

SIMULA - Ambiente de Simulação em Sistemas Multiagentes Reativos

**Rejane Frozza
Luis Otávio CamposAlvares
Frederico Sorio**

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Av. Bento Gonçalves, 9500
91501-970 Porto Alegre RS
Fax: (051) 336.5576

e-mail: [frozza,alvares,fsorio]@inf.ufrgs.br

RESUMO

Os sistemas multiagentes reativos são um tema de estudo da inteligência artificial distribuída e têm sido usados em pesquisas e estudos ligados a campos importantes de aplicação, gerando o desenvolvimento de sistemas não apenas para a área acadêmica, mas também para atender às necessidades do mercado industrial.

Este trabalho apresenta a definição e a implementação de um protótipo de um ambiente de simulação que possibilita o desenvolvimento de aplicações em sistemas multiagentes reativos. Este ambiente tem a finalidade de facilitar a criação de tais aplicações com o uso de agentes, atingindo um resultado satisfatório.

O ambiente definido é o SIMULA, que possibilita ao usuário criar suas aplicações através de elementos de uma interface gráfica. O usuário, na interação com o ambiente, determina os agentes envolvidos no problema e como eles agirão no processo de resolução do mesmo.

Apresenta-se, inicialmente, uma visão geral da área, seguida de breve descrição de três modelos de sistemas multiagentes reativos, com um quadro comparativo dos mesmos. A seguir, descreve-se, como objetivo principal, a definição do ambiente proposto para o desenvolvimento de aplicações baseadas neste paradigma, estabelecendo uma comparação com dois outros ambientes existentes.

1. INTRODUÇÃO

Os agentes reativos estão situados na área da inteligência artificial baseada em comportamento [MAE 93], que é uma nova abordagem para o estudo da inteligência, enfatizando a modelagem e a construção de sistemas para problemas que exibem alguma espécie de comportamento em seu domínio. Tem o objetivo de construir sistemas autônomos que operam em ambientes dinâmicos.

As pesquisas em sistemas multiagentes reativos estão gerando o desenvolvimento de sistemas não apenas para a área acadêmica, mas também para atender às necessidades do mercado industrial.

A disponibilidade de um ambiente de desenvolvimento de sistemas reativos traria facilidades para a implementação das aplicações que estão sendo exploradas neste campo da computação. Com o crescimento do interesse pela utilização de sistemas reativos para a solução de problemas, é importante que haja estímulo para a criação de sistemas reais, como o ambiente descrito neste trabalho.

O trabalho está organizado em seis seções. A seção 2 apresenta uma visão geral da área de sistemas multiagentes reativos. Na seção 3 são descritos brevemente três modelos de sistemas segundo este paradigma e é apresentado um quadro comparativo dos mesmos. O ambiente desenvolvido é introduzido na seção 4. A seção 5 aborda o protótipo do ambiente.

2. SISTEMAS MULTIAGENTES REATIVOS

A idéia principal, em um sistema multiagente reativo, é que um comportamento global inteligente possa ser alcançado a partir do comportamento individual dos agentes. Em um sistema multiagente, não é necessário que cada agente seja individualmente inteligente para alcançar um comportamento global inteligente. Para ilustrar essa idéia, utiliza-se o exemplo de uma colônia de formigas: mesmo que uma única formiga apresente poucas capacidades, pareça não ser muito inteligente, o comportamento de uma colônia de formigas, como um todo, é bem estruturado e fez com que as formigas sobrevivessem por milhões de anos.

Alexis Drogoul, em [DRO 93], apresenta a modelagem e a simulação de um sistema multiagente reativo aplicado ao comportamento de uma colônia de formigas.

Tudo que um agente reativo sabe sobre as ações e o comportamento dos outros membros da sociedade é percebido pelas mudanças no ambiente. Seguindo o exemplo das formigas, estas, usualmente, seguem um rastro químico quando buscam por comida, mas não existe um modelo de comunicação direta entre elas. Os agentes reativos possuem uma representação implícita do conhecimento sobre o ambiente e sobre os outros agentes: o conhecimento é representado pelo comportamento dos agentes reativos no sistema multiagente. São agentes baseados em comportamento: cada agente, individualmente, exibe comportamento de acordo com a situação na qual se encontra no ambiente de solução de algum problema. Quando o ambiente altera-se, os agentes reativos mudam seu comportamento.

