes, verificamos que, essas não necessariamente, dão indício sobre o comportamento do todo.

Em geral, se tomarmos os atributos de um modelo de sistema complexo numa área, por exemplo: ecologia, perceberemos que ele tem muitas características comuns a outras áreas, tais como sistema imunológico ou economia. O pesquisador Laureate I. Prigogine, premio Nobel de Química em 1977, e seus associados têm mostrado como os sistemas físicos e químicos podem exibir o mesmo tipo de comportamento de auto-organização que são típicos de sistemas biológicos (Resnick, 1994). Pesquisas têm comprovado que abordagens de sistemas complexos são inerentemente interdisciplinares; *insight* e resultados obtidos numa área podem ser transferidos para outra (Bramson, 2005).

Nos últimos anos, tem sido crescente o interesse no estudo sobre sistemas complexos e muitos centros de pesquisa tem se dedicado ao seu estudo: Santa Fe Institute⁵; NECSI –New

⁴ Joel Moses, Peter Senge, Joseph Sussman, J. Morley English, Reichtin and Maier, Flood and Carson, Coveney and Highfield, Edward O. Wilson, Katz and Kahn, Tom Hughes, David Warsh, John H. Holland, David Levy, A. O. Hirschman and C. E. Lindbloom, W. Brian Arthur, Murray Gell-Mann, Charles Perrow, John Sterman, Stuart Kauffman

⁵ <http://www.santafe.edu/>

England Complex Systems Institute⁶; COSI - Complexity in Social Science⁷, CSCS – The Center for the Study of Complex Systems⁸ (University of Michigan).

Para dar uma idéia mais geral sobre sistemas complexos, vamos analisar os seguintes exemplos: sistema imunológico, sistema nervoso central (SNC) e ecossistemas.

O sistema imunológico é capaz de perceber objetos estranhos ao organismo e, imediatamente, traçar uma estratégia de defesa. Como ele percebe esse elemento estranho e se prepara para enfrentar essa vulnerabilidade? Podemos dizer que essa identificação e má identificação são produtos da adaptação, mas esse processo adaptativo está longe de ser entendido de modo óbvio.

Consideremos o sistema nervoso central (SNC), composto por um grande número de células chamadas neurônios (que ocorrem numa gama de formas e atividades). O SNC consiste de milhões de neurônios e de centenas de tipos, cada um se dirigindo a centenas de contatos com outros neurônios, formando uma rede complexa. Os neurônios se comunicam através de pulsos de energia. Essa rede é similar ao sistema imunológico, agrega comportamento emergente, aprende com rapidez e se adapta às condições expostas. Embora a atividade de um único neurônio possa ser complexa, o comportamento de todo o sistema nervoso é muito mais complexo do que a soma das partes, atividades individuais de cada neurônio. O comportamento do sistema nervoso central depende muito mais das interações do que das ações. O número de interações – centenas de milhões de neurônios (cada um ligando centenas de outras interações simultâneas em milésimos de segundo) é uma experiência que supera a de qualquer outra máquina. Por exemplo, o mais sofisticado computador em comparação com o cérebro humano pode ser visto como um simples ábaco. Apesar dos grandes avanços, pouco se tem conseguido modelar em relação à complexidade do comportamento do cérebro (Johnson, 2003).

Ecossistemas são fenômenos que apresentam características semelhantes ao comportamento do cérebro humano e do sistema imunológico. Eles apresentam a mesma diversidade dos anteriores. Veja-se, por exemplo, a quantidade de organismos que se encontra em um metro cúbico de terra de uma floresta tropical. Ecossistemas estão em fluxo contínuo e exibem uma coleção de interações tais como: mutualismo, parasitismo, alterações biológicas, entre outras. A floresta pode manter sua diversidade por meio de um complexo conjunto de

⁶ <http://www.necsi.org/>

⁷ <http://www.irit.fr/COSI/>

⁸ <http://cscs.umich.edu/>

interações que reciclam seus nutrientes através do sistema e isso se torna um ciclo contínuo (Bar-Yam, 2004). Na Tabela 1 são apresentados alguns modelos e relações entre elementos salientes aos sistemas complexos⁹.

Tabela 1 - Exemplos e relações entre modelos de sistemas complexos

Exemplos Elementos	Sistema imunológico	Cérebro	Ecossistema	Sistema financeiro
Componentes	Material celular	Neurônios	Animais, indivíduos	Investidores
Organização	Organização celular	Rede de neurônios	Famílias, rebanho, cadeia alimentar	Empresas, associações, famílias
Feedback	Propagação da doença	Aprendizado	Sucesso ou falha	Compra, venda e troca
Heterogeneidade	Antígenos, anticorpos	Diferentes neurônios	Alimentação, raças, habitat,	Informação, preferência de riscos
Adaptação	Propagação ou redução da infecção	Comunicação, adaptação, aprendizagem	Acasalamento, estratégia de defesa, segurança	Aprendizagem, reação a anúncios
Comportamento emergente	Células saudáveis ou não, epidemia	Aprende e adapta as condições expostas	Extinção, castas, nicho	Movimento de mercado, inflação
Dinâmica	Propagação dos componentes	Troca de mensagens	Interação presa x predador, competição, cooperação	Ajuste de preços, movimento de estoque

Exemplos de sistemas complexos permeiam nossa vida diária e idéias sobre sistemas descentralizados estão se multiplicando em todos os níveis – na forma de organização das empresas, na composição das ferramentas computacionais, nas atividades econômicas, na organização familiar. No sistema educacional, indiretamente essas idéias estão presentes, um exemplo é a abordagem do ensino centrada no aluno, onde o professor não representa a autoridade central, mas sim um facilitador e um colaborador.

Sistemas complexos são aqueles cujos fenômenos e comportamentos emergem das interações de partes simples. Os sistemas complexos são formados por conjuntos de elementos interconectados, cujo comportamento coletivo surge de modo não óbvio (muitas vezes fugindo à intuição e de modo surpreendente), decorrentes das propriedades individuais dos elementos e de suas interconexões. São exemplos de sistemas complexos: colônia de

⁹ Adaptado de <http://cscs.umich.edu/old/complexity-eg.html>

insetos sociais, a composição das ferramentas computacionais, o mercado financeiro, a organização de uma cidade, bando de pássaros voando, reações do sistema imunológico, as relações familiares, congestionamento de trânsito, etc.. Em todos esses sistemas não há controle centralizado e os modelos são determinados pelas interações locais entre os componentes descentralizados (Resnick, 1994). A Figura 2 representa o ciclo num sistema complexo.

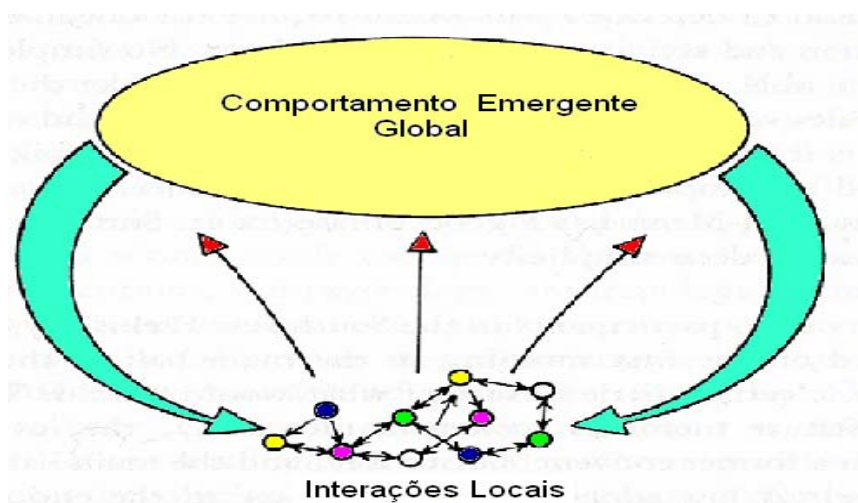


Figura 2 - Visão geral de um sistema complexo

Um dos objetivos desta tese é despertar a atenção das pessoas e, principalmente dos educadores, sobre a importância de se estudar e difundir os conceitos sobre sistemas complexos, além de iniciar um processo de sensibilização para o estudo de sistemas descentralizados. Mostrando que é possível e simples modelar situações nas quais os indivíduos têm comportamentos diferentes e complexos e apresentando mecanismos para seu estudo, que seja acessível a qualquer usuário.

Comparando aos exemplos de sistemas complexos citados apresentamos na Tabela 2 algumas características relacionadas aos sistemas complexos encontrada na literatura.

Tabela 2 - Características relacionadas aos sistemas complexos na literatura

Características Sistemas Complexos
Muitos elementos – agentes
Comportamento Simples – definido por regras
Movimento aleatório – randômico
Imprevisibilidade – interações não lineares
Sociedades heterogêneas – há diferentes agentes ou famílias
Auto-organização – o sistema se organiza sem um agente organizador. A organização surge espontaneamente a partir da desordem e não precisa ter um elemento organizador.
Os sistemas são caracterizados pela alta capacidade de adaptação
Comportamento emergente – o comportamento global surge das interações locais dos agentes no ambiente
Uma estrutura de realimentação ou de <i>feedback</i> , uma cadeia de causas e efeitos que forma um anel. Onde as mudanças são muitas vezes decorrentes do <i>feedback</i> que os agentes recebem como resultado de suas atividades.

Para possibilitar o estudo de sistemas complexos usaremos a tecnologia de Sistemas Multiagente (SMA), uma vez que essa tem sido amplamente utilizada para este fim por comportar as características dos sistemas complexos (Axelrod, 1997), (Bonabeau, 2002), (Epstein, et al., 1996), (Drogoul, et al., 1992), (Tang, et al., 2006), (Levy, et al., 2004).

2.1 2.1 Sistemas Multiagente (SMA)

Apesar de não haver consenso com relação à definição do termo “agente”, adotamos aqui a apresentada em (Woodridge, 2002), “um agente é um sistema de computador que está situado em algum ambiente, e que é capaz de ações autônomas neste ambiente para alcançar os seus objetivos¹⁰” (tradução nossa). Ressalva o autor que autonomia é central na definição de agente e também elemento de consenso. Por autonomia entende-se que o agente decide por si só se desempenha ou não determinada ação. Os componentes de um sistema multiagente

¹⁰ An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. (Segundo o autor essa definição é adaptada de Woodridge and Jennings, 1995).

são os agentes, com atributos que descrevem suas condições e características que governam seu comportamento.

Os agentes têm conjunto de atributos – estágios – que descrevem suas características. Os estágios podem ser formulados para representar os atributos de uma entidade real. Por exemplo, um agente projetado para imitar um inseto deve carregar o comportamento desse inseto, como movimento e comportamento. Normalmente, os atributos estabelecidos para definir alguma forma dos “agentes” são também atribuídos para os agentes individuais no sistema multiagente. O comportamento dos agentes é dirigido por regras de transição. Qualquer número de regras pode ser criado para governar as atividades dos agentes – que são as metas que os procuram satisfazer.

A pesquisa em Sistemas Multiagente (SMA) procura modelos e técnicas que coordenem as atividades de um grupo de agentes autônomos, de forma que eles trabalhem de forma cooperativa. Para (Bordini, et al., 2001), num SMA o enfoque principal é prover mecanismos para a criação de sistemas computacionais a partir de entidades de software autônomas, agentes, que interagem por meio de um ambiente compartilhado por todos os agentes de uma sociedade, e sobre o qual estes agentes atuam, alterando seu estado. Nos SMA os agentes são distinguidos em cognitivos e reativos.

2.1.1 Agentes cognitivos

Os agentes cognitivos têm uma representação simbólica explícita do seu ambiente, no qual podem argumentar e prever eventos futuros. Agentes cognitivos são dirigidos por intenções, isto é, por metas explícitas que conduzem seu comportamento e os tornam capazes de escolher entre possíveis ações. SMA cognitivos geralmente possuem poucos agentes, dado que cada agente é um sistema complexo e computacionalmente pesado, uma vez que engloba as características: percepção, ação, comunicação, representação, motivação, deliberação, raciocínio e aprendizagem (Bordini, et al., 2001), (Drogoul, et al., 1992), (Russel, et al., 2004). A Figura 3 apresenta a arquitetura para o modelo de agente cognitivo, adaptada de (Russel, et al., 2004).

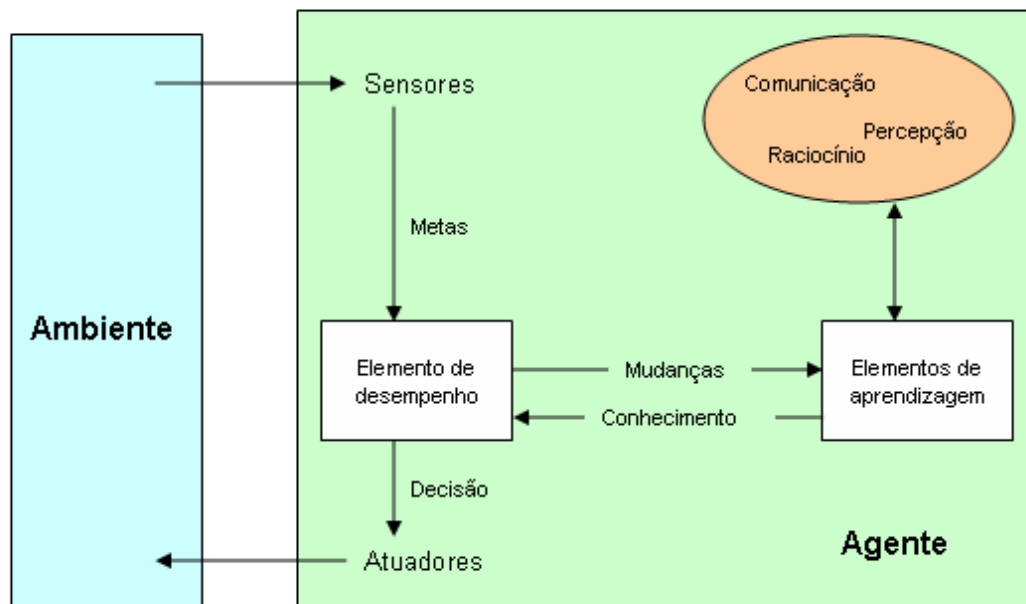


Figura 3 - Apresenta a arquitetura para o modelo de agente cognitivo, adaptada de (Russel, et al., 2004).

2.1.2 Agente reativo

Os agentes reativos simples são aqueles que selecionam ações com base na percepção atual, ignorando o restante do histórico de percepções (Russel, et al., 2004). O comportamento do agente reativo pode ser definido usando a noção que vem do campo da psicologia comportamentalista, estritamente definida com o esquema de “estímulo / resposta”. SMA reativos seguem a idéia de que um comportamento inteligente em um sistema emerge da interação entre um grande número de agentes simples. Ou seja, os agentes não são inteligentes individualmente, mas o comportamento global sim. Neste tipo de sistema, os agentes não possuem uma representação explícita do estado do ambiente e dos outros agentes, nem de suas ações passadas; eles têm comportamentos que podem ser descritos como autômatos finitos simples, e possuem um conjunto de regras que mapeiam percepções do ambiente diretamente em ações sobre este (Bordini, et al., 2001). Nos SMA reativos, o ambiente tem papel importante, uma vez que os agentes reativos não possuem memória. É a percepção do ambiente que permitirá a eles tomarem as decisões sobre suas ações. A Figura 4 apresenta a arquitetura para o modelo de agente reativo simples, adaptada de (Russel, et al., 2004).

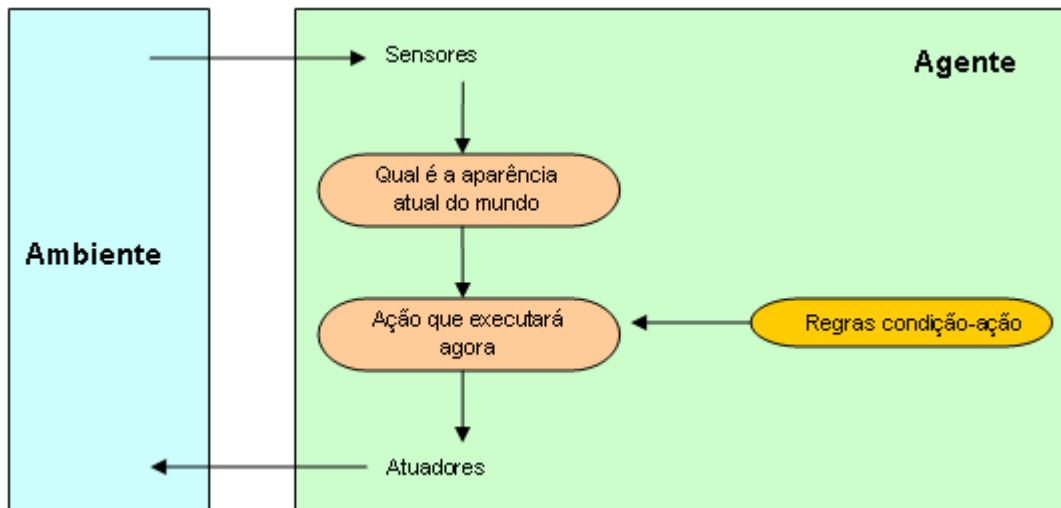


Figura 4 - Apresenta a arquitetura para o modelo de agente reativo simples, adaptada de (Russel, et al., 2004). O agente reativo simples age de acordo com uma regra cuja condição corresponde ao estado atual definido pela percepção¹¹

2.1.3 Aplicação de Sistemas Multiagente (SMA)

A pesquisa em SMA não está ligada apenas à comunidade de sistemas paralelos, ou à inteligência artificial, ou à teoria dos jogos e economia, ou às ciências sociais. Ela é de natureza interdisciplinar, transitando sob várias áreas do conhecimento¹². A área de aplicações de SMA também é bastante vasta: controle de tráfego aéreo, indústria, gerência de negócios, interação humano-computador, ambientes de aprendizagem, entretenimento, aplicações distribuídas, aplicações para Internet, simulação social, entre outras. A modelagem em SMA é uma abordagem promissora para projetar e simular essas situações.

Na modelagem baseada em agentes, consideramos como “agente” qualquer elemento para o qual desejamos atribuir um comportamento. Nesse tipo de modelagem os agentes possuem: representação interna dos dados (memória ou estágio); meios para modificar suas representações internas (percepções); e meios para modificar seu ambiente (comportamento).

A Figura 5 ilustra uma modelagem baseada em agentes. O observador determina os agentes (animados e inanimados) que comporão o modelo. Baseado em dados coletados do modelo no mundo real, ele determina o tipo de interação que deverá ocorrer entre os agentes.

¹¹ Função agente reativo simples (percepção) retorna uma ação.

estado ← interpretar entrada (percepção)

Regra ← regra correspondente (estado, regra)

Ação ← ação da regra (regra) retorna ação

¹²Para mais referências ver Wooldridge 2002; Bordini, Vieira, Moreira, 2001; Russell, Norvig, 2004; Drogoul, Ferber 1992

Cada agente possui um conjunto de regras específicas que definem o seu comportamento. Os agentes podem estar agrupados formando pequenas sociedades.

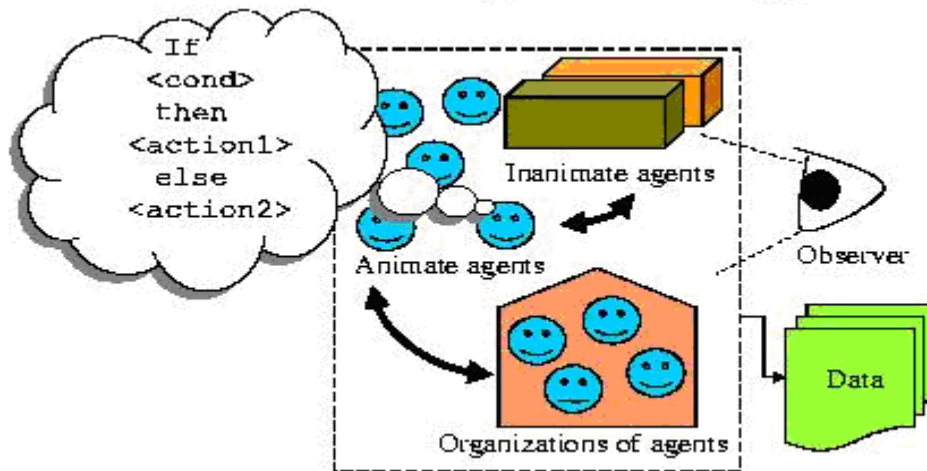


Figura 5 - Essa é uma abordagem “bottom up” chamada de Modelagem baseada em agente - Agent-based modelling (ABM)¹³

Nestas sociedades os agentes interagem uns com os outros gerando um comportamento mais complexo, difícil de ser alcançado apenas pelo comportamento individual. Neste trabalho focaremos o uso de sistemas multiagente em sistemas complexos.

2.1.3.1 Sistemas Multiagente e Sistemas Complexos

Em geral, para representar um sistema complexo por meio de sistema multiagente, basta considerar os agentes como reativos. Normalmente, o que distingue um sistema complexo de um sistema multiagente é o foco das propriedades definidas para os agentes:

- Reatividade – agentes inteligentes são capazes de perceber seu ambiente e responder oportunamente às mudanças que ocorrem para satisfazer o cumprimento de suas metas, (Woodridge, 2002);
- Proatividade – agentes inteligentes são capazes de exibir comportamento direcionado às suas metas tomando iniciativas para satisfazer seus objetivos, (Woodridge, 2002) e
- Habilidade social – agentes inteligentes são capazes de interagir com outros agentes (e inclusive com humanos) para satisfazer os objetivos para o qual foi projetado, (Woodridge, 2002).

¹³Fonte: “Swarm tutorial”. Disponível em: <http://www.swarm.org/csss-tutorial/titlefoil.html>. Acesso em: 25/09/2005.

No sistema complexo, os agentes, bem como o sistema, são adaptativos. Os sistemas adaptativos complexos são caracterizados pelo alto grau de adaptatividade, que lhes dão resiliência¹⁴ frente às perturbações. Uma propriedade-chave dos sistemas complexos é a emergência (Ruthen, 1993).

O estudo de sistemas complexos através de SMA nos ajuda a entender efeitos indiretos. Problemas que são difíceis de resolver usando abordagens tradicionais são muitas vezes difíceis porque as causas e os efeitos não são relacionados de modo óbvio. Assim, os ambientes de simulação multiagente tornam as idéias relacionadas a sistema complexos muito mais acessíveis, pois esses ambientes possibilitam as pessoas pensarem sobre as ações e interações de objetos individuais ou criaturas. Através da abordagem de simulação multiagente os estudantes podem explorar como mudanças de regras ou parâmetros que compõem o modelo podem levar a diferentes comportamentos do fenômeno. Essas mudanças possibilitam aos estudantes examinar mecanismos que estão subjacentes ao fenômeno que eles vêem no mundo real. Ao invés de aceitar as mudanças de fenômeno como “caixas pretas”, os aprendizes podem olhar dentro delas e então tentar reescrevê-las.

Mas a principal contribuição da modelagem multiagente vem das ferramentas que trazem novas representações e que são projetadas explicitamente para uso de simulação no computador. Nestas ferramentas de modelagem de simulação multiagente, são oferecidos centenas de objetos individuais agem em paralelo. Diferente das ferramentas computacionais que implementam os modelos matemáticos (por exemplo, resolução numérica através de equações diferenciais), as ferramentas baseadas em multiagente fornecem uma nova representação que são explicitamente planejadas para uso no computador – simulação multiagente (Bonabeau, 2002).

2.2 Simulação multiagente

Complementando o que foi dito na sessão anterior, a área de sistemas multiagente (SMA) estuda o comportamento de um grupo organizado de agentes autônomos que cooperam entre si para realizar atividades de resolução de problemas que estão além das suas capacidades individuais. Neste contexto, organização significa o estabelecimento de restrições

¹⁴ O termo resiliência, originário da Física, vem sendo utilizado por varias áreas. Em sistemas complexos, resiliência é a alta capacidade de adaptação dos agentes às situações do ambiente. Segundo dicionário Houaiss, o termo resiliência significa: capacidade de se recobrar facilmente ou se adaptar à má sorte ou às mudanças. Na Física se define que Resiliência é a capacidade que um elemento tem de retornar ao seu estado inicial, após sofrer uma influência externa. Na psicologia da educação, o termo resiliência é referido como processos que explicam a “superação” de crises e adversidades em indivíduos, grupos e organizações (Yunes, M. A. M. & Szymanski, H., na obra “Resiliência: noção, conceitos afins e considerações críticas” citada em (Tavares, 2001)

de comportamento dos agentes de modo que os mesmos trabalhem de forma cooperativa. O termo autônomo significa que o agente tem sua existência independente dos demais agentes e até mesmo do problema que será solucionado (Woodridge, 2002). Geralmente, as propriedades desejadas num sistema multiagente vêm do equilíbrio desses dois elementos.

Estes agentes são representações extraídas no contexto do “mundo real” e representadas num “mundo artificial”. Normalmente, um sistema multiagente pode ter milhares de agentes interagindo no mundo artificial representando algum fenômeno do mundo real, por exemplo: atores no mercado financeiro. Cada agente avalia independentemente sua situação e toma decisões baseado num conjunto de regras. Os agentes podem executar vários comportamentos de acordo com o sistema que ele representa, por exemplo: veículos e/ou pedestres em situações de tráfego, o comportamento de consumidores, a dinâmica da bolsa de valores, animais e plantas em um ecossistema.

Uma característica do modelo baseado em agentes é as ações repetitivas entre os agentes. Em geral, muitos modelos baseado em agentes representam um ciclo de vida completo (criação, reprodução e morte dos agentes) e as interação dos agentes com seu ambiente. A simulação no computador é então usada para descobrir propriedades do modelo e assim, ganhar entendimento dentro de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrões (Epstein, et al., 1996), (Axelrod, 1996).

Em geral, um modelo de simulação baseado em agentes consiste nas interações entre os agentes e entre estes e o ambiente. É importante ressaltar que mesmo num modelo simples é possível perceber o comportamento complexo que emerge das interações, sendo estas informações valiosas sobre a dinâmica do sistema. Isso torna possível explorar as conexões entre o comportamento no micro-nível de indivíduos e modelos em macro-nível que emergem das interações de muitos indivíduos. Modelos de simulação baseados em agentes mais sofisticados incorporam algoritmos evolucionários, redes neurais ou outras técnicas de aprendizagem para permitir aparições mais realísticas (Bonabeau, 2002).

2.2.1 Uso de simulação multiagente

Na construção de modelos de simulação baseado em agentes, consideramos como “agente” qualquer elemento para o qual desejamos atribuir um comportamento. Nesse tipo de modelagem os agentes possuem: representação interna dos dados (memória ou estágio); meios para modificar suas representações internas (percepções); e meios para modificar seu ambiente (comportamento).

Conforme visto na Figura 5, dois detalhes chamam a atenção: o primeiro é que cada agente possui um conjunto de regras específicas que definem seu comportamento. O segundo é que os agentes se agrupam para formar sociedades. Nas sociedades os agentes interagem uns com os outros gerando um comportamento mais complexo, difícil de ser alcançado apenas pelo comportamento individual. Uma pergunta que podemos fazer é: por que usar modelos de simulação baseada em agentes (e SMA)? A resposta seria a de que a modelagem baseada em agentes complementa e melhora a abordagem tradicional. Apesar de muito utilizados, os métodos matemáticos tradicionais segundo (Colella, et al., 2001):

- podem descrever propriedades macroscópicas de um sistema que já são conhecidas, mas não explicam a origem dessas propriedades;
- não podem ser facilmente extrapolados além do âmbito das equações;
- não suportam bem sistemas descontínuos;
- não suportam bem populações heterogêneas.

A simulação de modelos baseada em agentes geralmente envolve um ou mais tipos de agentes, em quantidades bastante variáveis em cada classe ou tipo, melhorando o estudo de populações heterogêneas. Ela é, em geral, interessada na dinâmica entre os agentes, e nos diferentes tipos de dinâmicas que podem resultar em diferentes ambientes. Assim, um dos objetivos da simulação multiagente é procurar entender as propriedades dos sistemas complexos através do uso de modelos.

Segundo (Axelrod, 1996) a simulação de modelos baseada em agentes tem sido considerada um modo de fazer ciência, que pode ser contrastada com os modos indutivos e dedutivos. Nesta mesma obra Axelrod utiliza e comenta a definição de simulação dada por Bratley, Fox e Schrage, segundo a qual “(...) simulação significa dirigir um modelo de um sistema com entradas satisfatórias e observar as correspondentes saídas” (tradução nossa). No entanto, ele ressalva que, apesar desta definição ser útil, ela esconde a explicitação dos diversos propósitos para os quais a simulação pode ser utilizada: predição, desempenho, treinamento, entretenimento, educação, prova e descoberta. Afirma, ainda, que a simulação como metodologia de pesquisa é um novo modo de conduzir a pesquisa científica, além da indução – descoberta de padrões em dados empíricos – e da dedução – conjunto de axiomas e a prova das conseqüências que podem ser derivadas dessas suposições.

A simulação, semelhante à dedução, inicia com um conjunto explícito de suposições, mas difere dela por não provar teoremas. Ao contrário, gera dados que podem ser analisados intuitivamente. Diferente da indução típica, a simulação dos dados vem de um rigoroso e especificado conjunto de regras, antes que tivesse sido diretamente mensurados no mundo real. Enquanto a indução pode ser usada para encontrar padrões nos dados, a dedução pode ser usada para gerar conseqüências das suposições, a modelagem da simulação pode ser usada como um auxílio na intuição.

Assim, um dos objetivos da simulação multiagente é buscar entender as propriedades dos sistemas complexos como na dedução: iniciando com um conjunto de hipóteses, mas não provando teoremas; e como na indução: gerando dados empíricos (simulados), mas não dados do mundo real. Em geral, a simulação multiagente objetiva extrapolar as experiências de pensamento, isto é, enriquece o entendimento sobre os processos fundamentais do sistema (ferramenta de análise).

Atualmente, o campo da simulação multiagente tem se expandido além de suas fronteiras e incorporado outros grupos de pessoas que trabalham numa gama de disciplinas, incluindo Ciências da Computação, Inteligência Artificial, Economia, Sociologia, Física, Biologia, Antropologia, entre outras. Muitos pesquisadores têm demonstrado os benefícios da aprendizagem, numa ampla variedade de fenômenos científicos, através da abordagem multiagente. Ressaltamos os trabalhos realizados nessa área através do “Media Lab” do Massachusetts Institute of Technology – MIT; do “Advanced Project Research Agency” através do Institute Santa Fe Institute; do “Center for connected learning and computer-based modeling” da Northwestern University; do “Advanced Learning Center” do Mesa State College, do “Center for the Study of Complex Systems” da University of Michigan, entre outros, todos usando ambientes de modelagem multiagente.

2.2.2 Por que usar simulação multiagente?

Um projeto na perspectiva multiagente pode oferecer um profundo entendimento de um fenômeno científico relevante. Por meio de um pequeno número de regras locais é possível capturar estruturas fundamentais subjacentes ao comportamento complexo dentro de um domínio. Com a simulação multiagente é possível fazer com que os processos possam ser visualizados em caráter micro – por meio de relações definidas por regras simples.

A modelagem baseada em agentes contrasta com as abordagens tradicionais de simulação que tipicamente envolvem a construção de conjunto de equações diferenciais

(Drogoul, et al., 1992). Segundo (Tang, et al., 2006), tais modelos, comumente chamados de “*equation-based models*” (*EBM*), têm sido amplamente aplicados sobre o comportamento de populações. Entretanto, os usuários tem utilizado a modelagem baseada em agentes por quatro razões principais:

- i. modelo baseado em agente é um modo natural para descrever sistemas compreendidos por entidades interagindo;
- ii. modelos baseado em agentes são flexíveis;
- iii. modelos baseado em agentes capturam o fenômeno emergente; e
- iv. modelo baseado em agentes fornece acesso a um nível mais alto de detalhes do sistema.

Em particular, as interações entre os agentes podem ser mais fáceis de modelar nos sistemas baseados em agentes do que nos sistemas baseados em equações, mesmo para os experientes nesta área, além de revelar os detalhes do sistema que está sendo modelado e que são obscurecidos pelos modelos baseado em equações (Tang, et al., 2006).

Os modelos de simulação multiagente estão baseados na idéia de que os programas exibem comportamento descrito pelos seus mecanismos internos, isto é, programa de instruções. Relacionando um indivíduo a um programa é possível simular um mundo artificial habitado por processos interativos. Podemos, então, alcançar simulações transportando a população de um biosistema real para um artificial na qual hipóteses particulares podem ser exploradas, repetindo experimentos do mesmo modo que num laboratório real, porém mais facilmente (sociedade artificial). Cada organismo de uma população é, então, representado por um agente cujo comportamento é programado por regras com todos os detalhes requeridos (Doran et al. apud (Drogoul, et al., 1992)), (Hogeweg, et al., 1985), (Collins, et al., 1991).

Entretanto, conforme sugere (Bonabeau, 2002), a modelagem baseada em agentes não é uma alternativa para a modelagem tradicional de equações diferenciais. Sendo importante definir “por que” e “quando” é útil usar a modelagem baseada em agentes, para que seja claro quando e como usar tal tecnologia. Para ele, o uso da simulação multiagente tem se tornado popular devido às facilidades de programar os modelos baseados em agentes. Mas o fato de a técnica de uso ser fácil não significa que os conceitos são fáceis. Embora a técnica seja relativamente simples, o aspecto conceitual é bastante profundo. Portanto, muita atenção é requerida para não haver uso indevido da modelagem baseada em agente.

Através da simulação multiagente é possível observar como os agentes individuais (pessoas, bactérias, insetos, nações, ou organizações) interagem entre si e com seu ambiente; é possível modelar situações nas quais os indivíduos têm comportamentos diferentes e complexos, podendo levar em conta tanto as propriedades quantitativas (parâmetros numéricos) quanto qualitativas (comportamento individual) do sistema representado no modelo. Essa descrição não revela a complexidade do comportamento individual, mas sim as características espaço temporal que emergem do coletivo, assumindo desse modo que interações entre simples indivíduos podem produzir um comportamento coletivo altamente estruturado e inteligente.

Assim, o uso da simulação multiagente é potencialmente útil em situações em que:

- i. o comportamento individual dos agentes não é linear e pode ser caracterizado por situações definidas por regras (“se” condição “então” ação). Em princípio, todas as coisas podem ser representadas por equações, mas a complexidade das equações cresce exponencialmente à medida que cresce a complexidade do comportamento dos agentes. Descrever o comportamento individual com equações torna-se algo impraticável (intratável)¹⁵;
- ii. as interações entre os agentes são complexas, não lineares, descontínuas ou discretas; quando o comportamento de um agente pode ser alterado drasticamente e descontinuamente por outros agentes. Por exemplo, o mercado financeiro ou situações de congestionamento de trânsito;
- iii. descrever o sistema através de atividades é mais fácil do que descrever através de processos. A modelagem baseada em agentes é muitas vezes o modo mais apropriado para descrever o que está acontecendo numa situação real e o modelador pode facilmente “conectar-se” ao modelo e se sentir “apropriado”;
- iv. a topologia no ambiente é crucial e a posição dos agentes não é fixa. Exemplo: situações de tráfego, colônia de insetos ou ecossistemas;
- v. as populações são heterogêneas, quando cada indivíduo é (potencialmente) diferente e as interações são heterogêneas e podem gerar efeitos em cadeia.

¹⁵ Intratável no número de variáveis e no uso de equações não lineares, desta forma torna-se impraticável o seu uso.

Embora o sistema investigado seja complicado (compostos por muitas relações), as hipóteses subjacentes ao modelo devem ser simples. A complexidade do modelo surge dos resultados simulados e não das hipóteses assumidas.

É importante ressaltar que uma boa simulação depende da expertise do usuário (engenheiro do conhecimento) em perceber quem são os agentes e como eles estão interagindo e descrevendo os modelos – a modelagem multiagente não é a solução para tudo-vai depender muito de como o observador percebe as ocorrências no mundo simulado. Quanto melhor for a experiência do usuário em transferir os aspectos do mundo real para o artificial, melhor será a representatividade do modelo simulado.

2.2.3 Requisitos para a modelagem de simulação multiagente

A construção de um modelo de simulação multiagente consiste em capturar as características essenciais do sistema complexo, por exemplo: seus elementos, suas propriedades, seus comportamentos e suas relações (entre os agentes e entre os agentes e o ambiente). Assim, para definir uma sociedade de agentes é necessário dispor de elementos que descrevam essas sociedades. Entre esses elementos deve ser considerado:

2.2.3.1 Definição do agente

A definição dos agentes é obtida com os elementos que descrevem as características do agente. Nesta categoria, destaque deve ser dado à:

- *forma* – a aparência do agente. Numa sociedade artificial, quanto mais próxima do modelo for a forma do agente, mais similar ao mundo real é a analogia;
- o *tamanho* – para cada modelo há a necessidade de caracterizar o agente num tamanho que possibilite a análise do comportamento;
- a *quantidade* – essencial para analisar o comportamento do modelo, pois se quisermos simular um comportamento de uma população, é interessante que tenhamos muitos agentes interagindo entre si;
- a *posição* – determinar a posição que cada agente ocupa no ambiente possibilita melhor definição das regras de comportamento. A posição de cada agente é definida por coordenadas cartesianas. A posição dos agentes pode ser determinada como um elemento pré-fixado ou de forma aleatória;
- a *cor* – usada para diferenciar os elementos componentes do modelo e melhorar a visualização;

2.2.3.2 Definição de comportamento

Alguns elementos necessários para definir o comportamento dos agentes são:

- *movimento* – forma como o agente percorre o ambiente;
- *percepção* – capacidade de o agente perceber outros agentes e ou objetos no ambiente;
- *colisão* – momento em que dois ou mais agentes tendem a ocupar a mesma posição no ambiente;

2.2.3.3 Definição do ambiente

O ambiente é o local onde o agente está posicionado e no qual desenvolve seu comportamento. Em geral, o ambiente é o palco onde ocorrem as interações entre os agentes e desses com o ambientes. Em sociedades artificiais, normalmente, usa-se uma grade (malha) sobre o plano cartesiano.

2.2.3.4 Identificação dos agentes e definição de variáveis

A identificação do agente é usada para o estabelecimento das regras de comportamento. Através da identificação é possível estabelecer o controle e formação de subconjuntos (famílias) e classes de agentes. Para descrição de modelos é importante estabelecer parâmetros relacionados aos agentes e ao modelo, denominadas de variáveis. As variáveis podem ser de diferentes tipos: lógicas, globais, locais e vinculadas aos agentes.

Além das características descritas acima, para definição dos agentes e ambientes, destacamos a necessidade de algumas ferramentas conceituais para facilitar o acompanhamento do modelo e possibilitar a extração de dados comportamentais e análise (gráficos, contadores, análise dos resultados, etc.). Ferramentas com esse teor facilitam a aprendizagem

No Apêndice A, é apresentada uma tabela descrevendo como cada um dos ambientes visitados trata cada um desses aspectos.

Uma das principais razões para expansão do uso da modelagem multiagente foi o crescente número e qualidade das plataformas de software desenvolvidas para simulação baseadas em agentes. Nos últimos anos, essas ferramentas têm tornado a modelagem baseada em agente mais atrativa para usuários de várias áreas. Na seqüência, apresentaremos algumas dessas ferramentas.

Capítulo 3

3 Ambientes de simulação multiagente

Ambientes de simulação têm sido amplamente utilizados para analisar modelos com variados propósitos, sendo a simulação multiagente uma alternativa para estudar ambientes dinâmicos como, por exemplo: a propagação de um incêndio em uma floresta; o comportamento de bando de aves voando; o comportamento de insetos sociais; e, também, para comparar os diversos cenários decorrentes da simulação.

Um modelo de simulação baseado em agentes é uma aproximação idealizada pelo modelador (pesquisador, estudante, usuário) de “como as coisas trabalham no mundo”, sendo os agentes as entidades computacionais, por exemplo: pessoas, animais, objetos inanimados, etc.. O modelador cria os agentes que são entidades representacionais de um fenômeno no mundo real. Para esta representação o modelador descreve as regras que expressam o comportamento dos agentes e também cria um ambiente virtual onde esses agentes operam. O modelador conduz o experimento de simulação observando o desempenho do comportamento dos agentes com outros agentes e dos agentes com o ambiente.

Uma das características importantes da simulação multiagente é o acompanhamento da dinâmica interativa dos agentes, elementos geralmente difíceis de serem observados em métodos analíticos e, muitas vezes, impossíveis de previsão em fenômenos decorrentes de reações individuais randômicas. Os valores randômicos de cada execução produzem diferentes resultados cada vez que o modelo é executado. Além disso, a simulação multiagente possibilita ao modelador explorar uma vasta gama de parâmetros, incluindo o comportamento dos agentes e dados do ambiente.

Existem vários ambientes de programação especificamente projetados para modelagem baseada em agentes, destacamos: StarLogo, Swarm, Repast, AgentSheets, NetLogo e SIMULA. Apresentamos também o ambiente Klik&Play, que apresenta condições suficientes para a modelagem multiagente. Esses ambientes de modelagem multiagente estão revolucionando a prática científica pela possibilidade de construção de modelos (Abrahamson, et al., 2005), (Blikstein, et al., 2005), (Levy, et al., 2005), (Blikstein, et al., 2005-a), (Levy, et al., 2004), (North, et al., 2006), (Minar, et al., 1996), (Railsback, et al.,

2006). Na seqüência apresentaremos um breve resumo das funcionalidades de cada ambiente mencionado.

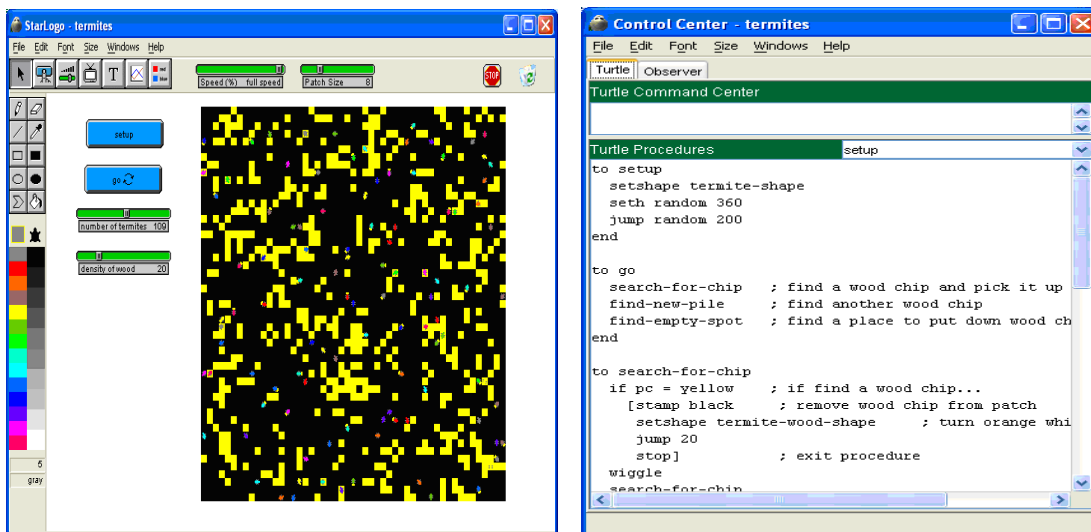
3.1 StarLogo

StarLogo [MIT Media Lab] é uma ferramenta de modelagem programável, especificamente projetado para a exploração de sistemas descentralizados, via simulação. O StarLogo foi criado pela equipe de desenvolvimento do StarLogo, MIT Media Lab, por Alan Epstein, Rebekah Wahba, and Robert Tau, sendo desenvolvido parte em Java e parte numa linguagem chamada YoYo, que é uma variante do Logo construída sobre Java para permitir programas híbridos e também para poder disponibilizar aplicações na web.

O StarLogo possibilita ao usuário criar e controlar o comportamento de tartarugas gráficas (*turtles*). As tartarugas se movem sobre o ambiente *patches* (grade onde as *turtles* se posicionam e se movimentam). A Figura 6, à esquerda, apresenta a interface gráfica do ambiente. O StarLogo é uma ferramenta baseada em agente (por exemplo, as *turtles* e os *patches* são agentes), e seu modelo de programação é procedural. A Figura 6, à direita, apresenta um fragmento de código que é executado na interface gráfica. A linguagem oferece um conjunto de comandos para que o usuário possa criar e controlar as *turtles* e os *patches*.

Na prática, o StarLogo é bastante fácil de usar (mesmo para quem tem pouca experiência em programação). O StarLogo oferece uma interface gráfica para auxiliar na escrita do código que define a simulação. Além disso, oferece vários elementos que possibilitam a criação de gráficos, definição de botões, *slider* que controlam a simulação e definem a entrada de dados (por exemplo, o número de agentes).

Entretanto, apesar de todas essas facilidades para a criação de gráficos, há alguns problemas associados a eles, por exemplo, apesar de ser possível a associação de vários elementos para serem plotados num mesmo gráfico, não é possível criar mais do que um gráfico. Com a versão atual do StarLogo 2.2, é possível e fácil exportar o modelo simulado para a web, através de *applet*. Existem muitos exemplos de simulação desenvolvidos em StarLogo disponíveis na internet, bem como existem grupos de apoio através de e-mail. O StarLogo pode ser acessado gratuitamente através <http://education.mit.edu/starlogo/>.



Recentemente, em janeiro de 2006, foi lançado o StarLogo TNG, a nova geração do StarLogo, escrito em linguagem C e Java. A ferramenta continua com a mesma filosofia, porém apresenta uma programação exclusivamente gráfica, nos moldes de blocos de encaixe. As funções são disponibilizadas numa interface gráfica e os usuários arrastam e montam as composições. As Figura 7 e Figura 8 mostram a nova interface do ambiente.

