

Instituto Militar de Engenharia

Mestrado em Sistemas e Computação

**Utilização do Ambiente SIMULA em
Sistemas a Eventos Discretos**

Wagner Tanaka Botelho

Prof. Cap. Antônio Eduardo Carrilho da Cunha

IME - Rio de Janeiro - RJ - Setembro/2003

Sumário

1	Introdução	5
2	Inteligência Artificial Distribuída	6
2.1	Agentes	6
2.1.1	Agentes Reativos	8
2.1.2	Agentes Cognitivos	8
2.2	Sistemas Multiagentes	8
2.2.1	Sistema Multiagentes Reativos	8
2.2.2	Sistema Multiagentes Cognitivos	9
3	Modelagem e o Ambiente Simula	12
3.1	Descrição do Problema	12
3.2	Objetivo	12
3.3	Funcionalidades do Ambiente	13
3.3.1	Módulo de Definição	13
3.3.2	Módulo de Execução	14
3.4	Comportamento Pré-definidos do Ambiente	14
3.5	Regras de Comportamento	15
3.6	Interface do Ambiente SIMULA	15
3.6.1	Restrições do Ambiente	16
3.6.2	Janela de Definição de Agentes: Primeira Etapa	17
3.6.3	Janela de Definição de Variáveis: Segunda Etapa	19
3.6.4	Janela de Definição dos Comportamentos dos Agentes: Terceira Etapa	19
3.6.5	Janela de Definição do Critério de Parada: Quarta Etapa	20
3.6.6	Janela de Definição das Dimensões do Ambiente: Quinta Etapa	22
3.6.7	Janela de Definição da Distribuição dos Agentes no Ambiente: Sexta Etapa	22
3.6.8	Janela de Execução	22
3.7	Definição dos Agentes para a Modelagem	23
3.7.1	Variável Global	24
3.7.2	Regras de Comportamento	25

3.7.3	Critério de Parada	26
3.7.4	Algoritmo	27
3.8	Modelagem por Autômatos	28
3.8.1	Agente Soldado	28
3.8.2	Agente Inimigo	29
3.8.3	Agente Bandeira	29
4	Resultados das Simulações	31
4.1	Cenários de Simulação	31
4.1.1	Primeiro Cenário	31
4.1.2	Segundo Cenário	32
4.1.3	Terceiro Cenário	33
4.1.4	Quarto Cenário	34
4.1.5	Tempo Médio	35
5	Conclusões	39

Listas de Figuras

3.1	Funcionalidades do Ambiente SIMULA	13
3.2	Interface do Ambiente SIMULA	16
3.3	Definição do Agente Soldado.	17
3.4	Definição do Agente Inimigo.	18
3.5	Definição do Agente Campo Minado.	18
3.6	Definição do Agente Bandeira.	19
3.7	Janela de Definição dos Comportamentos dos Agentes.	20
3.8	Janela de Definição das Regras de Comportamento dos Agentes.	21
3.9	Janela de Definição da Dimensão do Ambiente.	22
3.10	Janela de Definição da Distribuição dos Agentes no Ambiente.	23
3.11	Janela de Execução da Simulação Proposta neste Trabalho. .	24
3.12	Janela de Definição de Variáveis	25
3.13	Janela de Definição do Critério de Parada	27
3.14	Autômato para o Agente Soldado	28
3.15	Autômato para o Agente Inimigo	29
3.16	Autômato para Agente Bandeira.	29
4.1	Simulação Padrão Definida.	32
4.2	Resultado do Tempo Médio Total (em ciclos) de 10 Simulações nos 4 Cenários.	36
4.3	Resultados do Tempo Médio (em ciclos) para a Quantidade de Simulações que os Soldados Cumpriram a Missão.	37
4.4	Resultados do Tempo Médio (em ciclos) para a Quantidade de Simulações que os Soldados não Cumpriram a Missão. . . .	38

Listas de Tabelas

4.1	Resultado da Simulação com 10 Soldados.	33
4.2	Resultado da Simulação com 15 Soldados.	33
4.3	Resultado da Simulação com 20 Soldados.	34
4.4	Resultado da Simulação com 25 Soldados.	35
4.5	Valores do Tempo Médio (em ciclos) de 10 simulações.	35
4.6	Valores do Tempo Médio (em ciclos) para a Quantidade de Simulações que os Soldados Conseguiram Cumprir a sua Missão.	36
4.7	Valores do Tempo Médio (em ciclos) para a quantidade de Simulações que os Soldados não Conseguiram Cumprir a sua Missão.	37