O modo de funcionamento dos agentes reativos é por estímulo e resposta: a relação do agente reativo com seu ambiente ocorre através de respostas a estímulos recebidos. O agente executa uma determinada ação quando uma certa condição for satisfeita. Eles possuem capacidade de percepção e de comunicação: a comunicação ocorre através do ambiente, o qual é utilizado como meio de transmissão de mensagens. Tal comunicação não é dirigida a um determinado agente, mas é propagada a todos pelo ambiente, sem o uso de algum modelo de comunicação para o envio de mensagens. Outra característica importante dos agentes reativos é que eles não possuem memória das suas ações executadas no passado nem qualquer previsão das ações a serem executadas no futuro. Como não possuem capacidade de raciocínio, não

planejam suas ações e, sim, reagem a estímulos. Os sistemas multiagentes reativos tem, geralmente, um grande número de membros.

3. MODELOS DE SISTEMAS MULTIAGENTES REATIVOS

Neste trabalho são abordados três modelos de sistemas multiagentes reativos, que foram estudados e analisados com o objetivo de ressaltar as principais características de cada modelo, para determinar os elementos essenciais constituintes do protótipo de um ambiente de desenvolvimento de aplicações em sistemas multiagentes reativos.

3.1 Modelo da Funcionalidade Emergente

O modelo da Funcionalidade Emergente é descrito em [STE 90]. O objetivo do sistema multiagente reativo é que os agentes interajam através do ambiente e, desta forma, cooperem entre si para melhor realizarem as suas tarefas. Este modelo utiliza como idéias principais a da "subsumption architecture" de Brooks [BRO 91], que estabelece prioridades entre a execução dos comportamentos atribuídos aos agentes, e a da funcionalidade emergente, na qual um sistema com muitos componentes simples pode exibir um comportamento, como um todo, mais organizado do que o comportamento das partes individuais.

Segundo Steels, a funcionalidade está relacionada com o conceito de auto-organização de uma sociedade. A auto-organização é a capacidade que uma população de entidades (neste caso, um grupo de agentes) possui de se estruturar espacial e temporalmente em um ambiente. É uma modificação na topologia de um grupo de agentes, realizada de modo autônomo, permitindo a estes agentes se adaptarem ao seu ambiente através de uma nova organização. Pode-se considerar a estrutura da sociedade de agentes como a sua organização, e a realização das tarefas, por parte dos agentes, como a sua funcionalidade.

3.2 Modelo PACO

O Modelo PACO (Coordination Patterns) foi proposto por Demazeau [DEM 91] e utilizado em aplicações especializadas, como análise de imagens para a detecção de contornos e em generalização cartográfica [ALV 96]. Também se aplica à robótica para o controle de execução de trajetórias em robôs [DEM 93].

O modelo envolve um conjunto de agentes reativos que atuarão conforme o domínio da aplicação. Cada agente deste conjunto assume comportamentos definidos a priori e representa uma solução parcial do problema. Tal solução corresponde à posição de cada agente no ambiente. A solução global do problema é dada pela posição do conjunto de agentes.

A detecção de que uma solução foi encontrada é feita por um observador externo e não pelos próprios agentes, porque cada agente isoladamente não sabe se os outros agentes estão em uma posição estável ou não. Este observador externo pode ser o usuário do sistema ou um agente supervisor, com características diferentes dos outros agentes do sistema.

Neste modelo é introduzida a noção de campo que é o subconjunto do ambiente visível ao agente (a sua vizinhança). Para abordar tal noção, foram definidos três campos, para cada agente, que operacionalizam seu comportamento:

- *Campo de percepção* : guia as interações entre os agentes e o ambiente. É o subconjunto do ambiente que o agente pode perceber em um determinado momento. Por exemplo, o conjunto de agentes que o agente em questão sabe que existe.
- *Campo de comunicação* : guia as interações entre os agentes. É o subconjunto de agentes com quem o agente em questão pode se comunicar e que o influenciarão na execução de uma ação.

• *Campo de ação* : guia as ações de deslocamento do agente no ambiente. É o conjunto de agentes que podem sofrer uma ação do agente em questão.

3.3 Modelo da Eco-Resolução

O modelo da Eco-Resolução [FER 91] baseia-se em agentes com apenas dois tipos de comportamento: de *satisfação* e de *fuga*. Um eco-agente possui conhecimento local de seu próprio ambiente e comportamentos que modificam este conhecimento.

Quanto ao conhecimento local, um eco-agente sempre conhece: o seu estado de satisfação (quando o agente atinge a sua meta é verdadeiro e, caso contrário, é falso); a dependência temporária de um agente em relação a outro (relacionamento mestre-escravo) e os agentes que o impedem de agir.

Os comportamentos (ações) são simples e baseados no estado interno dos agentes. O estado interno reflete o posicionamento do agente no ambiente. Por exemplo, se ele está livre para agir ou não. O estado interno de um agente é representado pela tupla (s, f, l), onde 's' é o estado de satisfação, 'f' é o estado de fuga e 'l' é o estado de liberdade.