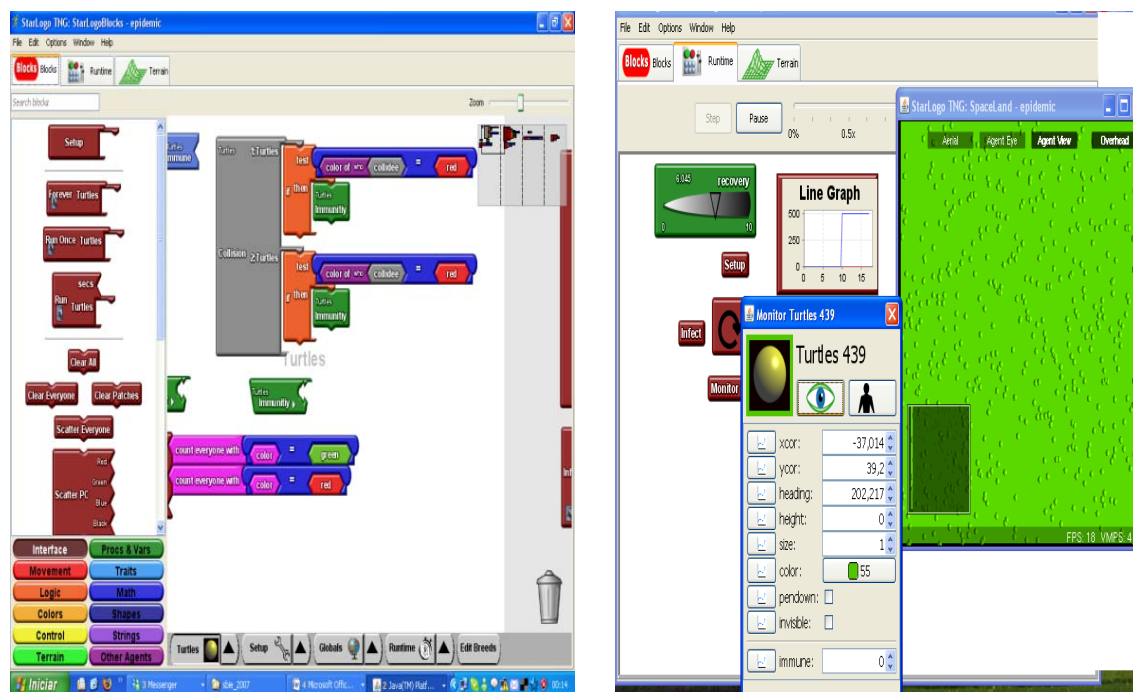


Figura 7 - Montagem do modelo em blocos (dir.) e visualização em 3D e inspeção comportamento agente (esq.)



Figura 8 - Peças de encaixe para montagem modelo

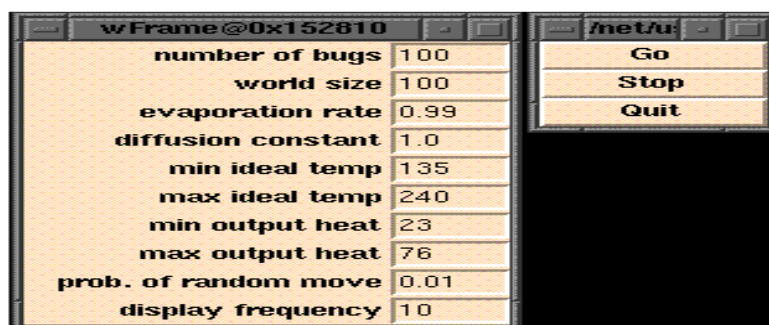
Apesar da interface gráfica, a concepção do modelo continua estruturada em comandos, que foram transferidos para blocos de encaixe. Assim, a lógica da programação continua vinculada à associação de comandos. Uma vantagem que pode ser vista neste modelo é que, nele, não há necessidade de saber os nomes do comando, mas a sua funcionalidade; como e onde usá-lo permanece preponderante.

3.2 Swarm

O Swarm (Minar, et al., 1996) foi originalmente desenvolvido pelo Santa Fe Institute (SFI), New Mexico, em 1994, com o objetivo de disponibilizar tanto um vocabulário quanto ferramentas para o desenvolvimento de modelos de simulação multiagente (ABMs). Atualmente, o desenvolvimento e gerenciamento do ambiente está sendo controlado pelo Swarm Development Group, Santa Fe – New Mexico – USA.

O Swarm foi concebido como uma biblioteca de software (Swarm) com métodos padronizados para programação de modelos de simulação multiagente. Para usar o ambiente o

usuário deve ter conhecimento tanto de programação quanto ser familiarizado com a metodologia orientada a objetos. Os programas podem ser desenvolvidos nas linguagens Objective-C ou Java. Não é fácil criar uma nova aplicação em Swarm. Entretanto, esse *framework* oferece várias peças de código que facilitam a construção do modelo: gerenciamento de memória, manutenção de listas, agenda de ações entre agentes e outros. O usuário constrói seu modelo de simulação incorporando vários objetos Swarm em seu próprio programa. A Figura 9 apresenta uma tabela de definição de atributos dos agentes no ambiente.



wFrame@0x152810	
number of bugs	100
world size	100
evaporation rate	0.99
diffusion constant	1.0
min ideal temp	135
max ideal temp	240
min output heat	23
max output heat	76
prob. of random move	0.01
display frequency	10

met/u:

Go

Stop

Quit

Figura 9 - Controle do modelo no ambiente Swarm

O usuário deve escolher a linguagem com a qual prefere trabalhar além de conhecer alguns códigos em Swarm. A versão em Java foi criada dez anos depois da versão em Objective-C, e apresenta mais vantagens em termos de avanço de linguagem. Logicamente, existem muito mais modelos em Objective-C do que em Java. A linguagem hoje também suporta JavaScript e Schema associado a Objective-C e Java.

No sistema Swarm, o componente principal é o “swarm”. Um “swarm” é uma coleção de agentes que possibilita agrupar hierarquias de agentes, controlar agenda de eventos, e sondar informações do estado atual do agente e de seu ambiente. Neste caso, o comportamento de alto nível do agente é definido por meio de um fenômeno emergente de agentes dentro do swarm. Essa abordagem multi-nível oferecida pelo Swarm faz com que ela se torne muito poderosa. No Swarm, os agentes podem ser utilizados para construir outros agentes e entender o mundo no qual estão imersos. No Swarm, existe um agente especial – o observador (atualmente são swarm – um grupo de agente e agenda de atividade) – cuja proposta é observar outros objetos via sonda de interface. Esses objetos podem fornecer dados tanto para análise em tempo real quanto armazenar para análise posterior. Possivelmente, o Swarm é a plataforma mais poderosa e flexível no momento. A ferramenta é *free software*,

contém tutoriais, exemplos e uma ampla comunidade de discussão. O ambiente está disponível no endereço <http://www.swarm.org/>.

A Figura 10 apresenta a interface gráfica da ferramenta e elementos de controle em um modelo de simulação. A Figura 11 apresenta a interface gráfica, evolutiva, do modelo da ratoeira seguido dos gráficos do comportamento desenvolvido no decorrer do tempo.

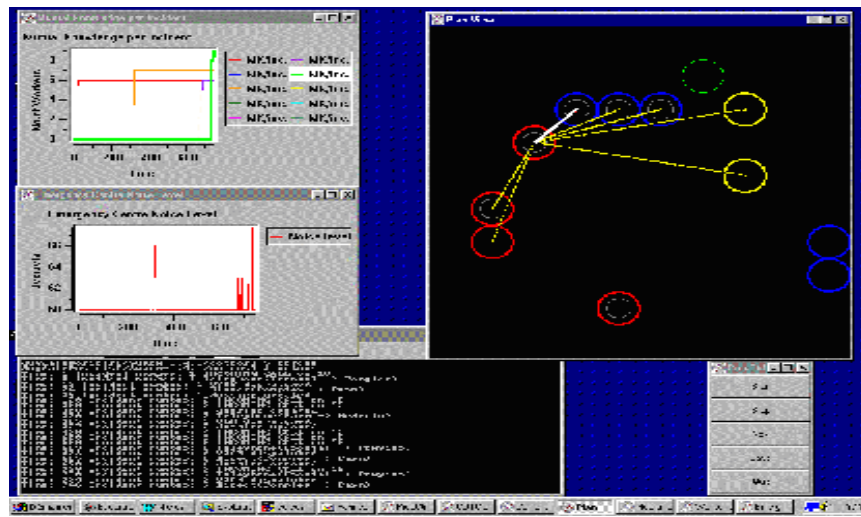


Figura 10 - Simulação desenvolvida no ambiente Swarm¹⁶

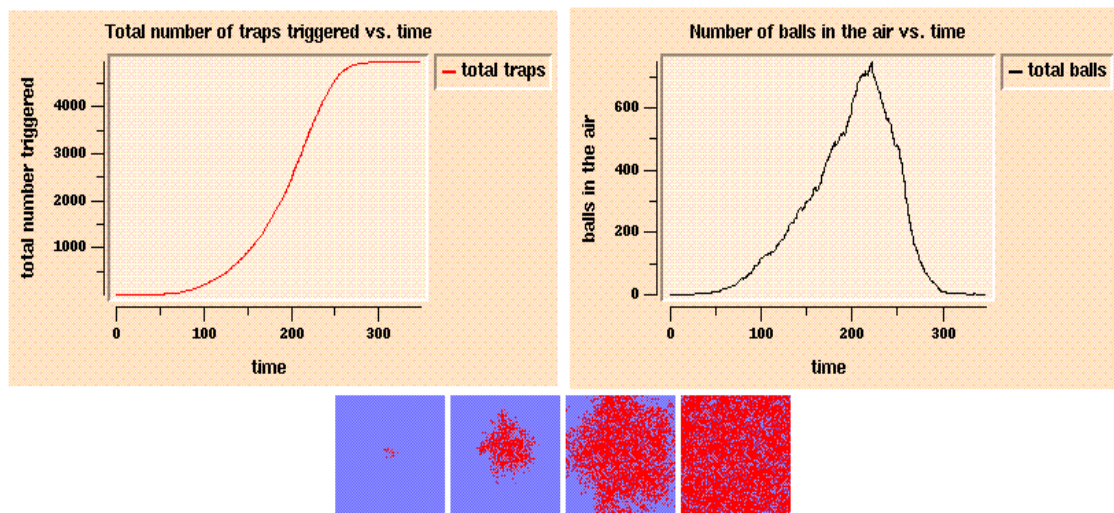


Figura 11 - Modelo de simulação ratoeiras – Swarm (acima: gráfico de comportamento do modelo; abaixo: visualização da seqüência de interações entre os agentes no decorrer do tempo)

¹⁶ Obtido de: “An Evaluation of Seven Software Simulation Tools for Use in the Social Sciences”, Julie Dugdale, GRIC-IRIT, Toulouse, France. Acessado em 12.05.06
<http://www.irit.fr/COSI/training/evaluationoftools/Evaluation-Of-Swarm.htm>

3.3 Repast

Repast (*Recursive Porous Agent Simulation Toolkit*) (Collier, et al., 2003), é um kit de ferramentas para modelagem de agentes. Foi desenvolvido pela University of Chicago, posteriormente mantido pela Argonne National Laboratory e, atualmente, é gerenciado pela Repast Organization for Architecture and Development (ROAD). O Repast é um kit ferramenta “*free open source*”. O Repast emprega muitos dos conceitos do Swarm, diferenciando deste no sentido de que tem implementações em várias linguagens e é construído com características adaptativas, tais como algoritmo genético e regressão. Os modelos também podem ser escritos em várias linguagens (incluindo Java, C#, Managed C++, Visual Basic.Net, Managed Lisp, Managed Prolog, and Python scripting). Segundo o ROAD a meta do Repast é

“Our goal with Repast is to move beyond the representation of agents as discrete, self-contained entities in favor of a view of social actors as permeable, interleaved, and mutually defining; with cascading and recombinant motives. We intend to support the modeling of belief systems, agents, organizations, and institutions as recursive social constructions.”

A ferramenta é *free software*, contém tutoriais, exemplos e uma ampla comunidade de discussão. Mais informações e o download podem ser adquiridas no endereço <http://repast.sourceforge.net/index.html>. A visualização do Repast é muito semelhante a ferramenta Swarm, afinal esse é uma variação do primeiro, como pode ser visto na Figura 12.

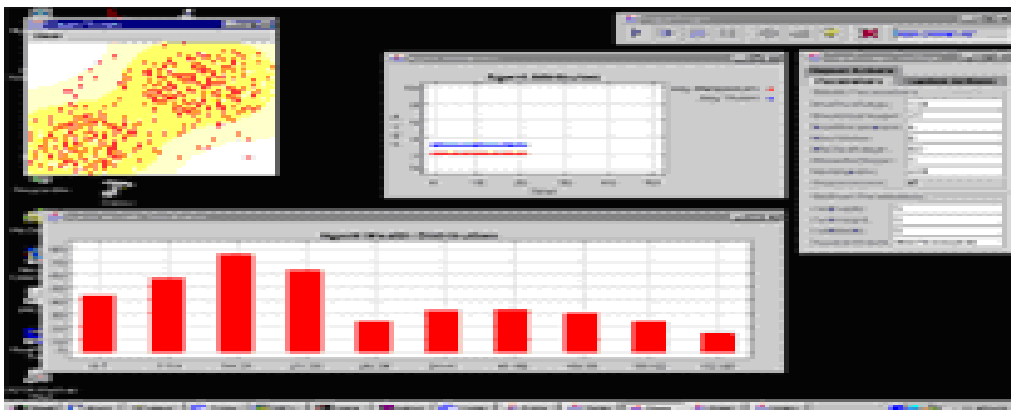


Figura 12 - Simulação desenvolvida no ambiente Repast

3.4 AgentSheets

AgentSheets (Repenning, 1993) é uma ferramenta de simulação e programação orientado a domínio, baseada em agentes que atuam sobre uma planilha. Ao invés das células da planilha serem ocupadas por números, são ocupadas por agentes. A simulação, então, ocorre sobre essa grade na qual vivem os agentes. A ferramenta AgentSheets usa o paradigma da programação visual, isto significa que não há um código baseado em texto, todo o desenvolvimento é feito através de uma interface gráfica (arrastando e colando elementos da caixa de ferramentas, etc.), conforme Figura 13. Os agentes são criados numa “galeria” e os comportamentos associados a eles são especificados por um conjunto de métodos e regras agrupados, Figura 14. As regras são baseadas no princípio do “SE” e “ENTÃO”. Cada regra representa um número de ações que são executadas, caso o valor da regra seja verdadeiro. É disponibilizada uma caixa de *ações* e outra de *condições* para serem aplicadas às regras de comportamento sobre os agentes.

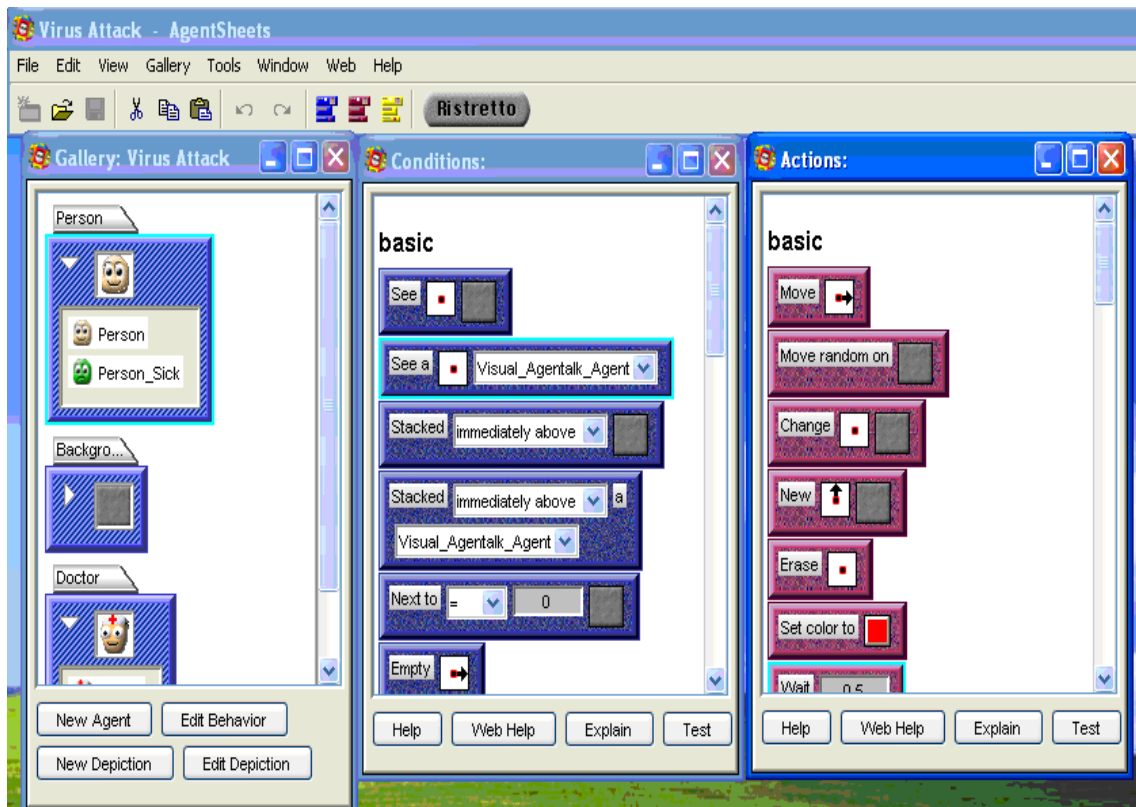


Figura 13 - Galeria de definição de agentes, caixa de condições, caixa de ações – AgentSheets



Figura 14 - Ferramenta de composição de regras

Dessa forma, a operação e o entendimento para desenvolvimento de modelos de simulação se tornam muito rápidos e fáceis. A Figura 15 apresenta a interface principal do ambiente – modelo em execução.

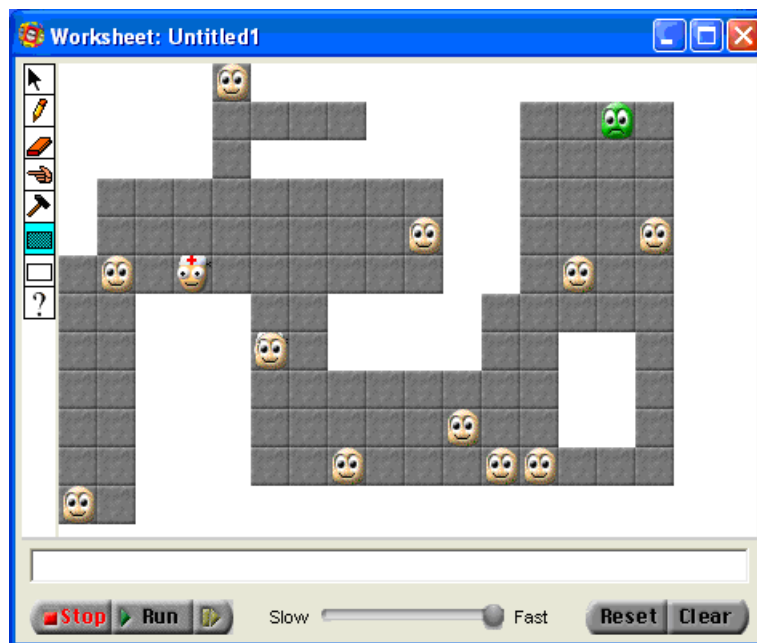


Figura 15 - Grade de distribuição dos agentes para executar a simulação

Entretanto, à medida que o modelo de simulação se torna mais profundo ou sofisticado, a fragilidade do AgentSheets se torna aparente. Considerando os modelos de simulação em

ciências sociais, duas limitações são particularmente visíveis e podem causar problemas com o uso dessa ferramenta: um agente não pode enviar informação para outro agente (por exemplo: transmitir uma quantidade ou o valor de um atributo interno); um agente não pode mudar o atributo de outro agente (é possível o agente examinar os atributos dos agentes vizinhos adjacentes a ele, mas não mudar - isso pode tornar problemático, se no modelo simulado houver a necessidade de influência de um agente sobre outro). Sem dúvida, há várias formas de trabalhar evitando esses problemas, mas se a necessidade de uso dessas situações se tornar muito freqüente, possivelmente tornará o modelo complicado e ineficiente. Alguns outros problemas identificados no uso dessa ferramenta são:

- dificuldade na identificação do posicionamento do agente sobre a grade;
- só é possível movimento em células adjacentes, não é possível executar saltos;
- como os agentes vivem sobre as grades, eles são exatamente do tamanho da grade;
- para grande quantidade de agentes é necessário definir os agentes com tamanhos pequenos. Os agentes em tamanho grande são úteis quando há necessidade de observar detalhes;
- a ferramenta foi idealizada para ser executada principalmente em Mac, a versão para Windows é ainda bastante acanhada e não dispõe de todos os recursos mencionados;
- os agentes não têm identificação única e nem é possível classificá-los numa família para estabelecer regras de comportamento específicas;

Por outro lado, é muito fácil, no AgentSheets, gerar um *applet* do modelo de simulação para ser executado no browser web. A ferramenta AgentSheets só está disponível na versão comercial, podendo ser acessada no endereço <http://www.agentsheets.com/>.

3.5 NetLogo

O NetLogo (Wilensky, 1999-2007), (Tisue, et al., 2004), (Tisue, et al., 2004(a)) é um ambiente integrado de modelagem multiagente para simulação de fenômenos naturais e sociais. É particularmente bem situado para modelagem de sistemas complexos desenvolvidos no decorrer do tempo. O NetLogo é uma geração decorrente da linguagem de modelagem multiagente que começou com o StarLogo. Sua linguagem é um dialeto Logo estendido para apoiar a programação de agentes e processamento paralelo. O NetLogo é escrito em Java e

pode ser executado em várias plataformas (Linux, Windows, Mac). Modelos individuais podem ser executados como *applets* Java dentro de um browser web. O Software pode ser carregado gratuitamente por meio do endereço <http://ccl.northwestern.edu/netlogo/>.

O NetLogo inclui uma interface gráfica para explorar, experimentar e visualizar os modelos, bem como linguagens de modelagem multiagente (MAML) usadas para modelos de autoria. Tais linguagens possibilitam ao usuário, facilmente, criar e manipular centenas ou milhares de agentes gráficos que trabalham paralelamente, e definir simples regras que governam seu comportamento e seu ambiente. Isso torna possível explorar as conexões entre o comportamento no micro-nível de indivíduos e modelos em macro-nível que emergem das interações de muitos indivíduos. O ambiente NetLogo tem dois tipos de agentes: estacionários, chamados de “*patches*”, e móveis, chamado de “*turtles*”. Há também o observador cujo objetivo é criar os agentes e gerenciar as ações desenvolvidas por esses, a Figura 16 mostra a interface gráfica do ambiente. Os agentes têm comportamento simples associado a eles, tais como procurar ou evitar um objeto ou outro agente. Através dessas regras simples, entretanto, surge um comportamento emergente complexo.

A linguagem do NetLogo foi desenvolvida com o princípio de “*low threshold*”, isto é, foi projetada para que pessoas com pouca experiência em linguagem de programação pudessem rapidamente aprender e construir seus próprios modelos. A Figura 17 apresenta um fragmento de código em NetLogo. Além disso, o ambiente, em si mesmo, inclui uma interface gráfica com avançadas características de visualização, ferramentas especializadas tais como BehaviorSpace, que possibilita aos usuários executar múltiplos experimentos e automaticamente coletar os dados para análise posterior. O NetLogo tem uma extensa documentação e tutoriais, bem como biblioteca de modelos em várias áreas do conhecimento, que podem ser usados e modificados.

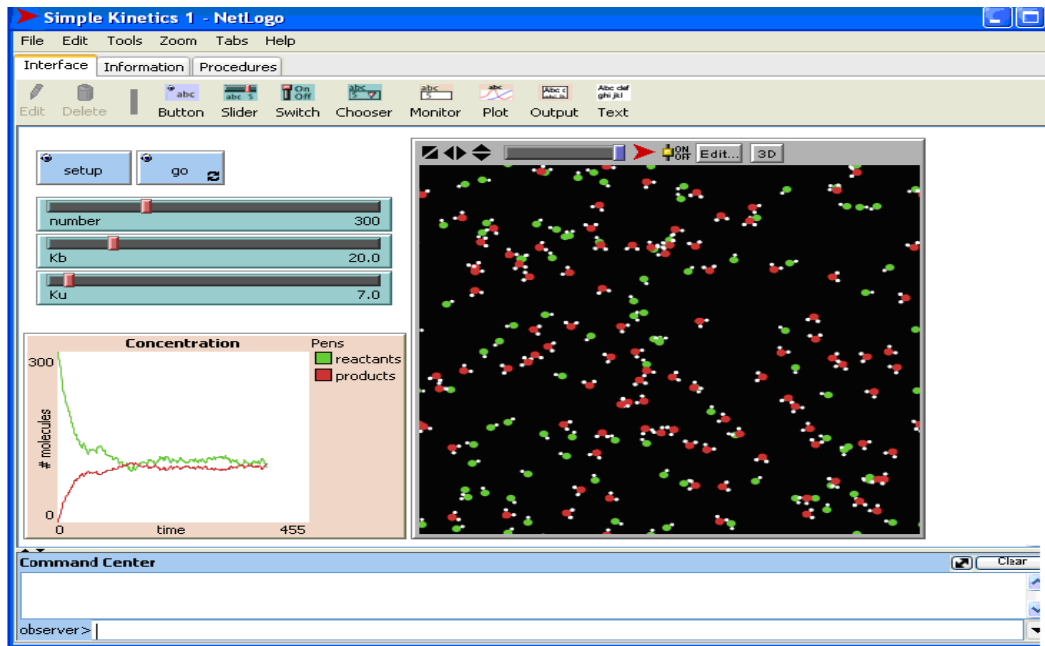


Figura 16 - Interface gráfica NetLogo

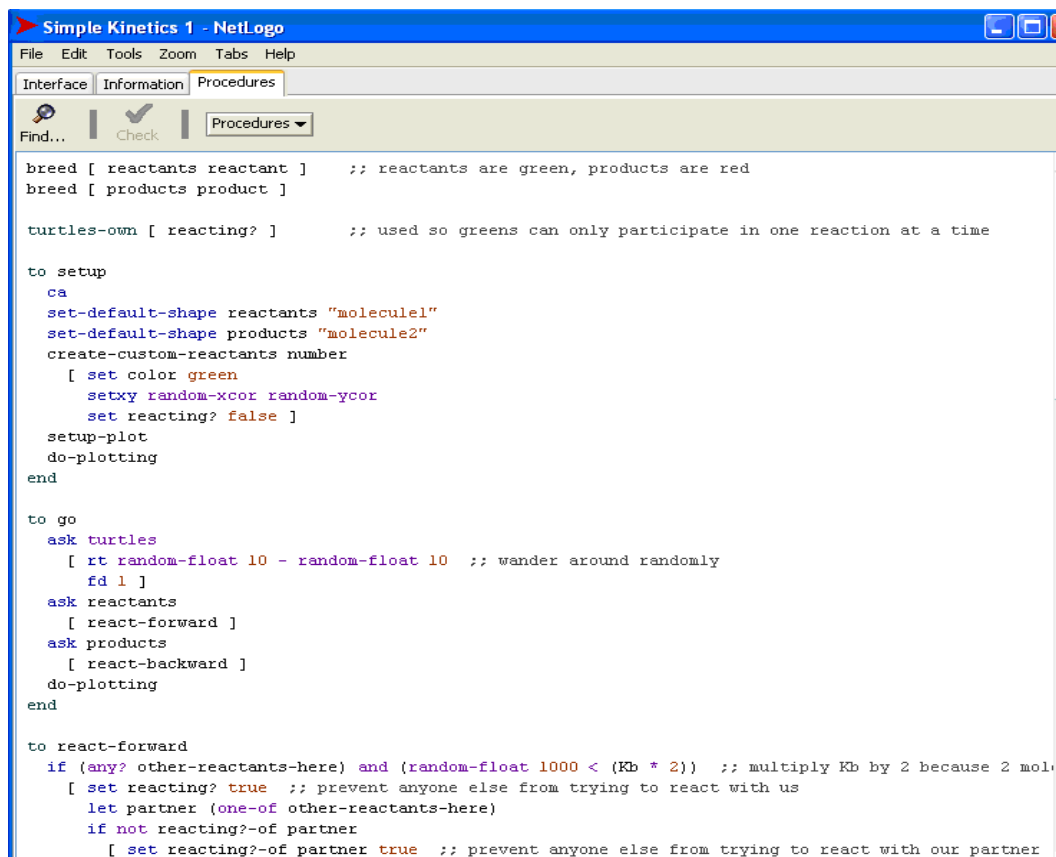


Figura 17 - Fragmento de código em NetLogo

3.6 Klik&Play

Apesar de não ser caracterizado como uma ferramenta de modelagem baseado em agentes, o Klik&Play¹⁷ será abordado neste trabalho pela contribuição para o desenvolvimento da nossa proposta e também pela possibilidade que vimos nele em atender a perspectiva da modelagem de simulação multiagente. O software Klik&Play foi desenvolvido pela empresa Clickteam, sendo seus fundadores François Lionet e Yves Lamoureux, e foi lançado em 1994. Este software foi concebido originalmente para autoria de jogos, sem a necessidade de conhecimento sobre programação convencional de computadores. Essa possibilidade permitia a criação de jogos de entretenimento e programas educacionais.

Podemos dizer que o Klik&Play é um sistema de programação orientado a declarações. O sistema permite ao usuário jogar e criar jogos, sendo pioneiro nessa linha. A programação no Klik&Play é realizada por meio de dois editores principais: um editor de cenas onde podemos descrever os personagens e as mobílias dos vários atos (níveis) de um jogo e um editor de eventos que possibilita descrever eventos para o jogo baseado na interação entre os personagens. O editor de eventos é como uma planilha, o que pode acontecer no jogo é mostrado horizontalmente (são as condições) e os personagens são mostrados verticalmente. Para definir o comportamento (ou ação) de um objeto quando uma condição ocorre, basta assinalar o espaço onde ocorre a interseção, veja Figura 18.

Todos os elementos (personagens e mobílias) dos jogos são entidades com propriedades de movimentação, interação, comunicação, etc. Por exemplo, um personagem possui dois tipos de propriedades importantes para a construção de um jogo, movimento e animação. Os elementos são armazenados em bibliotecas de onde podem ser arrastados com o mouse para compor o cenário (frame) onde está sendo criado o jogo. Na parte inferior da Figura 19, é possível observar uma aba com várias opções de elementos: criar agentes, criar cenário de fundo, contadores, relógios, etc.. Do lado esquerdo, fica disponível uma biblioteca de modelos de agentes, na parte superior da mesma figura, é possível acompanhar os agentes criados para o modelo e, na parte central, a arena onde ocorrem as interações.

Embora os tutoriais sobre a ferramenta não façam qualquer referência à multiagente, uma rápida inspeção nos permite inferir que podemos entendê-lo desta maneira. Sendo,

¹⁷É possível acessar e fazer download da ferramenta acessando <http://www.clickteam.com/eng/>. A ferramenta esta sendo distribuída por “Clickteam.com”.

portanto, adequado para criação de modelos, exceto pelas limitações inerentes à concepção do sistema. O Klik & Play possui milhares de usuários ao redor do mundo, tendo sido publicado em mais de catorze línguas e distribuído em muitos países. Para ser utilizado em escolas ou para fins educacionais, existe uma versão *freeware* “Klik&Play for schools”, <http://www.clickteam.com/English/klik&play.htm>. Países como a França têm adotado o seu uso, impulsionados pela facilidade de criação de jogos, uma abordagem lúdica para a aprendizagem. As escolas se apropriaram com facilidade deste ambiente, sem, no entanto, haver relatos de uso com esse enfoque de simulação multiagente.

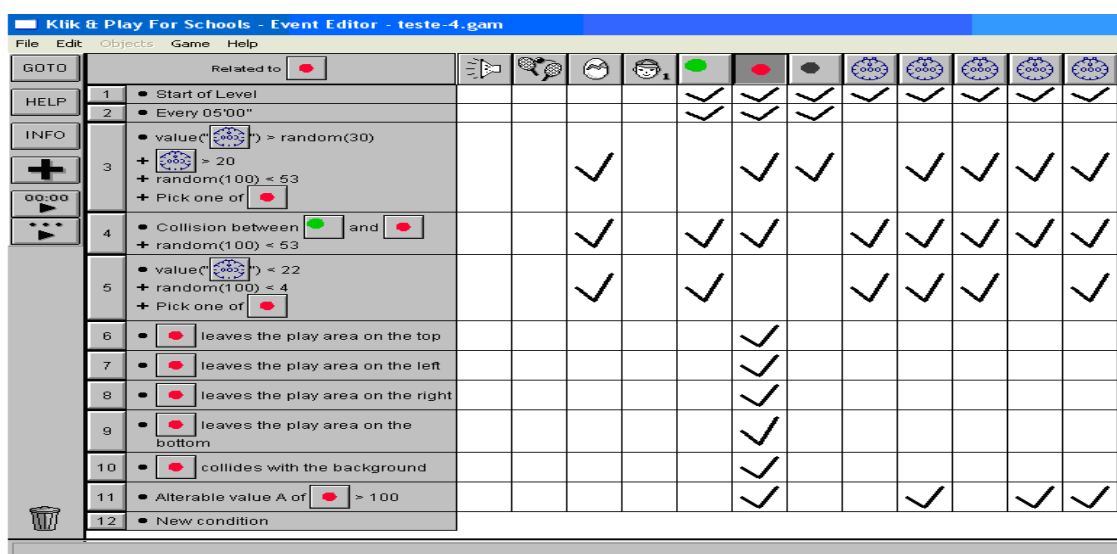


Figura 18 - Editor de Eventos Klik&Play

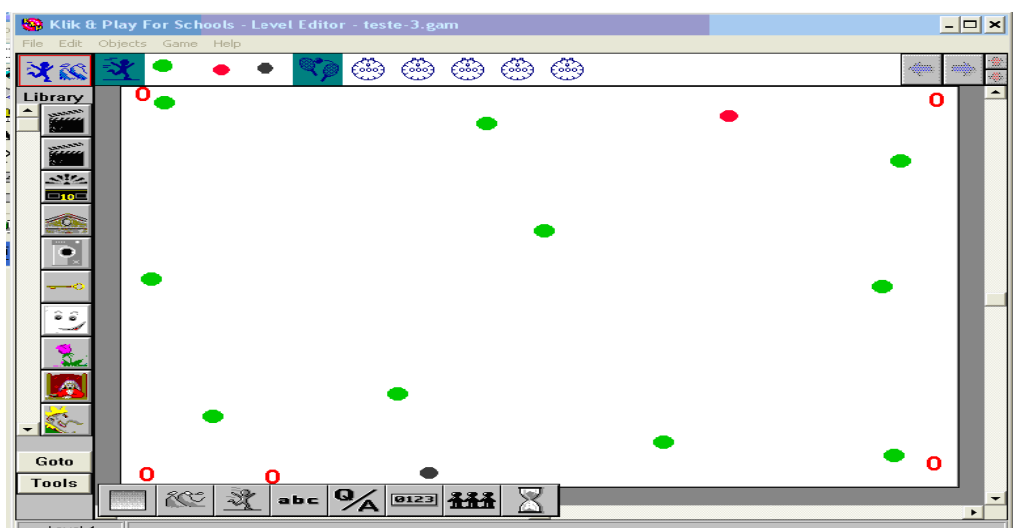


Figura 19 - Editor de nível / cenário execução modelo

3.7 SIMULA

SIMULA (Frozza, 1998) é um sistema que se destina ao desenvolvimento de aplicações com o uso de agentes reativos. A Figura 20 apresenta a funcionalidade do sistema. O usuário define uma situação inicial que determina como os agentes agirão. Feito isso, a ferramenta se encarrega de executar a simulação e apresentar uma situação final, atingida pela atuação dos agentes. O SIMULA destina-se a atender usuários que tenham conhecimento da tecnologia do uso de agentes para a construção de sistemas reais, científicos e simulações. E que tenham, também, um embasamento do que são agentes e de como eles podem atuar em sistemas.

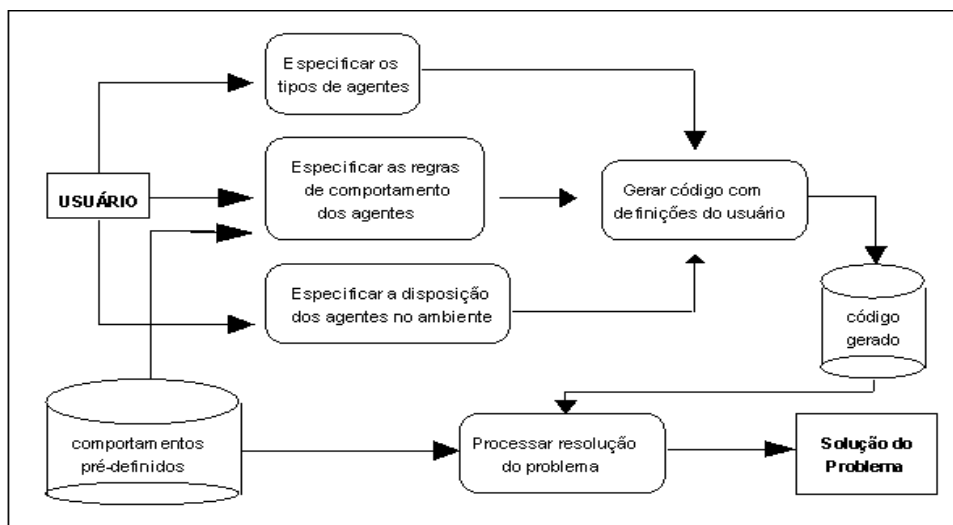


Figura 20 – Funcionalidades do ambiente SIMULA

O sistema é estruturado em dois módulos: o de definição e o de execução. O módulo de definição é composto de três etapas (Frozza, 1998), (Alvares, et al., 1997):

- *especificação dos tipos de agentes que farão parte da simulação*: nessa etapa o usuário especifica quais serão os agentes envolvidos na simulação, associando um nome para cada tipo de agente, uma figura e uma quantidade. Determina a percepção do agente, isto é, indica o raio sob o qual o agente consegue perceber outros agentes. Por exemplo, se um agente A tiver uma área de percepção igual a 2, significa que qualquer agente que estiver a duas posições do ambiente, ao redor do agente A, pode ser percebido por este;
- *especificação das regras de comportamento dos agentes*: as regras de comportamento são especificadas para cada tipo de agente e são elas que definem as ações dos agentes

no processo de simulação. Existem diversos comportamentos pré-definidos que podem ser utilizados no ambiente SIMULA. Esses são classificados em 3 tipos básicos: ativos, passivos e de estado. Os comportamentos ativos permitem a um agente atuar no ambiente e modificar o seu estado (por exemplo, “segue_maior_gradiente” e “foge_de_agente”); os comportamentos passivos permitem a um agente se movimentar no ambiente; e os comportamentos de estado estão relacionados à percepção do agente e às condições de execução de ações (“percebe_agente” e “atinge_agente”). Esta definição pode ser acompanhada pela Figura 21;

- *especificação da disposição dos agentes no ambiente*: possibilita ao usuário determinar a posição inicial de cada agente no ambiente no qual atuará. Essas podem ser fixas ou aleatórias.

O segundo módulo é o processamento e a execução das definições realizadas na etapa anterior. Esse módulo é composto por duas partes: uma encarregada de gerar o código e a outra de executar o código.

- *geração de código*: tomando por base as definições do usuário para a sua aplicação, associados com os comportamentos pré-definidos do ambiente é gerado um código, na linguagem Java, que será responsável pelo processo de simulação;
- *execução*: corresponde à execução do código gerado e representa a ativação do processo de resolução da aplicação definida. Este processo é mostrado de forma gráfica, através de uma simulação da atuação dos agentes no ambiente. Cada agente é representado por uma letra do nome do agente com uma cor, definida pelo usuário. O término do processo indica que os agentes atingiram uma situação final, que pode ser considerada como uma solução. O usuário acompanha a atuação dos agentes e fica a seu cargo analisar se o processo gerado, com as suas definições, equivale à aplicação modelada

A atuação dos agentes em um ambiente é modelada através de regras, que determinarão quais comportamentos dos agentes devem ser executados e sob que condições. Os comportamentos equivalem à chamada de um procedimento que executará uma ação do agente. Os elementos que compõem as regras são:

- *pré-condição* : são testes condicionais sobre o estado corrente do agente, que acionam ou não a execução da regra de comportamento associada ao agente;

- *ação-ativada* : representa o(s) comportamento(s) a ser(em) executado(s), se a pré-condição for verdadeira. O usuário pode utilizar os comportamentos ativos pré-definidos;
- *ação-condicional* : é um comportamento que representa uma condição dentro da execução da ação ativada. Por exemplo, define-se que o agente deve realizar um movimento randômico, mas antes de realizar esse comportamento, define-se que o mesmo deve verificar se há obstáculos em seu caminho;
- *pós-condição* : são os efeitos causados pela execução dos comportamentos. O usuário pode efetuar alterações ou atualizações de variáveis.
- *prioridade* : o usuário pode, através da prioridade, estabelecer a ordem de execução das regras, porque, a cada ciclo, o agente realiza apenas uma ação/comportamento.

O ambiente no SIMULA é definido a partir da especificação de agentes que o modelam, chamados agentes de ambiente por meio da colocação e remoção de pistas ou marcadores em uma posição, ação permitida a qualquer agente. Os agentes podem definir, modificar e utilizar variáveis (como energia ou tempo de vida). Deste modo, a interação de agentes com um ambiente é realizada através da interação entre agentes ativos, agentes de ambiente e marcadores.



Figura 21 - Janela de definição das regras de comportamento dos agentes

Após gerado o código, baseado nas regras de comportamento definidas pelo usuário, é possível executar a simulação do modelo. O código é gerado obedecendo a prioridade estabelecida nas regras de comportamentos. A Figura 22 mostra a interface de visualização do

ambiente. Cada ciclo¹⁸ corresponde à execução de uma regra de comportamento por cada tipo de agente.

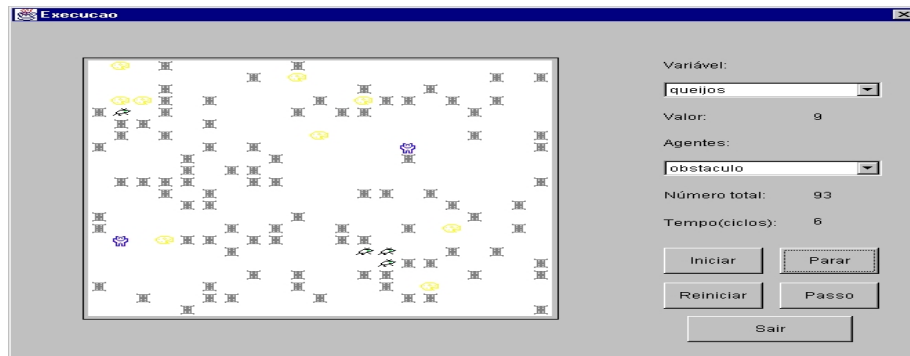


Figura 22 - Janela de execução do modelo de simulação - Exemplo Pets

Exemplo do uso do sistema SIMULA para simular um sistema de resgate com uso de robôs pode ser visto em (Bianchi, et al., 2000), a simulação em eventos discretos pode ser consultado em (Botelho, 2003). Para mais detalhes sobre o sistema consultar: <http://simula.sourceforge.net/>. O sistema SIMULA é um *open Java source* software, e foi projetada para auxiliar estudantes interessados em aprender sobre sistemas multiagente e simulação.

3.8 Considerações sobre as ferramentas

Baseado no levantamento das ferramentas visitadas e nos requisitos para a modelagem multiagente apresentados na sessão 2.2.3, podemos afirmar que a maioria dos sistemas de modelagem baseada em agentes segue o paradigma de “*framework* e biblioteca“. O *framework* – conjunto de conceitos para projetar e descrever a modelagem baseada em agentes; a biblioteca – conjunto de implementações de software para fornecer as ferramentas de simulação.

O sistema pioneiro foi o Swarm, cujas bibliotecas foram escritas em Objective-C e, posteriormente, disponibilizadas o Java Swarm. O Repast surgiu como uma implementação Java do Swarm, mas divergiu desse em relação às opções oferecidas, tanto de linguagem quanto de ferramentas. Recentemente o MASON¹⁹ está sendo desenvolvido como uma nova biblioteca de simulação multiagente escrito em Java. Esses ambientes têm se difundido pelo

¹⁸ Definir o que é ciclo – execução de todas as regras (rotinas)

¹⁹ Apesar de não ter sido relacionado, apresenta características similares aos ambientes Swarm e Repast.

fato de fornecerem uma padronização nos projetos de software e ferramentas para construção de modelos de simulação multiagente, sem limitar a complexidade ou tipo do modelo.

Entretanto, apresentam algumas limitações: a dificuldade de programar em Java, C++ ou Objective-C não é devido apenas a linguagem em si, mas também devido à complicação do ambiente onde essas linguagens são desenvolvidas (se baseado em linha de comando ou baseado num GUI). Carregar as bibliotecas ou as ferramentas para a modelagem já é uma barreira para programadores iniciantes, antes mesmo de tratar com as dificuldades próprias da linguagem de programação; ferramentas insuficientes para construir modelos, principalmente as relacionadas à representação do espaço; ferramenta insuficiente para executar e observar o experimento da simulação; além da falta de ferramenta para documentar o modelo.

Os sistemas descendentes da família Logo: StarLogo e NetLogo, seguiram um caminho completamente diferente. Sendo sua primeira proposta fornecer uma ferramenta de alto nível que possibilitasse, aos usuários, desde estudantes do ensino médio até pesquisadores, construir e aprender através da modelagem baseada em agentes. Neste intuito, os dois sistemas disponibilizam uma ótima interface gráfica e barra de ferramentas para controle do modelo (botões para acionamento e definição de variáveis, plots para construção de gráficos, monitores, caixa de mensagens, etc) e a linguagem derivada do Logo.

AgentSheets é um sistema comercial. A ferramenta usa o paradigma da programação visual que facilita a programação. Por ser uma ferramenta comercial, não tivemos acesso a maiores informações sobre seu funcionamento interno. Quanto ao Klik&Play, como dito acima, seu propósito é a construção de jogos, mas vimos nele uma possibilidade de modelagem multiagente.