Capítulo 1

Introdução

A área de Inteligência Artificial Distribuída (IAD) vem se destacando como geradora de teorias, metodologias e técnicas para o desenvolvimento de sistemas dinâmicos, com vários participantes e perspectivas, de grande escala e com interdependências mútuas. Como exemplo de sistemas com estas características citam-se Comércio Eletrônico (CE), Educação a Distância, sistemas de processamento de informação, etc.

Cada vez mais o uso de agentes está sendo inserida em atividades comerciais, pessoais, acadêmicas, etc.

Este relatório propõe um estudo dos Sistema Multiagentes com mais detalhes nos agentes reativos utilizando o ambiente SIMULA, mais especificamente o desenvolvimento de um Sistema Multiagente Reativo na simulação de uma "Querra".

Utilizaremos tanto a IAD com a tecnologia dos agentes como o Controle de Sistema a Eventos Discretos na modelagem do sistema.

No Capítulo 2 a área de Inteligência Artificial Distribuída é apresentada, destacando conceitos de agentes e Sistemas Multiagentes. No Capítulo 3 será apresentado o Ambiente SIMULA, que foi utilizado na simulação de agentes reativos e as regras de comportamentos dos agentes criados. No Capítulo 4 será apresentado os resultados das simulações. No Capítulo 5 será apresentado a conclusão.

Capítulo 2

Inteligência Artificial Distribuída

Estudos de Inteligência Artificial (IA) consideram como modelo de inteligência o comportamento individual humano, cuja ênfase é colocada na representação de conhecimento e métodos de inferência. Por outro lado, o modelo de inteligência utilizado pela Inteligência Artificial Distribuída (IAD) baseia-se no comportamento social, sendo a ênfase colocada em ações e interações entre agentes [13].

A IAD surgiu da junção entre as áreas de IA e Sistemas Distribuídos. Na IA a unidade de análise e desenvolvimento é um processo computacional com um único local de controle, um único foco de atenção e uma base de conhecimento, enquanto que a IAD analisa o desenvolvimento de grupos sociais de agentes que trabalham de forma cooperativa, objetivando resolver um determinado problema [9].

A IAD se diferencia da IA na medida em que traz novas e mais abrangentes perspectivas sobre representação do conhecimento, planejamento, resolução de problemas, coordenação, comunicação, negociação, etc [9]. Neste capítulo será apresentado uma visão geral sobre IAD, visando o embasamento teórico.

2.1 Agentes

Pode-se encontrar na literatura várias definições para o termo “agente”. Por exemplo, Etzioni e Weld definem um agente como “... um programa de computador que se comporta de forma análoga a um agente humano, tal como um agente de viagem ou um agente de seguro” [4]. Por sua vez, Russel e Norvig definem agente como “... qualquer coisa que pode perceber seu ambiente através de realizadores” [11]. É possível observar que a definição contextual de Etzioni e Weld tende a considerar os agentes como assistentes pessoais, enquanto a de Russel e Norvig apresenta uma abordagem geral e

didática [9].

Essa dificuldade em ter uma definição de agentes pode ser explicada pelas seguintes razões [9]:

- A interdisciplinariedade e a abrangência dos campos de pesquisa e comercial, nos quais a teoria de agentes pode ser estudada e aplicada;
- Agentes estão presentes no mundo real. Geralmente, os conceitos envolvidos com ambientes nos quais os agentes estão inseridos são imprecisos e incompletos. Esta característica torna difícil a categorização dos ambientes, e como consequência, também a categorização dos agentes que os representam.