Um exemplo de uso deste modelo em jogos é apresentado em [DRO 92].

3.4 Taxonomia dos Modelos

Para estabelecer uma comparação entre os modelos de sistemas multiagentes reativos estudados, foram ressaltadas características de cada um, mostrando suas diferenças e similaridades. A tabela 1 mostra o quadro de comparação entre os modelos.

4. O AMBIENTE DE DESENVOLVIMENTO SIMULA

Após o estudo das características dos modelos apresentados na seção anterior, definiu-se um ambiente para facilitar o desenvolvimento de sistemas multiagentes reativos. Este ambiente teve, desde o seu projeto, o objetivo de ser de fácil utilização, o que não impede sua utilização na simulação de problemas reais. Possibilita ao usuário criar aplicações em sistemas multiagentes reativos, através de elementos de uma interface gráfica, os quais permitem a este usuário definir os agentes envolvidos na simulação e como eles agirão no processo de resolução da simulação. O usuário define a sua aplicação criando um modelo para ela.

Tabela 1 - Comparação entre Modelos de Sistemas Multiagentes Reativos

MODELOS DE SISTEMAS MULTIAGENTES REATIVOS			
Características	Funcionalidade Emergente	PACO	Eco-Resolução
Agentes	<ul style="list-style-type: none"> . são reativos . possuem estados internos . se auto-organizam em seu ambiente . suas atividades emergem para uma ação coletiva 	<ul style="list-style-type: none"> . são reativos . possuem estados internos . podem ter uma organização inicial . possuem atributos (massa, posição,) 	<ul style="list-style-type: none"> . são reativos . são baseados no comportamento de animais . possuem estados internos em relação a sua satisfação, fuga e liberdade
Ambiente dos Agentes	<ul style="list-style-type: none"> . local de atuação dos agentes determinado pela aplicação . possui uma representação física 	<ul style="list-style-type: none"> . local de atuação dos agentes determinado pela aplicação . possui uma representação física 	<ul style="list-style-type: none"> . local de atuação dos agentes determinado pela aplicação . possui uma representação física
Interações	<ul style="list-style-type: none"> . entre agentes . entre agentes e ambiente . associadas à percepção do agente 	<ul style="list-style-type: none"> . entre agentes . entre agentes e ambiente . associadas à percepção do agente . baseadas em modelos físicos de forças 	<ul style="list-style-type: none"> . entre agentes . entre agentes e ambiente . associadas à percepção do agente
Comportamentos	<ul style="list-style-type: none"> . representam ações dos agentes . variam conforme o estado interno do agente . ligados à percepção dos agentes . baseados na <i>Subsumption architecture</i> 	<ul style="list-style-type: none"> . representam ações dos agentes . variam conforme o estado interno do agente . caracterizados pelas interações dos agentes . utilizam a noção de campos (vizinhança) dos agentes: de percepção, ação e comunicação . especificados por um Ciclo de Resolução 	<ul style="list-style-type: none"> . representam ações dos agentes . variam conforme o estado interno do agente . possui os comportamentos de satisfação e de fuga
Solução do Problema	<ul style="list-style-type: none"> . é a situação final do problema, quando os agentes realizaram suas tarefas 	<ul style="list-style-type: none"> . é encontrada quando os agentes atingem um estado de equilíbrio . gerada pelas soluções parciais de cada agente, que correspondem às posições de cada agente no ambiente 	<ul style="list-style-type: none"> . é a situação final do problema, quando os agentes atingem o seu estado de satisfação

4.1 Características do Ambiente de Desenvolvimento SIMULA

O ambiente 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.

Possui as seguintes características, detalhadas em [FRO 97]: apresenta uma interface gráfica que permite definir um problema como um sistema multiagente reativo; permite descrever os tipos de agentes envolvidos no problema ou na simulação, que agirão em busca de uma solução; permite descrever as regras de comportamento dos agentes definidos, que determinam como os agentes agirão (quais são as suas ações), para atingir uma solução, e quando (em que situação) a regra será ativada; permite situar os agentes em seu ambiente, determinando a posição inicial de cada um; possui comportamentos pré-definidos, usados para montar as regras de comportamento dos agentes e apresenta, de forma gráfica, o processo de resolução do problema ou da simulação.

4.2 SIMULA x Ambientes Existentes

Foram estudados dois ambientes de desenvolvimento de aplicações em agentes, que são o SIEME [MAG 96] e o SWARM [MIE 96]. A figura 1 apresenta uma comparação entre o SIMULA e estes ambientes.