É importante ressaltar que os sistemas examinados continuam sendo desenvolvidos e revisados. Nossa avaliação considera apenas as observações realizadas em: MASON version 10, Repast 3.1, Swarm 2.2, StarLogo 2.2, AgentSheets 2.5 e NetLogo 3.0. Algumas de nossas observações podem ter sido superadas nas versões atuais.

3.9 Por que desenvolver uma nova ferramenta para modelagem multiagente?

Embora existam muitas ferramentas para descrever modelos de simulação baseados em agentes, para muitos usuários, o desenvolvimento de tais modelos ainda representa um

obstáculo. Segundo (Ropella, et al., 2002), (Minar, et al., 1996), (Railsback, et al., 2006), esse problema resulta, em grande parte, da ausência de treinamento e habilidade no uso de software por pesquisadores (usuários) de muitos campos que usam a modelagem baseada em agente (por exemplo: biólogos, ecólogos, economistas, cientistas políticos, sociólogos, entre outros). Todas as ferramentas citadas, exceto o StarLogo TNG e AgentSheets, utilizam linguagem de programação baseada em texto para descrever o modelo.

Entendemos que, além da falta de treinamento dos usuários, os ambientes ainda podem proporcionar metáforas e ferramentas mais convenientes ao trabalho dos “resolvedores” de problemas. Fazendo um apanhado sobre cada uma das ferramentas citadas, podemos argumentar que os ambientes Swarm, Repast e MASSON, apresentam algumas limitações: dificuldade de uso, devido à necessidade de aprofundamento em linguagem de programação (Objective C, Java, Python, C#); ferramentas insuficientes para construir modelos, principalmente as relacionadas à representação do espaço; ferramenta insuficiente para executar e observar o experimento da simulação; além da falta de ferramenta para documentar o modelo.

Com relação aos ambientes StarLogo e NetLogo, ambos usam uma linguagem de programação própria derivada do Logo. Apesar da aparente facilidade da linguagem, à medida que os modelos vão ficando mais sofisticados, é percebida uma exigência no domínio do programa e uma capacidade de programação que estão, obviamente, além do que se pode esperar de alunos do ensino médio e usuários pouco familiarizados com programação.

Apesar da facilidade de programação visual do AgentSheets, a ferramenta se torna frágil à medida que os modelos ficam mais elaborados. As principais dificuldades apresentadas por esta ferramenta são: dificuldade na identificação do posicionamento do agente sobre a grade; os agentes só se movimentam em células adjacentes, não é possível saltos; como os agentes vivem sobre as grades, esses são exatamente do tamanho da grade; a ferramenta foi idealizada para ser executada principalmente em Mac, a versão para Windows é limitada; e principalmente, os agentes não têm identificação e nem é possível classificá-los numa família para estabelecer regras de comportamento específicas.

Além das características descritas para definição dos agentes e ambientes, sessão 2.2.3, destacamos a necessidade de algumas ferramentas conceituais para facilitar o acompanhamento do modelo e possibilitar a extração de dados comportamentais e análise

(gráficos, contadores, análise dos resultados, etc.). Ferramentas com esse teor facilitam a aprendizagem

Diante desses fatos, um dos aspectos que ainda requer estudos e desenvolvimentos é a construção dos modelos, segundo a ótica de um estudante (observador de sistemas) que seja leigo em programação. Em geral, a definição de famílias de agentes, suas propriedades relevantes, as interações interfamiliares e intrafamiliares, requer a escrita de código em alguma linguagem de programação, fator que impossibilita que muitos usuários leigos em computação utilizem os ambientes. Além disso, a observação do funcionamento desses sistemas ainda carece de melhores ferramentas de acompanhamento e análise.

Capítulo 4

4 Resolução cooperativa de problemas

A resolução de problemas é importante em muitos campos, tanto de domínio específico quanto genérico. Normalmente a compreensão de um problema requer a identificação dos dados oferecidos, o comportamento dos elementos componentes do problema, as restrições, as condições em que ocorrem entre os vários elementos e as relações entre elas. Invariavelmente a resolução de um problema requer alguma forma de partição do problema, isto é, divisão em problemas menores. Desse modo, métodos de resolução de problemas têm sido desenvolvidos há muitos anos. Segundo (Deek, 1997), (Deek, et al., 2000), apresentamos abaixo uma cronologia das principais metodologias de resolução de problemas:

- em (1910) o método científico de Dewey (1859 - 1952);
- em (1926) a visão criativa do método não sistemático de Wallas (1858 - 1932);
- em (1945) o trabalho de Polya (1897 - 1985), destacando a heurística por meio de: compreensão do problema, estabelecimento de um plano, execução do plano e retrospecto;
- em (1953) o método criativo de Osborn (1888 - 1966) sendo continuado (1967) por Parnes cujos passos são: “*the mess*”- significa dizer tudo sobre a situação problema e criar uma declaração que melhor defina o problema e, em sessões de *brainstorming*, encontrar a idéia, encontrar a solução e planejar a ação;
- em (1960) o trabalho de Simon (1916 - 2001) que viu o processo de resolução de problemas como quatro capacidades cognitivas²⁰: inteligência, projeto, escolha e implementação;
- em (1997) o método Dual Commom Model (DCM), desenvolvido por (Deek, 1997), integra a metodologia de resolução de problemas e a tarefa de desenvolvimento de

²⁰ Inteligência – capacidade de reconhecer a existência do problema, coletar informações pertinentes ao problema, e produzir uma acurada definição sobre o problema; Projeto – gerar possíveis soluções alternativas, incluindo soluções preliminares para cada projeto; Escolha - selecionar e implementar uma satisfatória solução e identificar alternativas; Implementação – executar a escolha para produzir uma solução.

programa com a habilidade e conhecimento cognitivo necessário a cada estágio do processo de resolução, sendo esse método derivado do modelo de Polya.

Métodos mais recentes foram desenvolvidos, visando a auxiliar estudantes de matemática, ciências e engenharia no refinamento do processo de resolução de problemas²¹ (subproblemas). Muitos outros modelos surgiram derivados dos já mencionados, mas não apresentaram grandes mudanças.

Em várias áreas (engenharia, informática, medicina, ensino, biologia, etc.) é possível ver o crescimento da complexidade dos problemas. Crescimento esse devido a disponibilidade de ferramentas para testar novas abordagens e estratégias e, também, devido a possibilidade de analisar aspectos antes não acessíveis, decorrente da falta de recursos e conhecimento. Com isso, um fator adicional tem crescido em importância: a colaboração. De fato, a crescente complexidade das aplicações tem provocado a necessidade do uso de equipes ou grupos para desenvolver estratégias e soluções para os problemas, uma vez que se torna inapropriado um único indivíduo lidar com um grande número de variáveis, etapas e métodos com adequada experiência ou nível de qualidade.

Um grupo que desenvolve o planejamento de uma atividade que resolverá um problema existente está, por definição, engajado numa resolução cooperativa de problema. Um modelo de resolução de problema cooperativo incluirá não apenas passos de resolução de problemas genéricos, tarefas de domínio específico, e requisitos de habilidades cognitivas, mas também a comunicação e a coordenação das atividades requeridas pelo ambiente cooperativo (Deek, et al., 2000).

²¹ É importante destacar que a distinção entre aprendizagem baseada em problemas e outras formas de aprendizagem ativa ou cooperativa muitas vezes é obscurecida porque elas partilham certas características comuns e abordagens híbridas, como métodos de adaptação de instrutores para sessões particulares. Entretanto, um componente essencial da aprendizagem baseada em problemas é que o conteúdo é introduzido dentro do contexto de um problema do mundo real. O contraste está na forma como as estratégias e os conceitos são apresentados. No ensino “tradicional” os problemas são apresentados no final do capítulo. Na abordagem de resolução de problemas os problemas precedem o capítulo. Na aprendizagem baseada em problemas, os estudantes trabalham em pequenos grupos, identificam o que eles sabem e, mais importante, quanto eles não sabem e devem aprender (questões de aprendizagem) para resolver o problema. Esses são os requisitos para o entendimento do problema e a tomada de decisão requerida para resolver o problema. A natureza do problema evita respostas imediatas. Os estudantes devem ir além dos livros textos para adquirir conhecimento, em outras fontes e entre seus companheiros de grupo. O primeiro papel do instrutor é instigar o grupo para a aprendizagem, e não fornecer respostas fáceis. Com a mudança advinda no formato de condução, diferentes formas de avaliação são exigidas, tais como avaliação no grupo.

Diante da liberdade de escolha de um método, Hohmann [1997, apud (Tommarello, 2002)] observa que o modo como o grupo apropria de um método num ambiente colaborativo difere substancialmente do modo como um indivíduo aplicaria esse mesmo método. Há muitos pontos a serem observados no trabalho colaborativo em relação à comunicação e a relação entre os membros do grupo, sendo um importante aspecto a forma como o indivíduo interage. No trabalho cooperativo, a comunicação tem papel decisivo, sendo importante também a identificação de tarefas, distribuição de tarefas, orientação do grupo, desenvolvimento dos planos, coordenação dos resultados produzidos pelas equipes, a integração das soluções, entre outros.

Grupos cooperativos parecem capazes de tratar com tarefas complexas mais eficientemente do que um indivíduo apenas. , particularmente porque um grupo terá uma ampla gama de habilidade e capacidades a mais do que um indivíduo (Finnegan, et al., 1996). Em um grupo, os membros envolvidos são afetados pela colaboração, porque um indivíduo aprende as habilidades e capacidades de outro membro do grupo.

É importante destacar que num processo de resolução cooperativa de problemas é importante integrar metodologias de resolução de problemas gerais com programa de desenvolvimento de tarefas, com habilidades e conhecimento cognitivo necessários para cada estágio do processo. O processo de resolução cooperativa de problemas pode ser analisado por duas vertentes: o processo cognitivo considerando os fatores resultantes dessas interações; e a estrutura cooperativa necessária para suportar o processo. O mapa apresentado na Figura 23, mostra o contexto no qual se dão as ações do trabalho cooperativo, apontando os elementos necessários e como eles se relacionam. Neste trabalho focaremos sobre o segundo, questão em aberto nesses sistemas.

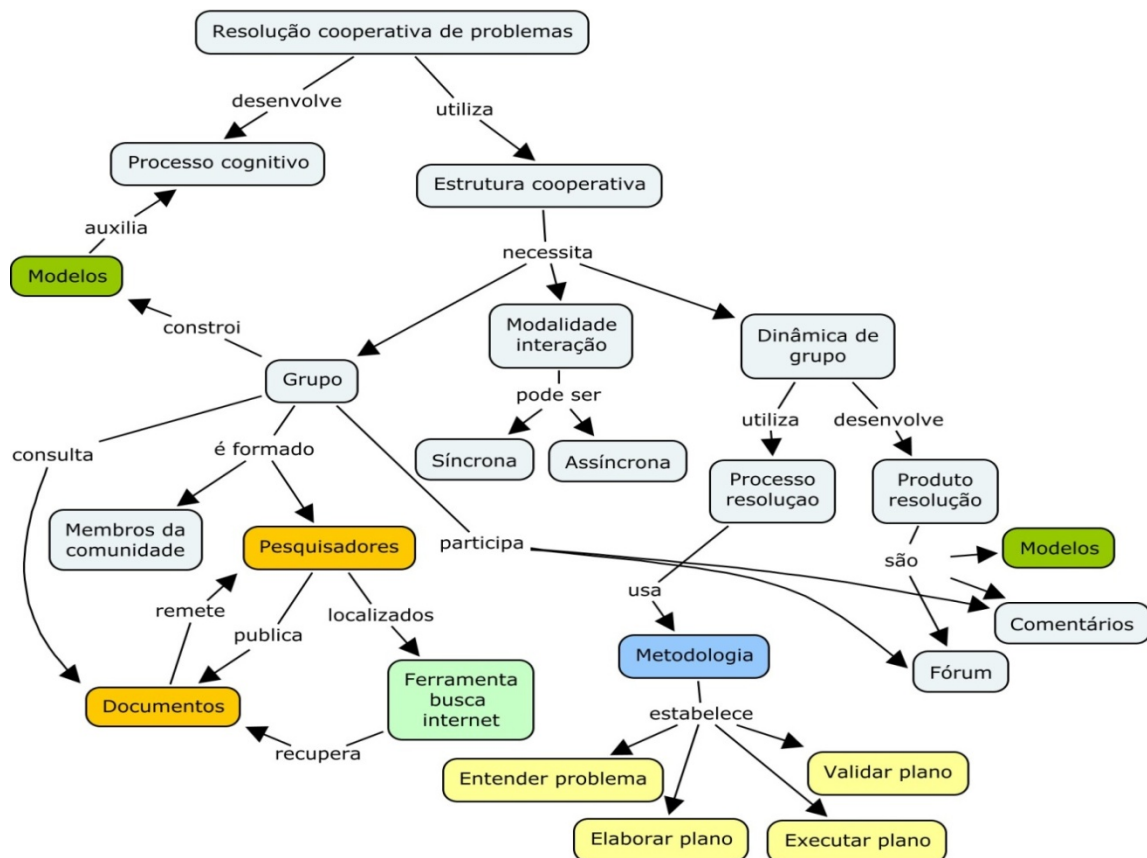


Figura 23 - Mapa conceitual do processo de resolução cooperativa de problemas

Processo cognitivo – um modelo de resolução cooperativa de problemas incluirá não apenas passos de resolução de problemas genéricos, tarefas de domínio específico, e requisitos de habilidades cognitivas, mas também a comunicação e a coordenação das atividades requeridas pelo ambiente cooperativo (Deek, et al., 2000). Durante o trabalho cooperativo há troca de experiência entre os membros envolvidos, de modo que eles aprendem as habilidades e capacidades um do outro. Hohmann (1997, apud (Tommarello, 2002)) observa que o modo como o grupo se apropria de um método de resolução num ambiente colaborativo difere substancialmente do modo como um indivíduo aplicaria esse mesmo método. Há muitos pontos a serem observados no trabalho colaborativo com relação à comunicação e com relação aos membros do grupo, sendo um importante aspecto a forma como o indivíduo interage com os outros.

Estrutura cooperativa – para apoiar o processo de resolução cooperativa três fatores são essenciais: a formação do grupo, a dinâmica de trabalho desse grupo e as ferramentas que apóiam essa interação.

Tratando de resolução cooperativa de problemas apoiada por computador, a área de sistemas *groupware* tem, amplamente, oferecido ferramentas para apoiar essa interação. Haja vista os ambientes existentes nessa linha e as facilidades por eles oferecidas para: gerenciamento de tarefas, compartilhamento de dados, interação síncrona (chat, vídeo conferência) e assíncrona (mensagens eletrônicas, fóruns), entre outras. Várias metodologias podem apoiar a dinâmica do grupo. Usando ferramentas apropriadas, podem ser levantadas questões em sessões de *brainstorming*, elaboração de planos e execução dos mesmos (a construção de modelos de simulação é uma alternativa), debate e votação das questões levantadas. Com relação à formação do grupo duas questões devem ser consideradas: quando os membros são previamente conhecidos e quando não são.

- membros conhecidos – nesse caso a formação do grupo ocorre agregando os elementos da comunidade que de uma forma ou de outra têm algum vínculo entre si. Esse é o caso que comumente ocorre nas empresas ou em ambientes educacionais. Para essa situação os ambientes que apóiam o trabalho cooperativo por computador atendem a essas necessidades de agregar pessoas.
- membros não conhecidos – nesse caso faz-se necessário um mecanismo de aproximação desses elementos. Os ambientes existentes que apóiam o trabalho cooperativo por computador ainda não dão conta dessa proposta.

4.1 Situando o problema

Observando o senso comum, quando as pessoas precisam de ajuda para resolver algum problema particular, procuram encontrar as pessoas certas para responder as suas perguntas. Mesmo quando temos uma pessoa em mente, ela pode não estar disponível e, então, outra alternativa deve ser procurada. Hoje em dia, devido ao fácil acesso à internet, esse recurso tem sido amplamente usado e, de modo geral, pode direcionar mais rapidamente as necessidades dos usuários. Essas buscas podem ser direcionadas a: informações adquiridas em base de dados (diretórios de pesquisa, currículos), usando filtros para fornecer apenas as informações relevantes; ou sistemas especialistas que automaticamente respondem questões previamente formuladas.

Entretanto, as atuais abordagens de contato não são apropriadas para a comunidade geral da internet. Primeiro, a alta tecnologia adotada por muitos sistemas existentes estão além

do alcance de muitos usuários. Segundo, a maioria dos sistemas supõe que os usuários devem saber de algum modo, onde encontrar a pessoa certa para a resposta (através de um endereço de email, local de trabalho, etc.). Porém, isso não é necessariamente verdade, pois podemos até saber que uma pessoa está na rede em algum lugar, mas sem o endereço completo ou sem conhecer o *site* no qual sabemos em que ela está hospedada, é praticamente impossível localizá-la. Um terceiro problema é que os sistemas de busca não estão totalmente integrados ao ambiente de *software* de trabalho cooperativo. Isto é, mesmo depois de um contato inicial com quem poderá responder a questão há uma dificuldade de se estabelecer a interação entre os interessados, pois eles podem não compartilhar o mesmo ambiente *groupware*.

Podemos concluir que, mesmo com as ferramentas de busca disponíveis e os ambientes *groupwares* para facilitar as interações, as pessoas têm grande dificuldade de contatar umas com as outras. Para pontuar essa questão tomamos dois fatores: como se forma o grupo num *groupware*?; e como encontrar os elementos do grupo?

Como nas interações presenciais, um pré-requisito essencial para a formação dos grupos, mesmo quando estão geograficamente dispersos, é que os componentes devem estar em contato uns com os outros. Os relacionamentos devem ser estabelecidos e muitas interações (incluindo as formais) devem ocorrer para que os elementos se encontrem e formem seus grupos. Ainda que haja um gargalo para que as interações à distância sejam espontâneas, os usuários que estão usando a rede (internet) estarão em desvantagem, a menos que encontrem uma ponte para superar a barreira da distância. Muitos fatores interferem para que haja esse contato. Para estabelecer um encontro à distância, as pessoas devem saber, pelo menos, o endereço eletrônico (e-mail), devem conhecer o *software* para estabelecer a conexão, conhecer e dispor de equipamento apropriado. Para uma videoconferência, por exemplo, as pessoas (em pequenos grupos) devem estar numa sala que tenha todo o equipamento necessário disponível e dispor de uma conexão de rede internet para estabelecer a comunicação.

As pessoas que desenvolvem ambientes *groupwares* estão criando sistemas com propostas especiais para suportar a comunicação, coordenação e o trabalho cooperativo. Eles têm sofisticado os sistemas de e-mail, ferramentas de coordenação, espaços para armazenar e compartilhar documentos, ferramentas de *brainstorming*, escrita colaborativa, e assim por diante. Entretanto, o modo de formar o grupo ainda é fortemente deficiente, nos casos em que

os membros não estejam partilhando um mesmo local de trabalho e não sejam conhecidos entre si

A formação dos grupos, quando seus membros não são previamente conhecidos, é, ainda, uma questão em aberto. Este trabalho busca, portanto, apresentar uma contribuição para preencher essa lacuna.

4.2 Formação do grupo

A formação de um grupo para o trabalho cooperativo é uma tarefa trivial, quando a negociação necessária para compor o grupo ocorre na empresa ou numa comunidade de elementos conhecidos, seja num ambiente real ou suportado por uma rede de computadores. Porém, formar um grupo num ambiente *on-line* no qual os elementos estão dispersos e o usuário ignora quem são os potenciais parceiros, é uma tarefa bastante complexa. Vamos buscar na literatura ferramentas que objetivam esse propósito.

4.2.1 Ferramenta QSabe

Entendendo que a quantidade de mensagens (perguntas, respostas, e opiniões) registradas em ambientes cooperativos alcança dimensões intratáveis, é necessário disponibilizar mecanismos para tratar desse problema. Para lidar com esse problema, o grupo GAIA desenvolveu o QSabe: um serviço cooperativo para apropriação e difusão do conhecimento através da Internet. A proposta do QSabe é ser uma ferramenta para a troca cooperativa de informações que tem o propósito de promover a aquisição (registro) e socialização (recuperação e disseminação) de conhecimento suportado por uma rede de computadores. O QSabe é uma ferramenta virtual de perguntas e respostas, na qual cooperam e colaboram vários participantes com o objetivo de adquirir e compartilhar conhecimento.

Segundo (Pessoa, 1997), utilizando a metáfora de ambientes *groupwares*, o *QSabe* atua como um assistente interpessoal ou um agente de interface, permitindo que cada membro do grupo participe dos trabalhos, formulando ou respondendo perguntas.

Diferente das listas de discussões e mesmo os FAQ's da rede internet que também se prestam a esses objetivos, o *QSabe* busca uma aproximação mais direcionada entre usuários que precisam da informação e especialistas/colaboradores que possuem a informação procurada (Menezes, et al., 2003). Utilizando teorias e técnicas de compreensão de texto, o *QSabe* é capaz de identificar o conteúdo de uma pergunta e, por meio de técnicas de

aprendizagem automática, aprender o perfil de cada respondedor/colaborador. Dessa maneira, o sistema é capaz de automatizar um serviço de perguntas e respostas. No atual estágio de desenvolvimento, o QSabe é um serviço de perguntas e respostas que age como um assistente interpessoal, permitindo a cada membro fazer parte do trabalho, formulando ou respondendo perguntas. A Figura 24 apresenta o modelo de interação utilizada na ferramenta.

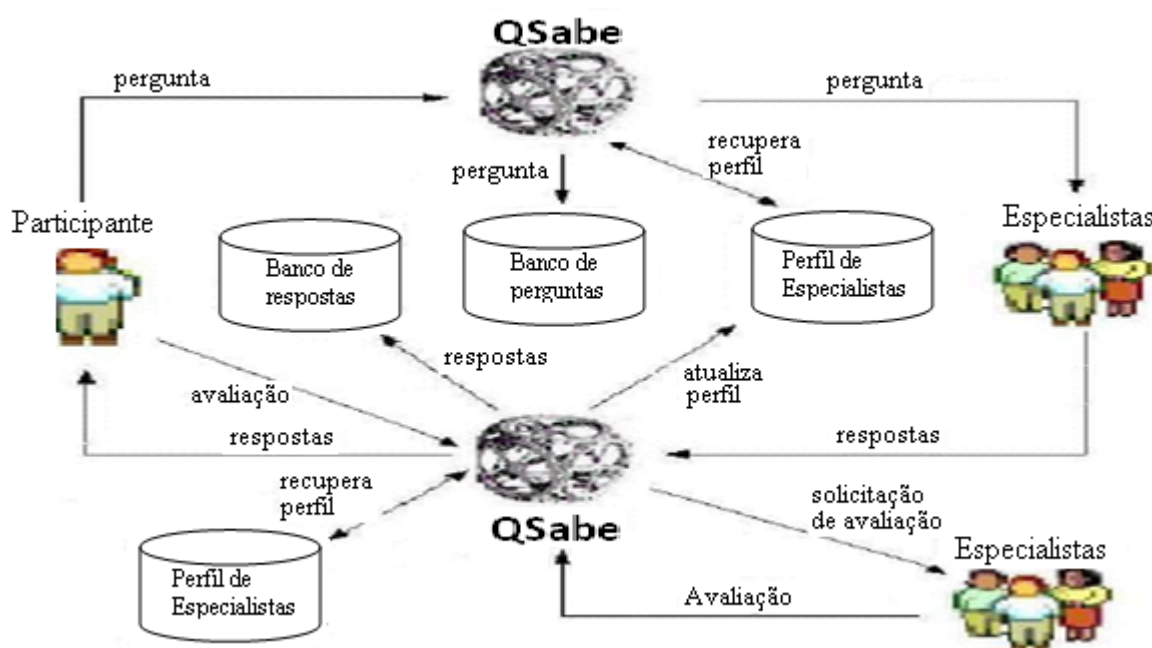


Figura 24 - Modelo de interação e aquisição de conhecimento no QSabe

4.2.2 Recuperando informação (máquinas de busca)

A WEB tornou-se um verdadeiro repositório de conhecimento humano e cultural, compartilhando com as pessoas idéias e informações antes nunca vistas. Esse universo sem fronteiras tem atraído a atenção de milhões de interessados. Ao mesmo tempo, o grande sucesso da web também tem introduzido novos problemas. A facilidade com que qualquer usuário pode hospedar seus próprios documentos em páginas na web nos remete ao termo sobrecarga de informação (*Information Overload*) (Kielgast, et al., 1997). Essa variedade de documentos disponíveis coloca o usuário com a difícil tarefa de separar o joio do trigo na busca por informação útil.

Muitos serviços de busca da web tais como Google™ (Brin, et al., 1998), Yahoo!®, MSNWebSearch™, foram introduzidos no intuito de ajudar as pessoas a encontrar informações na rede. Nesses sistemas, o usuário entra com uma consulta por meio de termos

chaves e o sistema retorna uma lista de páginas da web. Fica a encargo do usuário “varrer” essa lista de páginas para localizar as informações realmente relevantes, uma vez que diferentes assuntos ou diferentes aspectos do mesmo assunto estão misturados. Se o usuário não tiver uma referência do que deseja encontrar, essa busca pode ser improdutiva.

4.2.3 Redes sociais (*Social Network*)

Segundo (Garton, 1997), uma rede social (*social network*) ocorre quando pessoas ou organizações estão conectadas através de uma rede de computadores. Sendo uma abordagem focada na estrutura social. Uma rede social se dá por meio de um conjunto de pessoas (ou organização ou outras entidades sociais) conectadas por um conjunto de relações sociais, tais como: amizade, trabalho conjunto, troca de informação.

Atualmente, os *sites* de redes sociais têm crescido rapidamente. Esses *sites* são definidos, segundo (Boyd, 2007), como um serviço baseado na web que possibilita aos usuários:

- a) construir um perfil público ou semi-público dentro de um sistema delimitado;
- b) articular a lista de outros usuários com quem eles compartilham uma conexão;
- c) visualizar e transitar em sua lista de conexões e entre aquelas feitas por outras pessoas pertencentes ao sistema;

Desde 2002, centenas de *sites* de redes sociais têm surgido, tanto para orientação profissional (LinkedIn) quanto para orientação não profissional (MySpace, Facebook, Cyworld, Bebo, Tagged, Orkut, Friendster). Os sites de orientação não profissional apresentam características tipicamente relacionadas aos interesses dos usuários (Liu, 2007). De acordo com o relatório da “comScore” (2007), referente a um estudo sobre a expansão das redes sociais, a MySpace é apontada como o maior site de rede social, atraindo mais de 114 milhões de visitantes em junho de 2007.

Segundo (Boyd, 2007), o que torna as redes sociais um local exclusivo não é o fato de elas permitirem aos usuários encontrar pessoas estranhas, mas sim o fato de elas possibilitarem aos usuários se articularem na rede e se fazer visíveis em sua rede social. Essa ação pode resultar em conexões entre indivíduos que, de outro modo, não ocorreria. Nos grandes *sites* de rede social, os participantes não estão, necessariamente, procurando iniciar um novo relacionamento (geralmente com estrangeiros) ou procurando novas pessoas; pelo

contrário, eles estão primeiramente se comunicando com pessoas que já fazem parte de sua rede social (Boyd, 2007) (Ellison, 2007).

Embora existam exceções, as pesquisas insinuam que a maioria dos *sites* de redes sociais apóia, primeiramente, as relações sociais preexistentes (Boyd, 2007) (Ellison, 2007). Um *site* de rede social que alcançou muito sucesso no Brasil foi o Orkut²², este sistema permite a entrada de novos participantes se esses forem convidados por um usuário já cadastrado no sistema. O Orkut funciona basicamente por meio de perfis e comunidades. Os perfis são criados pelas pessoas ao se cadastrar. Essas pessoas indicam também quem são seus amigos. As comunidades são criadas pelos usuários e podem agregar grupos, funcionando como fóruns, com tópicos (nova pasta de assunto) e mensagens (que ficam dentro da pasta do assunto).

4.2.4 Uma alternativa para a formação de grupos para o trabalho cooperativo

A proposta apresentada pela ferramenta QSabe indica um caminho para encontrar a pessoa apropriada para auxiliar no processo de resolução de problema e um potencial participante do grupo, uma vez que consegue direcionar a questão formulada para o especialista respondedor.

Usando as ferramentas de busca na web, os usuários podem encontrar o documento relevante a sua consulta, mas o seu interesse também é o de contatar pessoas (especialistas) para formar um grupo de trabalho cooperativo. Os autores dos documentos recuperados pode ser uma alternativa para iniciar o trabalho, porém é necessário estabelecer o contato alegando o proposto e aguardar uma confirmação.

Os *sites* de rede sociais, conforme apresentado, se estabelecem principalmente por meio das relações sociais, tais como: amizade, trabalho conjunto e troca de informação, elementos que não descrevem bem o domínio de conhecimento do usuário. Em geral, na

²² <http://www.orkut.com>. Criado por Orkut Buyukokkten, ex-aluno da Universidade de Stanford e lançado pelo Google em janeiro de 2004, o software é uma espécie de conjunto de perfis de pessoas e suas comunidades. Nele é possível cadastrar-se e colocar fotos e preferências pessoais, listar amigos e formar comunidades. Os indivíduos são mostrados como perfis, é possível perceber suas conexões diretas (amigos) e indiretas (amigos dos amigos), bem como as organizações sob a forma de comunidades. No Orkut, cada novo membro, ao ser convidado, pode fazer seu perfil, colocando suas informações pessoais e adicionando amigos (que podem ser classificados em "amigos", "conhecidos", "melhores amigos", ou "desconhecidos", mas esta classificação não é disponibilizada para outras pessoas, onde todos aparecem simplesmente como "amigos") e juntando-se a comunidades das mais variadas (interesses, cidade, escolas etc.). Além disso, existem ferramentas de interação variadas, tais como sistemas de fóruns para comunidades, envio de mensagens para cada perfil, envio de mensagens para comunidades, amigos e amigos de amigos (normalmente utilizadas para spam) (Recuero, 2004).

descrição do perfil do usuário, são apresentados aspectos relacionados a afinidades, gostos e características pessoais. Essas características não são bons indicadores para descobrir as habilidades do usuário em questões de resolução cooperativa de problemas.

Dado o problema de interesse desta tese em agregar pessoas para o trabalho cooperativo e oferecer subsídios para a resolução de problemas, algumas das tarefas relacionadas à busca de informação podem muito bem ser adaptadas das ferramentas computacionais existentes e, a partir daí, serem automatizadas, ou seja, delegadas a algum processo de execução.

Entendemos que as ferramentas existentes não dispõem de recursos automatizados para envolver pessoas, quando se pretende resolver problemas e nem foram desenvolvidas com esse propósito. Entretanto, uma forma de automatizar o processo de recuperação de informação e contato com especialistas pode ocorrer baseada nas informações extraídas dos documentos recuperados para o usuário pelas ferramentas de busca convencionais e pelo processo de encontrar o respondedor adequado para a pergunta, Figura 25.



Figura 25 - Usuário apresenta uma questão de investigação - agente vai à busca de informação

Uma vez detectado um documento relevante à consulta, um mecanismo interno do sistema poderá varrer os dados à procura de um identificador do autor, que é um potencial especialista no assunto, ou de referências utilizadas por esse, num processo de encadeamento de informação. Detectado o contato (endereço de e-mail, por exemplo), o próprio sistema se incumbem de participar aquele autor sobre o interesse do usuário, sobre o assunto e informar sobre o ambiente no qual esse usuário está trabalhando, convidando-o a participar do grupo de resolução cooperativa de problema, Figura 26.

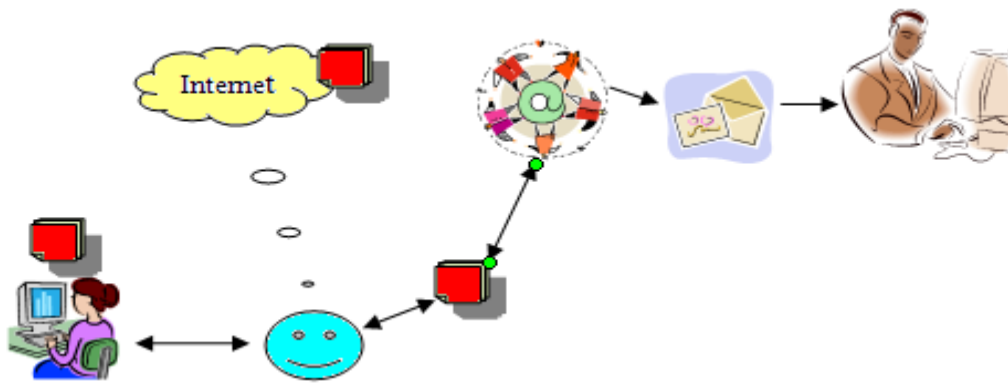


Figura 26 - Documentos de interesse selecionado - agente relaciona pessoas e envia convite para formar grupo

Aquelas pessoas que aceitarem o convite formarão o grupo de trabalho cooperativo, Figura 27.



Figura 27 - Pessoas aceitam convite para cooperar e forma grupo de trabalho cooperativo

4.3 Ferramentas para resolução cooperativa de problemas na internet

As ferramentas *groupware* habilitam os ambientes tecnologicamente e refletem mudanças fundamentais no uso dos computadores para resolver problemas e no uso de computadores para facilitar as interações humanas (Ellis, et al., 1991). Tais fatores têm feito da colaboração no desenvolvimento de sistemas uma necessidade, e não meramente uma opção técnica. Os sistemas *groupware* podem ser caracterizados como síncrono, assíncrono, ou a combinação de ambos. Num sistema *groupware* síncrono as ações são executadas em tempo real, apoiando a comunicação e a cooperação por meio de técnicas de mensagens instantâneas (*chat*, conferências). Nos sistemas assíncronos as ações são armazenadas e vistas depois, um exemplo são os e-mails.

As principais funções *groupware* são: compartilhar informações e documentos, troca de mensagens, formação de grupos, conferências e *chats*, agenda de grupos, gerenciamento de projetos, e apoio para o trabalho em equipe (Zwass, 1988 apud (Tommarcello, 2002)). Uma solicitação nos sistemas *groupware* é que eles forneçam interface coordenada para todos os participantes, uma abstração do acrônimo WYSIWIS – “*What You See Is What I See*”. Segundo (Stefik, et al., 1987). Para que as ferramentas de reunião sejam efetivas, todos os membros do grupo devem ser capazes de ver o trabalho de seus pares. Além disso, fornecendo interfaces coordenadas ou visões, essas ferramentas podem fornecer possibilidade para *brainstorming*, organização e avaliação.

Segundo (Tommarcello, 2002), o processo de resolução cooperativa de problemas necessita além de sincronização, de compartilhamento de arquivos, mensagens síncronas e assíncronas e características de segurança. A resolução cooperativa de problema requer direção específica para a resolução do problema, que inclui esforço para melhorar o processo cognitivo do grupo enquanto resolvem problema cooperativamente. Na Tabela 3 apresentamos alguns requisitos importantes para suportar o trabalho cooperativo.

Tabela 3 – Requisitos para um ambiente de resolução cooperativa de problemas

Ferramentas	Descrição
Compartilhamento de idéias	Espaço para que os usuários possam levantar suas idéias
Compartilhamento de documentos	Espaço para que os usuários possam armazenar informações pertinentes ao problema em tela e que seja acessível a todos os participantes do grupo
Mecanismos de interação	Ferramentas de comunicação entre os participantes do grupo
Planejamento e execução de tarefas	Ferramentas que apoiem a concretização das idéias discutidas
Avaliação e validação de tarefa	Ferramentas que suportam <i>feedback</i> as ações realizadas

As ferramentas comumente encontradas em ambientes que apoiam a resolução cooperativa de problemas suportados por computador, tanto para comunicação quanto para a coordenação de atividades, são apresentadas na Tabela 4:

Tabela 4 - Ferramentas de comunicação e interação

conferência	versão on-line para o encontro dos elementos do grupo. É uma ferramenta de comunicação síncrona. Uma característica dessa ferramenta de reunião é o fato de ela fornecer uma interface coordenada para todos os participantes;
Chat	nome dado a uma sala virtual, usando uma rede de computadores, para conversas e discussões através de mensagens digitadas, por grupos "virtuais" em tempo real. É um exemplo de comunicação síncrona em redes.
e-mail ou correio eletrônico	recurso que possibilita a troca de mensagens e arquivos de forma assíncrona, rápida e versátil. Em outras palavras, é um sistema de correspondência, ou seja, de troca de mensagens, via Internet;
Fórum	espaço de discussão pública on-line para que os usuários possam dar continuidade às discussões sobre um determinado assunto, postar perguntas, disponibilizar artigos e responder mensagens. No fórum geralmente é colocada uma questão, uma ponderação ou uma opinião que pode ser comentada por quem se interessar. As opiniões armazenadas podem ser lidas e comentadas;
Agenda	gerenciador de informação de forma livre: as informações não precisam estar estruturadas para serem lançadas na base de dados, o programa se encarrega disso. Geralmente as agendas são apresentadas em formato de planilha, coordenando datas e ações. O calendário está associado a uma agenda;
<i>brainstorming</i>	espaço compartilhado que possibilita aos membros da equipe gerar e compartilhar idéias. É também visto como uma estratégia criativa de resolução de problemas. Esta ferramenta pode ser utilizada em tempo real (de forma síncrona) ou não (de forma assíncrona). Normalmente, as idéias do grupo são disponibilizadas para votação.
Votação	ferramenta associada às idéias disponibilizadas pelos usuários do ambiente (podendo inclusive estar associado à ferramenta de <i>brainstorming</i>) para que eles possam avaliar a viabilidade e eficácia da mesma antes de ser levada à execução.
quadro de avisos (<i>writeboard</i>)	ferramenta do ambiente que possibilita aos usuários compartilhar avisos, informações, solicitações, etc. Bastante útil nos ambientes cooperativos para manter contato com os demais participantes.
Estante	ferramenta do ambiente que possibilita aos usuários compartilhar documentos de interesse comum.

Para obter um bom aproveitamento do conceito da divisão de problemas e da utilização de ferramentas que apoiem essa estratégia, é necessária uma forma de coordenar o

desenvolvimento de tais atividades. Um modelo de *groupware* que tem sido usado para apoiar sistemas síncronos e assíncronos é *Computer Supported Cooperative Work* (CSCW) e sua variante para casos de aprendizagem *Computer Supported Collaborative Learning* (CSCL).

4.4 Ferramentas de apoio ao trabalho cooperativo

Existem várias ferramentas que apóiam o trabalho / aprendizagem cooperativa por computador (CSCW – *Computer Supported Cooperative Work* / CSCL – *Computer Supported Collaborative Learning*). Descreveremos abaixo as características de alguns dessas ferramentas. Este levantamento não tem a pretensão de ser exaustivo, apenas sinalizar alguns elementos importantes nas ferramentas dessa categoria.

4.4.1 Groove (atual Microsoft® Office Groove®)

O Groove²³ (Tutorial, 2006-2007) é um sistema *groupware* que fornece a pequenos grupos de usuários (2 – 25), a possibilidade de compartilhar documentos, mensagens, aplicações simples e, especificamente, relacionadas ao projeto do grupo de modo seguro, suportado por uma rede VPN (*Virtual Private Network*), por meio do dispositivo de colaboração. Em abril de 2005, o sistema Groove foi adquirido pela Microsoft Corporation, passando a ser propriedade subsidiada da Microsoft, e adotando o nome de Groove Virtual Office. O seu objetivo é disponibilizar um local onde as pessoas possam, além de acessar informações e encontrar documentos, interagir umas com as outras. Trabalhando de modo síncrono ou assíncrono, on-line ou não. A formação do grupo de trabalho se faz por meio de convite expedido pelo usuário formador do grupo. Para compartilhar esse sistema de trabalho, todos os elementos do grupo devem ter instalado em suas máquinas o referido software.

Entre as ferramentas disponibilizadas podemos citar: serviço de mensagem (e-mail), mensagem instantânea baseada em texto (*chat*), compartilhamento de arquivo (texto, imagem, apresentações), *browsing* web, ferramentas de: *brainstorming*, estabelecimento de contato entre os membros do grupo (*contacts – agenda*) para discussão de trabalho, gerenciamento de tarefa, revisão de documento, calendário, bloco de anotações, e *sketchpad* (bloco de esboço). Cada membro do grupo pode definir dispositivos diferentes e usá-los simultaneamente, da mesma forma, podem escolher navegar juntos para trabalhar com o mesmo dispositivo ao mesmo tempo.

²³ <http://www.groove.net/home/index.cfm>

O dispositivo de discussão possibilita aos usuários trocar detalhes de idéias com os demais membros do projeto. O dispositivo de *brainstorming* permite à equipe esboçar idéias numa estrutura hierárquica. Para compartilhar, organizar e armazenar arquivo é oferecido o dispositivo documento (*files*), podendo esse também ser transferido por meio dessa ferramenta. Com o dispositivo agenda a equipe pode estabelecer as metas e datas para execução das tarefas. O dispositivo link permite aos elementos do grupo navegar (*browser*) a internet juntos. A identidade e características de cada membro do grupo são armazenadas no dispositivo de contato, o que possibilita um modo fácil de acessar mensagens. Dispositivos adicionais podem ser inseridos e partilhados no sistema.

4.4.2 GroupSystem

O GroupSystem²⁴ é uma aplicação *groupware* que dispõe de muitos dispositivos de uso colaborativo. Implantada desde 1989 quando a IBM e a University of Arizona, colaboraram para trazer essa ferramenta inovadora para o espaço comercial. Sua proposta é possibilitar aos grupos de trabalho ferramentas que possibilitem promover ajuda e inovação, de modo mais rápido e completo, no alcance de metas (GroupSystem, 2006-a) (GroupSystem, 2006). O GroupSystem deve ser instalado sobre um servidor partilhado para ser usado pelos membros do grupo numa *workstation* conectada, podendo ser usado tanto de modo síncrono quanto assíncrono.

Vários dispositivos de colaboração estão disponíveis, como por exemplo, a agenda que possibilita que as atividades sejam adicionadas, deletadas ou abertas. Após inserido um tópico, é aberta a ferramenta *brainstorming* e iniciada a sessão de discussão em que todos os participantes apresentam seu entendimento sobre o problema numa “folha de discussão” (uma janela do sistema). Os membros do grupo podem ir adicionando suas idéias sobre essa “folha”. O dispositivo de votação é ativado quando os membros do grupo acharem útil organizar as idéias desenvolvidas na ferramenta de *brainstorming* ou, simplesmente, para fornecer uma base para a decisão do grupo. O dispositivo de votação baseia-se numa escala de classificação de dez pontos, numa seleção múltipla de: sim/não, verdadeiro/falso, concordo/discordo (com um sintonizador, numa escala até 5 pontos) ou um método

²⁴ (www.groupsystems.com)

personalizado. Após completo o processo de votação, os resultados podem ser vistos clicando no ícone – resultados.

A ferramenta *Whiteboard* que simula o tradicional quadro branco pode ser usada para editar imagens e traçar esboços. Essa ferramenta oferece vários ícones com uma série de recursos (*paint*). São disponibilizadas dez páginas para esses esboços. O usuário pode imprimir ou salvar em arquivo. O dispositivo de prospecto (*Handout*) é essencialmente uma área para armazenar arquivos para o grupo (capacidade para cem arquivos de referência). Além disso, ela possibilita acessar, apenas para leitura, relatórios, arquivos multimídia, planilhas eletrônicas, material visual tipo gráfico, *chat* e diagramas. Ao acessar um documento, toda aplicação relacionada a ele será automaticamente aberta. Similar ao dispositivo *Handout* (prospecto) há o dispositivo relatório, sendo esse para armazenar projetos relacionados ao relatório. Ele permite a criação de arquivos de relatórios agregados, contendo dados, atividades gerais e recursos em uma pasta.

A ferramenta de opinião é essencialmente uma ferramenta de votação, porém mais simples e rápida para dar flexibilidade e, também, um meio informal de coletar opiniões dos membros do grupo. Além dessas, há uma pasta de utilidades (*Briefcase*), que possibilita acessar calculadora, bloco de notas, *clipboard*²⁵, etc. Essa ferramenta pode ser personalizada para adicionar outras aplicações comumente usadas, tais como e-mail ou processador de texto.

4.4.3 Lotus Notes (atual Lotus Domino)

O Lotus Notes²⁶ é um dos sistemas *groupware* pioneiros para uso geral. Ele fornece ao usuário dispositivos de e-mail, *chat*, *workflow*, calendários, lista de tarefas, e documentos compartilhados. Sua estrutura permite flexibilidade no uso desses elementos. O principal objetivo do software é oferecer facilidades para o gerenciamento do trabalho nas empresas. Associado ao sistema é oferecido o *Learning Space* que é um ambiente para desenvolvimento e distribuição de cursos à distância apoiado sobre o ambiente *Lotus Notes*. Desenvolvido pela *Lotus Education* e pela IBM, esse sistema oferece suporte a múltiplos idiomas e possui a capacidade de customização da estrutura navegacional. Desde a sua criação o Lotus tem sido constantemente aprimorado. Além dos serviços mencionados, o Lotus oferece também

²⁵ Uma área de memória que funciona como um depósito para as informações (imagens ou textos), que tenham sido copiadas ou extraídas de um documento, por último. Essas informações podem ser recuperadas pelo mesmo ou até por outro programa quando se emite o comando "Paste" (colar), garantindo a transferência ao usuário.

²⁶ (<http://www-128.ibm.com/developerworks/lotus/library/ls-NDHistory/>)

blogs²⁷, wikis²⁸, RSS Agregadores²⁹, gerenciamento de cliente por CRM³⁰ (*Customer Relationship Management*) e sistemas de *Help Desk*³¹ e, além disso, as organizações podem construir uma variedade de aplicações para o Notes usando o aplicativo "Domino Designer". Além de ser um sistema “*groupware*” o Notes/Domino é também uma plataforma para desenvolver aplicações otimizadas em formato cliente-servidor e aplicações web, além de oferecer dispositivos que facilitem a construção de qualquer tipo de aplicações “*workflow*”.