Na tentativa de conceituar o termo agentes, em [14] há uma proposta de adoção das noções fraca e forte de agência. A noção fraca considera um conjunto de propriedades/atributos que um software ou hardware deve apresentar para ser considerado um agente. As propriedades consideradas em [14] são as seguintes [9]:

- Autonomia: Com esta propriedade agentes operam sem a intervenção direta de seres humanos ou outros sistemas, e têm algum tipo de controle sobre suas ações e estados internos;
- Habilidade Social: Com esta propriedade agentes interagem com outros agentes através de algum tipo de linguagem de comunicação de agentes;
- Reatividade: Com esta propriedade agentes percebem seu ambiente (mundo físico, a Internet, outros agentes, etc), e respondem a mudanças que ocorrem neste ambiente;
- Iniciativa: Com esta propriedade agentes não agem simplesmente em resposta a seus ambientes, mas tomam iniciativa e por isto são capazes de exibir comportamento orientado a objetivos.

A definição da noção forte de agência considera, além das propriedades definidas na noção fraca, outras características relacionadas às características humanas, como por exemplo estados mentais (conhecimento, crença, intenção, obrigação), emoção, racionalidade, etc.

Neste trabalho será adotada a definição de agentes proposta por [3], que considera agentes como:

Definição: "...resolvedores de problemas que podem trabalhar em conjunto para resolver problemas que estão além de suas capacidades individuais" [3].

Na IAD existe uma divisão inicial dos agentes em reativos e cognitivos. Essa divisão será destacada nas próximas seções.

2.1.1 Agentes Reativos

Um agente reativo não usa raciocínio simbólico complexo, estruturas de memória e uma representação interna explícita do conhecimento. Assim, não possui um histórico de suas ações passadas, nem pode fazer previsão de atos futuros. Com essas restrições este agente somente percebe o ambiente externo e, baseado nos estímulos do ambiente, reage de uma forma pré-determinada [9].

A comunicação dá-se de forma indireta, através do ambiente externo. Devido a esta arquitetura baseada em estímulo-resposta, agentes reativos não precisam de sofisticados mecanismos de coordenação e comunicação. O comportamento do sistema é explicado pela existência de uma inteligência emergente [9]. Ainda em [9] tem-se algumas vantagens na modelagem de agentes reativos como: economia cognitiva, robustez e tolerância a falhas.

2.1.2 Agentes Cognitivos

Agentes cognitivos são capazes de raciocinar a respeito de suas intenções e conhecimentos, criar planos de ação e executá-los. Possuem modelos explícitos do mundo externo, estruturas de memória que permitem manter um histórico de ações passadas e fazer previsões de ações futuras, e um sistema desenvolvido de cooperação e comunicação [9].

Uma característica que diferencia do estímulo-resposta é que os agentes cognitivos possuem controle deliberativo, na medida em que conseguem deliberar qual ação será executada, e posteriormente executá-la.

2.2 Sistemas Multiagentes

Pesquisas em IAD são freqüentemente classificadas em duas grandes áreas: Resolução Distribuída de Problemas (RDP) e Sistemas Multiagentes (SMA). Nesse dois casos os problemas são resolvidos de forma cooperativa e distribuída, usando processos denominados agentes. Nas próximas subseções apresenta-se com mais detalhes os Sistemas Multiagentes reativos e cognitivos.

2.2.1 Sistema Multiagentes Reativos

A abordagem reativa foi introduzida por Brooks [1], no domínio da robótica. Um exemplo de sistema multiagentes reativos é uma colônia de formigas, onde cada formiga é uma entidade simples. Mas uma colônia de formigas pode realizar trabalhos tais como: procura de alimentos, transporte do alimento até o formigueiro, defesa da colônia, etc. A execução destes trabalhos é complexa, muito embora a estrutura de cada formiga seja simples.

A seguir são destacadas algumas características de agentes e sistemas multiagentes reativos:

- Não há representação explícita de conhecimento. O conhecimento dos agentes é implícito e se manifesta através do seu comportamento;
- Não há representação do ambiente. O seu comportamento se baseia no que é percebido a cada instante do ambiente, mas sem uma representação explícita deste;
- Não há memória das ações. Os agentes reativos não mantêm um histórico de suas ações, de forma que o resultado de uma ação passada não exerce nenhuma influência sobre suas ações futuras;
- Organização etológica. A forma de organização dos agentes reativos é similar a de animais como insetos e microorganismos, em oposição à organização social dos sistemas cognitivos;
- Grande número de membros. Os sistemas multiagentes reativos têm, em geral, um grande número de agentes, de ordem de dezenas, centenas ou mesmo milhões de agentes [13].