Sistema SIEME	Sistema SWARM	SIMULA
modelo da aplicação:		descrição dos agentes
descrição das ações		inicialização
interação com o usuário: através de uma linguagem de programação	interação com o usuário: através de uma linguagem de programação	interação com o usuário: janelas e menus
programação Smalltalk	programação C++	livre de linguagens tradicionais

Figura 1 - Comparação entre Ambientes

O SIEME é um simulador para sistemas multiagentes, totalmente integrado ao sistema de programação Smalltalk. Toda a especificação é realizada utilizando-se a linguagem Smalltalk, com o emprego de classes pré-definidas. O SWARM é um sistema integrado à linguagem de programação C++, que oferece uma biblioteca orientada a objetos para a construção de modelos. No SIMULA, o usuário está livre da tarefa de conhecer uma linguagem de programação específica para uso do ambiente. O usuário programa através da especificação de regras.

4.3 Funcionalidade do Ambiente SIMULA

O objetivo do SIMULA é diminuir o esforço de programação do usuário para criar suas aplicações, estimulando o mesmo a projetar novos sistemas com o uso de agentes reativos. Para isso, o ambiente oferece facilidades para a definição de problemas.

A figura 2 apresenta um esquema de como o usuário do ambiente SIMULA deve proceder para desenvolver suas aplicações em sistemas multiagentes reativos e de quais são as etapas internas de execução até ser atingida uma solução (situação final) para a aplicação descrita. O ambiente foi estruturado em dois módulos: módulo de definição e de execução.

Como pode-se observar na figura, a definição de uma aplicação compõe-se de três partes:

- *especificação dos tipos de agentes que farão parte do sistema*: o usuário especifica quais serão os agentes envolvidos no processo, associando um nome para cada tipo de agente. Também é definida a área de percepção do agente, que refere-se à área na qual o agente consegue perceber outros agentes;
- *especificação das regras de comportamento dos agentes*: as regras de comportamento são especificadas para cada tipo de agente e são elas que guiam as ações dos agentes no processo de simulação. Para especificar estas regras, o usuário utiliza os comportamentos pré-definidos do ambiente e também pode utilizar variáveis criadas por ele;
- *especificação da disposição dos agentes no ambiente*: é oferecida, ao usuário, a possibilidade de determinar a posição inicial de cada agente no ambiente no qual atuarão. Nesta etapa, o usuário define o número de agentes de cada tipo.

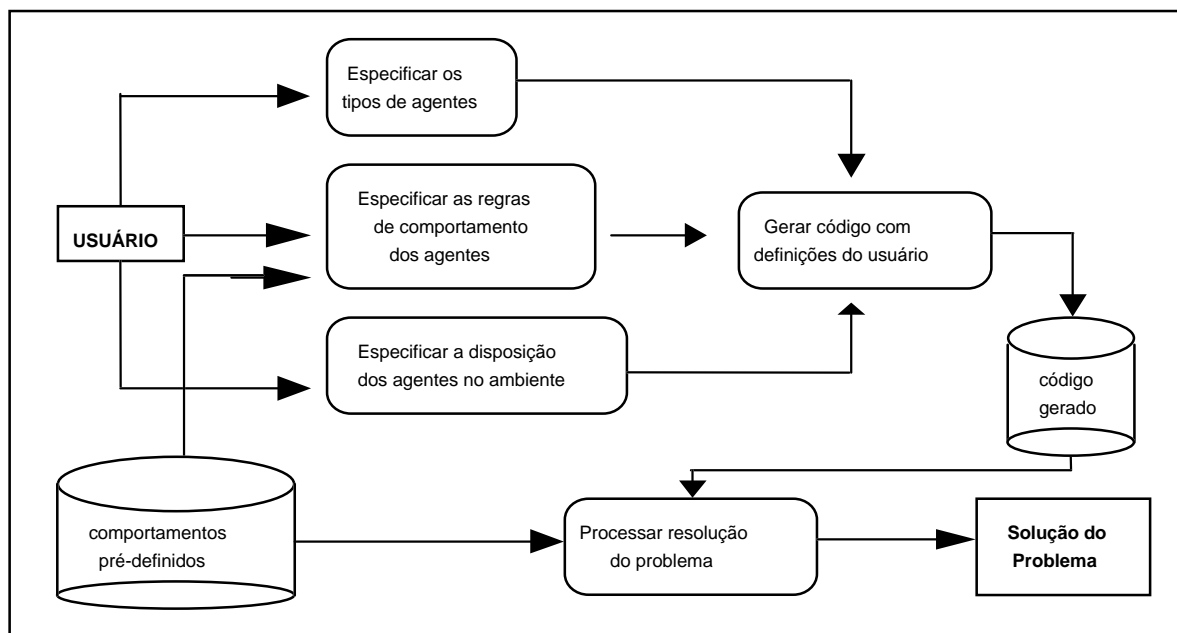


Figura 2 - Funcionalidade do Ambiente SIMULA

Após serem realizadas todas as definições necessárias, o usuário poderá acionar o módulo de execução do ambiente SIMULA. Este módulo divide-se em duas partes:

- *geração de código*: este processo utilizará as definições feitas pelo usuário para a sua aplicação, juntamente com os comportamentos pré-definidos do ambiente para, então, gerar um código, que será responsável pelo processo de resolução da aplicação;
- *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 e de seus movimentos/deslocamentos no ambiente.