4.4.4 AmCorA

O AmCorA³² (Netto, 2003), (Ferreira, et al., 2000) é um ambiente para aprendizagem cooperativa com o objetivo de integrar pessoas para proporcionar a aprendizagem on-line. A integração das pessoas se dá por meio de comunidades virtuais, chamadas de grupos. Para criar um grupo no AmCorA, basta fazer uma solicitação no site principal do ambiente. As comunidades são formadas por meio do agrupamento de pessoas que têm algo em comum. Os grupos inscritos hoje no AmCorA são formados, em sua maioria, por alunos e professores. Alguns grupos são formados por participantes de laboratórios de pesquisa, outros grupos têm por objetivo fornecer a professores e alunos orientados um espaço de trabalho onde podem

²⁷ Popularmente conhecido como blogs, é uma ferramenta de comunicação e informação na web. Os blogs caracterizam-se por serem páginas publicadas individualmente, expondo relatos da vida pessoal, como um diário pessoal. Além desta característica, eles possuem uma estrutura hipertextual e são constituídos por textos curtos e publicados em blocos padronizados e ordenados em ordem cronológica inversa (como um diário); cada publicação possui um link permanente de acesso; cada publicação pode ser enriquecida por um comentário; pode ser constantemente atualizada e as publicações mais antigas arquivadas; podem ser intertextuais e interdependentes, possuindo, assim, ligação com outros textos. (fonte: <http://pt.wikipedia.org/wiki/Blogs>)

²⁸ Wiki é o software colaborativo que permite a edição coletiva dos documentos usando um sistema que não necessita de que o conteúdo tenha de ser revisto antes da sua publicação. Através da ferramenta Wiki, um internauta pode se cadastrar em um Wiki e editar qualquer conteúdo apresentado em site do Wiki, manipulando apenas o browser de navegação. A edição do conteúdo é feita e o internauta salva a página no editor do Wiki e esta é atualizada automaticamente no site, sem a aprovação prévia. Os textos criados no Wiki não possuem “donos”, ou melhor, escritores próprios, estes são públicos e estão disponíveis para comunidade. O gerenciamento do conteúdo apresentado no Wiki não passa por nenhum administrador, parte do bom senso das pessoas que o constroem a exposição das idéias e conceitos de forma a não publicarem algum conceito incorreto ou danificar algo já produzido. (fonte: <http://pt.wikipedia.org/wiki/Wiki>)

²⁹ Os agregadores são programas que organizam as informações que são vistas de forma final pelo usuário. Tais programas são receptores de RSS Feed (agregadores de conteúdos que atualizam freqüentemente), uma tecnologia que permite a distribuição/recebimento de conteúdo (texto, som, vídeo) sem a necessidade de acessar um website para poder recebê-lo. (fonte: <http://pt.wikipedia.org/wiki/Agregador>)

³⁰ CRM é o acrônimo da frase em inglês *Customer Relationship Management*, é um sistema integrado de gestão com foco no cliente, constituído por um conjunto de procedimentos/processos organizados e integrados num modelo de gestão de negócios. (fonte: <http://pt.wikipedia.org/wiki/CRM>)

³¹ Serviço de apoio a usuários para suporte e resolução de problemas. No caso particular do ambiente Lotus está relacionada ao gerenciamento de tarefas.

³² (www.gaia.ufes.br/amcora)

compartilhar teses, dissertações e outros documentos importantes. No AmCorA, é possível a divisão dos grupos em subgrupos.

As ferramentas disponíveis no AmCorA estão relacionadas a informações pessoais e do grupo sendo os elementos de organização disponíveis: perfil, agenda, caixa postal, estante, escaninho, escrivadinha, e visualização dos grupos. Para o trabalho coletivo vários dispositivos são disponibilizados, dentre eles uma estante para compartilhar os trabalhos e atividades referentes ao grupo. Além disso, na caixa postal os membros do grupo podem discutir os temas de seu interesse. Há uma conexão direta com o e-mail particular dos membros o que permite ao sistema enviar a mensagem automaticamente para todos os componentes. Ainda merece destaque no grupo a ferramenta “Big Brother”. Esta ferramenta permite que sejam conhecidos os participantes que estão no momento “logados” no sistema. Por meio dessa facilidade os usuários podem trocar mensagens instantâneas. Outra característica importante desse ambiente é o apoio à criação de subgrupos.

4.4.5 CaMILE - Collaborative and Multimedia Interactive Learning Environment

CaMILE³³ (Gudizial, 1997-a), (Santoro, et al., 1998) é um sistema Web assíncrono de suporte à colaboração, com o objetivo de estimular a aprendizagem e é voltado para a análise em um nível alto de agregação e fóruns de discussões com grupos múltiplos ou uma classe inteira (Gudizial, 1997). Todos os acessos ao sistema são realizados por meio de browser Web que acessa um servidor único. A interface do sistema é baseada em formulários e é igual para todos os usuários.

As discussões no CaMILE são contextualizadas como em um *newsgroup*, porém, o contexto é persistente e está sempre disponível para os usuários, não "desaparecendo" após a visualização. CaMILE provê uma facilidade na qual os estudantes são solicitados a identificar o tipo de colaboração que estão apresentando (p. ex., uma questão, uma nova idéia, uma refutação, etc.) e são oferecidas sugestões de frases iniciais para serem usadas em cada um destes tipos de notas. De acordo com (Gudizial, 1997-a), uma diferença importante entre *newsgroups* e CaMILE é que esta ferramenta apóia a colaboração ancorada, ou seja, cada nota individual pode ser referenciada unicamente por meio de um *browser* Web. Isso quer dizer que o endereçamento direto de notas permite que páginas Web contenham hiperlinks para um

³³ <http://www.cc.gatech.edu/gvu/edtech/CaMILE.html>

contexto de discussão CaMILE. As âncoras funcionam como índices e lembretes do que estudantes discutiram sobre um determinado contexto.

Atualmente, muitas ações ancoradas no ambiente estão sendo movidas para o **Swikis** (<http://wiki.squeak.org/swiki/>). O Swikis é uma combinação de Squeak³⁴ + Wiki³⁵, = Swiki. Além das opções de escrita cooperativa do wiki com o Squeak, é possível a adição de *plugins* com características especiais (edição gráfica, imagens, mapas, etc.). Destacamos aqui a importância e disseminação de ferramentas Wiki utilizadas para identificar um tipo específico de coleção de documentos em hipertexto ou o software colaborativo usado para criá-lo.

4.5 Considerações sobre os sistemas

Dos sistemas citados dois foram projetados com fins educacionais e três com fins comerciais. Todos os sistemas oferecem as ferramentas básicas de interação síncrona e assíncrona, segundo sua finalidade de projeto. Alguns deles disponibilizam dispositivos mais apropriados para o apoio à resolução de problemas. O Lotus Note fornece assistência à coordenação do grupo, mas não fornece ferramentas específicas para atividades de cooperação em tempo real.

As características apresentadas pelo Groove facilitam o desempenho do grupo nos principais aspectos do processo de resolução de problemas. As tarefas podem ser facilmente identificadas, distribuídas, comunicadas e modificadas. A sua ferramenta de *brainstorming* permite aos usuários apresentar e discutir as diferentes idéias, porém não possibilita a votação para escolha.

O GroupSystem é um excelente sistema com dispositivos para uso em colaboração. Seu principal objetivo é apoiar as atividades de conhecimento colaborativo, semelhante ao planejamento estratégico e análise de risco. Oferece uma ferramenta de *brainstorming* associada com uma ferramenta de votação, requerido na resolução de problema para o processo de decisão. Aliado ao dispositivo de *whiteboard* o sistema oferece, ainda, uma

³⁴ Squeak é uma linguagem de programação derivada de Smalltalk-80. Squeak é a linguagem utilizada pelo servidor web Comanche, que contém um framework que torna possível desenvolver aplicações web integralmente em Squeak, sem a necessidade de executar um servidor web externo. Tanto o Comanche quanto o Swiki foram implementados por Mark Guzdial.

³⁵ A palavra Wiki é um termo cunhado por Ward Cunningham, originalmente WikiWikiWeb, criador do sistema em 1995, que em havaiano wiki-wiki significa: veloz, célere. O Wiki surgiu de um script criado por Ward Cunningham, que inicialmente funcionava no servidor com a finalidade de auxiliar na condução de grandes projetos de informática, onde todo o material relacionado ao projeto poderia ser atualizado constantemente por todos os membros da equipe, cada alteração é salva no sistema e todas as modificações poderiam ser revisadas.

ferramenta adicional – *paint*, que possibilita o usuário esboçar e editar imagens no *whiteboard*, que podem ser impressas ou salvas em arquivo.

Os sistemas de apoio à aprendizagem apresentam os dispositivos necessários para a interação entre os usuários, compartilhamento de material e projetos, acompanhamento e avaliação de atividades. Entretanto, não apresenta mecanismos para apoiar a resolução cooperativa de problemas, como discussão de idéias e elaboração de estratégias. A resolução cooperativa de problemas requer direção e apoio, que inclui o esforço para melhorar as atividades cognitivas do grupo enquanto engajado na resolução de problemas cooperativo, que esses sistemas não incorporam.

Em todos os sistemas abordados, percebemos que a formação do grupo ocorre quando os elementos já têm alguma referência sobre quem convidar. Nos sistemas de trabalho cooperativo as relações ocorrem baseadas nos elementos que compõem a empresa ou o projeto que está sendo desenvolvido. Nos sistemas de aprendizagem, os elementos do grupo também são previamente conhecidos ou as interações ocorrem na comunidade. A ferramenta de pergunta e resposta apresentada pelo QSabe indica uma direção para encontrar o especialista para responder a questão proposta, mas se limita a fornecer uma resposta e não um espaço de interação entre perguntador e respondedor com intuito de resolver um problema.

É importante salientar que neste trabalho não nos referiremos a outros softwares (*webblogs*, *wiki*) que tem como objetivo a escrita cooperativa e difusão de informação, pois fogem ao escopo do problema em vários aspectos:

- *wiki* - os cooperadores ficam no anonimato, não há uma formação de grupo de discussão. O material produzido (artigo) não mantém o vínculo com quem fez as alterações. Não há uma discussão efetiva sobre a resolução do problema.
- *blogs* - a construção de um blog não necessita ser cooperativa, a interação pode ocorrer ocasionalmente. Além disso, as alterações realizadas não ficam armazenadas e não há um controle de versão do material publicado.

Entendemos que, para haver maior produtividade na busca de solução cooperativa de problemas, os participantes do grupo devem apresentar conhecimento sobre o assunto tratado e dispor de evidências para auxiliar esse processo. Avaliando o modo como ocorre a formação

dos grupos nos sistemas apresentados, percebe-se que esses são, geralmente, formados por participantes de uma mesma comunidade ou que têm alguma afinidade. Diante disso, duas questões ainda permanecem em aberto nesses sistemas: o modo como recuperar documentos relacionados aos problemas em tela e, principalmente, como encontrar as pessoas (especialistas) para, num processo cooperativo, participar da resolução?

O trabalho aqui relatado procura diminuir a dificuldade enfrentada pela comunidade de usuários quando trata desses dois pontos em aberto: recuperação de informação e dificuldade de encontrar as pessoas que poderiam, em regime de cooperação, repassar conhecimentos e experiências. Além disso, outro ponto importante nos ambientes de resolução cooperativa de problemas é dispor de um local onde os resolvedores possam, além de discutir as idéias, formalizá-las dinamicamente por meio de esquemas e construção de modelos de simulação.

Capítulo 5

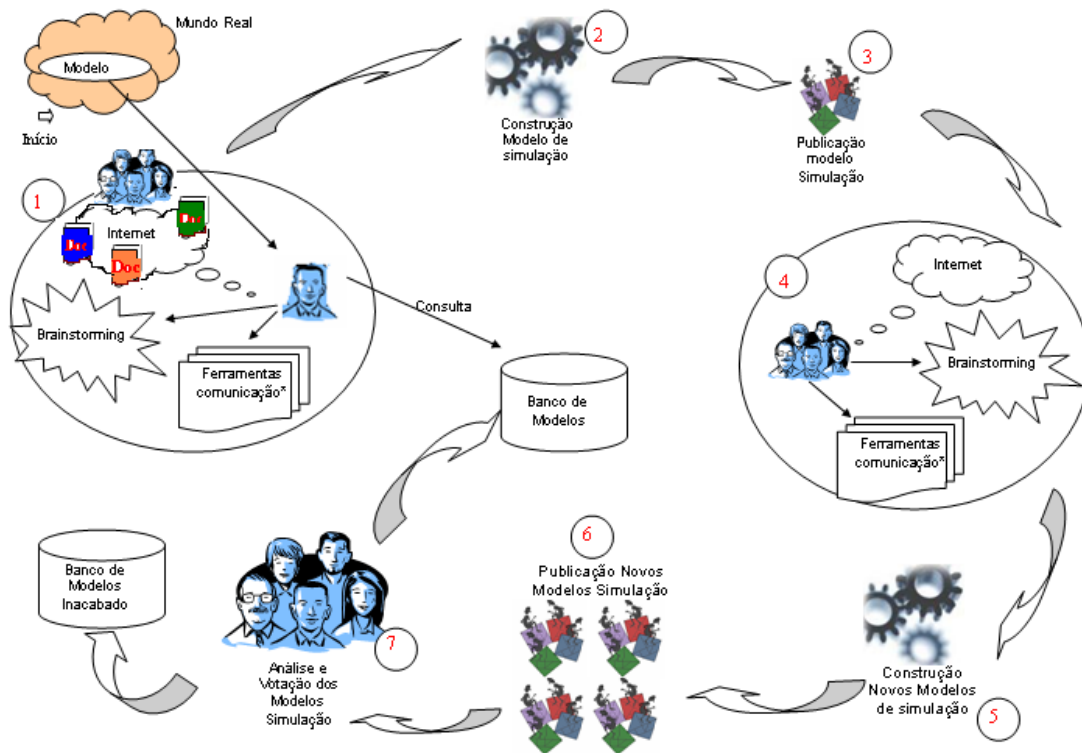
5 O ambiente AProSiMA

Seguindo a questão de investigação avaliamos sistemas de trabalho cooperativo e ferramentas de simulação multiagente e constatamos que os ambientes visitados, de forma amostral, não apresentam essas duas características simultaneamente. Além disso, a forma como os ambientes de trabalho cooperativo são adotados (estações locais) não viabiliza a formação de grupos com membros externos à comunidade com a qual eles se relacionam. O que pretendemos aqui é validar a possibilidade de ter um ambiente projetado para a resolução cooperativa de problemas e modelagem de simulação em um só local:

- para apoiar o trabalho cooperativo, o ambiente deverá ser desenvolvido com características para web e fazer uso das ferramentas *groupware* que venham auxiliar nesse propósito;
- para apoiar o trabalho cooperativo, o ambiente deverá possibilitar a formação de grupos internos e externos ao ambiente. Deve também oferecer informações textuais para elucidar o problema;
- para apoiar a modelagem de simulação multiagente, o ambiente deverá contar com uma ferramenta específica para construção de esquemas cooperativos que delineiam estratégias para construção dos modelos de simulação;
- para apoiar a modelagem de simulação multiagente, o ambiente deverá contar com uma ferramenta específica para construção de modelos de simulação, além de possibilitar a descrição dos modelos com o alcance daqueles baseados em uma linguagem de programação e com a facilidade de modelagem para usuários leigos em programação. Associado a isso pretendemos ainda disponibilizar um número maior de ferramentas conceituais para entendimento do modelo;
- para apoiar a resolução de problema, o ambiente deverá contar com uma metodologia de resolução de problema que oriente as pessoas para o objetivo desejado.

Assim, a concretização da proposta do ambiente resultou no **ambiente para resolução cooperativa de problemas baseado em simulação multiagente** denominado AProSiMA. O nome AProSiMA é um acrônimo para **Ambiente Problema Simulação MultiAgente**, o termo

resolução cooperativa é resultado do significado fonético de aproximar, que significa, entre outras coisas, estabelecer relações. A Figura 28, é uma idealização do funcionamento do ambiente.



* Ferramentas de comunicação disponíveis: Chat, E-mail, Fórum, Wiki.

Figura 28 –AProSiMA - Ambiente de resolução cooperativa de problemas baseado na simulação multiagente

5.1 Descrição do ambiente AProSiMA

O ponto de partida para utilização do ambiente ocorre quando um usuário se defronta com um problema do mundo real e, a partir daí, procura entendê-lo e de alguma forma representá-lo por meio de um esquema ou modelo para ganhar entendimento sobre esse problema. No início do processo é necessário coletar informações sobre o problema e, se possível, contatar com especialistas que possam auxiliar no processo de resolução cooperativa. É importante dispor de uma ferramenta de “*brainstorming*”³⁶ para ganhar *insights* e posteriormente organizar as idéias, (momento 1, Figura 28). A ferramenta de

³⁶ O termo *brainstorming* foi cunhado por Osborn em 1953 – para representar a técnica criativa de geração de idéias. Usaremos o termo *brainstorming* para representar um conjunto de ações que possibilitarão as interações entre os usuários do sistema para entendimento do modelo, levantamento de dados, esboços dos modelos, questionamentos, etc.. Essas ferramentas ainda serão implementadas.

simulação disponível no ambiente possibilitará ao usuário (ou ao grupo) construir o seu próprio modelo de simulação (momento 2, Figura 28), (nos capítulos posteriores daremos mais detalhes do processo de construção do modelo de simulação). Após construída a primeira versão do modelo de simulação, o usuário deve publicá-lo (momento 3, da referida figura) para que os demais participantes do ambiente possam analisá-lo, inferir sobre ele e apresentar novas versões, num trabalho cooperativo para uma melhor abrangência do modelo (momento 4 e 5, Figura 28). Depois, numa sessão conjunta, discutir os procedimentos para efetivação da idéia.

Esse processo é acompanhado sistematicamente pelas publicações das contribuições realizadas pelos cooperadores (momento 6, Figura 28). Periodicamente, usando uma ferramenta de votação, os participantes do ambiente votam os modelos apresentados, classificando-os com relação à maior proximidade com o objetivo a ser simulado (momento 7, Figura 28). Os demais modelos são arquivados num “*banco de modelos incompletos*”, pois podem dar origem a soluções de outros problemas. Os modelos desenvolvidos são armazenados num “*banco de modelos*”, servindo de referência para novos problemas, conforme pode ser seguido pela Figura 28. Todo esse processo é contínuo, uma vez que um problema não acaba em si mesmo.

As ferramentas de comunicação, que estarão disponíveis no ambiente, possibilitarão aos usuários traçar detalhes das idéias com os demais membros envolvidos no projeto. A ferramenta *brainstorming* (Esquematizador) possibilitará à equipe (ou ao próprio modelador) esboçar as idéias numa estrutura hierárquica. As idéias desenvolvidas na sessão de *brainstorming* poderão ser organizadas e armazenadas juntamente com o modelo construído. O processo de construção do modelo, bem como as discussões sobre sua modelagem será acompanhado através de uma ferramenta de publicação colaborativa – Wiki³⁷ e em fóruns.

5.2 Ferramentas do ambiente AProSiMA

A materialização de cada passo proposto para o ambiente resultou nas macro-funcionalidades apresentadas na Figura 29. Algumas foram separadas em pacotes devido às

³⁷ Resumidamente, Wiki é uma ferramenta que permite que documentos sejam editados coletivamente com uma linguagem de marcação muito simples apenas por meio da utilização de um navegador web. Nossa pretensão é disponibilizar um Wiki que suporte tabelas, imagens, fórmulas, ou até elementos interativos, como votações e jogos.

suas afinidades funcionais. As ferramentas que estarão disponíveis no ambiente AProSiMA e suas funcionalidades são:

1. **Pacote de cadastro** – dispositivo que possibilitará o acesso do usuário ao sistema e a todos os recursos disponíveis.
2. **Pacote de formação de grupo** – esta ferramenta possibilita aos usuários criar e controlar os grupos a que pertence. É facultativa ao usuário a criação de grupos. Uma vez formado o grupo, é possível consultar todas as ações realizadas pelo usuário e os demais membros que participaram da ação.
3. **Pacote QSabe2** – é um gerenciador de perguntas e respostas. Esta ferramenta será desenvolvida nos moldes apresentado em 4.2.1 e adaptada às características do ambiente AProSiMA.
4. **Pacote de comunicação** – neste pacote estão inseridas as seguintes ferramentas: chat; fórum; wiki³⁸; mensagem de e-mail. Essas ferramentas têm como objetivo possibilitar a interação entre os usuários do ambiente. Durante o desenvolvimento das atividades, essas ferramentas serão os canais de comunicação para a atividade cooperativa, que pode ocorrer de forma síncrona ou assíncrona;
5. **Mural** – essa ferramenta é utilizada para estabelecer um contato rápido com todos os participantes do ambiente, inclusive com aqueles que ainda não interagiram. Através dela será possível postar dúvidas, convidar pessoas, consultar respostas, fazer anúncios sobre alguma descoberta, etc.
6. **Pacote simulador (NetPlay)** – este pacote descreve a ferramenta que tem por finalidade construir modelos de simulação baseados em multiagente sem a necessidade de conhecimento em linguagem de programação;
7. **Pacote Gerenciador** – este pacote tem a finalidade de recuperar informações e contatar pessoas para o processo de resolução cooperativa de problema. Esse pacote conta com um “agente” que desempenha a função de recuperar documentos e localizar pessoas através desses documentos;

³⁸ Através de consenso, passou a ser chamado "wiki" o software colaborativo que permite a edição coletiva dos documentos. Os textos são editados coletivamente por meio de um navegador web, usando uma linguagem de marcação simples.

8. **Pacote esquematizador** – o objetivo desta ferramenta é possibilitar aos diferentes usuários um local compartilhado onde eles possam discutir as possibilidades do fenômeno em estudo. A ferramenta possibilitará aos usuários formalizar suas discussões por meio de esquemas. Numa sessão cooperativa, todos poderão visualizar simultaneamente o esquema em construção e discutir sobre ele. Durante o processo de discussão todos os esquemas traçados podem ser recuperados durante a sessão. Ao final da discussão, os esquemas construídos serão salvos e guardados num banco de esquemas.

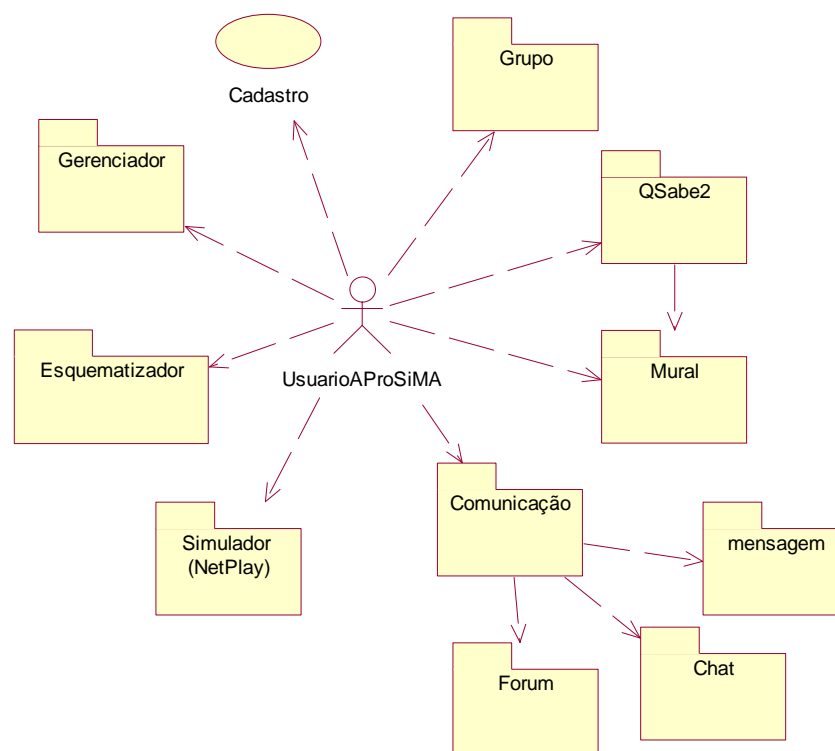


Figura 29 – Macro funcionalidades do ambiente AProSiMA

A Figura 29 apresenta os relacionamentos entre as ferramentas disponíveis no ambiente AProSiMA. A Figura 30 apresenta as chamadas entre cada pacote, que pode ser acompanhado pela descrição abaixo:

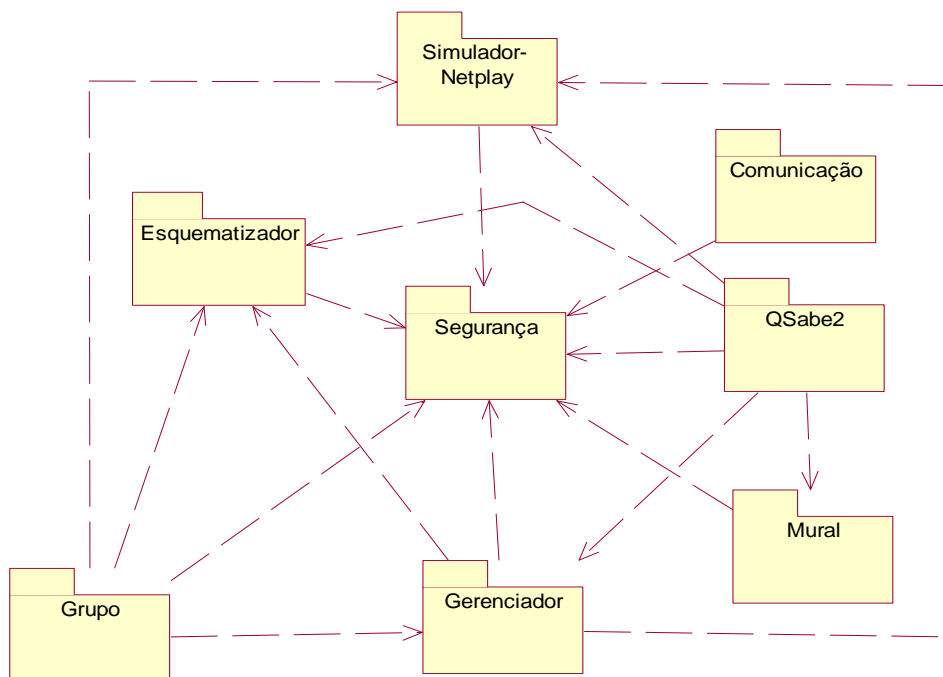


Figura 30 - Diagrama de pacotes – relacionamento

As chamadas que o pacote “Grupo” faz aos pacotes:

- “Segurança” é basicamente para recuperar o perfil do usuário, por meio da descrição de sua área de interesse, para auxiliar na formação do grupo com mesma afinidade;
- “NetPlay” é basicamente para obtenção da identificação (ID) dos usuários e a descrição dos modelos, para contatar pessoas com interesses comuns;
- “Gerenciador” é basicamente para obtenção da ID dos usuários e para relacionar os documentos por ele recuperados, que descreve interesses comuns;
- “Esquemmatizador” é basicamente para obtenção da ID dos usuários e os esquemas que desenvolveu, relacionados ao mesmo interesse e também para indicar os componentes do grupo de discussão representada por aquele esquema;

As chamadas que o pacote “esquemmatizador” faz ao pacote “segurança” é basicamente para obtenção da ID dos usuários.

As chamadas que o pacote “Simulador - NetPlay” faz ao pacote “segurança” é basicamente para obtenção da ID dos usuários que dará controle às ações realizadas.

As chamadas que o pacote “Gerenciador” faz aos pacotes:

- “Simulador – NetPlay” é basicamente para obtenção da ID dos usuários e a descrição dos modelos, para o caso de recuperação de modelos similares a um novo problema;
- “Esquematizador” é basicamente para obtenção da descrição dos esquemas desenvolvidos, para o caso de recuperação de casos (esquemas) similares a um novo problema, e da ID dos usuários;
- “Segurança” é para vincular as ações definidas por cada usuário;

As chamadas que o pacote “QSabe2” faz aos pacotes:

- “Segurança” é basicamente para obtenção da ID dos usuários e analisar o perfil descrito em sua área de interesse;
- “Gerenciador” é basicamente para avaliar os documentos recuperados, cujos autores não foram convidados, no intuito de analisar se dentro daqueles documentos pode ser encontrado algum especialista para auxiliar na resposta às perguntas propostas, e relacionar a ID do usuário;
- “Simulador – Netplay” é basicamente para avaliar os modelos descritos no intuito de analisar se, por meio de sua descrição, é possível encontrar algum especialista para auxiliar nas respostas às perguntas propostas, e relacionar à ID do usuário;
- “Mural” para avaliar as perguntas e respostas cadastradas no ambiente e, a partir daí, sugerir uma resposta ou fazer os outros caminhos possíveis com a ferramenta;

O pacote “Mural” vincula-se apenas ao pacote “Segurança”, pois este é quem dá acesso ao usuário para utilizar as ferramentas. No pacote mural o usuário pode postar suas perguntas ou consultar respostas aos problemas já resolvidos.

O pacote comunicação estabelece o vínculo com os usuários através do pacote “Segurança”.

Para validar a proposta, estamos implementando o ambiente em partes. O primeiro protótipo conta com a implementação das seguintes ferramentas: Cadastro, Gerenciador e Simulador - NetPlay cujos detalhes apresentaremos neste trabalho. A ferramenta Esquematizador está em fase final de implementação. A ferramenta para criação de grupos está sendo implementada, assim como a ferramenta QSabe2. Essas estarão disponíveis no segundo protótipo. A implementação das ferramentas tem sido realizada por alunos do curso

de graduação em Ciência da Computação da Universidade Federal do Espírito Santo – UFES, como projeto de final de curso.

5.2.1 Ferramenta para criação de Grupo

Esta ferramenta tem o propósito de oferecer serviços que possam facilitar a criação dos grupos de usuários (listar cada participante do ambiente com suas preferências e atuações, enviar convites para formação de grupos, agendar horários, etc.). Ao iniciar o processo de criação de um grupo é importante que os usuários levem em consideração as afinidades entre os elementos do grupo, como por exemplo, a área de atuação, os trabalhos já desenvolvidos, a descrição feita pelo próprio usuário com relação a sua área de trabalho e preferências, entre outros.

Levando em consideração as necessidades acima sobre a formação do grupo, a ferramenta de criação de grupo deve estar interligada às demais ferramentas do ambiente para possibilitar que esses dados estejam acessíveis a cada usuário. Para a criação de um novo grupo o usuário deverá fornecer o nome e uma descrição resumida do propósito deste grupo. Isso servirá para a identificação do grupo, pelos usuários pertencentes ao sistema ou pelos usuários recém ingressados, convidados, que procuram um grupo com algum tema específico no qual deseja cooperar.

Um usuário pertencente a um grupo poderá enviar convites a outros usuários do ambiente para que os mesmos se associem ao grupo em questão. Também será possível convidar pessoas que não estejam cadastradas no sistema, ampliando, assim, a lista de cooperadores e possíveis especialistas.

Qualquer usuário pertencente ao ambiente poderá pesquisar por grupos cadastrados, consultar o nome do grupo, a descrição dos trabalhos realizados por ele, o usuário criador do grupo e a lista de seus usuários. Se houver algum grupo de interesse, o usuário poderá se associar a ele, selecionando-o de uma lista de grupos a que não esteja vinculado.

Após a formação do grupo, qualquer usuário membro poderá enviar mensagens aos demais membros. Para isso, ele terá duas opções: enviar um e-mail, bastando para isso, escrever o corpo da mensagem que, em seguida, o sistema se encarregará de enviá-la aos demais usuários; ou postando um aviso no mural do grupo.

Para facilitar o encontro dos usuários com mais afinidade ao problema a ser resolvido, a ferramenta possibilita consultar as demais ferramentas do ambiente (NetPlay, Gerenciador, Esquematizador) para verificar os trabalhos lá realizados. Uma visualização das macro funcionalidades da ferramenta pode ser acompanhada através da Figura 31.

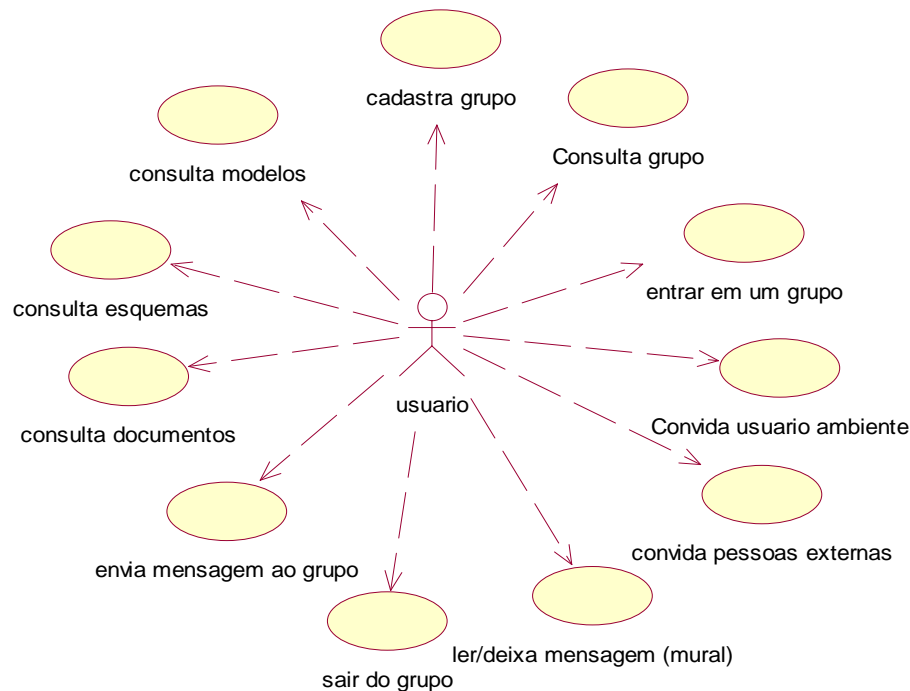


Figura 31 – Macro funcionalidades da ferramenta para formação de Grupos

5.2.2 Ferramenta QSabe2

Esta ferramenta será implementada, tomando-se como base o trabalho realizado por (Pessoa, 1997), adaptado às necessidades do ambiente AProSiMA. A sessão 4.2.1 apresenta as características da ferramenta original.

Para o ambiente AProSiMA a ferramenta QSabe2 atuará com algumas modificações com relação ao QSabe, como por exemplo, os participantes do ambiente podem tanto perguntar quanto responder questões propostas. A escolha do especialista respondedor será baseada através das características descritas pelo usuário no momento do cadastro e pelas ações realizadas por ele no ambiente (construção de modelos e esquemas, documentos selecionados como relevantes a uma consulta). Essas ações é que darão uma referência ao agente da ferramenta (QSabe) para selecionar o respondedor mais apropriado (contexto). As

ações realizadas no ambiente alimentarão o perfil de cada usuário. Perfil este obtidos por meio da descrição de suas ações. Este processo será realizado através de similaridade entre perguntas e habilidades dos usuários descritas em suas atividades.

À medida que as ações forem realizadas no ambiente, o perfil do respondedor e o nível de confiança nas respostas será alimentado com o *feedback* de suas respostas, dado pelo usuário que postou a pergunta e pela avaliação dos demais especialistas do ambiente. Os especialistas são aqueles usuários também selecionados para responder a questão. Nesse caso, as respostas dadas por cada especialista é avaliada entre seus pares.

As macro-funcionalidades da ferramenta QSabe2 são apresentadas na Figura 32, separadas em pacotes devido às suas afinidades funcionais

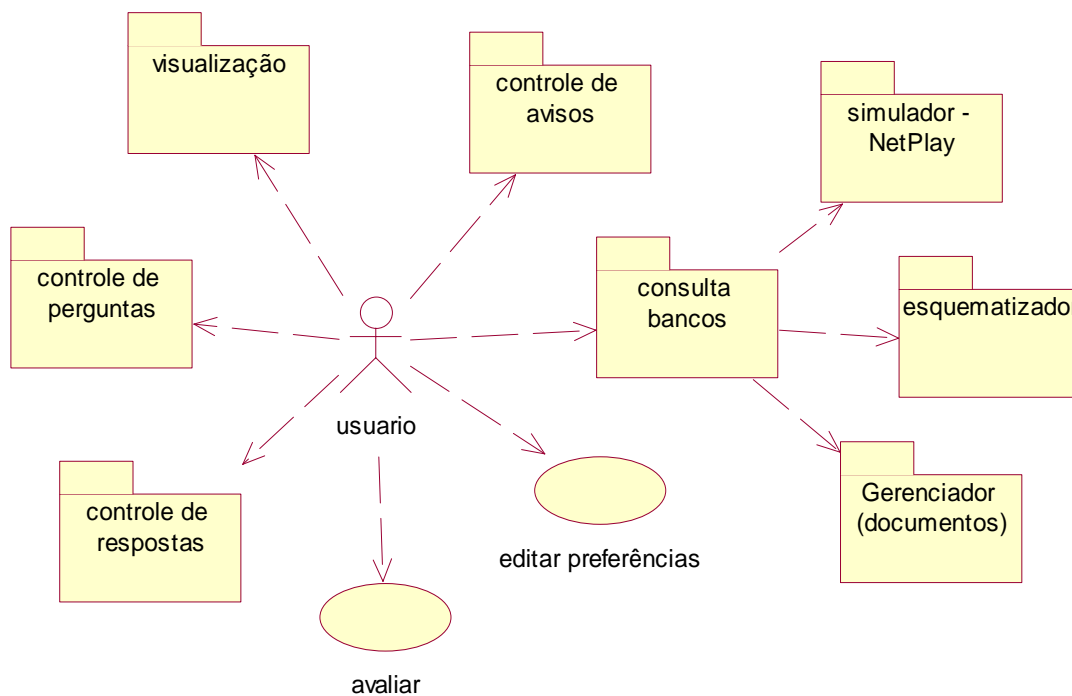


Figura 32 - Macro-funcionalidade da ferramenta QSabe2

As ações possíveis para cada usuário da ferramenta QSabe2 podem ser visualizadas por meio da Figura 33

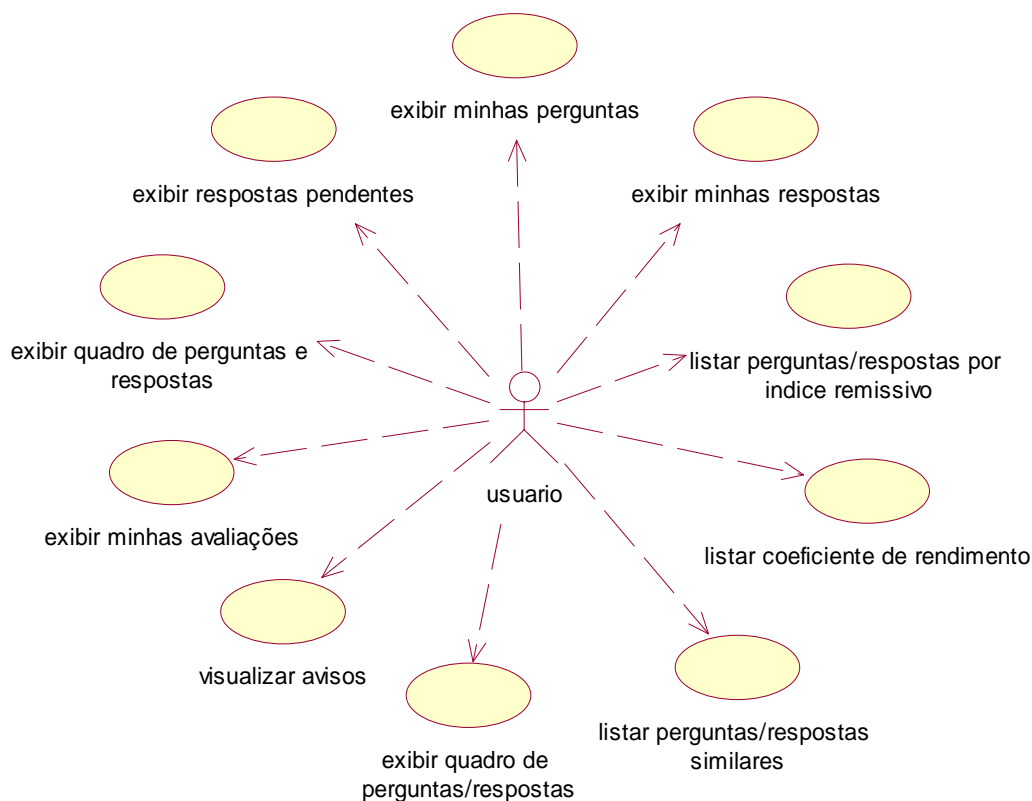


Figura 33 - Ações do usuário da ferramenta QSabe2

5.2.3 Ferramenta Esquematizador

O Esquematizador é uma ferramenta que tem como objetivo auxiliar o trabalho cooperativo no sentido de possibilitar aos usuários discutirem suas idéias de forma síncrona (fazendo uso de um chat) e formalizá-las por meio de esquemas gráficos (diagrama de fluxo - *flowchart*), agregando conhecimento e sinalizando um caminho de resolução de problemas.

Definimos um Esquema como uma representação esquemática de um processo de solução de problema, realizada por meio de um gráfico para ilustrar de forma descomplicada a transição de informações entre os resolvidores do problema. Em geral, um esquema é uma documentação dos passos discutidos no processo de resolução de problemas. Através de um esquema é possível seguir uma seqüência lógica para a elaboração do modelo de simulação.

A ferramenta Esquematizador deve, basicamente, oferecer duas funcionalidades principais aos usuários da ferramenta:

- possibilitar aos usuários do ambiente criar, de forma síncrona e cooperativa, esquemas referentes ao problema em discussão. Esses esquemas seriam formados basicamente por diagramas de fluxo, conforme pode ser observado na Figura 34.

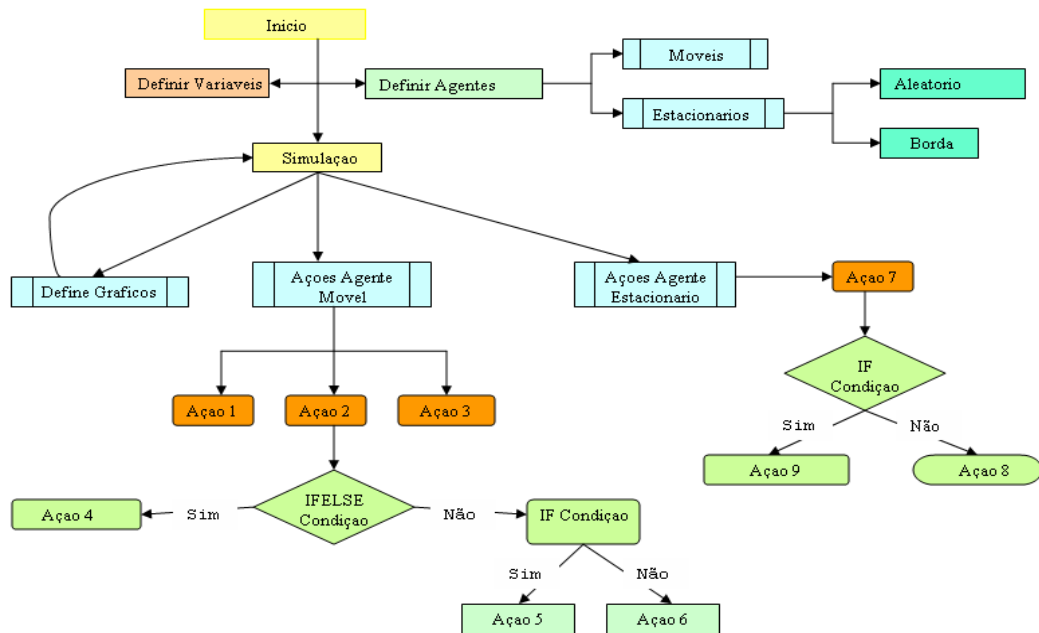


Figura 34 – Sugestão de como deve ser um esquema

- possibilidade de reutilização de esquemas. Quando um esquema criado for salvo, ele será inserido no banco de esquemas. Posteriormente, estes esquemas poderão ser acessados e, além de poderem ser visualizados, poderão ser reutilizados para criação de um novo esquema. A reutilização de um esquema pode ocorrer sem modificação alguma, ou como ponto de partida para novos esquemas, através de sucessivas modificações. O reúso de um esquema pode estar relacionado ao mesmo assunto ou a outro assunto que possa ter tomado por base a discussão anterior. Esse processo significa iniciar um novo esquema a partir de uma estrutura já existente (qualquer nó da árvore) que apresente relação com o problema a ser discutido.

Quando o usuário acessar a ferramenta, poderá escolher se deseja criar um novo esquema ou se deseja pesquisar por esquemas já existentes. Escolhendo criar um novo esquema, será necessário nomeá-lo e descrever sobre seu objetivo. Uma vez iniciado o processo de construção do esquema, o usuário criador poderá dar acesso a outros usuários do ambiente para que eles possam trabalhar de forma cooperativa. Durante todo o processo é

possível manter um diálogo síncrono com todos os cooperadores. Na Figura 35, podemos ter uma visualização do que seria desejável a essa ferramenta.