Os comportamentos são associados a cada um dos tipos de agentes definidos pelo usuário. Todos os agentes do mesmo tipo agem da mesma forma, possuindo os mesmos comportamentos.

Há três classes de comportamentos pré-definidos no ambiente SIMULA e que são usados para montar as regras de comportamento dos agentes: comportamento ativo, comportamento passivo e comportamento de estado.

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.

A figura 3 mostra o que deve ser especificado em cada regra de comportamento. Cada tipo de agente pode ter várias regras de comportamento.

REGRAS DE COMPORTAMENTO	
pré-condição	: < condições para execução do comportamento >
ação-ativada	: < comportamento a ser executado >
ação-condicional	: < comportamento embutido na ação ativada >
pós-condição	: < atualizações de variáveis >
prioridade	: < ordem para a execução das regras >

Figura 3 - Elementos de uma Regra de Comportamento

A seguir, são descritos os elementos que compõem as regras:

- *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;
- *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 este comportamento, define-se que o mesmo deve verificar se há obstáculos em seu caminho;
- *pós-condição* : são os efeitos causados pela execução dos comportamentos;
- *prioridade* : o usuário pode, através da prioridade, estabelecer a ordem de execução das regras, porque, a cada ciclo, a agente realiza apenas uma ação/comportamento.

5. O PROTÓTIPO DO AMBIENTE SIMULA

Desenvolveu-se um protótipo do ambiente SIMULA, com o objetivo de verificar se as características contempladas eram suficientes e satisfatórias para a simulação de sistemas, envolvendo a atuação de agentes reativos na solução de problemas. O protótipo foi desenvolvido na linguagem JAVA.



Figura 4 - Janela Principal do Ambiente : Exemplo “Pets”

A figura 4 mostra que a interface do SIMULA é constituída por um menu principal com quatro opções, através das quais o usuário realiza as definições e visualiza a simulação da aplicação modelada.

A primeira etapa a ser realizada pelo usuário, para a definição de sua aplicação, é definir os agentes que irão compor o ambiente de solução do processo. A figura 5 mostra a tela para definição de um agente.

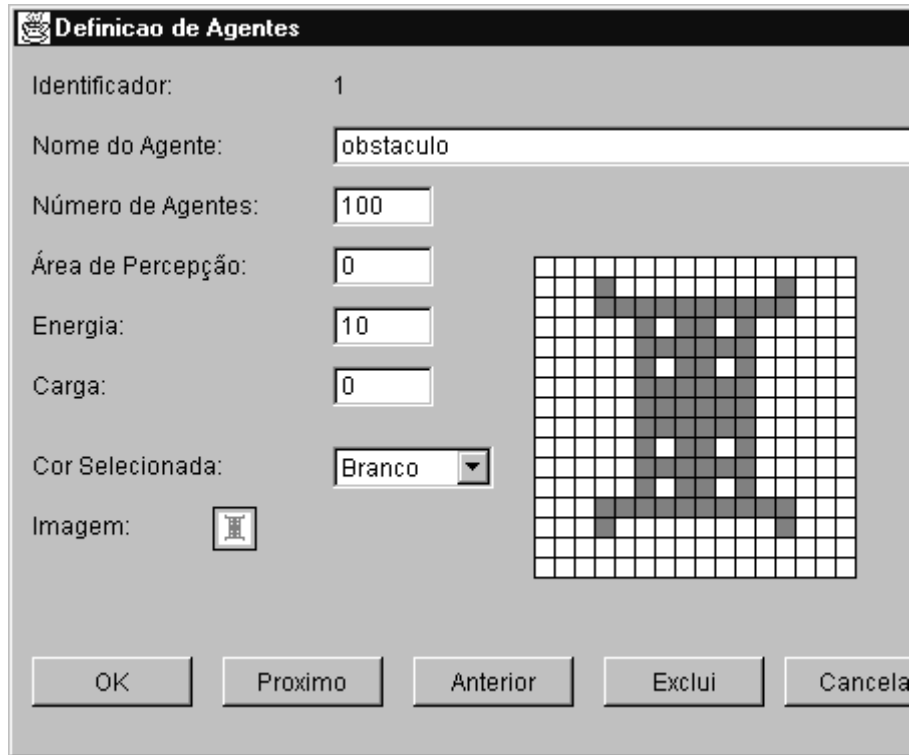


Figura 5 - Janela de Definição de Agentes

Após definir os agentes, o usuário poderá passar para a próxima etapa que é a definição dos comportamentos de cada tipo de agente. A definição de comportamentos aos agentes é feita através de regras que determinarão as ações a serem executadas pelos agentes e sob que condições.

A figura 6 mostra que a definição de comportamentos envolve a seleção do tipo de agente, definido anteriormente, a especificação da prioridade da regra e a criação da regra de comportamento.

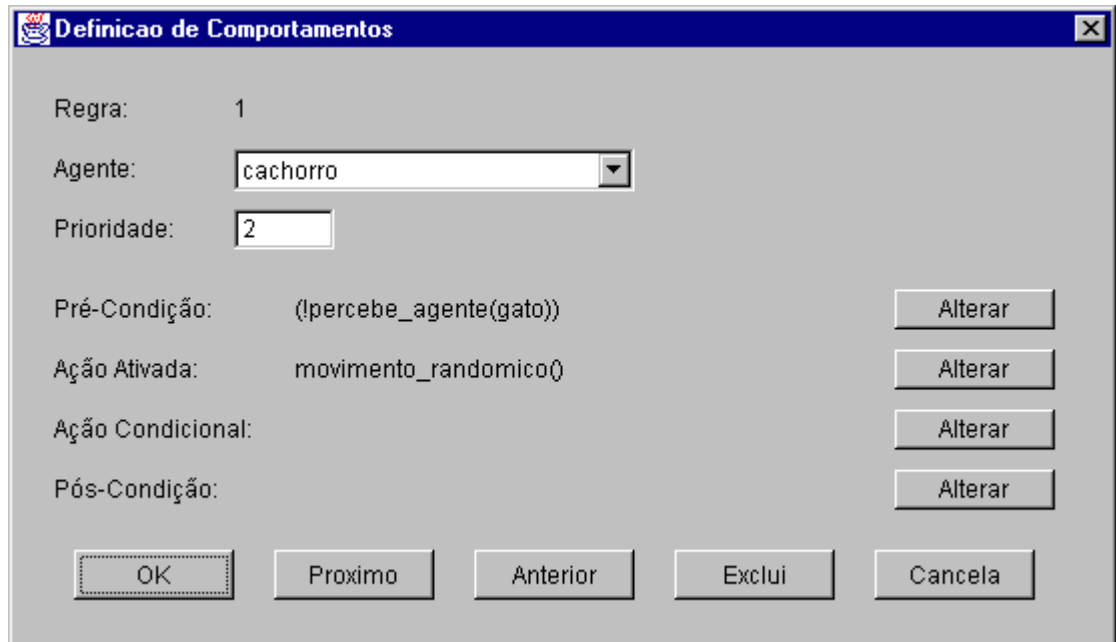


Figura 6 - Janela de Definição dos Comportamentos dos Agentes

Através da janela de definição da distribuição dos agentes no ambiente, como mostra a figura 7, o usuário poderá definir uma posição inicial para cada agente. Esta posição deve estar nos limites da matriz do ambiente dos agentes. Para os agentes que não tiverem uma posição inicial definida pelo usuário, o sistema se encarrega de gerar posições aleatórias para os mesmos.

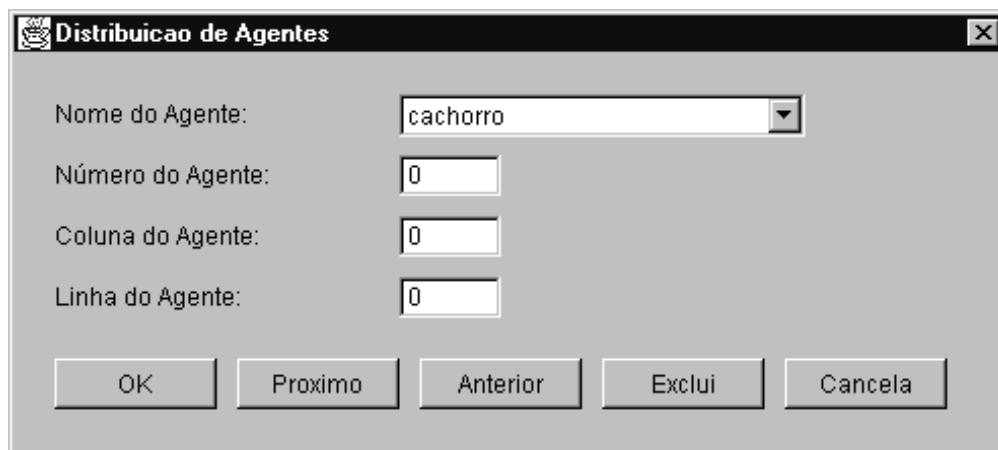


Figura 7 - Janela de Definição da Distribuição dos Agentes no Ambiente

Após serem realizadas todas as definições, o usuário deverá ativar a geração do código através do menu *Executar*. O código é gerado com as regras de comportamento na ordem de prioridade estabelecida. O arquivo gerado denomina-se CODIGOS.JAVA.