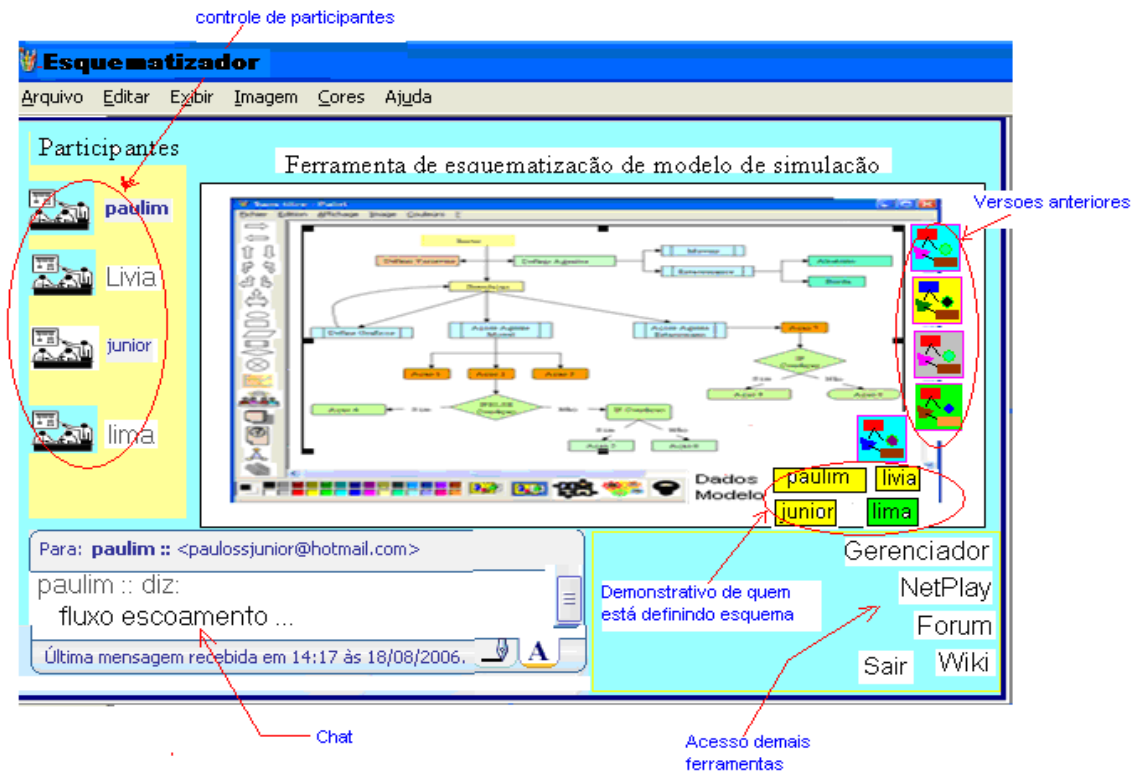


Figura 35 - Sugestão de como será a interface da ferramenta Esquemizador

A Figura 36 mostra as formas de interação do usuário com a ferramenta por meio de macro funcionalidades. O ator Usuário representa qualquer pessoa que queira fazer uso da ferramenta.

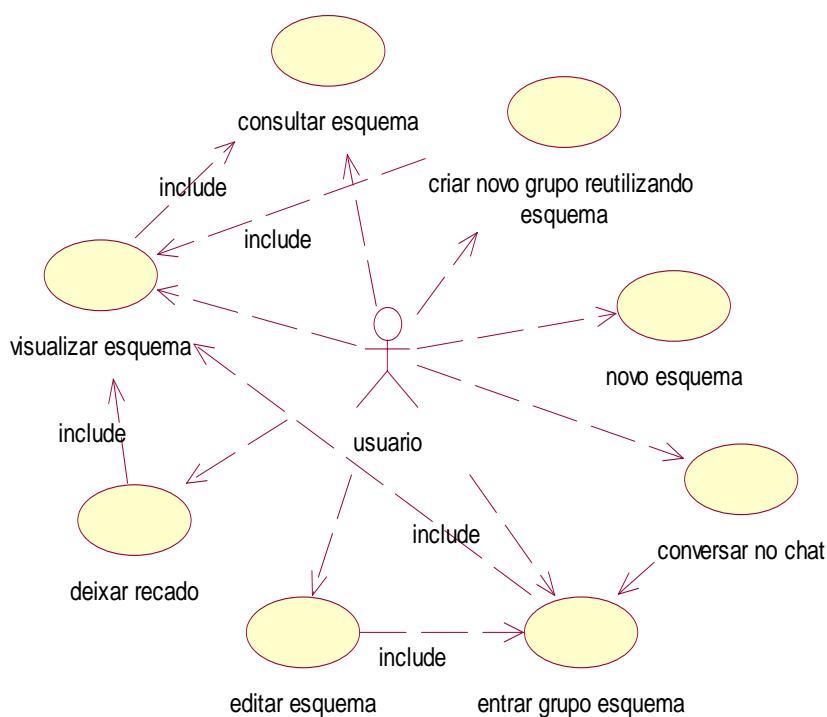


Figura 36 – Ações do usuário para a ferramenta Esquematizador

5.3 Ferramenta de comunicação

As ferramentas necessárias para compor o pacote comunicação são: chat, e-mail e fórum. Essas ferramentas seguem o padrão convencional utilizado por outros ambientes de interação entre usuários.

- a) chat – para possibilitar que os usuários se comuniquem entre eles de forma síncrona;
- b) e-mail – ferramenta para contato assíncrono entre os usuários;
- c) fórum – local onde os usuários poderão fazer seus comentários, tirar dúvidas, formalizar e conduzir uma discussão sobre um problema em questão.

As ações possíveis de serem executadas pelos usuários na ferramenta comunicação podem ser acompanhadas por meio da Figura 37.

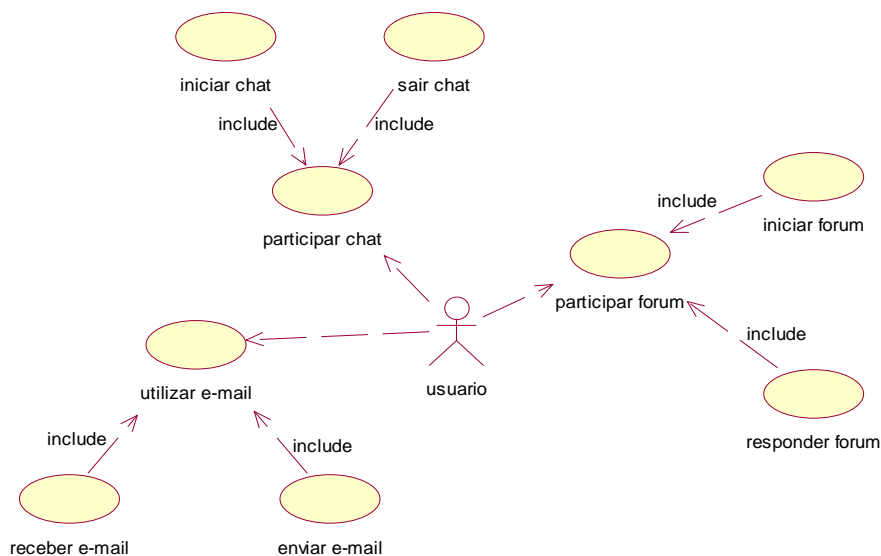


Figura 37 - Ações do usuário para ferramenta comunicação

5.3.1 Ferramenta Mural

Essa ferramenta é o dispositivo de contato direto entre os usuários do ambiente. Nela estarão expostos os problemas levantados pelos usuários e as respectivas respostas, quando essas já tiverem sido apresentadas. Os avisos na ferramenta mural são caracterizados como anúncios de descobertas ou solicitações a outros participantes do ambiente, como por exemplo: (fulano de tal diz) “*encontrei uns documentos interessante sobre formigas cortadeiras*”. As ações possíveis de serem executadas na ferramenta mural podem ser acompanhadas pela Figura 38.

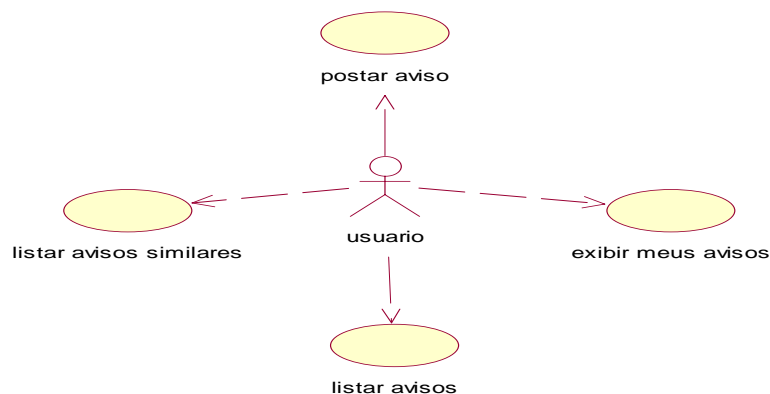


Figura 38 - Ações do usuário para ferramenta Mural

5.4 Ferramenta Gerenciador

Para atender as necessidades do ambiente AProSiMA, em localizar pessoas e documentos, desenvolvemos a ferramenta Gerenciador que atua por meio do “agente Fred”. O “agente Fred” trabalha juntamente com os usuários do ambiente na tarefa de recuperar informações pertinentes aos problemas em destaque, bem como localizar pessoas para auxiliar no processo de resolução cooperativa de problema. Esse agente tem, então, duas funções distintas: recuperar informações (documentos) e localizar pessoas através destes documentos (especialistas), conforme pode ser acompanhado pela Figura 39.

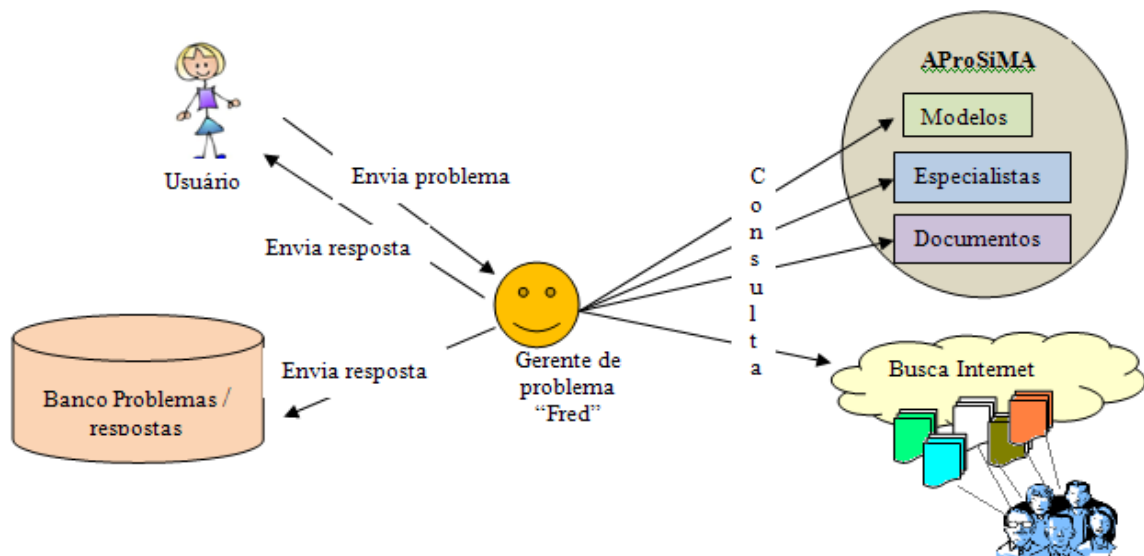


Figura 39 - Estrutura interna do Agente Fred

- Recupera informações – nesta função o agente tem a incumbência de receber os problemas (consultas) postados pelos usuários e sair em busca de informações que possam vir a resolvê-los. Para tanto, terá acesso à consulta aos bancos de modelos, banco de especialistas e bancos de documentos, todos já disponíveis no ambiente. Porém sua pesquisa maior se dá por meio da Internet, da qual ele, além de trazer os dados retornados do buscador associados a um resumo, ficará responsável por processá-los. Este processamento ocorre após uma referência do usuário. Por exemplo, Fred (o agente interno da ferramenta) indexará todos os documentos retornados e, através de processamento interno, retornará aqueles mais similares ao documento sugerido (*ranking*). O usuário dispõe, assim, de uma massa de documentos mais adequadas ao

seu interesse. Os documentos selecionados pelos usuários passarão a compor a base de documentos do ambiente.

- Localizar pessoas – trabalhando paralelamente às consultas dos usuários à Internet, o agente Fred monitora os documentos recuperados à procura de contatos (endereços de *e-mail*). Ao detectar essas pessoas, automaticamente, ele emite um convite para as mesmas, apresentando o ambiente e as convidando para compor o corpo de especialistas. Essa ação tem o intuito de auxiliar no processo de criação de grupos cooperativos de resolução de problema, para o qual seu conhecimento possa ser útil.

Pela Figura 40 é possível visualizar as ações que são realizadas na ferramenta Gerenciador.

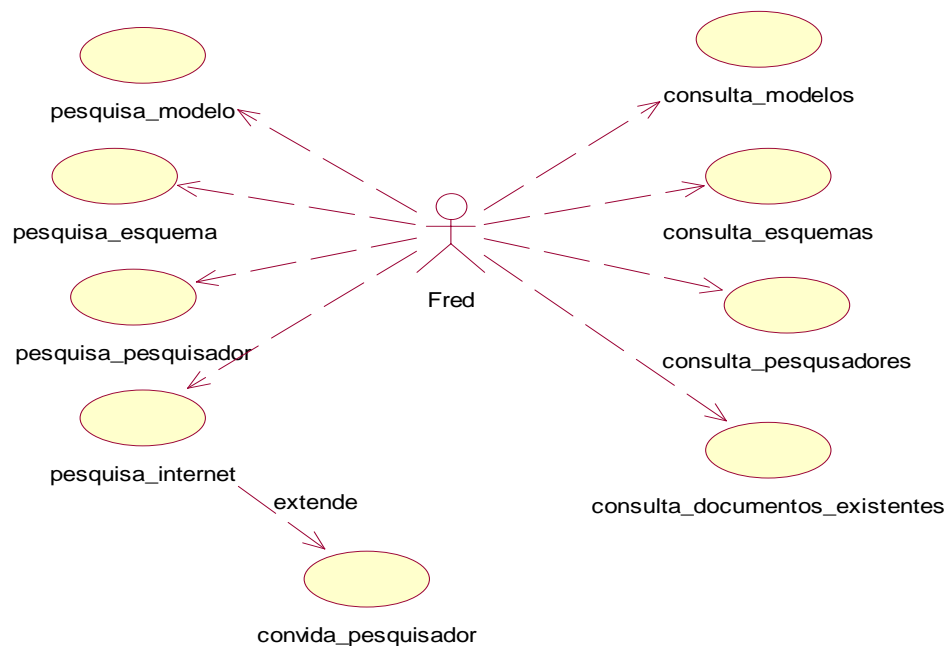


Figura 40 - Diagrama de caso de uso Gerenciador

A função da ferramenta Gerenciador é, basicamente, de recuperação de informação. Sendo essa ação realizada em dois locais distintos: no próprio ambiente (modelos, esquemas, documentos e pesquisadores) e através da internet (documentos e pesquisadores). O acompanhamento de como se dá o processo de busca na internet pode ser visualizado pela Figura 41.

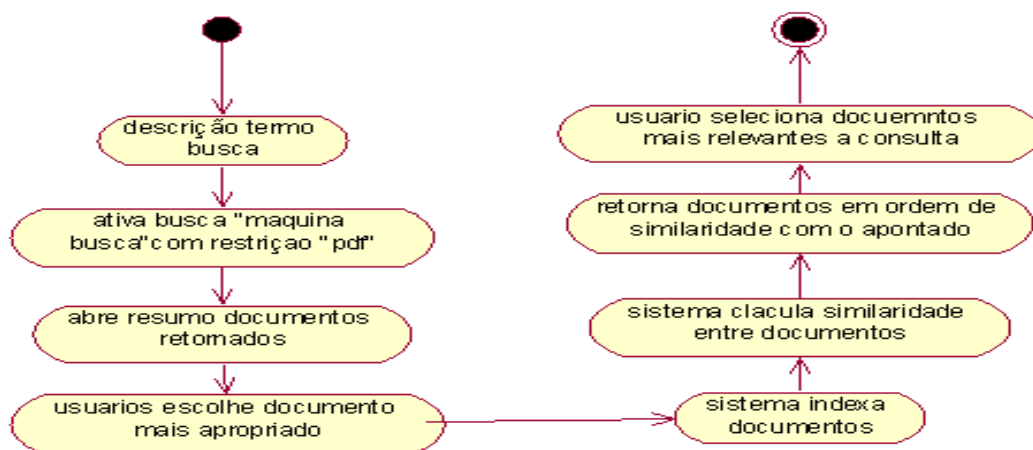


Figura 41 - Diagrama de atividade - pesquisa Internet

Para facilitar a realização do processo apresentado acima, Figura 41, utilizamos para implementação da ferramenta Gerenciador a API³⁹ (*Application Programming Interface*) do Lucene⁴⁰, que realiza as atividades de indexação, pesquisa, e similaridade entre documentos. O funcionamento do Lucene pode ser acompanhado pela Figura 42. A ação desta API ocorre a partir da necessidade de indexação dos documentos.

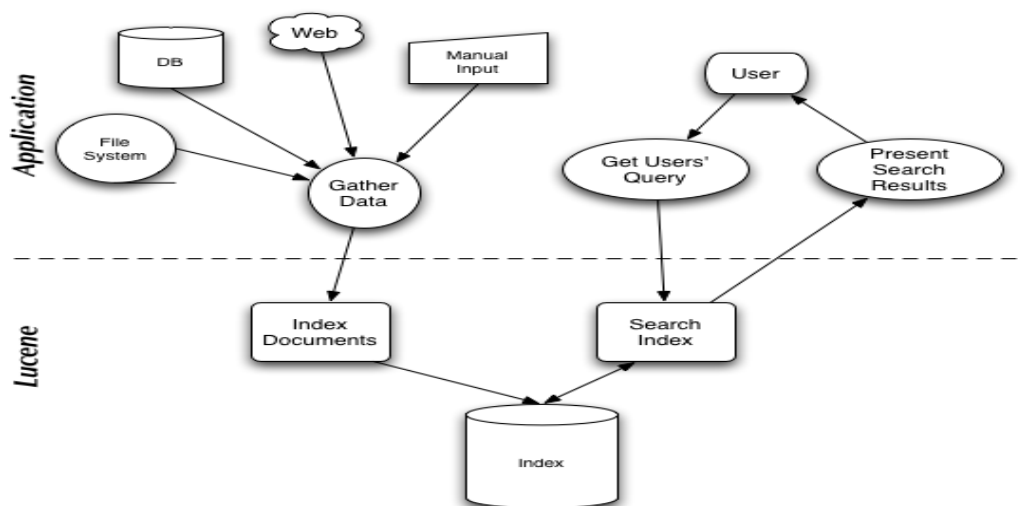


Figura 42 - Utilização da API do Lucene

³⁹ **API – Application Programming Interface** (ou **Interface de Programação de Aplicativos**) é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades por programas aplicativos -- isto é: programas que não querem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços (<http://pt.wikipedia.org/wiki/API>).

⁴⁰ Lucene é uma biblioteca de recuperação de informação open source, ou seja, é uma biblioteca de pesquisa e indexação. Foi originalmente implementada em Java por Doug Cutting, e hoje é apoiado pela Apache Software Foundation, e disponibilizada pela Apache Software License (Gospodnetić, 2004).

5.4.1 Sobre o funcionamento da API Lucene

A ferramenta Gerenciador está diretamente integrada à API do Lucene. Por meio da ferramenta Gerenciador é possível a busca na base de dados do ambiente AProSiMA ou pela internet, mediante o termo de busca indicado pelo usuário. Quando a busca ocorre na base de dados do ambiente, a ação é imediatamente realizada, conforme mostra a Figura 42. Quando a busca é realizada na internet, fazemos uso, primeiramente, de uma máquina de busca de uso público (para essa aplicação usamos o Yahoo! Cadê?). Esse fato ocorre, pois o Lucene não é um *web crawler*⁴¹, por esse motivo usamos uma máquina de busca para trazer os documentos para serem tratados por ela (indexação, pesquisa e similaridade entre documentos). Para restringir nossa busca, selecionamos apenas documentos no formato *pdf*.

De modo geral, o Lucene oferece uma API que requer entendimento mínimo sobre indexação e pesquisa em documentos textos. É necessário apenas manusear essas classes para usar o Lucene dentro de sua aplicação. Lucene é uma biblioteca Java, por esse motivo não faz restrições sobre o que está sendo indexado ou pesquisado, essa é uma característica que o coloca em vantagem em relação a outras aplicações de pesquisa. Para usar o Lucene não é necessário se preocupar com a origem dos dados, seu formato ou seu idioma, basta que estes dados estejam convertido em formato de texto. Isso significa que o Lucene pode indexar e pesquisar dados armazenados em: páginas web sobre servidores web remotos, documentos armazenados em sistemas locais, simples arquivos texto, documentos do Microsoft Word, arquivos PDF ou HTML, ou qualquer outro formato de arquivo de que possa ser extraído informações textuais.

Similarmente, com auxílio do Lucene é possível indexar e pesquisar dados armazenados em base de dados, possibilidades que muitas bases de dados não oferecem; mensagens de e-mail; mensagens instantâneas; chats; página wiki, entre outros. Além disso, a API do Lucene permite que a aplicação trate das regras de negócio para o domínio do problema enquanto encobre a complexidade da implementação da indexação e pesquisa. O Lucene pode ser pensado como uma camada sob a aplicação, como mostrado na Figura 42. Além disso, o Lucene permite adicionar capacidades de indexação e pesquisa à aplicação, por

⁴¹ Os *web crawler*, também conhecido como *web spider* ou *web robot*, é um programa ou um script automático que rastreia a web de modo metódico e automático. Muitos sites, em particular os sites de busca, usam os *spider* como uma forma de fornecer e atualizar seus dados. Os *web crawler* são usados principalmente para criar uma cópia de todas as páginas visitadas que posteriormente serão processadas pelas máquinas de busca (indexação e download) fornecendo assim uma pesquisa rápida (http://en.wikipedia.org/wiki/Web_crawler).

exemplo, é possível pesquisar por: “+camarão +arroz -pudim, maçã -torta +tigre”, “animal: macaco AND comida: banana”, e assim por diante.

Para descrever melhor como a ferramenta Gerenciador trabalha, faremos um breve apanhado sobre recuperação de informação.

5.4.2 Recuperação de informação

Há séculos o ser humano tem tido por prática armazenar informações para utilizá-las posteriormente. Com o aumento cada vez mais rápido de informações um dos grandes problemas das pessoas é armazenar e, posteriormente, encontrar os arquivos armazenados de modo fácil e rápido. A organização e localização de informações relevantes é uma tarefa complicada. Os sistemas de recuperação de informação (RI) surgem, neste contexto, como um processo no qual os documentos são organizados na forma de um grande catálogo – *índex*. Um *índex* é uma antiga e bastante eficaz estrutura de dados utilizada para catalogar um conjunto de palavras e onde cada palavra aponta para um determinado documento para acesso posterior (Baeza-Yates, et al., 1999). A indexação de documentos é um processo básico utilizado no âmbito de Recuperação de Informação (RI).

Neste contexto, estudos na área de RI têm como objetivo, além de buscar documentos, trazer aqueles que realmente são importantes para o usuário. Para realizar tal tarefa vários modelos têm sido explorados. Os três modelos clássicos de RI são: booleano, vetorial e probabilístico (Baeza-Yates, et al., 1999). Esses modelos consideram que cada documento é descrito como um conjunto representativo de palavras chaves chamado de *termos indexados*. Esses termos (palavras) indexados são utilizados como forma de acesso e recuperação de informação. Essa indexação pode ser utilizada de maneira específica, pois os termos indexados são normalmente os substantivos, porque estes carregam significado em si mesmo e sua semântica é fácil de entender e recuperar. Os substantivos são considerados as palavras chaves dos documentos. Aqueles termos que trabalham principalmente como complementares, tais como: adjetivos, advérbios e conectivos, normalmente, não são úteis como termos indexados. Esses termos são denominados *stopwords*. De maneira mais genérica, o sistema pode indexar qualquer tipo de palavra encontrada no documento. Tanto os substantivos como os termos complementares podem ser utilizados em situações diferentes, dependendo da necessidade do contexto em que se encontra a aplicação (Baeza-Yates, et al., 1999).

Através desses documentos indexados é possível, então, que um sistema de RI possa comparar o termo de busca dado pelo usuário, que também será indexado, com os demais termos existentes em seu catálogo e recuperar aqueles documentos que são apontados pelo index. O processo de indexação de documentos é uma maneira que nos permite armazenar informações de forma estruturada para posterior recuperação.

As três principais técnicas de indexação são (Baeza-Yates,1999):

- arquivos invertidos (*inverted files*) - armazena uma lista de documentos para cada palavra;
- árvore de sufixos (*suffix trees* or *suffix arrays*) – são rápidos para pesquisar por frases, mas são difíceis de construir e manter;
- arquivos de assinatura (*signature files*) – foram populares na década de 80, é essencialmente uma lista de pares que consiste de um documento e palavras, colecionadas por um documento.

Atualmente, a melhor escolha para as aplicações têm sido os arquivos invertidos (Baeza-Yates, et al., 1999). A estrutura de um arquivo invertido é composta por dois elementos: o vocabulário e a ocorrência. O vocabulário é o conjunto de todas as palavras diferentes do texto. Para cada palavra, uma lista de todas as posições em que essa palavra aparece no texto é armazenada. O conjunto de toda aquela lista é chamado de ocorrência. Essa posição pode referir à palavra ou ao caractere. A Figura 43 apresenta um exemplo de index invertido.

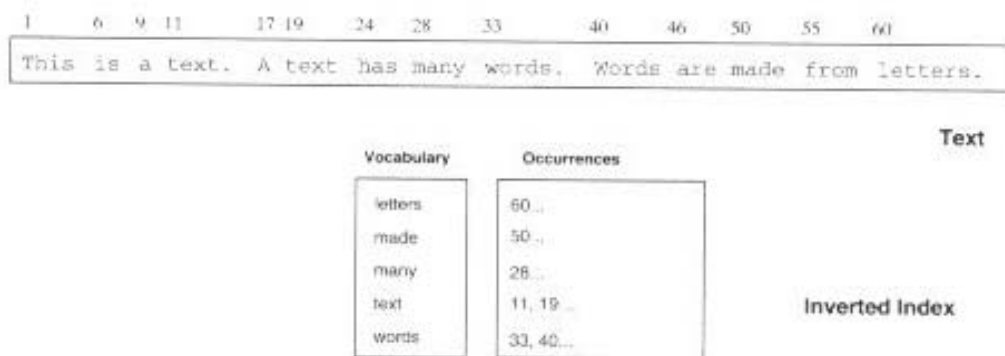


Figura 43 - Exemplo de um texto e um index invertido construído sobre ele. As palavras são convertidas para minúscula e algumas não são indexadas (Baeza-Yates,1999).

Uma vez que os termos dos documentos estão indexados, a pesquisa (*searching*) é o processo de olhar esses termos no índice para encontrar o documento onde ele aparece. A qualidade da pesquisa é tipicamente descrita usando métricas de “*precision*”⁴² e “*recall*”⁴³. Em geral, o “*recall*” mede o quanto a procura do sistema acha documentos relevantes, considerando que “*precision*” mede o quanto os filtros do sistema desconsideram os documentos irrelevantes. Entretanto, outros fatores devem ser observados em relação à pesquisa: velocidade de busca, capacidade de, rapidamente, pesquisar uma grande quantidade de textos, suporte para busca por um único termo ou vários termos, ranking dos resultados e também o uso de uma sintaxe amigável para composição da consulta (Gospodnetic, et al., 2005).

Quando se trata de recuperar informações disponíveis na Internet, utiliza-se um recurso denominado “máquina de busca” (“*web search engine*”). Cada máquina de busca tem um algoritmo específico de recuperação de informação, mas todas utilizam indexação e pesquisa. Entretanto, antes de realizar a indexação, é necessário coletar os documentos que estão disponíveis na web. Esta ação é realizada por meio dos *web crawlers* (também conhecidos como *web spider* ou *web robot*) que são programas ou scripts automatizados que “varrem” (browser) a *world wide web* de modo metódico e automatizado. Os *web crawler* são, principalmente, usados para criar uma cópia de todas as páginas visitadas, que, posteriormente, serão processadas pela máquina de busca que indexará as páginas carregadas, para possibilitar uma busca rápida. Os *web crawlers* são a parte central de uma máquina de busca, e detalhes de seu algoritmo de rastreamento é mantido como segredo de negócio (Yates, et al., 2005).

5.4.3 Atuação da ferramenta Gerenciador

Para nosso objetivo, entendemos a web similar a um grafo dirigido, sendo os links as arestas (ou arcos) e as páginas web os nós (ou vértices) que contêm as informações. Neste sentido, as máquinas de busca realizam de fato um trabalho fantástico, ou seja, elas possibilitam navegar sobre este grafo e nos traz centenas de informações, em particular, textos. Mas o que fazer com todos esses textos? Será que todos eles servem? As máquinas de

⁴² *Precision* – é definido como a relação entre documentos relevantes recuperados e todos os documentos recuperados pela pesquisa. (Baeza-Yates, et al., 1999)

⁴³ *Recall* – é definido como a relação entre documentos relevantes recuperados e todos documentos relevantes. (Baeza-Yates, et al., 1999).

busca retornam ao usuário uma lista “ranqueada” dos documentos relevantes como resultado da pesquisa. Entretanto, segundo (Barfouroush, et al., 2002), poucos desses resultados podem ter valor para o usuário. Por essa razão pode ser útil usar alguma informação do usuário, no contexto da informação, para melhor se aproximar do seu objetivo.

As máquinas de busca na internet, segundo (Barfouroush, et al., 2002), aceitam a consulta do usuário como um conjunto de palavras chave dentro de um formulário de expressão booleana. Infelizmente, o método de comparação retorna, geralmente, uma grande quantidade de resultados de baixa qualidade. Os mesmos autores afirmam que a formulação de algumas perguntas no formulário de expressão booleana é difícil ou impossível. Uma das grandes dificuldades é o fato de que as pessoas encontram na linguagem natural uma forma de expressar a sua necessidade de informação, mas com palavras chaves nem sempre é possível cobrir a complexidade semântica do que se deseja expressar.

Com o intuito de trazer ao usuário os documentos mais próximos de seu interesse, a ferramenta Gerenciador tem a finalidade de agrupar os documentos retornados da busca, de forma mais aproximada possível. Para realizar esta ação, entendemos que usar apenas alguns termos chave não aproxima os documentos suficientemente. Assim, mediante a indicação do usuário sobre um documento relevante, a ferramenta Gerenciador processará todos os documentos encontrados comparando-os a todo o documento referenciado, e não apenas alguns termos chave. Com esse novo processamento, o resultado indicará uma seqüência ordenada de documentos mais similares àquele escolhido.

Uma vez selecionado os documentos relevantes ao interesse do usuário para a resolução do problema em tela, duas opções são oferecidas:

- a) simplesmente armazenar o documento para consultas posteriores pelo próprio usuário ou por outro usuário que tenha interesse no mesmo assunto;
- b) armazenar o documento e solicitar ao agente Fred que faça uma busca pelo autor do documento e outros pesquisadores referenciados no texto, para que seja emitido um convite para participar do processo de resolução cooperativa, indicando que um usuário (ou grupo) está trabalhando sobre o assunto e que o documento referido foi importante para o desenvolvimento do problema, sobre o assunto que o autor tem domínio de conhecimento.

Com esta ação entendemos estar localizando e envolvendo as pessoas geograficamente distribuídas, que têm domínio de conhecimento, para formar o grupo de resolução cooperativa de problemas. Paralelamente, estamos encontrando o respondedor mais apropriado para atender as questões propostas, aumentando a comunidade de resolvedores de problemas.

Até este momento do trabalho, não houve preocupação com relação à otimização do sistema no sentido de desempenho da ferramenta, mas métodos computacionalmente exeqüíveis serão analisados e implementados nos protótipos subsequentes.

5.5 Ferramenta de simulação – NetPlay

O desenvolvimento de um projeto usando a simulação multiagente consiste, em termos gerais, em reproduzir artificialmente um fenômeno natural, cujo objetivo é ganhar melhor entendimento sobre o fenômeno descrito e enriquecer nossa compreensão acerca de processos fundamentais que podem aparecer em uma variedade de aplicações.

A ferramenta **NetPlay** está sendo desenvolvida no intuito de possibilitar e estimular os usuários a criarem, cooperativamente, modelos de simulação, na perspectiva do uso da modelagem multiagente. Neste sentido, o modelador representa a sua aplicação criando um modelo para ela. A modelagem consiste em representar um problema ou uma situação real, utilizando, para isso, um grupo de agentes reativos que interagem entre si e com o ambiente no qual está inserido. Através do modelo é possível estudar o comportamento dos agentes, analisar o resultado de suas interações e as conseqüências advindas das tomadas de decisão, baseadas nos dados de entrada no modelo.

O modelador define a situação inicial do seu modelo, representando através da composição dos agentes e do ambiente, e determinando como os agentes se comportarão com base nas regras de interação entre esses. Após estas definições, o ambiente se encarrega de executar a simulação e apresentar uma situação final, atingida pela atuação dos agentes. O ambiente possibilita testar várias situações, por meio da mudança de parâmetros e alteração dos dados que definem o modelo.

A maioria dos ambientes de modelagem de simulação baseado em agentes, citado no Capítulo 3, exige conhecimento e o uso de linguagem de programação específica. O ambiente **NetPlay** está sendo desenvolvido para fins didáticos, sendo um dos seus objetivos a fácil utilização. A proposta de uma nova ferramenta de simulação multiagente – NetPlay, tem

como propósito superar as dificuldades apresentadas nas ferramentas já existentes. Para tanto propõe:

- a) uma nova linguagem, que seja acessível a usuários leigos em linguagem de programação;
- b) uma nova metáfora de modelagem, de modo que os usuários não precisem conhecer a teoria de agentes;
- c) uma nova forma de interação entre os usuários, usando a web como plataforma de desenvolvimento; e
- d) novas ferramentas de apoio ao entendimento do modelo.

Assim, a proposta da ferramenta é possibilitar e estimular os usuários a criarem, cooperativamente, modelos de simulação, na perspectiva do uso da modelagem multiagente, da capacidade de reúso dos objetos projetados e visando diminuir o esforço da programação, para que os usuários possam visualizar o comportamento resultante das interações definidas e fazer conjecturas em relação ao modelo e tomada de decisão.

5.5.1 Linguagem de programação

Programar um computador em uma linguagem de programação convencional, como por exemplo; as linguagens baseada em texto, requer um sentido preciso da estrutura de programação, o que não é uma prática para muitos usuários. Um estilo de programação que é fácil de ser entendida por todos, principalmente por não programadores ou novatos em programação, é a programação visual (VP).

Tanto nos programas visuais quanto nas linguagens visuais, os objetos de interesse são representados por figuras e símbolos. Uma das razões para usar representações gráficas é o fato de possibilitar aos usuários aplicarem o conhecimento de como o ícone do objeto é manipulado na interface, sem se preocupar como sua representação é manipulada no programa. Outra razão para usar linguagem visual é que os programas visuais são mais fáceis de entender do que os baseados em texto (Chang, 1987), (Modugno, et al., 1994).

Um sistema de programação visual apresenta a estrutura do programa em duas ou mais dimensões, usando expressões visuais tais como gráficos, quadros, figuras, ícones, entre outros. Vários pesquisadores têm defendido o uso da programação visual, descrevendo as

muitas vantagens em relação às linguagens de programação textual, (Smith, 1997), (Smith, et al., 2001), (Repenning, 1993), (Glinert, et al., 1990), (Lieberman, 2001), (Repenning, et al., 2000), (Koelma, et al., 1992), (Resnick, 2007).

De modo geral, uma linguagem visual é uma representação pictorial de entidades conceituais e operações, sendo essencialmente uma ferramenta por meio da qual os usuários compõem sentenças usando objetos visuais ou ícones. Segundo (Koelma, et al., 1992) seu layout e uso de ícones parece ficar mais próximo do modo de pensar humano, o que simplifica a transformação da representação usada pela mente, ao pensar sobre um problema para a representação usada na programação do problema. A linguagem visual torna um caminho mais curto e fácil para compreender, aprender, usar e recordar. O uso de elementos pictoriais é importante porque, freqüentemente, as imagens dão mais significado do que textos: uma única imagem pode valer por mil palavras (o inverso também pode ser verdadeiro).

Os ícones, geralmente, se referem à imagem física de um objeto (Chang, 1987). Uma sentença é um arranjo espacial de objetos ícones e / ou conjunto de ícones que normalmente descrevem uma entidade conceitual complexa ou uma seqüência de operações. O objeto ícone representa entidades conceituais ou um grupo de objetos ícones que são arranjados de um modo particular. A combinação dos ícones é realizada através de operadores. De modo geral, uma operação é uma expressão como $x1 \text{ op } x2 = x3$, onde os ícones $x1$ e $x2$ são combinados através do operador op , formando $x3$. Para mais detalhes sobre a programação visual, recomendamos (Chang, 1987), em que se pode estudar sobre os objetos, as operações e a gramática da linguagem visual.

Usando ferramentas de programação visual, é possível construir aplicações sem apelo à programação textual. A programação visual pode ser considerada também como uma ponte para tornar a programação mais fácil. Isto faz com que o usuário possa se concentrar mais na resolução de problemas e menos na escrita dos programas. Geralmente, as linguagens visuais são sempre orientadas a objetos e/ou a eventos. Descreveremos a seguir o modelo de programação a ser utilizado na ferramenta NetPlay.

5.5.2 Ambiente de desenvolvimento

O ambiente de desenvolvimento é onde o modelador relaciona os elementos do modelo, é onde ele define os componentes do modelo. Fazendo analogia com a programação orientada

a objetos há: agentes (que são as classes), as regras de comportamento (que são os métodos) e as características de todos os agentes (que são os atributos). A arquitetura⁴⁴ do ambiente, Figura 44, para a concepção do modelo não exige conhecimento algum do modelador sobre o uso de agentes ou linguagem de programação. Cada passo da arquitetura será comentado na seqüência.

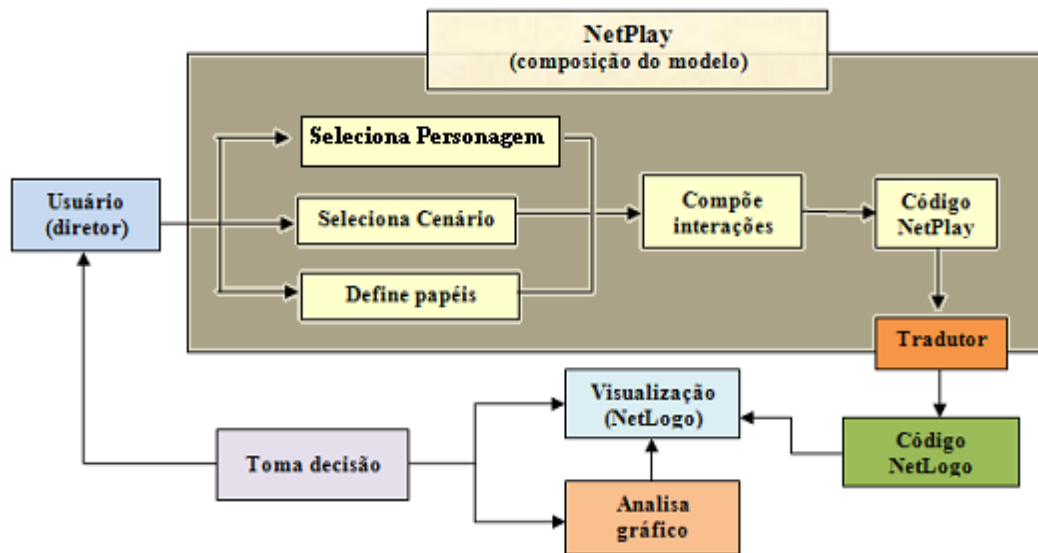


Figura 44 – Estrutura interna do NetPlay

5.5.3 Descrição dos Modelos

A descrição de um modelo de simulação multiagente no NetPlay pode ser comparada a uma peça teatral. Usando essa comparação, destacamos os seguintes elementos: a) o diretor – é o sujeito que caracteriza o cenário e os personagens e, também, orchestra toda a peça, neste trabalho representado pelo usuário; b) o cenário – é definido pelos elementos que compõem o palco, local onde a peça se desenvolve; c) os personagens – são os elementos vivos da peça, que são representados por atores, que agem de acordo com o seu script.

No modelo de teatro há várias formas de intervenção do diretor para conduzir a encenação, são elas:

- teatro de fantoche (manipulação total do modelador);

⁴⁴ Enquanto um modelo *explica* o comportamento de um sistema, a arquitetura diz como esse sistema opera. Evidentemente, um sistema pode ser explicado segundo vários modelos e um mesmo modelo pode ter várias arquiteturas que, por sua vez, podem ter várias implementações computacionais.

- teatro sem script (o diretor caracteriza os personagens, representados por atores, e estes desempenham o papel – o diretor os observa e faz intervenções);
- teatro com script e intervenção (os atores sabem seus personagens e o diretor vai interferindo no decorrer da apresentação)

Para este trabalho, adotamos o modelo de teatro sem script. Após selecionar os personagens e os objetos para compor o cenário, o diretor passa, então, a descrever o comportamento de cada um dos personagens, definindo como deve ser a sua atuação na peça. Essa discriminação está relacionada com os atributos conforme mencionados no tópico 2.2.3, para identificar aspectos físicos, comportamento e ações. Essas características definem os elementos básicos dos personagens (agentes).

Neste modelo o diretor (usuário) seleciona os personagens e os objetos para compor o cenário, Figura 45. Após essa seleção, o diretor passa a descrever o comportamento de cada um dos personagens, definindo como deve ser a sua atuação na peça, e posicionando os objetos no palco. Feito isso, passa à composição das regras de interação entre os componentes da peça. O diretor faz uso de um conjunto de regras descentralizadas, isto é, cada personagem tem o seu próprio script (conjunto de composições). As regras são definidas por condições e ações que modelam o comportamento dos personagens. No palco ocorrem as interações entre os personagens (ator x ator) e com os elementos que compõem o cenário (ator x cenário), segundo os seus scripts. A interação entre os vários personagens e os elementos que compõem o cenário faz emergir o comportamento descrito no modelo (peça teatral).

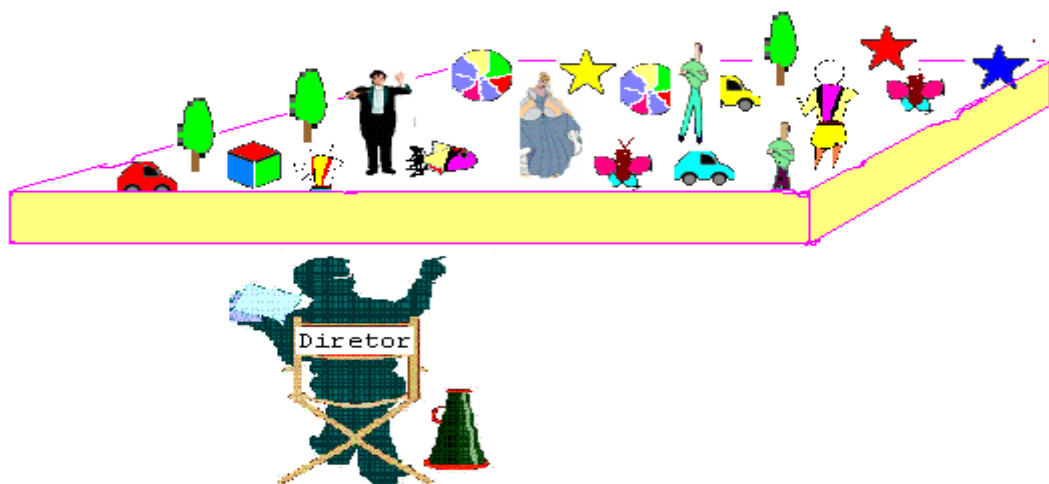


Figura 45- Metáfora do teatro - Diretor selecionando equipe para encenar peça

De modo geral, a maioria dos usuários sabe descrever claramente as características, o comportamento e a forma de interação dos personagens (agentes), porém não sabem programar de que modo essas características e essas interações podem ocorrer. Assim, no NetPlay, o usuário descreve o modelo usando elementos desta metáfora e escolhendo os comportamentos típicos para compor cada papel. O ambiente de simulação passa, então, a ser visto como uma metáfora do teatro de representação, onde o diretor (usuário) assiste o desenvolvimento das interações e estuda o modelo simulado que representa um retrato do mundo real, podendo interferir e tomar decisões sobre as possibilidades e ocorrências.

5.5.4 Definição dos agentes

O NetPlay possibilita a criação de três tipos de agentes (personagens, segundo a metáfora): móveis, estacionários e borda. Como o próprio nome sugere, os agentes móveis têm a capacidade de percorrer o ambiente que é definido pelos agentes estacionários, isto é, os que compõem o ambiente de interação. O agente borda é utilizado para delimitar uma região no ambiente. O diretor seleciona os personagens segundo o *script* que define o comportamento a ser executado. A Figura 46 representa a criação dos agentes no NetPlay e a definição de seu comportamento, essa ação é vista com a escolha do ator que representará o personagem. O diretor define quais os personagens que iniciarão a peça e a quantidade de atores necessária.

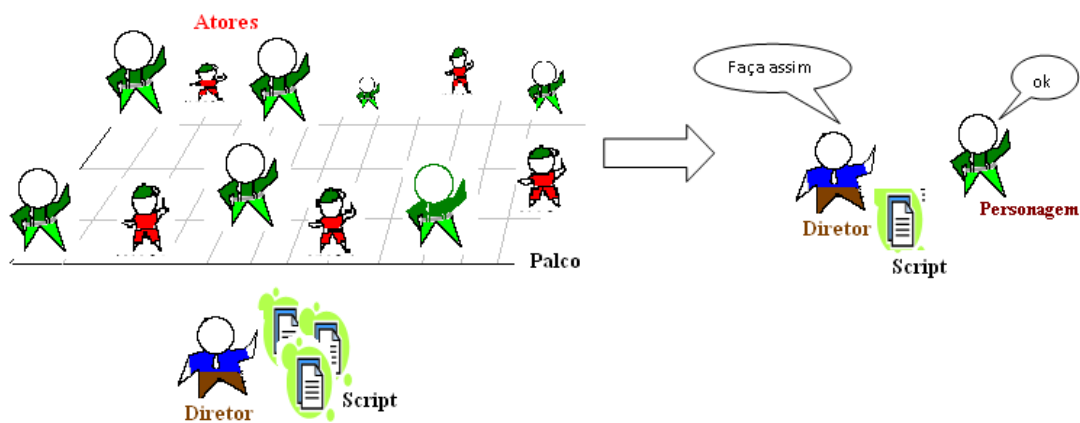


Figura 46 - Criação de agentes no NetPlay

Os personagens são a coleção de agentes que partilham o mesmo espaço, interagindo e se comunicando uns com os outros. Os agentes são equipados com um mecanismo de comunicação (regras de condição x ação) que lhes permite interagir uns com os outros no

mesmo espaço partilhando. Os personagens, nesse modelo, são agentes reativos, que possibilitam a manipulação direta baseada em regras de comportamento predefinidas.