Depois disso, o usuário poderá ativar a execução da simulação através do menu *Executar*. A figura 8 ilustra um exemplo de simulação. Nesta janela, pode-se verificar as

variáveis e seus valores, o número de agentes de cada tipo e o número de ciclos executados. Um ciclo corresponde à execução de uma regra de comportamento por cada tipo de agente.

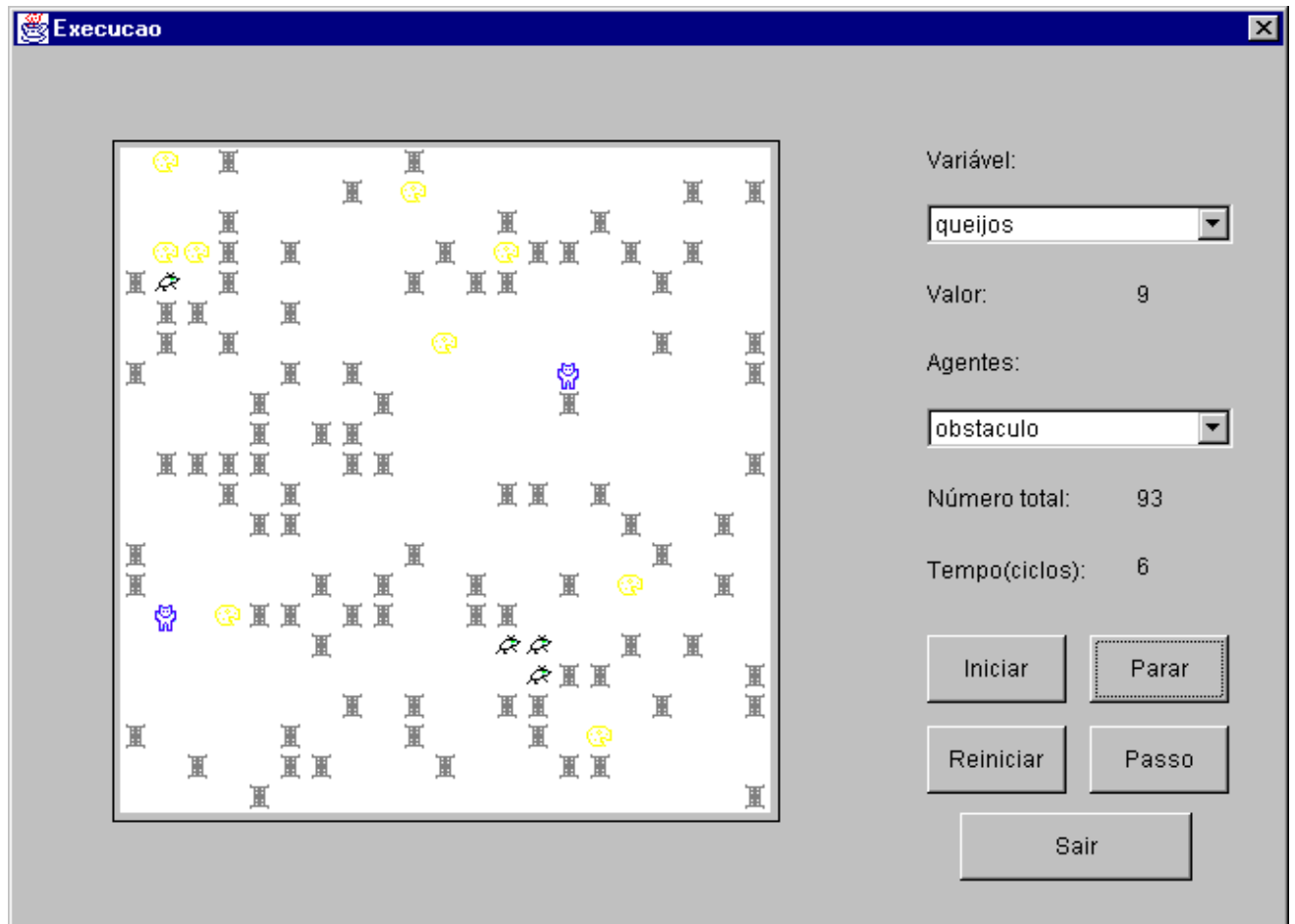


Figura 8 - Tela de execução

6. Conclusões

Nos últimos anos houve um crescimento do estudo e da utilização da abordagem de sistemas multiagentes reativos, nas mais diversas áreas de aplicações. Este interesse surgiu da necessidade de aplicar novas técnicas e conceitos para a construção de sistemas e para auxiliar no desenvolvimento dos mesmos.

Os objetivos, para a construção de um ambiente de simulação com agentes reativos, foram incentivar cada vez mais os usuários a modelarem seus problemas para serem resolvidos com o uso de agentes reativos e facilitar o trabalho dos usuários, livrando-os da tarefa de programação e envolvendo-os apenas na definição de seus problemas como um conjunto de agentes possuindo comportamentos que regem as ações dos agentes.

Como vantagens do ambiente de desenvolvimento SIMULA, pode-se citar:

- o uso de uma interface gráfica para as definições das aplicações dos usuários, tornando mais amigável a interação do usuário com o ambiente;
- o uso de um editor gráfico para desenhar os agentes envolvidos na simulação;

- o processo de resolução da aplicação definida pelo usuário é feita de forma gráfica, permitindo ao usuário acompanhar visualmente a atuação dos agentes durante a simulação;
- o usuário não precisa conhecer uma linguagem de programação tradicional para desenvolver sua aplicação. A tarefa do usuário é modelar a sua aplicação através de agentes, definindo-os e especificando suas regras de comportamento

Dando continuidade a este projeto, já está cumprida a etapa prevista de integração de um editor gráfico de agentes e implementação do ambiente em Java. As próximas etapas previstas são a realização da análise sintática das regras no momento de sua definição e a realização de um módulo para a definição e integração de novos comportamentos ao ambiente.

7. Referências

- [ALV 96] ALVARES, Luis Otavio; BAEIJS, C; DEMAZEAU, Y.. **SIGMA: Um Sistema Holístico de Generalização Cartográfica**. Anais GIS BRASIL 96. Curitiba, 6-10 maio, 1996.
- [BRO 91] BROOKS, Rodney A.. Intelligence Without Reason. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI-91), 12., 1991, Sydney. **Proceedings...** Austrália, 1991.
- [DEM 91] DEMAZEAU, Yves. **Coordination Patterns in Multi-Agent Worlds: Applications to Computer Vision and Robotics**. IEE Colloquium on Intelligent Agents, IEE, London, February, 1991.
- [DEM 93] DEMAZEAU, Yves. La Plate-forme PACO et ses applications. In: JOURNÉE NATIONALE DU PRC-IA SUR LES SYSTÈMES MULTI-AGENTS, 2., 1993, Montpellier, FR. **Annales...** [s.l. : s.n.], 1993.
- [DRO 92] DROGOUL, Alexis; DUBREUIL, Christophe. Eco-Problem-Solving Model: Results of the N Puzzle. In: WERNER, Eric; DEMAZEAU, Yves (Eds.). **Decentralized A.I.3**. Amsterdam: North-Holland, 1992.
- [DRO 93] DROGOUL, Alexis. **De La Simulation Multi-Agents à La Résolution Collective de Problèmes**. Thèse de L'Université Paris VI. Institut Blaise Pascal, LAFORIA TH93/05, França, 23 de novembre de 1993.
- [FER 91] FERBER, Jacques; JACOPIN, Eric. The Framework of Eco-Problem Solving. In: DEMAZEAU, Yves; MULLER, Jean-Pierre (Eds.). **Decentralized A.I.2**, Amsterdam: North-Holland, 1991.
- [FRO 97] FROZZA, Rejane. **SIMULA - Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos**. Dissertação de Mestrado. CPGCC/UFRGS. Porto Alegre, 1997.
- [MAE 93] MAES, Pattie. Behavior-Based Artificial Intelligence. In: From Animals To Animats, INTERNATIONAL CONFERENCE ON THE SIMULATION OF ADAPTIVE BEHAVIOR, 2., 1993. **Proceedings...** Cambridge: MIT Press, 1993.
- [MAG 96] MAGNIN, Laurent. **SIEME - Simulateur d'environnement pour systèmes multi-agents**. Disponível por WWW em <http://www-laforia.ibp.fr/~magnin/these/sieme.html> (novembro 1996).
- [MIE 96] MINAR, Nelson et al.. **The SWARM Simulation System: a toolkit for building multi-agent simulations**. Disponível por WWW em <http://www.santafe.edu/projects/swarm/overview/overview.html> (novembro 1996).

[STE 90] STEELS, Luc. Cooperating Between Distributed Agents Through Self-Organisation. In: DEMAZEAU, Yves; MULLER, J.P. (Eds.). **Decentralized A.I.**, Amsterdam: North-Holland, 1990.