5.5.5 Definição de variáveis

As variáveis, neste modelo, estão relacionadas aos papéis que os agentes desempenharão. Elas são utilizadas para guardar valores e auxiliar na simulação, por exemplo: computar a idade, armazenar um valor, computar os ciclos do modelo, entre outras ações. Por meio das variáveis o diretor pode alterar o comportamento dos atores e, conseqüentemente, da simulação.

5.5.6 Regras de comportamento

Como já mencionado nos capítulos anteriores, o comportamento que os agentes descrevem num sistema complexo são relativamente simples, as suas interações é que proporcionam um fenômeno mais elaborado. Para definir o comportamento dos agentes, mapeamos as características predominantes nos modelos do mundo real e os sintetizar em situações que pudessem ser combinadas, de modo a mapear o maior número possível de relações comportamentais.

Em geral, os comportamentos mais visíveis nos modelos de sistemas complexos estão relacionados à percepção de outros agentes ou objetos, e do posicionamento de algum outro objeto ou o seu próprio. Os atributos (forma, cor) dos agentes também são elementos que possibilitam definir comportamentos. As ações de reprodução e destruição possibilitam alterações de comportamento dos agentes. Da mesma forma, a inserção de variáveis ampliam as possibilidades de relações comportamentais. Desse modo, seguindo o modelo de ferramenta que estamos apresentando, criamos as seguintes possibilidades para definição de comportamento dos agentes:



percepção de outros ;



valor de variável;



posicionamento;



recuperar valor de variável;



forma, cor;



reproduzir - gerar um novo;



destruir, morrer;

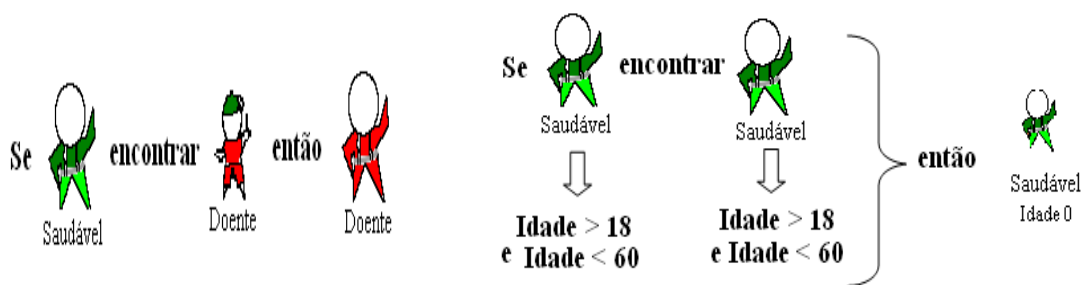


alterar posição;

Com esses elementos, o diretor tem a sua disposição uma série de comportamentos que podem ser utilizados para fazer novas combinações e para compor os relacionamentos. Os comportamentos dos agentes que definem a simulação são obtidos através de regras lógicas “*se condição*” ... “*então ação*”. As **composições**, que descrevem o comportamento dos agentes, são compostas em duas etapas: a primeira determina as situações que representam as condições e, posteriormente, as situações que representam as ações decorrentes dessa. A Figura 47 (a) (baseada no modelo de transmissão de vírus numa população), representa a seguinte regra de comportamento: *se* um agente **saudável** *encontrar* (significa perceber outro agente à distância 0) um agente **doente**, *então* ele se *contamina* e também fica **doente**, (condição = encontrar; ação = contaminar).

Cada uma dessas opções remete a uma construção, seja selecionando os elementos (por exemplo, a percepção indica que dois agentes devem se encontrar a uma determinada distância), seja compondo uma expressão.

Para as composições que necessitam de mais de uma condição, é utilizado os operadores lógicos (e) e (ou). A Figura 47 (b) indica que, se dois agentes saudáveis se encontrarem e a idade de ambos estiver no intervalo ente 18 anos e 60 anos, eles vão gerar um descendente com idade zero. Uma mesma condição pode levar a mais de uma ação. Da mesma forma, há ações que não necessitam satisfazer nenhuma condição, por exemplo: incrementar tempo ou atualizar variável idade.



a) Composição de contaminação b) Composição com duas condições e uma ação

Figura 47 - Composições (regras de comportamento)

5.5.7 Ferramentas de monitoramento

São oferecidas duas ferramentas para monitoramento do modelo: monitores e gráficos. Através dos monitores é possível inspecionar a variação dos agentes no decorrer da simulação, geralmente por meio de uma contagem. Paralelamente ao palco de visualização do movimento dos agentes, o comportamento do modelo pode ser observado através da plotagem num gráfico (linha, barra ou pontos) das variáveis que controlam o comportamento dos agentes. A Figura 48 representa a análise do diretor ao ver o desenvolvimento dos personagens no palco. O diretor acompanha o desenrolar das ações observando os elementos apresentados pelos monitores e pelo gráfico.



Figura 48 - Visualização do modelo

5.5.8 Realização dos Modelos

A realização do NetPlay requer uma máquina abstrata onde seja possível encenar as peças concebidas por seus usuários. Para agilizar a implantação do sistema, estamos traduzindo os modelos desenvolvidos em NetPlay para programas em NetLogo, conforme pode ser acompanhado pela Figura 44. O tradutor é o conjunto de bibliotecas que decodifica as características definidas na linguagem de programação visual e a transforma em uma linguagem de programação baseada em texto. A estratégia tem por finalidade experimentar o modelo antes da concepção de uma máquina abstrata específica.

A tradução do modelo poderia ser feita para qualquer linguagem, mas optamos pelo NetLogo por economia de tempo em implementação desnecessária de interfaces, uma vez que o ambiente NetLogo já traz embutido nele toda essa estrutura. Desta forma, para visualizar o funcionamento do sistema, têm-se duas opções: a) acompanhar através do ambiente NetLogo,

usando o código gerado pelo NetPlay ou, b) utilizar um *browser* para visualizar o *applet* do modelo.

5.5.9 Ferramenta de análise gráfica

Em geral, durante a execução de uma simulação ocorrem muitos ciclos. Devido à quantidade de pontos gerados e plotados no gráfico, uma visualização mais pontual torna-se difícil de precisar. Além do mais, é importante para os usuários ter ferramentas de acompanhamento que os auxiliem no entendimento do modelo.

Desta forma, incorporamos à ferramenta de construção de modelos de simulação um dispositivo de análise gráfica que permite avaliar precisamente: os pontos de interseção entre todas as variáveis plotadas; os pontos de coordenadas máximas e mínimas; os pontos principais de cada variável e as coordenadas relativas à mesma. Para uma visão mais geral, todos os pontos são apresentados simultaneamente num mesmo quadro.

Outro elemento de destaque do dispositivo de análise gráfica é a possibilidade de exportar os dados em forma de planilha, para que possa ser lido em outras ferramentas (*nome_arquivo.csv*). Esta facilidade permite que os dados sejam trabalhados isoladamente e novos gráficos possam ser construídos de acordo com o interesse do usuário.

Capítulo 6

6 Três implementações para um mesmo modelo – comparação de ferramentas

Testamos a funcionalidade e abrangência da ferramenta de construção de modelos de simulação multiagente NetPlay, comparando com duas outras ferramentas existentes RepastS [<http://repast.sourceforge.net/>] e NetLogo [<http://ccl.northwestern.edu/netlogo/>]. Para essa comparação usamos o modelo “presa / predador”, um representante do sistema complexo.

6.1 Situando o modelo

O modelo selecionado representa uma variação do comportamento presa/predador usando três agentes: lobos, ovelhas e grama. Tanto os lobos quanto as ovelhas andam aleatoriamente sobre uma grade, ambiente representado pela grama, e o movimento dos animais tem um custo em termos de perda de energia. Os lobos e as ovelhas precisam se alimentar para repor sua energia. Eles morrem, se o seu nível de energia for reduzido a zero. O lobo come a ovelha, se ambos estiverem localizados na mesma posição espacial. Essa possibilidade de ação faz aumentar a energia do lobo. Similarmente, a ovelha pode comer a grama se estiver localizada na mesma posição em que vive a grama, ganhando uma quantidade de energia. Uma vez que a ovelha come a grama, essa precisa renascer para que a ovelha possa comê-la novamente. A taxa de renascimento da grama é contada após a grama ter sido comida numa determinada localização. A reprodução dos animais é modelada por meio de um processo aleatório, desde que a quantidade de energia acumulada satisfaça a condição de reprodução. Com a reprodução é criado um novo animal (filho) de seus pais, dividindo a quantidade de energia do agente pai pela metade, ficando pai e filho com quantidades iguais de energia.

6.2 Modelagem Repast

A implementação do modelo no ambiente Repast segue o paradigma de “*framework* e biblioteca“. O *framework* é o conjunto de conceitos para projetar e descrever a modelagem baseada em agentes; a biblioteca é o conjunto de implementações de software para fornecer as ferramentas de simulação.

Para o desenvolvimento deste modelo selecionamos a linguagem Java e utilizamos o ambiente Repast Symphony (Repast S, <http://repast.sourceforge.net/>). A versão do Repast S, segundo seus desenvolvedores, tem como propósito ser acessível a usuários com pouca habilidade em programação. Assim, a implementação do modelo no Repast S (Tatara, et al., 2006) consiste de várias classes, que incluem, especificamente, classe de agente e dados carregados. Parte do modelo é criada em “*plain old Java objects*” (POJOs), usando o Eclipse como ambiente de desenvolvimento integrado – IDE (*Integrated Development Environment*).

Os componentes do modelo POJO podem representar qualquer coisa, mas são usados geralmente para representar os agentes do modelo. A Figura 49 apresenta a interface gráfica do Eclipse para descrever os elementos do modelo (classes, atributos). A Figura 50 apresenta fragmentos do código escrito na linguagem Java para a criação da classe *ContextCreator* para o modelo presa/predador, gerado paralelo à descrição gráfica. Nesta classe é criado o contexto, ou melhor, a grade de projeção para o contexto do modelo, representado no plano cartesiano. Em seguida são criados os agentes lobos e ovelhas e os parâmetros da simulação. Por fim, é definido o agente grama em cada localização da grade.

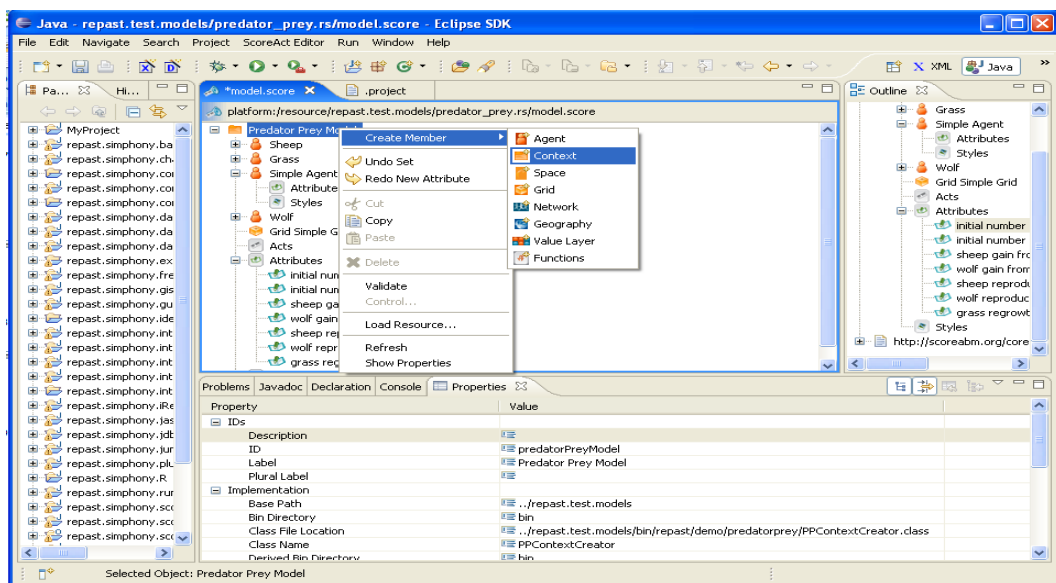


Figura 49 - Interface gráfica do Eclipse para definição dos elementos do modelo

As três classes de agentes, no modelo, estende a classe *SimpleAgent*, que contém os métodos comuns relacionados ao movimento e morte, por exemplo. A classe *SimpleAgent* também contém métodos específicos do Repast, como `@ScheduleMethod`. O movimento sobre a grade é realizado pelo método “*move*”, que recupera o objeto *Grade*, no contexto do agente,

obtendo suas coordenadas sobre a grade. O agente se move aleatoriamente sobre o ambiente, selecionando uma posição na grade entre os seus oito vizinhos adjacentes. O método “*die*” (morte do agente) ocorre quando a quantidade de energia decresce ao valor zero ou, no caso da ovelha, quando esta é comida pelo lobo. O processo de morte do agente se dá pela simples remoção do agente do contexto da simulação.

```
public class PPContextCreator implements DataLoader {
    public Context create(Object creatorID) {
        int xdim = 50;
        int ydim = 50;

        Context<SimpleAgent> context =
            Contexts.createContext(SimpleAgent.class, creatorID);
        Projections.createGrid("Simple Grid", context, new
            DefaultGridParameters<SimpleAgent>(
                new RandomGridAdder<SimpleAgent>(), true, xdim, ydim));
        Parameters p = RunEnvironment.getInstance().getParameters();
        int numWolves = (Integer) p.getValue("initial number of wolves");
        p = RunEnvironment.getInstance().getParameters();
        int numSheep = (Integer) p.getValue("initial number of sheep");
        for (int i = 0; i < numWolves; i++) {
            Wolf wolf = new Wolf();
            context.add(wolf);
        }
    }
}
```

Figura 50 – Fragmento da criação da classe ContextCreator no Repast S para o modelo presa/predador

Uma vez criadas as classes essenciais para o modelo, o usuário pode carregá-lo para o Repast S executar. O ambiente disponibiliza uma variedade de botões na barra de ferramentas para carregar o modelo: *setup* (acionar), *playing and pausing* (iniciar e pausar), *stepping* (passo a passo), *stopping* (parar) e *resetting* (reiniciar). Os componentes do modelo são mostrados numa estrutura de árvore, do lado esquerdo da interface, permitindo ao usuário criar e adicionar vários componentes ao modelo, Figura 51. Durante a execução, o usuário pode alterar alguns parâmetros do modelo sem alterar o código: número de agentes ou propriedades que são comuns às classes de agentes. A Figura 52 mostra a interface gráfica. É possível visualizar uma ou mais representações gráficas (gráficos de pontos ou interações entre os agentes) em uma mesma tela de visualização em 2 ou 3 dimensões.

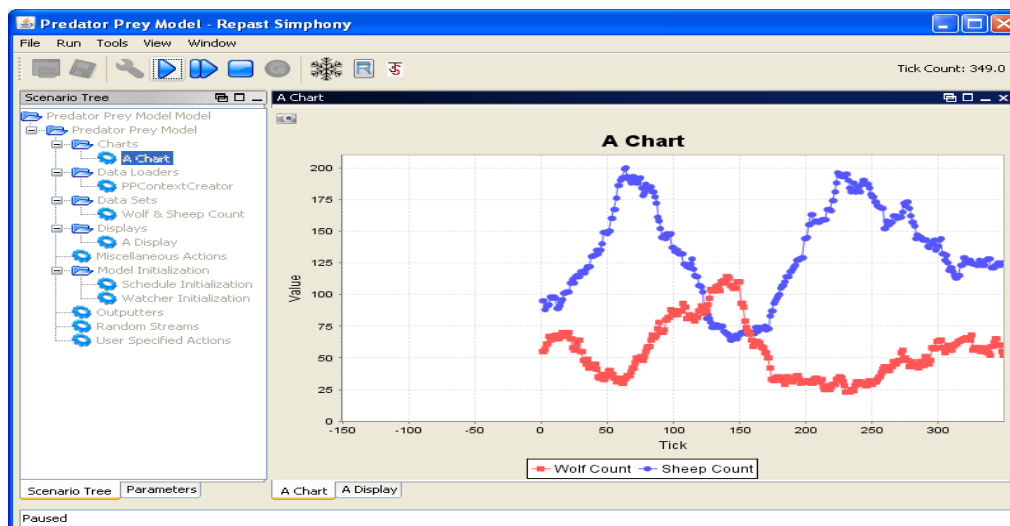


Figura 51 - Árvore de cenários e visualização gráfica

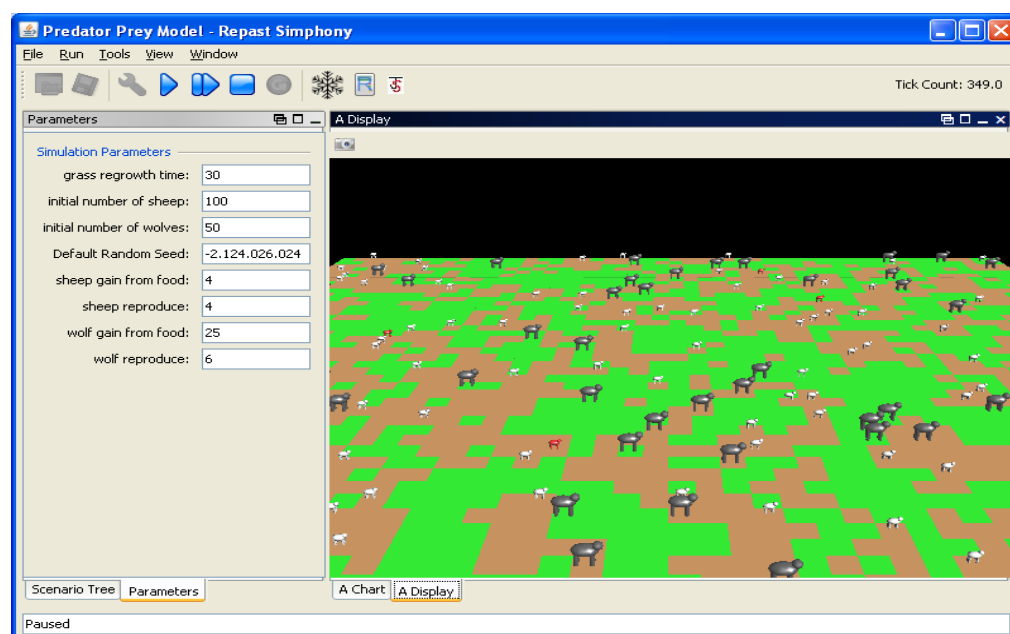


Figura 52 - Interface gráfica 3D e controle de parâmetros

6.3 Modelo NetLogo

O modelo desenvolvido no ambiente NetLogo segue o modelo (Wilensky, 1998). A linguagem utilizada no NetLogo é uma variação da linguagem Logo, sendo esta estruturada e procedural. Todas as ações podem ser definidas por procedimentos que são acionados mediante referência ao procedimento que as contém. O ambiente oferece vários recursos gráficos que auxiliam na programação. As variáveis são de três tipos: globais (*globals*), próprias dos agentes (*turtles-own* ou *patches-own*) e locais. As duas primeiras devem ser

declaradas nas primeiras linhas do código. É possível estabelecer formas diferentes para os agentes através da definição de raças (*breed*), que faz com que o modelo fique mais atrativo. As formas podem ser selecionadas na biblioteca de formas.

A definição do ambiente, grade onde ocorrem as interações, é dada pelos agentes estacionários (*patches*). Não é necessário, aqui, definir essa grade, pois ela já é componente do ambiente. Entretanto, o modelador pode redefinir suas dimensões através da interface gráfica, apenas alterando seus valores. A criação do ambiente fica definida pelo preenchimento dos espaços da grade usando a alteração de cor, através do comando *ask patches [set pcolor green]*, por exemplo. Observe que a solicitação foi realizada no procedimento de criação – *setup*, isto é, de domínio do agente *Observer*. Neste caso foi necessário solicitar uma ação por outro agente, o que exige o uso do comando *ask* “agente”.

Para definir a quantidade de agentes lobos e ovelhas, foi utilizada uma variável global: *initial-number-wolves* e *initial-number-sheep*, respectivamente (definida na interface do ambiente como *slider* – para controlar a quantidade de agentes sem alteração de código), conforme pode ser visto na Figura 53. A criação dos agentes foi realizada usando o comando *set-default-shape sheep "sheep"* para definir forma e *create-custom-sheep initial-number-sheep*, para a criação propriamente dita dos agentes. Os atributos dos agentes foram definidos neste mesmo bloco: cor, tamanho, posição no ambiente e definida a variável energia (*energy*), com a qual o agente inicia o modelo – definida randomicamente conforme código.

O acionamento do modelo é definido pelo procedimento de iniciar, denominado “*to GO*”, neste caso atribuído ao agente móvel *turtles*, configurado pelo botão de acionamento na interface. Este procedimento chama todos os demais procedimentos do modelo. Cada um desses procedimentos define ações específicas segundo padrões da linguagem. A Figura 54 é um recorte da linguagem de programação.

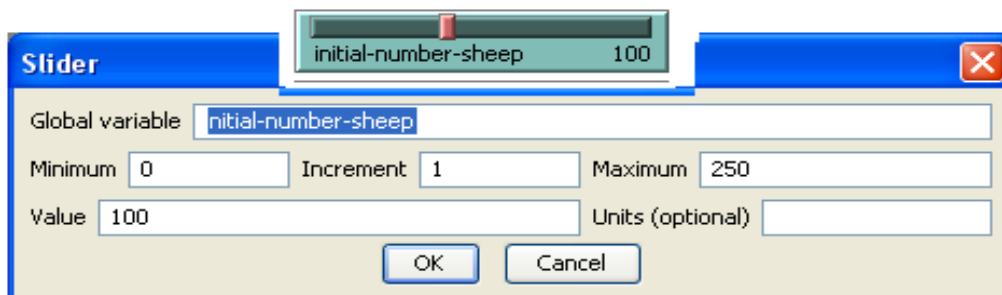


Figura 53 - Definição de variável tipo slider

O movimento dos agentes se dá pelo procedimento *to move*. O movimento é caracterizado avançando uma unidade para frente, após definida uma direção de forma aleatória numa variação de ângulo estipulada. O procedimento *to death* define a morte do agente, que ocorre caso a quantidade de energia do agente se reduza a zero. O agente ovelha e o agente grama podem morrer, caso sejam comidos. A morte do agente é realizada através da remoção do agente, para as ovelhas, e alteração de cor para a grama.

```

to go
  if not any? turtles [ stop ]
  ask sheep [ move
    if grass? [
      set energy energy - 1
      eat-grass
    ]
  ]
  if grass? [ ask patches [ grow-grass ] ]
  do-plot          ;; plot populations
  display-labels
  set ticks ticks + 1
end

```

Figura 54 – Recorte código ambiente NetLogo

A Figura 55 apresenta a interface de visualização do ambiente com todos os elementos acima destacados. Do lado esquerdo ficam os botões de controle da simulação e o gráfico. Do lado direito fica a animação do modelo em 2D. O ambiente possibilita também a visualização em 3D, basta acionar o botão na barra de ferramentas.

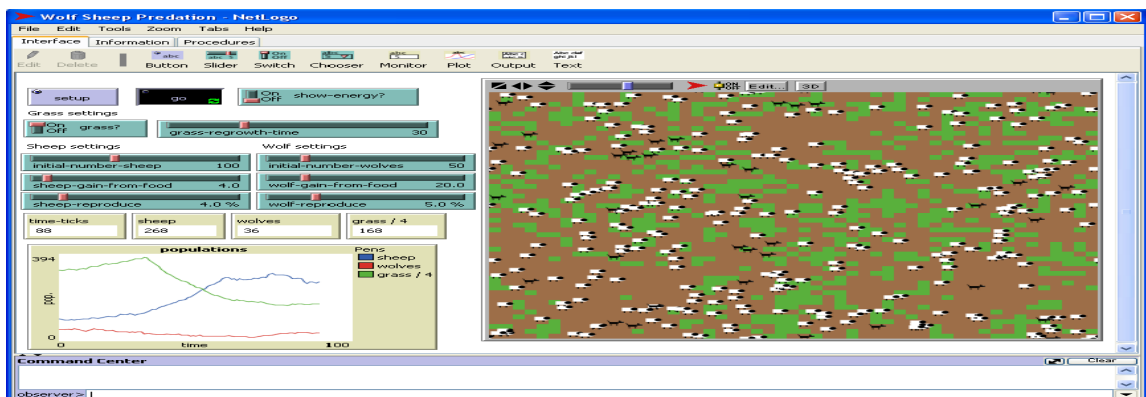


Figura 55 - interface gráfica NetLogo

6.4 Modelo NetPlay

A forma de modelar no NetPlay se assemelha às ações de um diretor de teatro ao montar uma peça. Para que isso ocorra efetivamente ele tem a sua disposição um conjunto de personagens e o cenário. Para montar a peça, ele seleciona os personagens e determina seu papel (o seu comportamento); monta o cenário e posiciona cada personagem segundo o script.

Após, estabelece as regras de interação entre os personagens e entre esses e o ambiente; assiste o desenvolvimento das interações e estuda o modelo simulado.

Para concretizar essa metáfora, o NetPlay adota o paradigma visual, ou seja, o modelador define os elementos do modelo de simulação por meio da seleção de objetos e atributos pré-definidos, preenchendo caixas de diálogo e compondo regras de comportamento e interação desses elementos através do preenchimento de relações entre eles. Cada objeto definido tem uma biblioteca de primitivas para compor os procedimentos que governam o comportamento e as interações dos agentes. Por se tratar de uma ferramenta para web, o NetPlay é um conjunto de páginas dinâmicas que possibilita ao usuário construir o modelo e exportar para o servidor de execução, além de possibilitar a visualização de modelos já criados pelo usuário em momentos anteriores.

No NetPlay os agentes podem ser definidos de três tipos: móveis, estacionários ou borda. Ao definir o tipo de agente, imediatamente são relacionados aos seus atributos (cor, forma, tamanho, posição e tipo de movimento, para os agentes móveis). Cada um desses atributos é definido selecionando elementos da biblioteca, conforme Figura 56. Ao selecionar o agente estacionário, é definido um grau de dispersão do mesmo pelo ambiente.

Para o caso de agente móvel ou estacionário, a distribuição dos agentes pelo ambiente pode ser determinada através de atribuição aleatória ou posição específica, definida por coordenadas cartesianas e selecionada na grade que representa o ambiente. A quantidade de agentes é definida utilizando uma variável tipo *slider* “numOvelhas”, que permite controlar a quantidade de agentes através da interface gráfica durante a execução do modelo. O movimento dos agentes é atribuído durante a sua criação. É permitido três tipos de movimento: passear, refletir e parado. O movimento passear simula um andar aleatório definido por uma quantidade de passos à frente e uma direção, mediante a variação de ângulo dado de 0 a 360. Esse procedimento gera um movimento aleatório, pois a cada ciclo esse processo é realizado. O movimento refletir faz com que o sentido e a direção do movimento do agente, quando esse colide com algum objeto ou outro agente, sejam redirecionados (refletido) num ângulo previamente selecionado pelo usuário. A seleção do movimento parado faz com que o agente se torne imóvel durante a execução do modelo. A Figura 56 mostra a criação de um agente móvel com seus atributos e comportamento individual, ou seja, a composição do personagem.



Figura 56 - Definição de agente Ovelha - NetPlay

A definição de variáveis obedece a padrões semelhantes às do agente, preenchimento de caixas de diálogo. São definidos três tipos de variáveis: lógica (recebe valor verdadeiro ou falso), *slider* (podem ser modificadas sem necessidade de alterar o código – dial) e valor (admite valores numéricos ou expressões), conforme Figura 57. As variáveis auxiliam na montagem do comportamento do personagem.



Figura 57 - Definição de variáveis (apresenta todas as variáveis criadas no modelo, operadores matemáticos e condições relacionadas à contagem dos agentes e suas características), basta o usuário selecionar quais delas usará para escrever a expressão.

As regras de comportamento dos agentes são definidas pela composição de regras lógicas *se condição ... então ação*. As **composições**, que descrevem o comportamento dos

agentes, são compostas em duas etapas selecionando os ícones que representam as condições e, posteriormente, os ícones que representam as ações decorrente dessa, conforme Figura 58.



Figura 58 - Composição para regras de comportamento

Da mesma forma dos ambientes anteriores, o lobo se alimenta da ovelha e esta se alimenta da grama, quando ambos estão sob a mesma posição. A Figura 59 representa a composição “comeGrama”. A composição é formada pela condição – ovelha percebe grama à distância zero, e as ações resultantes são:

- a) criar uma agente terra, para indicar a remoção do agente grama;
- b) incrementar a quantidade de energia à variável “energia”, própria da ovelha.

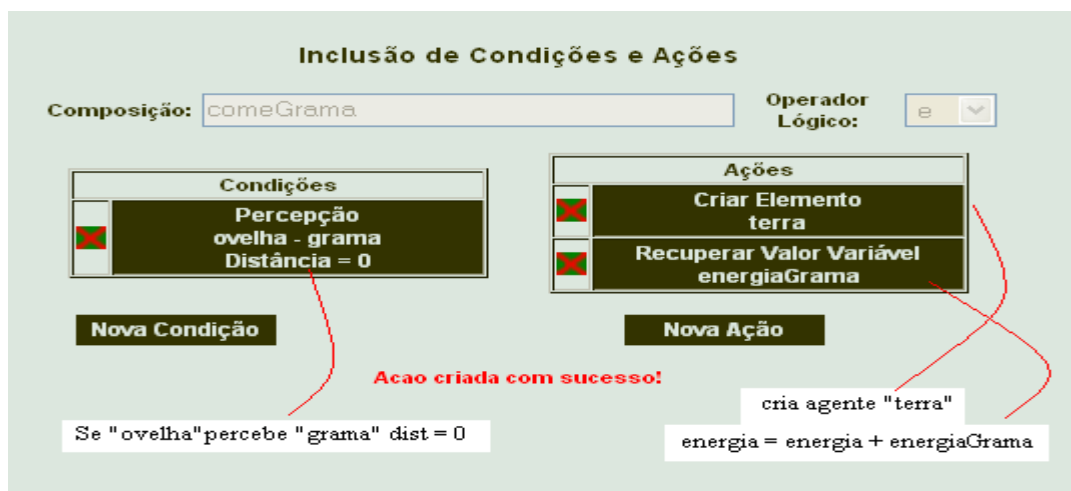


Figura 59 - composição “comeGrama” = alimentação ovelha

A morte do agente ovelha é simulada por meio de sua remoção, mediante satisfação da condição “ganhoEnergia < 0”. O agente ovelha também pode morrer, caso seja comido pelo lobo, conforme citado acima, porém o processo de remoção é o mesmo.

A ferramenta NetPlay não trabalha com código em formato de texto, todas as ações são realizadas semelhantes às citadas. Para a visualização dos modelos, o NetPlay ainda não tem uma estrutura própria com esse propósito. Desse modo, os conceitos do NetPlay são traduzidos pelo ambiente para um outro programa, usando bibliotecas e padrões escritos para o NetPlay. Assim, fazemos uso de um ambiente auxiliar, no caso o NetLogo (Wilensky, 1999-2007), para visualizar nossas ações. Feito isso, a visualização do modelo pode ser observada como um *applet* através no próprio browser, ou no ambiente NetLogo, veja Figura 60.

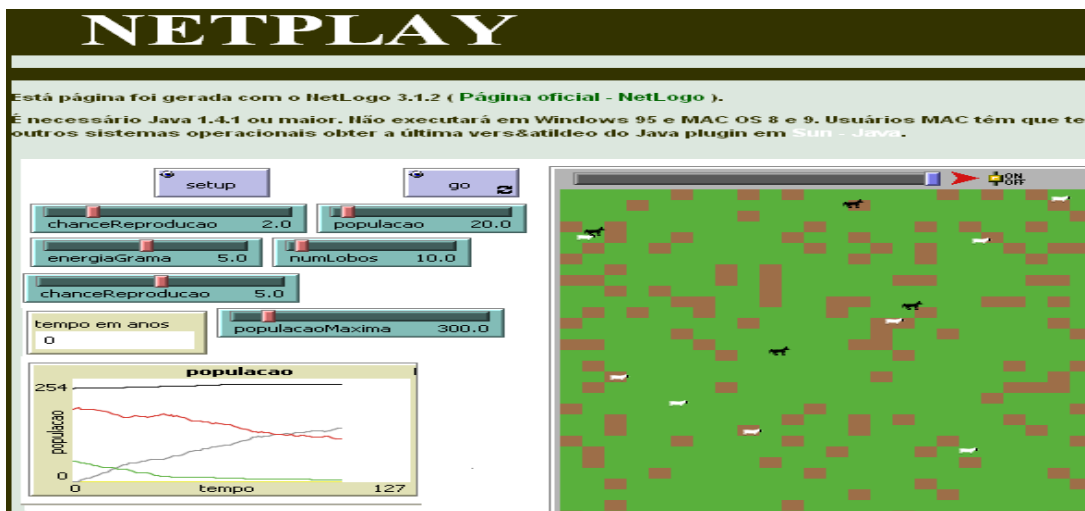


Figura 60 - Applet do modelo solicitado pelo NetPlay

6.5 Considerações sobre as ferramentas

As três implementações fazem com que o modelo tenha o mesmo grau de possibilidades. Entretanto, o que as torna realmente diferentes é a forma como os modelos são construídos e apresentados. O resultado dessa comparação mostra que os objetivos e filosofias dos desenvolvedores desses ambientes diferem, e essas diferenças são refletidas no próprio ambiente. Resumimos aqui nossa interpretação, baseada em nossa experiência com o desenvolvimento do modelo e o levantamento de requisitos para o desenvolvimento da ferramenta NetPlay.

Segundo os desenvolvedores, o Repast S foi desenvolvido para atender vários objetivos, mas claramente apóia o domínio das ciências sociais, incluindo ferramentas

específicas para esse domínio. O ambiente Repast S prioriza a implementação do modelo e, separadamente, fornece uma interface visual para observação e condução do experimento, o que não é uma ação trivial. A tentativa de fazer o Repast S mais acessível para usuários não experientes em programação ainda não foi atingida, pois toda construção do modelo é suportada pela programação orientada a objetos e linguagem Java.

O NetLogo reflete sua herança como uma ferramenta educacional com o propósito de ser fácil de usar e apresentar boa documentação e exemplo. O ambiente agrega num só local todo o processo de desenvolvimento do modelo através da escrita de código, visualização do comportamento e documentação sobre o modelo (alterando abas). Sua linguagem apresenta uma estrutura de alto nível e adota o modelo procedural. O ambiente apresenta várias facilidades para composição dos modelos. Apesar das facilidades, à medida que os modelos vão ficando mais sofisticados, é exigido maior conhecimento tanto na linguagem quanto no domínio do ambiente.

O NetPlay é uma ferramenta para construção de modelos de simulação multiagente para ser usado na web, cuja programação usa o paradigma visual e cuja metáfora para descrição do modelo é bem acessível. Essa forma de programar, acreditamos, possibilita a muitos usuários desenvolver seus próprios modelos, pois, nessa construção, a preocupação é direcionada para o modelo em si e não para a programação. Essa é uma iniciativa para remover a barreira da sintaxe de programação, disponibilizando uma programação fácil – programação visual.

Embutida em sua estrutura o NetPlay tem um tradutor, uma ferramenta prática que executa automaticamente, e de forma transparente para o usuário, a geração do código que, por sua vez possibilita que o modelo seja visualizado como um *applet* ou exportado para visualização em outro ambiente. O tradutor é o conjunto de bibliotecas que decodifica as características definidas na linguagem de programação visual utilizada no NetPlay e a transforma em uma linguagem de programação baseada em texto para que o modelo seja executado, pois o NetPlay não tem ainda uma interface de visualização.

A Tabela 5 apresenta alguns aspectos (conceitos e termos) utilizados por cada um dos ambientes avaliados, relacionados com as facilidades de modelagem para leigos em programação.

Tabela 5 - Avaliação entre os ambientes

Aspecto	Repast S	NetLogo	NetPlay
Definição agente	Baixo	Médio	Alto
Definição comportamento	Baixo	Médio	Alto
Definição ambiente	Baixo	Alto	Alto
Elementos de visualização	Alto	Alto	Alto
Ferramentas de análise e comparação	Não possui	Não possui	Possui

6.6 Características adicionais

Além do exposto, o NetPlay apresenta duas características que merecem destaque: ser uma ferramenta web e apresentar um dispositivo de análise gráfica. Em primeiro lugar, o NetPlay é uma ferramenta que está inserida no ambiente web AProSiMA, o que possibilita que os usuários geograficamente distribuídos possam, cooperativamente, participar da discussão e construção de modelos e analisar os modelos já desenvolvidos, além de contar com toda a estrutura de resolução cooperativa de problemas proposta pelo ambiente.

O fato de poder ser acessado, via web, de qualquer lugar, sem a necessidade de instalação de programa na máquina cliente, possibilitar a visualização dos modelos já criados pelo usuário ou outros criados em momentos anteriores, possibilita uma maior flexibilidade entre os usuários e reúso dos modelos que podem ser acessados e modificados. A visualização dos modelos é apresentada em formato de árvore de criação, Figura 61.

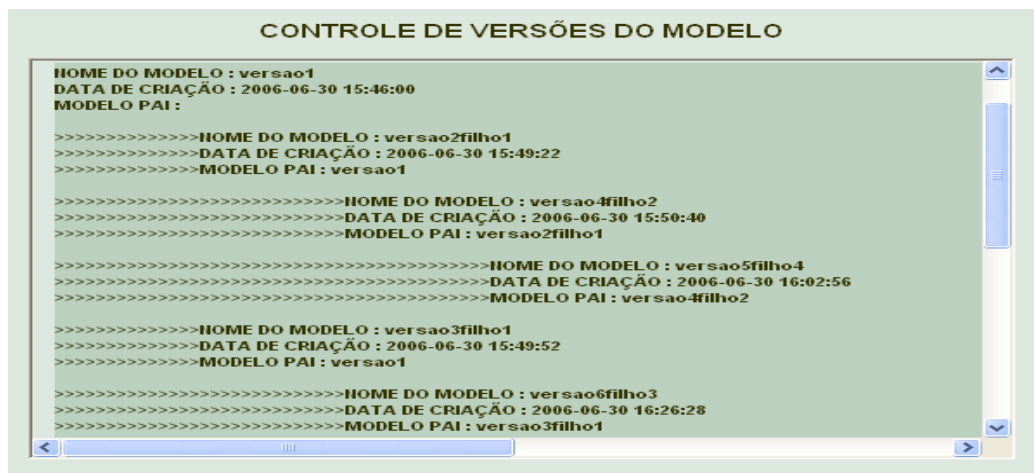


Figura 61 - Controle de versões

Em segundo lugar, foi incorporado à ferramenta NetPlay um dispositivo para análise gráfica. Com esse dispositivo é possível analisar detalhadamente todos os pontos plotados no gráfico que descrevem o comportamento do modelo. Pontos de máximos e mínimos, interseção de curvas e posição específica de cada valor plotado, são alguns dos elementos mostrados. Essa característica é importante, pois, sendo um modelo de simulação composto por muitos ciclos, é difícil acompanhar com precisão os dados plotados no gráfico. Além disso, é possível exportar esses dados, apresentados em tabelas (*nome_arquivo.csv*), para outras ferramentas, caso o usuário pretenda realizar uma análise pormenorizada ou estabelecer novas relações, Figura 62.

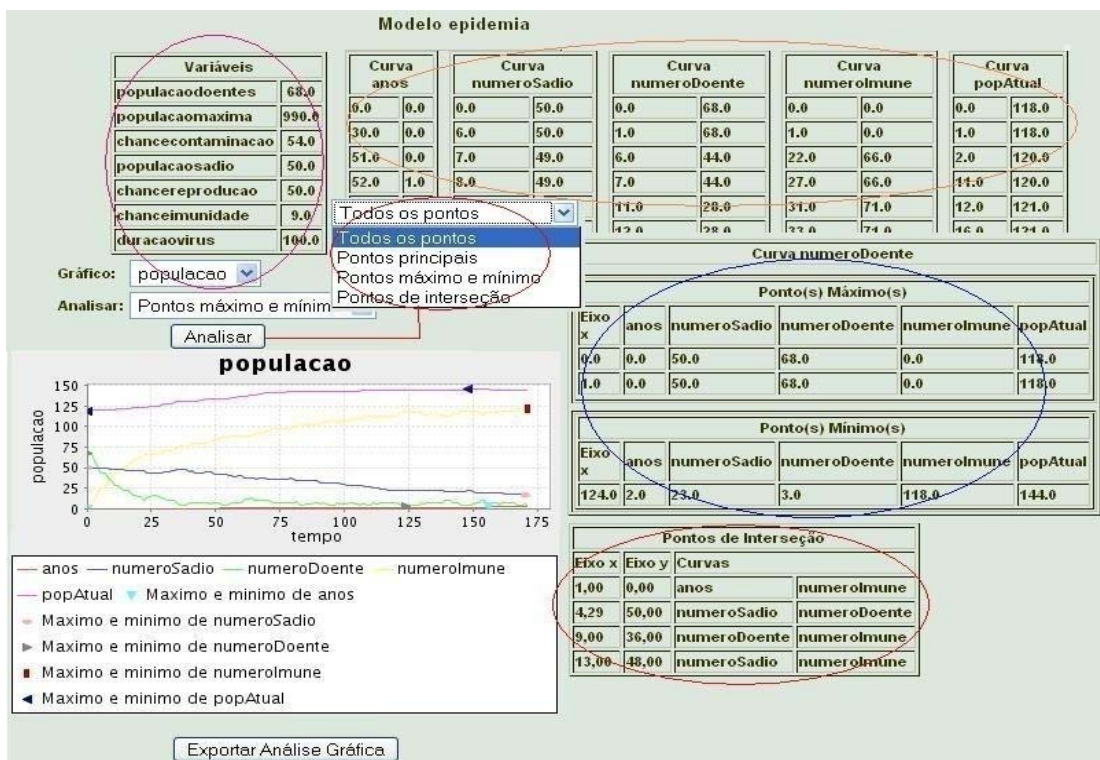


Figura 62 - Dispositivo de análise gráfica

Capítulo 7

7 Aplicação da ferramenta de simulação - NetPlay

Foi ofertada, no primeiro semestre de 2007, no Departamento de Informática da Universidade Federal do Espírito Santo – UFES, no curso de Computação (Ciência e Engenharia) e para o curso de Mestrado (Informática e Engenharia Elétrica), a disciplina optativa “Simulação baseada em sistemas multiagente – aplicações em educação“, para seus discentes, tendo três objetivos principais:

- i. testar a ferramenta para construção de modelos de simulação – NetPlay;
- ii. iniciar um processo de sensibilização para o estudo de sistemas descentralizados, mostrando que é possível e simples modelar situações nas quais os indivíduos têm comportamentos diferentes e complexos;
- iii. fazer um comparativo, com relação à forma de construção de modelos, entre as ferramentas NetLogo e NetPlay.

Participaram da disciplina 8 estudantes, sendo 6 do curso de graduação em ciência da computação, 1 do curso de engenharia da computação e 1 do curso de mestrado em Engenharia Elétrica, ênfase em automação. A disciplina teve duração de 45 horas, sendo um encontro semanal com duração de 3 horas, num total de 15 encontros.

A disciplina fundamentou-se na participação cooperativa. A metodologia adotada priorizou: leitura e discussão sobre sistemas multiagente e sistemas complexos; a exploração de modelos de simulação multiagente e posterior discussão sobre os dados observados; construção intensiva de novos modelos de simulação multiagente, utilizando as ferramentas disponíveis; apresentação dos modelos implementados.

Para iniciar a disciplina e provocar os estudantes com relação à abordagem de ensino e instigar seus conhecimentos, apresentamos e discutimos sobre um problema comumente abordado em vários níveis de ensino e observável no dia a dia. Este problema trata da relação num ecossistema entre presa e predadores.

7.1 Abordagem analítica x abordagem simulação multiagente

A fim de concretizar um exemplo e trazer para uma situação mais próxima das vivenciadas pela maioria dos estudantes nos cursos de graduação, apresentamos o modelo “presa x predador”, adaptado de (Wilensky, 1998), discutimos como o mesmo é abordado nos cursos de graduação e o que não é explorado nesses cursos. Este modelo foi desenvolvido nas duas ferramentas citadas.

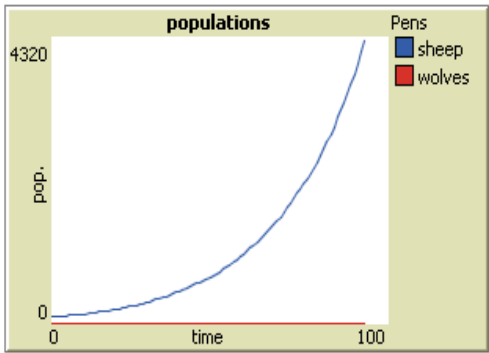
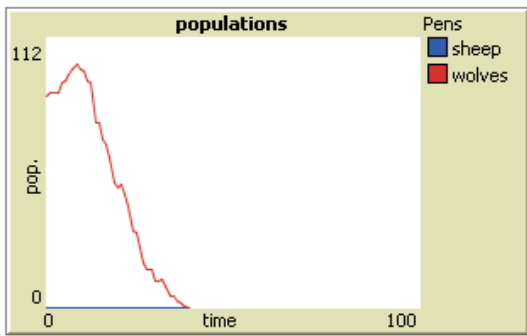
O problema explora a estabilidade presa/predador num ecossistema. Tais sistemas são chamados: a) **instáveis** - quando eles tendem a resultados que levam à extinção de uma ou mais espécies envolvidas; b) **estáveis** - quando tendem a se manter no decorrer do tempo, apesar das flutuações ocorridas nas populações. A dinâmica da interação entre as populações (presa e predador) tem sido tópico de interesse na comunidade de biólogos. Esse modelo procura mostrar as oscilações que ocorrem nas populações de herbívoros (presas) e carnívoros (predadores) em um sistema, cuja descrição é a seguinte: Suponha um modelo de presa / predador, onde duas espécies interagem: uma como presa e outra como predador.

Nos cursos de graduação, em geral, a abordagem segue a seguinte linha: A população das presas é dada por $x(t)$ e a população dos predadores é dada por $y(t)$. Nós estamos interessados no comportamento das duas populações. De modo simplificado, não estamos, neste modelo, considerando variáveis como ganho ou perda de energia. Assim, o sistema é representado no decorrer do tempo, conforme as funções:

$$\begin{cases} x'(t) = ax \\ y'(t) = -by \end{cases}$$

O sistema pode ser analisado de acordo com a presença dos elementos constituintes, conforme Tabela 6.

Tabela 6 - Comportamento do sistema “presa / predador”

Comportamento do sistema Presa / Predador	
Ausência de predadores	Ausência de presa
$x' = ax$ para algum $a > 0$	$y' = -by$ para algum $b > 0$
 <p>The graph shows the population of sheep (blue line) increasing exponentially over time from 0 to 100. The y-axis is labeled 'pop.' with a value of 4320. The x-axis is labeled 'time' with values 0 and 100. A legend indicates 'sheep' in blue and 'wolves' in red. The wolf population remains at 0.</p>	 <p>The graph shows the population of wolves (red line) decreasing exponentially over time from 112 to 0. The y-axis is labeled 'pop.' with a value of 112. The x-axis is labeled 'time' with values 0 and 100. A legend indicates 'sheep' in blue and 'wolves' in red. The sheep population remains at 0.</p>
Conseqüência: a população de presas cresce exponencialmente.	Conseqüência: a população dos predadores tende a extinção, ou seja, a eliminação total da espécie devido à ausência de alimento.

Claro que esse sistema pode ser facilmente resolvido por meio de equações diferenciais. Assim, o modelo pode ser considerado muito simples, pois é possível a integração dos dois lados das equações. Além disso, outros elementos não influenciam no sistema e, não envolve variáveis randômicas. Uma solução para o sistema é obtida por:

$$\begin{cases} \frac{dx}{dt} = ax \\ \frac{dy}{dt} = -by \end{cases} \Rightarrow \begin{cases} \frac{dx}{ax} = dt \\ \frac{dy}{-by} = dt \end{cases}$$

$$\begin{cases} \int \frac{dx}{ax} = \int dt \\ \int \frac{dy}{-by} = \int dt \end{cases} \Rightarrow \begin{cases} a \ln x = t \\ -b \ln y = t \end{cases} \Rightarrow \begin{cases} x = e^{at} \\ y = e^{-bt} \end{cases}$$

Evidentemente, quando o tamanho da população de presa e predador é comparado no decorrer de muitas gerações, tendemos a encontrar oscilações regulares nessas populações que

são as fases, ou seja, quando uma cresce a outra tende a declinar, e vice-versa. Muitos modelos matemáticos têm sido propostos para explicar essas oscilações.

Assim, as explorações científicas e educacionais para o sistema presa x predador, comumente usadas nos cursos de graduação tradicionais, são baseadas em conjunto de equações diferenciais, conhecidas como equações de Lotka-Volterra⁴⁵. Ou seja, as mudanças de densidade da população do predador (y) e na densidade de população de presa (x) podem ser descritas com as seguintes equações diferenciais:

$$\begin{cases} \frac{dx}{dt} = \alpha x - k_1 xy & (1) \\ \frac{dy}{dt} = k_2 y - \beta xy & (2) \end{cases}$$

sendo: y é o número de indivíduos de algum predador (exemplo lobo);

x é o número de indivíduos de alguma presa (exemplo ovelha);

t representa o crescimento das duas populações no tempo; e

α , β , k_1 e k_2 são parâmetros que representam a interação entre as duas espécies.

Cada equação descreve a razão pela qual uma dada variável (por exemplo, a densidade da população de presa) muda no decorrer do tempo. A equação de Lotka-Volterra aqui apresentada descreve as mudanças de densidade na população de presas (x) e na população de predadores (y). Nessa equação α representa a taxa de nascimento das presas, β representa a taxa de mortalidade dos predadores e k_1 e k_2 são constantes. Observe que o tamanho da população e a densidade da população são proporcionais umas as outras.

Analisando a equação (1), o termo (αx) representa a taxa de nascimento das presas multiplicado pela densidade de população de presas. Deste fato decorre que o crescimento da densidade é devido ao nascimento de novas presas. No termo ($k_1 x y$), a frequência da interação entre a população de presa e a população de predador é determinada em decorrência

⁴⁵ O modelo de presa-predador de Lotka-Volterra é um modelo de importância histórica na modelagem matemática de sistemas ecológicos. Esse modelo foi desenvolvido independentemente por Lotka (1925) e Volterra (1926). A lei de Lotka-Volterra propõe que “quando as populações estão envolvidas em feedback negativo com outras espécies, ou até mesmo com componentes de seu ambiente”, é provável que se observe uma dinâmica oscilatória (cíclica). Um exemplo de feedback negativo é quando um aumento na população de uma espécie de presa gera um aumento na população de seus predadores (através do aumento da reprodução), e isso, por sua vez, retroalimenta para reduzir a população de presas através do aumento da mortalidade devido à predação (Berryman, 2002, 2003, citado por (Haemig)).

do decrescimento da densidade de presa e devido ao consumo das presas pelos predadores. A alteração na densidade da população de presas é, então, calculada pela subtração entre o total de mortes e o total de nascimento de presas.

A equação (2) pode ser analisada em termos similares à primeira, embora nesta equação o nascimento de predadores depende das interações entre presas e predadores, enquanto que a morte de predadores não.

É importante notar que a equação (1), que descreve a população de presas, contém a variável y , que descreve a densidade da população de predador, e vice-versa. A equação assim formulada descreve a relação entre os dois elementos, indicando como a densidade (tamanho) de uma população depende da densidade (tamanho) da outra. Especificamente, o aumento no número de predadores faz com que aumente a natalidade de predadores; o aumento na população de predadores causa a morte das presas. Esse comportamento caracteriza uma flutuação cíclica entre as populações de presa e predadores, Figura 63.

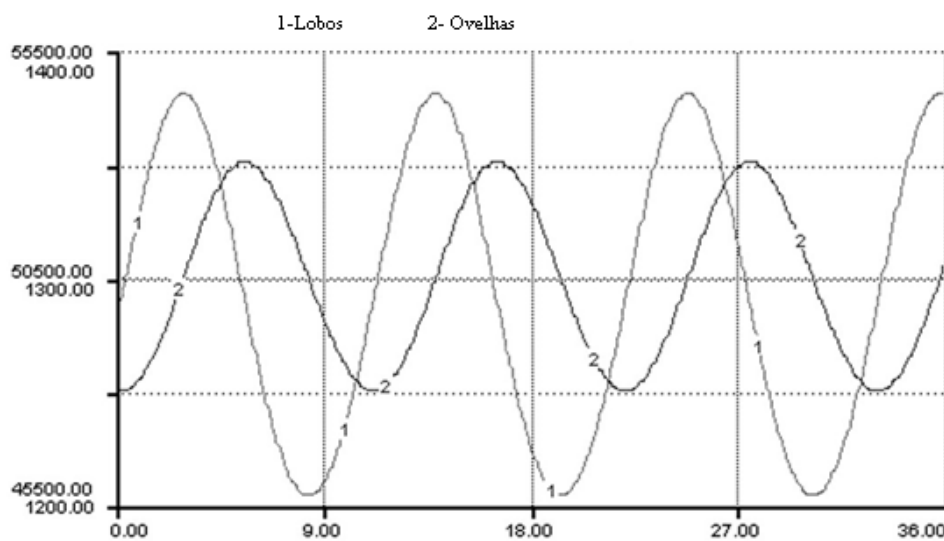


Figura 63 - Resultado da equação de Lotka - Volterra para o modelo presa x predador com populações iniciais de 50.000 e 1.250, respectivamente

Nesta disciplina, “Simulação baseada em sistemas multiagente – aplicações em educação“, não era interesse aprofundar no estudo dessas equações, pretendia apenas notar que a abordagem clássica descreve as flutuações cíclicas entre as duas populações, especificando propriedades tais como: natalidade, frequência de interação e densidade.

A nova abordagem que pretendemos dar para o problema implica ver o fenômeno por outro ângulo. Ao invés de descrever relações entre as populações, estamos interessados em especificar o comportamento dos *indivíduos*. A relação relevante aqui é: que tipo de ação deve ter um predador individual ou uma presa individual para que as populações desses indivíduos exibam características de oscilação? Outra forma de pensar sobre as ações dos indivíduos seria considerar as *regras de comportamento* que cada indivíduo pode seguir para que, em nível de população, alcance o resultado. Nesse levantamento observamos comportamentos como: o movimento dos animais; perda de energia ao se movimentar no ambiente à procura de alimento; ganho de energia ao se alimentar; condições de reprodução;

Aparentemente, o modelo construído baseado nessas regras assemelha-se ao proposto pelas equações de Lotka-Volterra, mas ao analisar o modelo executado, considerando o comportamento individual e o fenômeno emergente dessas interações, percebe-se que as oscilações entre as populações são observadas de forma diferenciada, pois o comportamento individual interfere no todo, veja Figura 64.

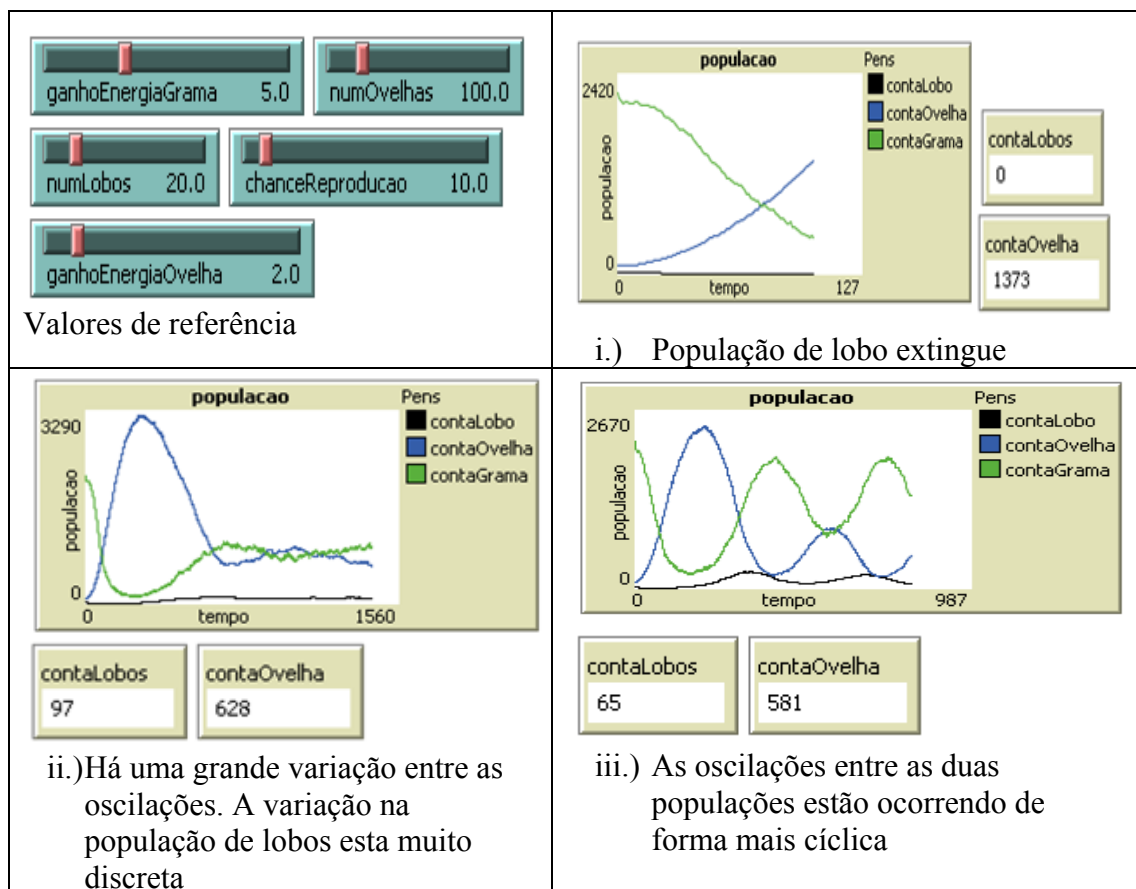


Figura 64 – Três comportamentos diferentes para o mesmo conjunto de regras no modelo presa/predador

Assim, vários parâmetros podem ser levantados neste problema. O modo simplificado não retrata a realidade do sistema. Podemos indagar:

- quanto ao crescimento dos herbívoros (presas);
- quanto ao crescimento dos carnívoros (predadores);
- se ao diminuir o número de presas este número afeta o sistema;

O importante é perceber que, mesmo que a fonte seja constante, o sistema oscila devido à defasagem entre produção e consumo. Entretanto, se olharmos o sistema com os demais elementos como:

- taxa de reprodução;
- morte dos elementos;
- ganho de energia com alimentação;
- razão de reposição, entre outros.

Veremos que o sistema passa a apresentar muitas variáveis independentes que influenciam umas as outras. E o comportamento individual dos elementos da população interfere no todo. Os valores já não são mais previsíveis. Qualquer alteração numa das variáveis, interfere no comportamento do modelo de modo não proporcional. Não é possível prever exatamente o momento em que a população de presas terá o dobro da população de predadores, por exemplo. Detalhe desse comportamento pode ser acompanhado por meio da Figura 64.

Este é um resultado clássico visto através da lente do fenômeno emergente. Através do modelo de simulação, o modelador controla o comportamento dos indivíduos em micro-nível e observa o resultado que emerge na população em macro-nível. Através da experimentação dinâmica dessas conexões, um melhor entendimento do sistema presa/predador pode ser alcançado. Essa abordagem, geralmente, não é levantada em sala de aulas tradicionais, ficando uma riqueza de conteúdo inexplorada.

Através da simulação, podemos explorar o modelo, atribuindo diferentes valores às suas variáveis livres e, assim, avaliar seus efeitos sobre o ecossistema. Podemos deixar o sistema funcionando e anotar o que acontece no decorrer do tempo. Podemos também alterar as variáveis com diferentes valores e observar as diferenças no comportamento do sistema.

Podemos, ainda, observar que uma das alterações imediatas desse sistema é o fato de o crescimento da população de animais se dar em detrimento do alimento e, também, uma queda nessa população se operar diante da escassez do alimento, o que leva a uma oscilação dessas populações. Com alterações das variáveis, é possível perceber que, para determinados valores, é possível estabelecer um equilíbrio entre as populações e, para outros valores, há um desaparecimento imediato de uma das populações. Da mesma forma, podemos fazer diferentes variações, com diferentes valores para as variáveis e observar as diferenças no comportamento do sistema.

É importante salientar que a observação direta do mundo real, parece ser o mais adequado, por nos colocar em contato direto com os fenômenos. Entretanto, algumas vezes, isso não é possível pela dificuldade de repetição de certos fenômenos, pelos custos associados, ou pelo risco que pode representar para seus observadores e/ou observados. Diante desses motivos é importante dispor de programas de computador, que apresentem comportamentos que nos permitam exercitar o pensamento analítico e também compreender sistemas específicos.

Utilizando as ferramentas apresentadas, cada tipo de indivíduo que nos interessa observar é representado no sistema. Os indivíduos são agrupados em famílias e cada família possui suas próprias características. Os indivíduos interagem em um determinado ambiente, segundo uma lógica própria dessas famílias, provocando com isso diferentes comportamentos para o sistema como um todo. O comportamento dos indivíduos e, conseqüentemente, do sistema como um todo, é dependente de certos parâmetros (variáveis) que podem ser controladas pelos observadores. Para perceber como os indivíduos e, conseqüentemente, como o sistema se comporta, dispomos de alguns instrumentos de apoio à percepção: monitores, *plots*, caixa de textos, análise gráfica, além do controle de variáveis.

Com a criação do modelo podemos explorar várias versões desse sistema, desde uma mais simples, com apenas uma variável de entrada até outras mais complexas. Em cada um desses sistemas é possível observar, registrar dados e tirar conclusões sobre o sistema observado. Todos os elementos citados devem estar à disposição do ensino, desde o nível mais básico, e nas diversas disciplinas para serem trabalhadas interdisciplinarmente, motivo para disponibilizar uma ferramenta que não dependa de aprendizado de linguagem de programação.

Entretanto, é importante salientar que as duas abordagens são importantes e devem ser aplicadas mediante os objetivos desejáveis. A abordagem da simulação para analisar um modelo é oposta à abordagem analítica, cujo método de análise do sistema é puramente teórico. Numa abordagem analítica ressalta-se a segurança dos dados. Numa abordagem de simulação ressalta-se a flexibilidade e comodidade da operação. A Tabela 7 apresenta uma comparação entre as duas abordagens.

Tabela 7 - Comparação entre abordagens: analítica e de simulação

Comparação entre abordagem analítica e abordagem de simulação		
Elementos destaque	Abordagem analítica	Abordagem de simulação
Segurança	X	
Flexibilidade		X
Comodidade		X
Exige especialidade	X	
Não exige conhecimento em programação		X
Exige conhecimento matemático	X	

7.2 Elementos observados durante o curso

Com relação à aprendizagem de sistemas complexos pudemos perceber o interesse dos alunos com relação à nova forma de ver os fatos, que até então passavam despercebidos. Por exemplo, ao analisar o modelo “presa x predador”, usando a abordagem analítica (equações diferenciais) e a de simulação multiagente, para explorar o comportamento resultante das interações entre os agentes, os estudantes puderam perceber mais nitidamente que as oscilações dependiam, criticamente, dos parâmetros e como essas influenciavam o modelo. Apesar do conhecimento matemático de que dispunham, perceberam que a exploração com modelos de simulação multiagente é nitidamente diferente e muito mais acessível, enquanto a exploração com equações é muito mais abstrata, pois trabalha com símbolos.

Elemento que merece destaque nessa experiência foi o fato de despertar nos alunos as diferentes formas de análise de comportamento do modelo: na forma analítica as relações se dão em termos do comportamento da população no decorrer do tempo; enquanto que, na forma de simulação multiagente, as ações e interações estão mais relacionadas ao

comportamento individual dos agentes, que é mais próxima da nossa forma de pensar. Outros aspectos que despertaram o interesse dos estudantes:

- a) a possibilidade e facilidade de analisar os modelos com a introdução de muitos fatores aleatórios e várias inter-relações entre as variáveis ou agentes;
- b) que pequenas alterações nos parâmetros se refletiam em grandes alterações no comportamento do modelo;
- c) perceberam que as alterações do modelo foram realizadas pela aleatoriedade do comportamento dos agentes e não pela imposição do que o agente deveria fazer;
- d) estudar sistemas multiagente em situações definidas por relacionamento por meio de regras simples e não como protocolos e programas elaborados.

7.3 Atividades durante a disciplina

Outros modelos foram discutidos em sala de aula (movimento de veículos em um posto de combustível; autômato celular – “jogo da vida”; comportamento de vírus numa população) para despertar nos alunos o interesse pelas características dos sistemas complexos.

Após debate, os alunos observaram que, nos modelos trabalhados, algumas características eram proeminentes:

- existência de muitos elementos (agentes);
- controle descentralizado;
- auto-organizado;
- interação entre os muitos elementos;
- comportamento definido por regras simples;
- imprevisibilidade;
- autopoiese⁴⁶;

Além dos modelos previamente selecionados, cada discente desenvolveu um modelo de seu interesse, representativo de um fenômeno complexo, utilizando inicialmente a ferramenta NetPlay e, posteriormente, a ferramenta NetLogo, para um modelo mais completo.

⁴⁶ Autopoiese é capacidade de um sistema de se organizar de tal forma que o único produto seja ele mesmo

Foram apresentados 3 modelos relacionados ao comportamento de insetos (abelhas e cupins). Um deles pretendia observar o comportamento interno de uma colméia, por meio das atividades desenvolvidas em cada ciclo de vida das abelhas. Os outros dois pretendiam observar o comportamento externo dos insetos (abelhas e cupins) com relação à coleta de alimentos e defesa da colônia, respectivamente. Dois modelos tinham objetivos de avaliar o desempenho dos agentes em situações aleatórias: um simulando uma equipe de troca de pneus num *pitstop* de fórmula um; e o outro avaliando a performance de jogador e goleiro em situação de marcação de pênalti. Para avaliar o comportamento de animais durante a jornada de migração, foi modelada a “marcha dos pingüins”, baseada no filme de mesmo título. Em caráter organizacional foi modelado o movimento de um armazém: recebendo, depositando, organizando e despachando cargas. Para avaliar a influência dos semáforos no tráfego de veículos, foi criado um modelo que mapeava a interferência desse em condições de fluxo de trânsito. Detalhes de cada modelo podem ser acompanhados pela Tabela 15, Apêndice B.

Durante todo o processo de construção dos modelos, os alunos foram instigados a examinar os seus propósitos, tanto pelo mediador como pelos demais estudantes num trabalho cooperativo. Todo o material produzido neste curso está disponível numa página de construção cooperativa <http://simulacaomultiagente.pbwiki.com/>.

7.4 Considerações sobre a disciplina

Exemplos como os tratados pelos alunos fazem parte do nosso dia-a-dia e, geralmente, não são estudados na sua totalidade. Nosso sistema de ensino, de modo geral, não prioriza as propriedades dos sistemas complexos, como por exemplo, a “emergência”, cujas características são:

- capacidade de se organizar sem um princípio organizacional centralizado;
- o estudo de partes separadas do sistema não leva ao entendimento do todo;
- o sistema não será definível pela soma de suas partes, mas por uma propriedade que emerge deste seu funcionamento;
- os sistemas complexos são identificados pelo que eles fazem;

Além da propriedade de emergência, é possível verificar também as seguintes propriedades:

- imprevisibilidade – um sistema complexo é fundamentalmente não determinístico. É impossível, antecipadamente, prever o comportamento do sistema, mesmo conhecendo as funções que o compõem;
- dinamismo e capacidade de se auto-organizar e se reestruturar – um sistema complexo tem a capacidade de se auto organizar. É difícil, se não impossível, estudar suas propriedades através da decomposição de suas partes. As interações permanentes com o ambiente e sua capacidade de auto-organização possibilitam a sua reestruturação.
- um sistema complexo possui propriedades semelhantes aos sistemas distribuídos (em sentido conexionista), isto é, algumas de suas funções não podem precisamente ser localizadas. Além disso, os relacionamentos existentes entre os elementos de um sistema complexos são de curto e longo alcance, não-lineares e possuem *feedback* (positivo e negativo) simultaneamente.
- o estudo sobre fenômenos complexos envolve uma exploração interdisciplinar e não um estudo num campo bem definido.

Foi observado pelos alunos que as atividades em um modelo de simulação consistem nos eventos, que são ativados em determinados pontos do tempo e, desse modo, afetam o estado geral do sistema. Esses pontos são ativados de modo randômico, de maneira que nenhuma entrada fora do sistema é requerida. De modo geral, numa abordagem de simulação os componentes do programa consistem de entidades que combinam eventos relacionados dentro de um processo.

Durante todo o processo de construção dos modelos, os alunos foram instigados sobre os seus propósitos, tanto pelo mediador como pelos demais alunos num trabalho cooperativo. Com as discussões e os modelos desenvolvidos pelos alunos constatamos a importância do estudo de sistemas descentralizados, através de modelos de simulação, para que os alunos possam pensar, não apenas no processo do fenômeno, mas também explorar conceitos e idéias que até então lhes eram despercebidas.

7.4.1 Avaliação da ferramenta

Durante o desenvolvimento da disciplina, os alunos iniciaram a construção dos modelos utilizando a ferramenta NetPlay. Esta utilização contribuiu bastante para ajustes na própria implementação da ferramenta. Entretanto, uma versão corrigida da ferramenta

NetPlay só foi possível disponibilizar após o término da disciplina. Assim, para avaliar o alcance da ferramenta não podíamos contar com a disponibilidade de todos os alunos que cursaram a disciplinas, pois os mesmos haviam concluído o curso e já estavam engajados em outros projetos. Mesmo assim, contamos com a contribuição de três alunos que gentilmente atenderam a nossa solicitação para construir um novo modelo de simulação e responder ao questionário proposto sobre a utilização da ferramenta NetPlay.

A proposta do questionário era avaliar cada uma das etapas da construção do modelo: criação de agentes, com seus atributos e comportamentos; criação de variáveis; composição de regras de comportamento e interações; acompanhamento dos dados gerados pelo modelo; e ferramentas auxiliares para interpretação de dados. As respostas obtidas dos alunos estão mapeadas na Tabela 8. A primeira coluna da tabela representa a formulação da pergunta; a coluna do meio representa a opção mais votada e a terceira coluna as considerações feitas pelos alunos com relação à ferramenta NetPlay.

Quanto ao uso da ferramenta NetPlay foi consenso que a mesma tem uma interface muito amigável, sendo adequada, principalmente, para pessoas que não possuem domínio em programação. Apresenta grande facilidade de representação do modelo através da composição dos elementos e possibilidades de relacionamentos. O fato de ser uma ferramenta para web possibilita maior mobilidade do usuário e o compartilhamento de modelos em tempo real.

Tabela 8 - Questionário de avaliação da ferramenta NetPlay

Descrição	Consideração	Observações
Com relação à criação de agentes e seus atributos		
Quanto à criação de agentes e características (nome, cor, tamanho, forma, quantidade)	Fácil utilização e atende aos requisitos necessários para compor as características dos agentes que compõem o modelo	
Quanto aos tipos de agentes disponíveis (móveis, estacionários, borda)	Conseguem mapear as necessidades do modelo	Um tipo de agente importante é o “plano de fundo”, que seriam aqueles não mapeados pelos agentes inseridos.
Quanto ao movimento descrito pelos agentes (passear, refletir e parado)	Conseguem mapear as necessidades do modelo	
Quanto ao posicionamento dos agentes no ambiente	Conseguem mapear as necessidades do modelo	O posicionamento dos agentes (móveis e estacionários) quando distribuídos de forma aleatória, se dá de maneira fácil. Contudo, o posicionamento dos agentes em posições específicas pode se tornar um pouco enfadonho, uma vez que o posicionamento é realizado individualmente para cada agente.
Com relação à criação de variáveis		
Quanto à criação de variáveis	Fácil e atende aos requisitos necessários para compor as características do modelo	Apesar de não ser tão intuitivo, em pouco tempo se aprende como deve ser feito
Com relação aos tipos de variáveis (lógica, valor, slider)	Conseguem mapear as necessidades do modelo	
Quanto à edição de variável tipo valor (editar expressão) se dá de modo:	Fácil e atende aos requisitos necessários para descrever as variáveis do modelo	A presença das possíveis funções a serem usadas facilita a criação

Com relação às regras de comportamento		
Quanto à definição de regras de comportamento		
Com relação às composições (condições e ações) que definem o comportamento do modelo	Fácil e atende aos requisitos necessários para compor as características do modelo	A criação é bem intuitiva o que facilita a interação. Entretanto, há limitações do tipo: SE condição FAÇA ação. Poderia existir: SE condição FAÇA ação1 SENÃO ação2
Com relação às possibilidades de condições (percepção, posicionamento, cor do agente, valor de variável, comparação de variáveis) para descrever o comportamento do modelo:	Conseguem mapear as necessidades do modelo, mas poderia ser melhorado	Algumas composições são limitadas, por exemplo, percepção de outro agente sempre é fixo na igualdade, isto, se agente1 está a uma <u>distância igual a xxx</u> do agente1 faça xxx. Poderia ter uma composição comparando-se a: >, >=, <, <= etc.
Com relação às possibilidades de ações (criar elemento, destruir elemento, alterar cor, alterar forma, incrementar/decrementar variável, recuperar valor de variável, alterar posição, alterar movimento, parar movimento) para descrever o comportamento do modelo:	Conseguem mapear as necessidades do modelo, mas deixa a desejar em alguns pontos.	Possui algumas limitações como a criação/destruição de 1 só elemento por vez
Com relação às ferramentas de acompanhamento dos dados gerados pelo modelo (monitor, gráfico e texto)		
Com relação aos monitores	Conseguem mapear as necessidades do modelo	
Com relação ao gráfico	Conseguem mapear as necessidades do modelo	
Com relação ao texto	Conseguem mapear as necessidades do modelo	

Com relação à análise de dados		
Com relação à ferramenta de análise dos dados (analisar gráfico)	Fácil utilização. Bastante didática e intuitiva para construção de modelos de simulação por leigos em programação	

Para complementar as opiniões dos alunos com relação à facilidade de construção de modelos de simulação, no âmbito de sistemas complexos, construímos os modelos: Têrmites e Epidemia.

O modelo Têrmites foi adaptado de (Wilensky, 1998), que se inspirou no comportamento de insetos sociais (têrmites) carregando e empilhando madeira. Este modelo é composto de dois tipos de agentes: têrmites e blocos de madeira. As têrmites são agentes moveis e os blocos de madeira agentes estacionários randomicamente distribuídos no ambiente. No modelo, cada têrmite segue um conjunto de regras simples que podem ser resumidas em: vagar aleatoriamente pelo ambiente; localizar madeira, procurar por pilha de madeira, depositar madeira num local vazio da pilha. Agindo desta forma as pilhas de madeiras são formadas, mas não determinadas em locais específicos.

Algumas vezes as têrmites recolhem material de uma pilha já existente e levam para uma outra, o que faz a estratégia parecer contra produtiva. Se as pilhas fossem protegidas, o modelo poderia se resumir num conjunto de muitas pequenas pilhas ao invés de uma grande pilha. No decorrer do processo, o que se pode observar é uma grande pilha arredondada. Esse projeto é um exemplo de estratégia descentralizada, pois não há nenhuma têrmite comandando as ações e nem um local pré-definido para que as pilhas sejam alocadas. Durante todo o fenômeno cada têrmite segue o conjunto de regras estabelecidas, mas a colônia como um todo realiza uma tarefa bastante sofisticada.

O modelo Epidemia simula a transmissão e perpetuação de um vírus numa população humana. Vírus é um micro-organismo que pode infectar outros organismos biológicos. Como exemplos de doenças que atingem seres humanos e são causadas por vírus podemos citar AIDS, caxumba, sarampo, hepatite, ebola, dengue, poliomielite e febre amarela. Segundo (Yorke, et al., 1979) há uma variedade de fatores que podem influenciar a sobrevivência e transmissão de um vírus em uma população, entre eles estão: tamanho da população; proporção de pessoas imunes a essa doença; renovação da população (taxa de natalidade); forma de transmissão; tempo de contágio dessa doença (semanas que uma pessoa consegue sobreviver sem tratamento). Esse modelo é uma adaptação de (Wilensky, 1998) que tem o mesmo propósito.

O modelo retrata que através do comportamento dos agentes, definido por regras simples, pode desencadear um fenômeno não previsível. Este modelo é um exemplo de sistema complexo, pois o fenômeno é determinado não por alguma autoridade centralizada, mas por interações locais dos componentes descentralizados. O fenômeno epidemia surgiu (em parte) devido às relações entre as pessoas e às variáveis do modelo.

Ressaltamos que um dos principais propósitos para a construção da ferramenta NetPlay foi possibilitar que até pessoas com pouco conhecimento em programação pudessem construir seus próprios modelos de simulação multiagente. Desse modo, buscamos sempre privilegiar a facilidade na construção dos modelos. Com relação ao uso da ferramenta NetPlay para a construção de modelos percebemos que:

- todas as vezes que criamos um agente móvel, criamos uma raça para ele. Em modelos com um único agente móvel isso talvez não fosse necessário. No modelo Epidemia, por exemplo, os agentes doentes, imunes e saudáveis poderiam ser de uma única raça, diferenciados pela cor e por algumas variáveis locais;
- algumas variáveis poderiam ser locais a todos os agentes móveis. No modelo Epidemia, por exemplo, todos os agentes têm uma variável que contabiliza a idade deles, pois não é possível ter variáveis locais a todos os agentes, só a uma raça. Neste caso, tivemos que criar duas variáveis, *idadeSaudaveis* e *idadeDoentes*, para guardar a idade dos agentes móveis. No modelo Térmites preferimos usar outro artifício para caracterizar o agente “desocupado”, criando a raça desocupado e alterando sua cor.
- como não dispomos ainda de uma interface para visualização, tivemos que limitar nosso “mundo” (ambiente de execução) a uma grade de coordenada máxima de 25 unidades nos dois eixos. Para modelos que necessitam de uma visualização com um menor número de pixels (como o caso do modelo térmites), deixou a desejar em relação aos demais ambientes disponíveis;
- não inserimos nas composições a possibilidade de controle de repetição como: ENQUANTO (*while*) e de seleção como: SE ... ENTÃO ... SENÃO (*if...else*). Preferimos optar em fazer composições somente com o comando SE (*if*) para tornar o modelo mais simples de ser construído.

Capítulo 8

8 Conclusões

A sociedade do Conhecimento está, hoje, assentada sobre dois pilares: a cooperação e o reuso. A complexidade dos problemas que hoje se apresentam torna-os intratáveis por indivíduos ou equipes diminutas. Por outro lado, é cada vez mais óbvio que no mundo das soluções digitais, tudo deve e pode ser reutilizado, principalmente o conhecimento para resolução de problemas.

Avanços nos meios tecnológicos têm melhorado as técnicas para ajudar as organizações a reunir, analisar, entender, armazenar, utilizar e reutilizar as informações em seus processos de resolução de problemas. Os ambientes de trabalho cooperativo suportados por uma rede de computadores têm evoluído no sentido de possibilitar a interação dos usuários e o gerenciamento de suas ações. É, assim, uma evolução lógica estendê-los para executar e resolver problema.

Atestado que grupos colaboradores apresentam um alcance mais amplo nas resoluções de problemas e que o conhecimento compartilhado, em grupo mediado por redes de computadores, tem atingido uma amplitude maior na busca por soluções, esse tem se tornado o tema central dos sistemas de apoio à resolução de problemas.

Assim, entendemos que nosso projeto endereça diretamente a esses problemas, à medida que otimizou a busca de parceiros e a recuperação de informações para elucidação do problema, sistematiza a discussão, oferece elementos que viabilizam o registro e a recuperação de informação, possibilita a construção de modelos de simulação e a interação entre os usuários.

Como levantado no início deste trabalho, uma das dificuldades encontradas com relação ao processo de resolução cooperativa de problemas relaciona-se, primeiramente, a formação do grupo de resolução. Percebemos a necessidade de melhorar as ferramentas de apoio à interação das pessoas para um processo de resolução cooperativa de problemas, no sentido de:

- possibilitar mecanismos para a localização de especialistas, geograficamente distribuídos e desconhecidos, e meio de contato;

- compartilhar um local comum que não dependesse de instalação de software e também não dependesse de um software específico e comum a todos os participantes;
- disponibilizar um mecanismo que auxiliasse a busca de material informativo a fim de elucidar o problema;
- possibilitar aos desenvolvedores condições de discutir idéias e formalizar as estratégias de solução por meio de um esquema;

Nesse sentido, o ambiente apresentado nesta tese vem ao encontro do suprimento dessas necessidades. Com a ferramenta QSabe2, melhoramos as possibilidades de localização de especialistas e usuários, geograficamente distribuídos, com interesses afins. Aprimoramos o modo de localização do especialista respondedor através das características descritas pelos próprios usuários no questionário de identificação e pelas ações realizadas no próprio ambiente (construção de modelos e esquemas, documentos selecionados como relevantes a sua área de interesse, respostas às consultas, etc.). Essas ações darão uma maior referência ao agente da ferramenta (QSabe) para selecionar o respondedor mais apropriado (contexto). As ações realizadas no ambiente alimentarão o perfil de cada usuário, obtidos através das descrições das mesmas.

A ferramenta Esquemizador possibilita aos usuários discutir as soluções para os problemas propostos e registrar as estratégias desenvolvidas. Esse esquema pode direcionar a construção do modelo de simulação ou a reutilização das estratégias para problemas similares. Associada à ferramenta de Gerenciamento de Grupo, para a formação do grupo e agendamento dos componentes do mesmo que participarão das discussões, a ferramenta Esquemizador tende a tornar mais dinâmica a busca por soluções.

A ferramenta de Gerenciamento de Grupo possibilita aos usuários do sistema mapear o perfil de cada componente do ambiente e, assim, detectar quais deles têm as características mais apropriadas para a formação do grupo que discutirá um determinado assunto. Com isso, a composição dos grupos vai além da escolha dos componentes por simples afinidade.

Com a utilização da ferramenta Gerenciador, tanto o grupo quanto os usuários individuais poderão recolher material para a elucidação do problema, seja no próprio ambiente ou através de busca na internet (por meio dos mecanismos disponíveis de recuperação de informação), de modo mais relacionado ao seu interesse. Essa ferramenta propicia um resultado de busca mais direcionado ao interesse do usuário, isso devido à possibilidade de similaridade entre os documentos encontrados proporcionada pela

ferramenta. Outro elemento que contribui para localizar o material de interesse e o pesquisador mais apropriado é o armazenamento do material selecionado no banco de dados do ambiente.

Utilizando a ferramenta de modelagem e simulação é possível transferir as estratégias discutidas para a solução do problema para o modelo de simulação. Essa ação permite aos interessados no problema analisar os dados, as relações e as conseqüências advindas das estratégias propostas para a solução do problema.

Um elemento que merece destaque no desenvolvimento desta tese é o fato de todas as ferramentas terem sido desenvolvidas para uso web, o que possibilita a maior interação entre os usuários e uma maior troca de conhecimento sobre os problemas em tela.

Com relação aos problemas relacionados a sistemas complexos, percebemos o quanto é importante a sua divulgação e conscientização, tanto no meio acadêmico quanto para a sociedade de modo geral, uma vez que idéias sobre descentralização e auto-organização estão se espalhando pela sociedade e sendo tomadas como modelo para representar os problemas em várias áreas do conhecimento.

Apesar do crescimento do uso de modelos descentralizados, muitas pessoas ainda são resistentes a tais idéias, possivelmente pelo fato da falta de incorporação das mesmas como objeto de ensino e discussão com estratégia de solução de problemas.

Para muitos usuários, a disseminação de tais conceitos permanece um entrave, devido à ausência de treinamento e habilidade para utilizar as ferramentas existentes de construção de modelos de simulação. Assim, a criação de uma nova linguagem de programação (linguagem visual e baseada em eventos) para construção de modelos de simulação, por meio da ferramenta NetPlay, possibilitará aos usuários construir seus próprios modelos de simulação. Com essa iniciativa, cada usuário (modelador) poderá construir e tratar seus próprios modelos e problemas direcionados aos seus interesses.

Nossa proposta de tornar a construção de modelos mais voltada para o fenômeno em si possibilitará aos usuários, com pouca habilidade em programação de computadores, construir seus próprios modelos. Quando o usuário capturar o comportamento do modelo real e procurar estratégias para representá-lo num modelo computacional, estará desenvolvendo um trabalho de síntese. As respostas obtidas através da simulação possibilitará que ele vá refinando o modelo até que este capture o fenômeno em estudo e fique o mais

próximo possível do modelo real. Assim, na construção do modelo a preocupação do usuário deve ser esta: capturar o comportamento do fenômeno, e não se preocupar em escrever isso numa linguagem de programação.

Entendemos que, superada a barreira da utilização de ambientes de simulação, para construção de modelos de sistemas complexos, os usuários se sentirão mais à vontade em representar seus modelos e discuti-los com outros. Essa ação desencadeará uma reflexão sobre ações que até então não eram observadas. Com isso acreditamos conseguir sensibilizar um maior número de pessoas sobre a importância do estudo de sistemas complexos. Esses, por sua vez, serão novos adeptos do uso da modelagem de simulação multiagente para a construção de sistemas complexos, em que as ações se tornam cíclicas.

Assim, com este trabalho, acreditamos ter conseguido atingir os seguintes pontos:

- disponibilizar um ambiente para resolução cooperativa de problemas que não depende de instalação de software específico e seja disponível na web;
- mecanismos que possibilitem a interação entre pesquisadores desconhecidos; facilidade na construção de modelos de simulação multiagente;
- mecanismos que possibilitem a elucidação dos problemas, através de material disponível sobre o assunto;
- direcionamento das questões a especialistas mais relacionados ao problema;
- espaço para discussão e formalização das idéias geradas;
- gerenciamento dos grupos formados no ambiente e, evidentemente, mecanismos de comunicação entre os integrantes do ambiente.

8.1 Com relação à sensibilização para o estudo de sistemas complexos

Para alguns alunos o estudo de sistemas complexos será objeto de interesse e fascínio, que o motivará a aprender mais sobre o assunto. Entretanto, para a maioria dos alunos (e elementos da sociedade), o ensino de sistemas complexos não deve ser encarado apenas como uma forma prática, aplicada para resolver questões dos problemas cotidianos. Os conceitos de sistemas complexos e as metodologias, como a modelagem baseada em agentes, cobrem um vasto conjunto de fenômenos naturais e sociais que vai desde fenômenos meteorológicos a congestionamento de trânsito, oferecendo uma gama de modelos para que os educadores possam ajudar os estudantes a aprender sobre importantes conceitos e competências. Assim,

entendemos que a simulação pode ser direcionada para conteúdos tanto na área de ciências sociais quanto na de ciências naturais, incluindo biologia, física, matemática, química, ciência da computação, economia, sociologia, psicologia social.

Além disso, a vantagem de um ambiente web para apoiar a resolução cooperativa de problemas que disponibiliza uma ferramenta para construção de modelos de simulação inclui várias possibilidades:

- manter um banco de problemas que sirva de base para novos modelos
- disponibilizar modelos que servirão de introdução a uma nova geração de estudantes de sistemas complexos
- contatar com pessoas que desenvolveram modelos de problemas similares

Com esta tese, conseguimos disponibilizar um ambiente de software com características web que atende ao processo de resolução de problemas, desde a coleta de material para elucidação do mesmo, a formação do grupo de discussão e o local partilhado para formalização das estratégias de solução, até a modelagem de simulação do problema e posterior análise dos dados. Assim, entendemos que o ambiente apresentado nesta tese atinge os objetivos propostos no seu início e que será bastante útil a vários segmentos educacionais.

8.2 Frutos desta tese

Durante o desenvolvimento desta tese foram publicados os artigos:

- Azevedo, L. L. ; Ramiro, T. B. ; Monteiro, V. P. C. ; Teixeira, S. ; Oliveira, E.; *Recuperação de Informação através do processo de aproximações sucessivas*. In: XXI Congresso Brasileiro de Biblioteconomia, Documentação e Ciência da Informação, 2005, Curitiba, PR. Anais do XXI CBBD, 2005.
- Ramiro, T. B. ; Oliveira, E. ; Azevedo, L. L. ; Monteiro, V. C. P. ; Teixeira, S.; *Atribuindo títulos de assuntos na categorização automática de documentos*. In: XXI Congresso Brasileiro de Biblioteconomia, Documentação e Ciência da Informação, 2005, Curitiba, PR. Anais do XXI CBBD, 2005. 16.
- Azevedo L.L., Menezes, C.S., Pantarolo, E., Magdalena, B.C, *Ambientes Computacionais Orientados a Agentes para apoio à Aprendizagem Baseada em Simulação*; SBIE 2005.

- Azevedo, L. L.; Menezes, C. S.; Pantarolo, E.; *Tutorial: Exploração do ambiente NetLogo*. - SBIE 2005.
- Azevedo, L. L.; Andrade, A.; Velasco, D.; Barbiero, L.; Cury, D.; Menezes, C. S.; *NetPlay: um ambiente para simulação multiagente*, Renote. Vols. 4, Nº 2, Dezembro, 2006
- Azevedo L.L., Menezes, C.S.; *AProSiMA - Um ambiente na web para resolução cooperativa de problemas baseado em simulação multiagente*, Renote, V.4 Nº 2, Dezembro, 2006
- Azevedo L.L., Menezes, C.S.; *NetPlay: uma ferramenta para construção de modelos de simulação baseado em multiagente*, SBIE 2007

Desenvolvemos com alunos do curso de Ciências da Computação da Universidade Federal do Espírito Santo, co-orientações em trabalhos de conclusão de curso (projetos finais – monografias):

- ALEXANDRE LUIZ DE SOUZA ANDRADE (2006/1): NetPlay: um ambiente de resolução cooperativa de problemas baseado em simulação multiagente
- DIEGO CAMPAGNARO VELASCO (2006/2): Descrevendo interações no ambiente NetPlay
- LÚCIO MARCOS BARBIERO (2006/2): NetPlay - tradução para NetLogo, visualização da execução e análise de dados implícitos de modelos
- FERNANDO FIOROT MASSARIOL (2006/2): Gerenciador – Uma ferramenta de gerenciamento e recuperação de informação no âmbito do ambiente AProSiMA

Trabalhos em andamento, referentes ao período letivo de 2007/2:

- ISAU LAUERS JR.: Ferramenta Esquematizador
- RICARDO BRIOSCHI: Ferramenta para formação e controle de Grupos
- FELLYPE SUCHI RAMOS: Ferramenta QSabe2

Das implementações resultantes dessas monografias resultaram o ambiente AProSiMA, que está disponível para uso em: [http:// 200.137.66.20/aprosima](http://200.137.66.20/aprosima).

8.3 Trabalhos futuros

De acordo com o nosso conceito sobre problema: *um problema não termina em si, ele é o ponto de partida para novas investigações e descobertas*, concluímos este trabalho e levantamos novas investigações e trabalhos futuros, diretamente ligado a esse, no sentido de:

- ampliar as possibilidades de recuperação de informação utilizando, por exemplo, mecanismos que levem em consideração a semântica dos textos;
- estender a possibilidade de busca por especialista usando para tal: grupos e projetos de pesquisa, periódicos, anais, jornais, publicações, livros, em sistemas do tipo Lattes, etc.;
- otimizar o mecanismo de recuperação de informação com relação ao processo de agrupamento de documentos similares;
- desenvolver uma interface própria para a ferramenta NetPlay de modo a torná-lo independente de outros softwares e que seja nos mesmos moldes da composição dos modelos – fácil utilização;
- desenvolver um mecanismo que possibilite a passagem da linguagem gráfica utilizada na ferramenta Esquemizador, para a linguagem visual utilizada pela ferramenta NetPlay;
- acrescentar novas funcionalidades à ferramenta NetPlay de modo a possibilitar uma maior disponibilidade de ações e condições para compor as regras de comportamento dos agentes, como por exemplo: novos tipos de movimento, utilização do mouse de modo interativo, acréscimo ou retirada de agentes no decorrer da simulação;
- criar um mecanismo de validação das regras de comportamento descritas nas expressões da ferramenta NetPlay;
- desenvolver um mecanismo que possibilite capturar as posições específicas dos agentes móveis e permita direcioná-lo para outra posição, de modo independente
- desenvolver um mecanismo que possibilite capturar os elementos da vizinhança de um agente (todos, quatro (+), direita, esquerda, acima, abaixo);

Do ponto de vista educacional, trabalhar junto aos professores do ensino fundamental e médio das escolas de Barra do Garças e junto aos alunos do curso de licenciatura em Matemática e licenciatura em Biologia do campus do Médio Araguaia – UFMT, na conscientização da importância do ensino de sistemas complexos através do desenvolvimento e exploração de modelos. Neste trabalho será utilizado o ambiente AProSiMA, proposto nesta

tese, como forma de avaliação contínua do mesmo e levantamento de requisitos para sua atualização.

Outro aspecto que pode ser vislumbrado com esta pesquisa é a possibilidade de a ferramenta de simulação estabelecer a comunicação com elementos externos, permitindo que a atuação do usuário seja mais interativa. Esse elemento possibilitaria ao usuário interferir no modelo durante a sua execução. Entretanto, mais importante é que ele possa ir alterando as regras de comportamento dos agentes através da aprendizagem de novas regras. Essa possibilidade seria variações do que muitos pesquisadores denominam de “simulação participativa” (Participatory Simulations) (Colella, et al., 2002), (Borovoy, et al., 1996), (Eisenberg, et al., 1996).

Além disso, esta pesquisa poderá abrir novos caminhos no sentido da criação de ambientes que possam tratar de problemas em outras áreas do conhecimento e disponibilizando novas ferramentas de modelagem.

Outro aspecto que pode ser vislumbrado é a criação de ontologias de problemas para a classificação e identificação de problemas, bem como a indicação de soluções. A possibilidade da procura por problemas e sua resolução seria muito mais dinâmica. Com o suporte de ontologias, seria facilitada a criação de agentes inteligentes para atuar como auxiliar ao usuário nos processos de resolução de problemas, seja coletando material e disponibilizando para o usuário, seja na seleção de casos similares. A criação de agentes inteligentes também poderá ser direcionada para auxiliar na estruturação do problema e de sua solução.

Referências bibliográficas

Abrahamson D. and Wilensky U. ProbLab goes to school: Design, teaching, and learning of probability with multi-agent interactive computer models [Conference] // Proceedings of the Fourth Conference of the European Society for Research in Mathematics Education. - San Feliu de Guixols - Spain : <http://ccl.northwestern.edu/papers/>, 2005.

Alvares L. O. and Sishman J. S. Introdução aos sistemas multiagentes [Online] // Programa de Pós-Graduação em Computação (material de apoio) .- UFRGS, 1997. - http://www.inf.ufrgs.br/~alvares/SMA/Apostila_SMA.ps..

Axelrod R. The complexity of cooperation: agent-based models of competition and collaboration [Book]. - Princeton, NJ : Princeton Univ. Press, 1996.

Axelrod Robert Resources for agent-based modeling [Book Section] // The complexity of cooperation: Agent-based models of competition and collaboration. - Princeton : Princeton University Press, 1997.

Azevedo L.L., Menezes, C.S., Pantarolo, E., Magdalena, B.C, Ambientes Computacionais Orientados a Agentes para apoio à Aprendizagem Baseada em Simulação [Article] // SBIE2005. - nov. 2005.

Azevedo Livia Lopes [et al.] NetPlay: um ambiente para simulação multiagente [Journal] // Renote. - Porto Alegre : UFRGS, 2006. - Dezembro : Vols. 4, N° 2.

Azevedo Livia Lopes AProSiMA - um ambiente para resolução cooperativa de problemas baseado em simulação multiagente [Online]. - 2006. - http://www.inf.ufes.br/~liviaufmt/qualificacao/versao_final_qualificacao.pdf.

Azevedo Livia Lopes, Menezes Crediné Silva and Pantarolo Edilson Tutorial: Exploração do ambiente NetLogo. - Juiz de Fora - MG : [s.n.], 2005.

Baeza-Yates R. and Ribeiro-neto B. Modern Information Retrieval [Book]. - USA : Addison Wesley, 1999.

Barfourosh A. A. [et al.] Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition [Article] // Information Retrieval on the WWW and Active Logic. - [s.l.] : <http://citeseer.ist.psu.edu/barfourosh02information.html> , 2002.

Bar-Yam Y. Dynamics of complex systems [Book]. - Reading, MA : Addison-Wesley., 1997.

Bar-Yam Y. Making things work: solving complex problems in a complex world [Book]. - USA : NECSI-Knowledge Press, 2004.

Bianchi R. and Reali-Costa A. H. Uso do ambiente SIMULA para Simulação do Domínio de Resgate RoboCup [Online] // Laboratório de Técnicas Inteligentes Departamento de Engenharia da Computação . - Universidade de São Paulo, 2000. - <http://www.lti.pcs.usp.br/~rbianchi/publications/mappel2000.pdf>.

Blikstein P. and Wilensky U. Less is More: Agent-Based Simulation as a Powerful Learning Tool in Materials Science [Conference] // Proceedings of the IV International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005). - Utrecht - Holland : <http://ccl.northwestern.edu/papers/>, 2005-a.

Blikstein P., Abrahamson D. and Wilensky U. NetLogo: Where We Are, Where We're Going [Conference] // Proceedings of Interaction Design & Children, Boulder, . - Boulder - Colorado : <http://ccl.northwestern.edu/papers/> , 2005.

Bonabeau E. Agent-based modelling: methods and techniques for simulating human systems [Article] // Proceedings of the National Academy of Science. - 2002. - Vol. 99(3).

Bordini R. H. and Moreira A. F. JAI - fundamentos de sistemas multiagente [Article] // JAI - Fundamentos de Sistemas Multiagentes. In: Carlos Eduardo Ferreira. (Org.). As Tecnologias da informação e a questão social. Anais do XXI Congresso da SBC. - Fortaleza : Anais do XXI congresso da SBC, 2001.

Borovoy R. [et al.] Real Meets Virtual: Blending Real World Artifacts with Computational Media [Article] // CHI'96, ACM Press. - Vancouver, April 1996.

Borovoy R. [et al.] Things That Blink: Computationally Augmented Name Tags [Article] // IBM Systems Journal. - 1996. - pp. 35 (3), 488-495..

Botelho W. T. Utilização do ambiente simula em sistemas a eventos discretos [Online] // Instituto Militar de Engenharia – IME, Rio de Janeiro. - 2003. - http://www.dep.ufmg.br/professores/miranda/EPD030/relatorio_Control.pdf .

Boyd D. M., Ellison, N. B. Social network sites: Definition, history, and scholarship [Article] // Journal of Computer-Mediated Communication. - 2007. - 13(1), article 11. - <http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html>.

Bramson Aaron L About the Science of Complexity [Online] // Center for the Study of Complex Systems. - University Michigan, 2005. - www.cscs.umich.edu.

Brasil Secretaria de Educação Fundamental Parâmetros curriculares nacionais: introdução aos parâmetros curriculares nacionais [Book]. - Brasília : Secretaria de Educação Fundamental: MEC/SEF, 1997.

Brin S. and Page L. The anatomy of a Large-Scale hypertextual web search engine // Computer Networks and ISDN System. - 1998. - Vol. 30.

Chang S. Visual languages: A tutorial and survey [Article] // IEEE software. - 1987. - Vol. 4 (1).

Colella V. S., Klopfer E. and Resnick M. Adventures in modeling: exploring complex, dynamic system with StarLogo [Book]. - NY : Teachers college Press, 2001.

Colella V., Borovoy R. and Resnick M. Participatory Simulations: Using Computational Objects to Learn about Dynamic Systems [Online] // MIT Media Lab. - 2002. - <http://llk.media.mit.edu/papers/part-sims/>.

Collier N., Howe T. and North M. Onward and Upward: The Transition to Repast 2.0 [Article] // Electronic Proceedings. - Pittsburgh- USA : [s.n.], 2003.

Collins R. J. and Jefferson D. R. Ant Farm: Towards Simulated Evolution [Book Section] // Artificial Life II / book auth. Langton C.. - [s.l.] : Addison-Wesley, 1991. - Vol. <http://citeseer.ist.psu.edu/collins91antfarm.html> .

Deek F. P. An integrated environment for problem solving and program development // Ph.D. Dissertation (Unpublished). - New Jersey : New Jersey Institute of Technology, 1997.

Deek F., McHugh J. and Turoff M. Problem Solving and Cognitive Foundations for Program Development: An Integrated Model [Article] // submitted for review to the Journal of Cognitive Science. - 2000. - <http://cblis.utc.sk/cblis-cd-old/2003/>.

Delors J., I. Al Mufti I. Amagi, R. Carneiro, F. Chung, and B. Geremek W. Gorham, A. Kornhauser, M. Manley, Q. M. Padrón, M. A. Savané, K. Singh, R. Stavenhagen, W. S. Myong, N. Zhou Educação: um tesouro a descobrir. Relatório para a UNESCO da Comissão Internacional sobre Educação para o século XXI [Book]. - [s.l.] : UNESCO, MEC, 2006. - Vol. 10 ed..

diSessa A. A. Artificial worlds and real experience. In Artificial intelligence and Education [Book Section] // Learning Environments and Tutoring Systems / book auth. Yazdani R. W. Lawler and M.. - Norwood, NJ : Eds. Ablex Publishing Corp., 1987.

Drogoul Alexis and Ferber Jacques Multi-Agent Simulation as a Tool for Modeling Societies: Application to Social Differentiation in Ant Colonies [Online]. - 1992. - maio 2006. - <http://www-poleia.lip6.fr/~drogoul/papers/Drogoul.Maamaw92.pdf> .

Eisenberg M. [et al.] Real Meets Virtual: Blending Real World Artifacts with Computational Media [Article] // CHI'96, ACM Press. - Vancouver, April 1996 1996.

Eisenberg M., Mackay, W., Druin, A., Lehman, S., & Resnick, M. Real Meets Virtual: Blending Real World Artifacts with Computational Media [Article] // CHI'96, ACM Press. - Vancouver, April 1996.

Ellis C. A., Gibbs S. J. and Rein G. L. Groupware: some issues and experiences [Article] // Communications of the ACM. - 1991. - Vol. 34 n° 1.

Ellison N., Steinfield, C., Lampe, C. The benefits of Facebook "friends": Exploring the relationship between college student's use of online social network and social capital [Article] // Journal of Computer-Mediated Communication. - Indiana : [s.n.], 2007. - 12(3). - <http://jcmc.indiana.edu/vol12/issue4/ellison.html> : Vol. article 1..

Epstein J. M. and Astell R. L. Growing artificial societies: social science from the bottom up [Book]. - Cambridge, MA : MIT Press, 1996.

Ferreira W. G. [et al.] Ambiente didático na internet de dimensionamento de estruturas metálicas [Online]. - 2000. - fev 2005. - http://www.univap.br/iasee/anais/trabalhos/artigoNEXEM_IASEE_2.pdf.

Finnegan P. and O'Mahony L. Group Problem Solving and Decision Making: an Investigation of the Process and the Supporting Technology [Article] // Journal of Information Technology. - 1996. - Num. 3., - Vols. Vol. 11, pp. 211-221.

Flores-Mendez Roberto A. Towards a standardization of multi-agent system framework [Article] // Crossroads, vol. 5, n° 4, ACM Press. - 1999. - pp. 18-24.

Frozza R. SIMULA - Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos [Online]. - Universidade Federal do Rio Grande do Sul, 1998. - <http://simula.sourceforge.net/>.

Garton L., Haythornthwaite, C., Wellman, B. Studying Online Social Networks [Article] // JCMC - Journal of Computer-Mediated Communication. - [s.l.] : Indiana University, 1997. - 3(1).

Gell-Mann Murray O quark e o jaguar: aventuras no simples e no complexo, Trad. Alexandre Tort [Book]. - Rio de Janeiro : Rocco, 1996.

Glinert E. P., Kopache M. E. and McIntyre D. W. Exploring the general-purpose visual alternative [Article] // Journal of visual languages an computing. - 1990. - march.

Gospodnetic O. and Hatcher E. Lucene in action [Book]. - [s.l.] : Manning Publications Co., 2005.

Gospodnetić O., Hatcher, E. Lucene in Action [Book]. - 2004.

GroupSystem Group Intelligence Guide [Online]. - 2006-a. - www.groupsystem.com.

GroupSystem How to improve your organization's group intelligence [Online]. - 2006. - www.groupsystem.com.

Gudizial M. Anchored Collaborative Learning Environments, CaMILE: Collaborative and Multimedia Interactive Learning Environment [Article] // College of Computing. - [s.l.] :

Georgia Institute of Technology EduTech Institute GVU Center, 1997-a. - Vol.
<http://www.cc.gatech.edu/gvu>.

Gudizial M. Information ecology of collaborations in educational setting: influence of tool [Article] // Proceedings of computer Support for Collaborative learning. - 1997. - Vol.
<http://www.oise.utoronto.ca/cscl/papers/guzdial.pdf> .

Haemig P. D. Leis da Ecologia Populacional. [Online] // ECOLOGIA.INFO 23. - Haemig, P. D.. - 2007. - <http://www.ecologia.info/leis-ecologia-populacional.htm>.

Hogeweg P. and Hesper B. Interesting events and distributed systems [Article] // SCS conf. Distributed simulation. - San Diego-USA : [s.n.], 1985. - Vol. 1.

Jacobson M. J. and Wilensky U. Complex Systems in Education: Scientific and Educational Importance and Implications for the Learning Sciences [Article] // The Journal of the Learning Sciences. - [s.l.] : Lawrence Erlbaum Associates, 2006. - 15(1).

Jacobson M. J. Butterflies, traffic jams, and cheetahs: problem solving and complex systems [Conference] // Annual Meeting of the American Educational Research Association conference. - New Orleans : [s.n.], 2000.

Johnson S. Emergencia: a vida integrada de formigas, cérebros, cidades e softwares [Book]. - Rio de Janeiro - RJ : Jorge Zahar, 2003.

Kielgast S. and Hubbard B. A. Valor agregado à informação - da teoria à prática // Ciência da Informação. - 1997. - Vol. 26.

Koelma D., van Balen R. and Smeulders A. SCIL-VP: a multi-purpose visual programming environment [Conference] // SIGAPP Symposium on Applied Computing: Technological Challenges of the 1990's . - Kansas City - Missouri - USA : In Proceedings of the 1992 ACM, 1992.

Levy S. T. and Wilensky U. An analysis of student' patterns of exploration with NetLogo models embedded in the Connected Chemistry environment [Conference] // The annual meeting of the American Educational Research Association. - Montréal, Quebec - Canada : <http://ccl.northwestern.edu/papers/>, 2005.

Levy S. T., Kim H. and Wilensky U. Connected Chemistry - A study of secondary students using agent-based models to learn chemistry [Conference] // Modeling Across the Curriculum (MAC): Technology, Pedagogy, Assessment, & Research". in Annual meeting of the American Educational Research Associations. - San Diego - USA : <http://ccl.northwesterns.edu/papers/>, 2004.

Lieberman H. Your wish is my command: programming by example [Book]. - Sann Francisco - USA : Morgan Kaufmann, 2001.

Liu H. Social network profiles as taste performances [Article] // Journal of Computer-Mediated Communication. - 2007. - 13(1), article 13. - <http://jcmc.indiana.edu/vol13/issue1/liu.html>.

Menezes C. S. [et al.] QSabe - um ambiente inteligente para roteamento de perguntas em uma comunidade virtual de esclarecimento [Article]. - Santiago - Chile : [s.n.], 2003.

Menezes Crediné S. Práticas pedagógicas e prática docente: Parte II - Análise de ecossistemas // (nota de aula - impresso). - 2006.

Menezes Crediné Silva, Tavares Orivaldo Lira and Pessoa José marques QSabe - Trocando Experiências sobre Informática Educativa em uma Rede de Educadores [Journal] // SBC-ie nº 2. - 2003.

Milton Rosa and Orey Daniel C. Modelagem matemática: Como tudo começou...? [Online]. - 2005. - fevereiro 2006. - www.csus.edu/indiv/o/oreyd/sylabi/comotudocomecou.doc.

Minar N. [et al.] The Swarm Simulation System, A Toolkit for Building Multi-Agent Simulations [Online]. - 1996. - 2006. - citeseer.ist.psu.edu/minar96swarm.html.

Modugno F. and Myers A. Pursuit: Visual Programming in a Visual Domain. [Report]. - [s.l.] : Carnegie Mellon University School of Computer Science Technical Report, no. CMU-CS-94-109. <http://citeseer.ist.psu.edu/modugno94pursuit.html>, 1994.

Netto H. V. Agregando flexibilidade e configurabilidade ao ambiente AmCorA // Dissertação Mestrado (Engenharia Eletrica). - Vitória : Universidade Federal do Espírito Santo - UFES, 2003.

North M. J., Collier N. T. and Vos J. R. Experiences creating three implementations of the repast agent modeling toolkit [Article] // Transactions on Modeling and Computer Simulation. - [s.l.] : ACM Pres., 2006. - jan.. - Vol. 16. - pp. 1- 25.

Nunamaker J. Collaborative Computing: The Next Millennium [Article] // Computer. - 1999. - 9. - Vol. 32.

Papert S. Tomorrow's classrooms? [Online] // Times educational supplement. - Works by Seymour Papert, Ph.D., 1982. - <http://www.papert.org/articles/TomorrowsClassrooms.html>.

Pessoa José Marques Desenvolvimento de software orientado a agentes: uma experiência com agentes de interface // Dissertação Mestrado (Mestrado em Informática). - Vitória : Universidade Federal do Espírito Santo, 1997.

Polya George A arte de resolver problemas: um novo aspecto do método matemático [Book]. - Rio de Janeiro : Interciência, 1995.

Railsback Steven F., Lytinen Steven L. and Jackson Stephen K. Agent-based simulation platforms: Review and development recommendations [Online]. - 2006. - 2006. - <http://www.humboldt.edu/~ecomodel/documents/ABMPlatformReview.pdf>.

Recuero R. C Teoria das Redes e Redes Sociais na Internet [Article] // XXVII INTERCOM. - Porto Alegre - RS : PUC-RS, 2004.

Repenning A. and Perrone C. Programming by example: programming by analogous examples. [Article] // Commun . - [s.l.] : ACM, 2000. - 43. - Vols. 3 pg. 90-97.

Repenning Alexander Agentsheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments // Ph.D. Dissertation. - [s.l.] : University of Colorado at Boulder, Department of Computer Science, 1993.

Resnick M. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten [Conference] // Proceedings of the 6th ACM SIGCHI Conference on Creativity & Amp: Cognition. - Washington : ACM press, 2007.

Resnick Mitchel and Wilensky Uri Beyond the deterministic, centralized mindsets: New thinking for new sciences [Article] // Presented at the annual conference of the American Educational Research Association. - Atlanta, GA. : [s.n.], 1993.

Resnick Mitchel Decentralized Modeling and Decentralized Thinking [Article] // Modeling and Simulation in Precollege Science and Mathematics. - 1999. - Editor: W. Feurzeig and N. Roberts. - pp. 114-137, Springer: New York..

Resnick Mitchel Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds [Book]. - [s.l.] : Cambridge, MA: MIT Press., 1994.

Ropella G. E. P., Railsback S. F. and Jackson S. K. Software engineering considerations for individual-based models [Article] // Natural resource modeling. - [s.l.] : Spring, 2002. - 1. - Vol. 15.

Russel S. and Norvig P. inteligência Artificial [Book]. - [s.l.] : Campus, 2004. - Vol. Segunda edição.

Ruthen R. Adapting to Complexity [Article] // Scientific American. - 1993. - http://www.modelingcomplexity.org/pols490/adapting_to_complexity.pdf .

Santoro F. M., Borges M. R. S. and Santos N. Um framework para estudo de ambientes de suporte à aprendizagem cooperativa [Article] // Simposio Brasileiro de informática na Educação - SBIE. - 1998.

Smith D. C. Pygmalion: A computer program to model and stimulate creative thought [Book]. - Bassel - Stuttgart : Birkhauser Verlag, 1997.

Smith D. C., Cypher L. and Tesler L. Novice programming comes of age [Book Section] // Your wish is my command: programming by example / book auth. Libermann H.. - San Francisco - USA : Morgan Kaufmann Publishers, 2001.

Stefik M. [et al.] [Stefik et al, 1987] Stefik, M., Foster, G., Bobrow, D. Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings [Article] // Communications of the ACM. - 1987. - Vol. 30 N° 1.

Sussman J. M. Ideas on complexity in system - twenty views [Online]. - Massachusetts Institute of Technology, 2000. - 2005. - <http://esd.mit.edu/wps/esd-wp-2000-02.pdf>.

Tang Yuying, Parsons Simon and Sklar Elizabeth Agent-based modeling of human education data [Article] // AAMAS. - Hokkaido - Japan : ACM, 2006. - May-8-12.

Tang Yuqing, Parsons Simon and Sklar Elizabeth Agent-based modeling of human education data [Article] // Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. - Hokkaido - Japan : ACM, 2006. - may. - pp. 129-131.

Tatara E. [et al.] An Introduction to Repast Modeling by Using a Simple Predator-Prey Example [Conference] // Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects. - Argonne - IL-USA : [s.n.], 2006.

Tavares J. Resiliência e Educação (pp. 13-42) [Book]. - São Paulo : Cortez, 2001.

Tisue S. and Wilensky U. NetLogo: A Simple Environment for Modeling Complexity [Conference] // International Conference on Complex Systems. - Boston : [s.n.], 2004.

Tisue S. and Wilensky U. NetLogo: Design and Implementation of a Multi-Agent Modeling Environment [Conference] // SwarmFest. - Evanston, Illinois : Center for Connected Learning and Computer-Based Modeling Northwestern University, 2004(a).

Tommarello Joanna DeFranco Literature review of collaborative problem solving and groupware in the software development domain, State of the Art Review [Online]. - 2002. - 05 23, 2006. - http://web.njit.edu/~hiltz/SOTA_JDT.doc.

Tutorial Groove [Online] // Office Online. - Microsoft Office Groove, 2006-2007. - <http://office.microsoft.com/pt-br/products/HA101687221046.aspx>.

Wilensky Uri NetLogo Virus model [Online] // Center for Connected Learning and Computer-Based Modeling. - Northwestern University, Evanston, IL., 1998. - 2005. - <http://ccl.northwestern.edu/netlogo/models/Virus..>

Wilensky U. and Reisman K. Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories--an Embodied Modeling

Approach [Article] // Cognition and Instruction. - [s.l.] : Lawrence Erlbaum Associates, 2006. - Vol. <http://ccl.northwestern.edu/papers/wolfshe>.

Wilensky U. NetLogo [Online] // Center for Connected Learning and Computer-Based Modeling. - Northwestern University, Evanston - IL, 1999-2007. - 2005-2007. - <http://ccl.northwestern.edu/netlogo>.

Wilensky U. NetLogo Termites model. - Northwestern University, Evanston, IL. : Center for Connected Learning and Computer-Based Modeling, 1998.

Wilensky U. NetLogo Virus model. // Center for Connected Learning and Computer-Based Modeling. - [s.l.] : Northwestern University, Evanston, IL., 1998.

Wilensky Uri NetLogo Wolf Sheep Predation model [Online] // Center for connected learning and computer-based modeling. - Northwestern University, 1998. - 2006. - <http://ccl.northwestern.edu/netologo/models/RabbitsGrassWeeds>.

Woodridge M. An introduction to multiagent system [Book]. - Chichester - England : John Wiley & Sons Ltd, 2002.

Yates R. B. [et al.] Crawling a Country: Better Strategies than BreadthFirst for Web Page Ordering [Article] // WWW2005, May 10–14. - Chiba, Japan : [s.n.], 2005.

Yorke J., Nathanson N. and Pianigiani G., Martin, J. Seasonality and the requirements for perpetuation and eradication of viruses in populations [Article] // Journal of Epidemiology. - 1979. - Vols. volume 109, pages 103-123.

Apêndice A

Tabela 9 - Tabela de definição de agente – características de determinação

Descrição		NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets
D E F I N I Ç Ã O d o A G E N T E	Forma	Biblioteca de formas "shape editor", ferramenta para criar objetos "paint".	Biblioteca de figuras "picture editor", e ferramenta para criar objetos "paint".	Definido por um método "buildObjects". Utiliza comandos da biblioteca que cria objetos gráficos (subclasses da biblioteca de gráficos). O componente principal é o "swarm", que é uma coleção de agentes que possibilita agrupar hierarquias de agentes, controle de agenda de eventos, e sondar informações do estado atual do agente e de seu ambiente.	Disponibiliza uma interface para que o usuário possa desenhar a forma que dará ao seu agente.	Biblioteca de figuras "Gallery", e ferramenta para criar objetos "paint".
	Cor	Ferramenta "shape" disponibiliza uma paleta cores.	Na ferramenta "picture editor", há uma paleta de cores.	Idem	Disponibiliza opções de cores para serem selecionadas	Disponibiliza paint para desenhar formas e palhetas de cores.

Descrição		NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets
D E F I N I Ç A O D O A G E N T E	Tamanho	Tanto os modelos da biblioteca quanto os desenhados podem ser redimensionados pelo usuário	Tanto os modelos da biblioteca quanto os desenhados podem ser redimensionados pelo usuário	Idem	Definido pelo usuário.	Tanto os modelos da Gallery quanto os definidos pelo usuário podem ser redimensionados. A dimensão do agente define a célula do ambiente.
	Quantidade	Definido através de código: - <code>crt 10</code> (cria 10 agentes) - <code>crt agentes</code> (cria o número de agentes controlado pelo slider do parâmetro agente)	Os agentes são definidos um a um, e controlado pelo usuário.	A biblioteca do Swarm cria uma seqüência de classes que guardam e refinam a capacidade de criar objetos de simulação, gerenciamento de memória e agenda de suas atividades.	Determinada pelo usuário, através do preenchimento de um campo pré-estabelecido.	São posicionados um a um sobre a tabela que representa o ambiente.
	Posição	No modo default, os agentes são posicionados no centro da tela, uns sobre os outros. A distribuição randômica é feita através de comandos: - <code>setxy (random-float screen-size-x) (random-float screen-size-y)</code> - <code>fd random "valor"</code> . Também é possível definir a posição através de coordenadas específicas	O usuário que define a posição do objeto sobre o ambiente, usando o mouse. Pode-se criar os objetos definindo as coordenadas.	A biblioteca do Swarm cria uma seqüência de classes que guardam e refinam a capacidade de criar objetos de simulação, gerenciamento de memória e agenda de suas atividades.	Definida pelo usuário, pode ser fixa ou aleatória.	Definida pelo usuário, que posiciona o agente nas células através do controle do mouse.

Tabela 10 - Tabela de descrição de comportamento

Descrição	NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets	
REGRAS de COMPORTAMENTO	Movimento	Definido através de código: -fd "passos" (anda a frente "passos") - rt "ângulo" (gira direita "ângulo") -lt "ângulo" (gira esquerda "ângulo")	Disponibiliza a tabela de opções. Controlado pelo usuário. Controlado pelo computador.	Definido por métodos "buildActions". Os comandos definidos no método dizem aos agentes, ou grupos de agentes, qual deve ser o seu comportamento. Esses comandos são instancias da Classe "ActionGroup", que significa que eles serão repetidos se os grupos forem repetidos. As repetições são controladas pela agenda.	Definida pelas regras pré-definidas através da tabela de comportamento	Definida pela tabela de comportamento. Usando a composição de condição / ação.
	Percepção	É possível programar para que o modelo verifique a existência de um agente (móvel ou estacionário) no raio delimitado	Não possibilita	Idem	Definida pelas regras pré-definidas através da tabela de comportamento	Definida pela composição de condições de comportamento. O agente só percebe um outro se for adjacente a ele.
	Colisão	Determinado quando dois ou mais agentes se posicionam sobre a mesma posição no ambiente	Determinado quando dois ou mais agentes se posicionam sobre a mesma posição no ambiente	Idem	Definida pelas regras pré-definidas através da tabela de comportamento	Definida pela composição de condições de comportamento.

Tabela 11 - Tabela de definição de ambiente

Descrição	NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets
Ambiente	Uma grade (patch) bem definida por coordenadas cartesianas.	Não é bem delimitado	Definido pelos agentes estacionários	Definida pelos agentes que o modelam, denominados de agentes de ambiente.	O ambiente é definido como um agente. O usuário determina o local de posição das células de forma livre. Com essa opção é fácil desenhar formas para simular o piso do ambiente. Ao posicionar um agente sobre o "piso" este ocupa toda a célula.

Tabela 12 - Tabela de identidade dos agentes e definição de variáveis

Descrição	NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets
Identidade dos agentes	Cada agente tem uma identidade	Não há controle	Cada agente tem uma identidade	Os agentes têm identidade própria.	Não há controle
Variáveis	Definidas como globais, lógicas, locais, vinculadas aos agentes, sendo em quantidade indeterminada.	Vinculadas aos agentes e no máximo de três	Definidas através de classes. Pode ser visualizada de duas formas: DefaultProbeMaps - esta apresenta uma janela, gráfica, da classe com todas as variáveis pertencentes aquela classe. CustomProbeMaps - é uma janela personalizada, gera uma interface do objeto, mostrando apenas os métodos que a classe entende	Disponibiliza uma tabela para definição de variáveis a serem usadas pelos agentes. São oferecidos atributos específicos para a criação das variáveis: tipo, valor, nome.	Vinculadas à composição das ações / condições. São pré-estabelecidas.

Tabela 13 - Tabela de definição de elementos gráficos – ferramentas conceituais

Descrição	NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets	
E L E M E N T O S G R Á F I C O S	Gráficos	Disponibiliza ferramenta para construção automática de gráfico	Não disponibiliza	Fornecer uma variedade de classes e protocolos que geram interfaces gráficas: gráfico de linha, histogramas, movimento de simulação.	Não determinado	Não disponibiliza
	Monitores	Possui ferramenta para construção automática de monitoramento, cuja expressão pode ser programada.	Só disponibiliza contadores (sem referência)	Disponibiliza classes e protocolos para gerar monitores de acompanhamento.	Disponibiliza apenas algumas para acompanhamento das variáveis e agentes.	Oferece uma janela "Tiggers" com ações pré-definidas, bastante elementar.
	Botões de acionamento	Disponibiliza ferramenta para gerar botões de disparo de ações	Não disponibiliza	Idem	Disponibiliza apenas os botões de acionamento, pausa, e parada	Disponibiliza apenas os botões de acionar e parar a simulação.
	Informação externa	Possui ferramenta para exportar dados para programas externos	Não disponibiliza	Idem	Não determinado	Não disponibiliza
	Texto	Disponibiliza ferramenta para inserir texto informativo na interface gráfica	Não disponibiliza	idem	Não determinado	Não disponibiliza

Tabela 14 - Tabela de aspectos gerais – outros elementos

Descrição	NetLogo / StarLogo	Klik&Play	Swarm / Repast	SIMULA	AgentSheets
Fundo (backdrop)	Em termos gerais não possui. O ambiente é determinado pelos agentes estacionários, definido pelo usuário conforme modelo.	Biblioteca de modelos	Em termos gerais não possui, o ambiente é definido através da programação de objetos estacionários.	Em termos gerais não possui. O ambiente é determinado pelos agentes estacionários, definido pelo usuário conforme modelo.	Não possui. Porém pode ser simulado pela disposição dos próprios agentes.
Linguagem	Derivada Logo	Edição de eventos	Objetive C e Java Os modelos desenvolvidos no Repast podem ser construídos em outras linguagens além das citadas: C#, Managed C++, Visual Basic.Net, Managed Lisp, Managed Prolog, and Python scripting	Java	Programação visual
Biblioteca de modelos	Ambos ambientes Possuem muitos exemplos, classificados por áreas de conhecimento. Entretanto, a biblioteca do NetLogo é bem mais extensa.	Discreta. À medida que os modelos são construídos, amplia-se a biblioteca de modelos.	O Repast possui muitos modelos. O Swarm possui em número mais restrito	-	Os modelos desenvolvidos são armazenados numa pasta de modelos.

Vantagens	Muito fácil de usar e executar modelos. Possui uma grande comunidade que oferece suporte.	Fácil de usar. Possibilita a inserção de agentes na biblioteca de modo muito fácil, que possibilita o do agente.	O Repast possui muitos usuários, bom suporte pela comunidade.	Fácil de usar	Fácil de usar
Apoio da comunidade acadêmica e apoio de manutenção	Alto. Existe grupo de desenvolvimento extremamente ativo. A comunidade do StarLogo lançou uma nova versão em junho/2006, incluindo o OpenStarLogo, que é uma versão open source do ambiente. A última versão do NetLogo, a 3.1, ocorreu em abril de 2006.	Desconhecido	Swarm - Não é bem apoiado para a comunidade de Ciências Sociais. A manutenção é relativamente baixa. Repast - muito alto, principalmente da comunidade de Ciências Sociais. Manutenção ativa	Desconhecido	Desconhecido
Limitações	Modelos criados não podem ser facilmente estendidos	Cada modelos é criado para um fim específico, não podem ser facilmente estendidos. A versão disponível livre é para o âmbito acadêmico.	Os sistemas são bastante versáteis. Os modelos podem ser facilmente estendidos, desde que tenha conhecimento em programação.	Modelos criados não podem ser facilmente estendidos	Modelos criados não podem ser facilmente estendidos. É um software comercial e foi projetado para uso especial em MAC, a versão windows é limitada.

Desvantagens	Modelos muito complicados (que necessitem de algoritmos específicos) excede a capacidade do NetLogo	Bastante limitado, só possibilita descrever modelos definidos com regras simples.	Swarm é extremamente difícil de instalar e executar no sistema operacional Windows. Ambos exigem conhecimento de linguagens de programação.	Modelos limitados as condições pré-definidas no ambiente.	Bastante limitado, só possibilita descrever modelos definidos com regras simples.
Resolução cooperativa	O NetLogo apresenta uma ferramenta adicional HubNet, que possibilita vários usuários participarem simultaneamente na exploração do modelo, que pode estar ligado a uma rede de computadores o através de calculadora gráfica TI-83+	-	-	-	
Outras características	NetLogo Possibilita a visualização dos modelos em 3D, facilidade em gerar applet dos modelos.	-	Possui vários algoritmos embutidos: algoritmo genérico, probabilísticos, seleção randômica.	-	Possibilita gerar applet facilmente.

**Devido às similaridades entre alguns ambientes, apresentamos suas características numa mesma coluna das tabelas acima.

Apêndice B

Tabela 15 - Modelos desenvolvidos durante disciplina

Modelo	Característica
Armazém	O objetivo do modelo é dar ao "analista" (observador), uma visão de como o armazém vai se comportar no futuro. É levando em conta a quantidade máxima de estocagem e a capacidade de chegada e saída de carga. O analista poderá definir a quantidade de caminhões que chegam e que saem. O recebimento dos produtos é aleatório. No caso de um armazém de produtos diversos, consideramos que somente produtos iguais podem ser armazenados juntos (numa pilha máxima de 10 unidades). Os produtos são armazenados e retirados numa ordem seqüencial
Colméia-1	O objetivo desse modelo é observar o comportamento interno de uma colméia de abelhas relacionado às variáveis: temperatura, umidade, número de elementos. O controle das variáveis é definido pelo comportamento das abelhas - movimentação dentro da colméia. O modelo pretende mapear o posicionamento das abelhas dentro da colméia segundo o seu ciclo de vida (ovo, larva, pupa, adulta). Em cada estágio a abelha desenvolve um comportamento simples.
Colméia-2	<p>O objetivo desse modelo é observar o comportamento externo das abelhas com relação ao patrulhamento e busca de alimento. O destaque deste modelo é trajetória entre o alimento e a colméia demarcada pela dispersão do feromônio.</p> <p>Característica: cada abelha segue regras simples, mas a colméia como um todo representa um comportamento sofisticado.</p> <p>Comportamento: as abelhas ao sair da colméia andam aleatoriamente em busca do alimento que está depositado em algum lugar do ambiente. Ao encontrar o alimento, a abelha retorna a colméia deixando um rastro (feromônio) indicativo de onde este foi encontrado. Esse rastro ao ser detectado por outras abelhas conduz a fonte de alimento e retorno a colméia. Quanto mais abelhas seguem esse rastro, mais forte esse fica e, conseqüentemente, atrai maior número de abelhas. Essa demarcação faz com que as abelhas esgotem a fonte de alimento em ordem, isso é, há uma condução para que todas permaneçam numa mesma fonte. A busca de uma nova fonte só ocorre quando a anterior esgota. Nessa lógica, elas sempre buscam a fonte de alimento mais próxima a colméia. Ao terminar a fonte de alimento, o rastro vai se dispersando ate desaparecer.</p>

	<p>Caso a fonte esteja muito distante e o número de abelhas for pequeno e fatores influenciarem na dispersão do feromônio, a fonte de alimento demora mais tempo para ser esgotada.</p> <p>Para cada tipo de abelha há uma estratégia de comportamento, neste modelo foi priorizado apenas o rastro deixado pelo feromônio para conduzir as abelhas do ninho à fonte de alimento. Entretanto, foi implementado diferentes taxas de feromônio, isto é, pode ser controlado se o mesmo é mais denso ou não.</p> <p>Coisas percebidas: como no mundo real, se duas fontes de alimento estão eqüidistantes, as abelhas sempre direciona a uma delas e não as duas ao mesmo tempo.</p>
Controle de tráfego de veículos	<p>O objetivo é modelar o controle de tráfego de veículos de forma a identificar os possíveis tipos de controle de tráfego de uma determinada estrutura viária, analisar os resultados obtidos quando os diferentes tipos de controle são utilizados e verificar que tipo de controle é mais adequado para aquela estrutura. Com esse modelo pretende-se responder as seguintes questões: a) Qual o número de vezes que um veículo pára em semáforos durante a sua viagem?; b) Qual o tempo de parada de um veículo durante a sua viagem?; c) Qual o tempo de viagem de um veículo?</p> <p>Em situação experimental, foi modelado um cruzamento com fluxos diferenciados, um mais intenso que outro. numa via era permitido três correntes de fluxo em cada direção, na outra via apenas dois fluxos em cada direção. A intensidade do fluxo era aleatória e o tempo de parada no semáforo era controlado pelo número de veículos em posição de aguardando.</p> <p>Obs. um modelo de comportamento similar esta sendo desenvolvido pelo aluno como dissertação de mestrado. Neste trabalho o modelo esta sendo desenvolvido em linguagem de programação C++. Ao analisar as formas de implementação em cada um dos modelos, o aluno ficou surpreso com o fato de regras tão simples descrever um comportamento tão complexo. Além da facilidade de programação, tanto no ambiente NetLogo quanto no NetPlay.</p>
Cupins	<p>O objetivo desse modelo é representar o comportamento de uma colônia de cupins, destacando as classes de operário - que coletam alimento, e soldados - responsável pela guarda do ninho e proteção aos operários durante a coleta de alimentos</p>
Pênalti	<p>O modelo simula o treinamento de chute de pênalti com o objetivo de capturar a probabilidade de acertos do jogador. Os elementos observados serão: direção do chute, intensidade do chute, posicionamento do goleiro. Por outro lado, procura uma relação de</p>

	<p>melhor aproveitamento do goleiro. Comparar os resultados e deduzir quais estratégias utilizar para ter um maior aproveitamento nas cobranças de pênalti? O que é mais importante, força ou direção ? Dá para prever a movimentação do goleiro num determinado instante?</p>
Pingüins	<p>Baseado no filme "marcha dos pingüins" pretende-se com esse modelo reproduzir o comportamento dessas aves durante o período de migração. Modelar o comportamento em várias etapas (adulto e jovem, macho e fêmeas) com o objetivo de entender a luta pela preservação da espécie.</p>
Pitstop	<p>O problema é definido como: relatório de eficiência de trocas de pneus. Procura-se avaliar a Medida de eficiência de cada mecânico para efetuar a troca de pneus em um determinado tempo. Este tempo é aleatório, uma vez que podem ocorrer erros na tarefa do mecânico. Foi estipulado um tempo ideal de troca de pneus de dez segundos. A cada troca, calcula-se o tempo total da troca e, a partir do tempo desta troca, define-se se foi uma troca em um tempo aceitável, ou não. A melhor troca vai sendo registrada.</p> <p>Característica: o modelo apresenta comportamento completamente aleatório.</p> <p>Obs. com esse modelo é possível fazer uma avaliação do desempenho de cada um dos trocadores e computar qual o melhor desempenho. é atribuído um tempo ideal para as trocas e essa é a meta a ser atingida.</p>

Anexo 1

Questionário sobre o uso da ferramenta NetPlay.

Desenvolva um modelos de simulação referente a sistemas complexos usando o NetPlay e, por favor, responda as questões:

Faça uma breve descrição sobre o modelo e os elementos que podem ser observado, por exemplo:

- a) Agentes móveis;
- b) Agentes estacionários;
- c) Movimento;

Utilizando a ferramenta NetPlay para a criação de modelos, pode-se concluir que:

Com relação aos agentes:

- a) A criação de agentes (nome, cor, forma, tamanho, quantidade, posição e movimento) se dá de modo:
 - i) Fácil e atende aos requisitos necessários para compor as características dos agentes que compõem o modelo;
 - ii) Fácil, mas não atende aos requisitos necessários para compor as características dos agentes que compõem o modelo;
 - iii) Difícil, mas atende aos requisitos necessários para compor as características dos agentes que compõem o modelo;
 - iv) Difícil e não atende aos requisitos necessários para compor as características dos agentes que compõem o modelo;

Justifique sua resposta: _____

b) Os tipos de agentes (móveis, estacionários e borda):

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de agente? _____
- iv) Qual a sua característica? _____

c) Os movimentos disponíveis para o agente (passear, refletir e parado):

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de movimento? _____
- iv) Qual a sua característica? _____

Com relação à criação de variáveis:

- a) A criação de variáveis (lógica, slider e valor) se dá de modo:
 - i) Fácil e atende aos requisitos necessários para compor as características do modelo;
 - ii) Fácil, mas não atende aos requisitos necessários para compor as características do modelo;
 - iii) Difícil, mas atende aos requisitos necessários para compor as características do modelo;
 - iv) Difícil e não atende aos requisitos necessários para compor as características do modelo;Justifique sua resposta: _____

b) Os tipos disponíveis para as variáveis (lógico, slider e valor):

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de variável? _____
- iv) Qual a sua característica? _____

c) A criação da variável valor (editar expressão) se dá de modo:

- i) Fácil e atende aos requisitos necessários para descrever as variáveis;
 - ii) Fácil, mas não atende aos requisitos necessários para descrever as variáveis;
 - iii) Difícil, mas atende aos requisitos necessários para descrever as variáveis;
 - iv) Difícil e não atende aos requisitos necessários para descrever as variáveis;
- Justifique sua resposta: _____

Com relação às composições (condições e ações) que definem o comportamento do modelo:

- a) A criação das composições (condições e ações) que definem o comportamento dos agentes no modelo se dá de modo:

- i) Fácil e atende aos requisitos necessários para compor as características do modelo;
- ii) Fácil, mas não atende aos requisitos necessários para compor as características do modelo;
- iii) Difícil, mas atende aos requisitos necessários para compor as características do modelo;
- iv) Difícil e não atende aos requisitos necessários para compor as características dos agentes que compõem o modelo;

Justifique sua resposta: _____

b) As possibilidades de condições (percepção, posicionamento, cor do agente, valor de variável, comparação de variáveis) para descrever o comportamento do modelo:

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de condição? _____
- iv) Qual a sua característica? _____

c) As possibilidades de ações (criar elemento, destruir elemento, alterar cor, alterar forma, incrementar/decrementar variável, recuperar valor de variável, alterar posição, alterar movimento, parar movimento) para descrever o comportamento do modelo:

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de ação? _____
- iv) Qual a sua característica? _____

Com relação às ferramentas de acompanhamento dos dados gerados pelo modelo (monitor, gráfico e texto):

a) Monitores

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de ação? _____

b) Gráficos

- i) Conseguem mapear as necessidades do modelo;

- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de ação? _____

c) Textos

- i) Conseguem mapear as necessidades do modelo;
- ii) Deixa a desejar com relação à descrição dos modelos;
- iii) Sugere algum outro tipo de ação? _____

Com relação à ferramenta de análise dos dados (analisar gráfico)

- i) Captura os dados necessários para análise e cruzamento de dados (todos pontos, pontos principais, pontos de máximo/mínimo, pontos de interseção);
- ii) Não captura os dados necessários para análise e cruzamento de dados (todos pontos, pontos principais, pontos de máximo/mínimo, pontos de interseção);
- iii) Apresenta boa visualização com relação aos dados apresentados;
- iv) Não apresenta boa visualização com relação aos dados apresentados;
- v) Sugere algum outro tipo de análise de dados? _____

Com relação à interface da ferramenta

- a) Fácil utilização. Bastante didática e intuitiva para construção de modelos de simulação por leigos em programação
- b) Media utilização. Media didática e media intuição para construção de modelos de simulação por leigos em programação;
- c) Difícil utilização. Pouco didática e intuitiva para a construção de modelos de simulação por leigos em programação;

Sinta-se à vontade para maiores considerações.

Obrigada!