Existem alguns ambientes de desenvolvimento para sistemas multiagentes reativos, como o sistema SEIEME - Simulateur Evènementiel Multi-Entités [8], o sistema Swarm [10] e o ambiente SIMULA [6].

2.2.2 Sistema Multiagentes Cognitivos

Os SMAs cognitivos são baseados em modelos organizacionais humanos, como grupos, hierarquias e mercados [13]. Segundo [5] apud [13] as principais características dos agentes cognitivos são:

- Mantém uma representação explícita de seu ambiente e de outros agentes da sociedade;
- Podem manter um histórico das interações e ações passadas;
- A comunicação entre agentes é feita através do envio e recebimento de mensagens;
- Seu mecanismo de controle é deliberativo, ou seja, tais agentes raciocinam e decidem sobre quais objetivos devem alcançar;
- Seu modelo de organização é baseado em sistemas sociológicos, como as organizações humanas;
- Uma sociedade contém tipicamente poucos agentes.

Organizações de Agentes

As organizações de agentes devem ser analisadas através da ocorrência de interações sociais. Estas interações permitem prever a formação dinâmica destas organizações.

Segundo [2] apud [13] existem duas classes de modelos para as interações sociais:

- Modelos descendentes (top-down);
- Modelos ascendentes (bottom-up).

Os modelos descendentes levam em consideração que de princípio os agentes já têm um problema a resolver. Tendo a cooperação pré-estabelecida como uma hipótese de partida, as organizações sociais são limitadas por uma organização pré-existente, guiando os agentes para atingir seus objetivos. Neste modelo, a alocação de tarefas pode ser feita de modo estático ou dinâmico. As desvantagens desta abordagem são a ausência de uma perspectiva dinâmica, um subjetivismo social e uma ênfase na comunicação como motor das interações sociais. Estes modelos são utilizados em RDPs.

Nos modelos ascendentes os agentes não têm necessariamente um objetivo a atingir. As interações sociais são estabelecidas dinamicamente como forma de atingirem seus próprios objetivos. Estes modelos são mais utilizados em um SMA.

Interação entre Agentes

Grande parte dos trabalhos em interações entre agentes cognitivos baseia-se na teoria de atos de fala (do Inglês speech act theory) [12] apud [13]. Esta teoria propõe uma categorização de primitivas de comunicação que são associadas às suas consequências. Alguns exemplos são: inform, ask-about, answer, promise, propose, etc. A linguagem KQML é uma linguagem baseada em atos de linguagem, é uma das mais utilizadas na área de IAD.

Um ponto importante e freqüentemente abordado nos atos de fala são os protocolos de interação. Este protocolos servem para estruturar as trocas de mensagens entre agentes e são concebidos com um caráter genérico, pois podem ser utilizados em outras aplicações.

Existem algumas ferramentas para prover a comunicação entre os agentes cognitivos, pode-se citar o SACI (Simple Agent Communication Structure). Na próxima subseção será apresentada algumas características dessa ferramenta.

SACI (Simple Agent Communication Structure)

O desenvolvimento de Sistemas Multiagentes normalmente requer uma comunicação entre agentes em um ambiente distribuído, o que não é uma tarefa

elementar, pois é necessário conhecimento sobre protocolos de rede, portas TCP/IP, programação distribuída e algumas outras tecnologias.

Saci é uma ferramenta desenvolvida no intuito de facilitar a comunicação entre agentes distribuídos. Trata-se de um conjunto de classes e facilidades que podem ser usados para ajudar no desenvolvimento de sociedades de agentes distribuídos. Ele possui como principais características:

- Procedimentos para a composição, envio e recebimento de mensagens KQML;
- Pelo fato dos agentes serem identificados por um nome, eles podem enviar mensagens para outros agentes apenas utilizando o nome do receptor. Este serviço é conhecido como “serviço de páginas brancas” ou “serviço de nome de agente”.
- Agentes podem conhecer outros através do “serviço de páginas amarelas”, onde podem registrar seus serviços com um facilitador e perguntar para o mesmo encontrar quais serviços são oferecidos por cada agente;
- Os agentes podem ser implementados como applets e rodarem em browsers;
- Os agentes podem ser ativados remotamente;
- Os agentes podem ser monitorados em eventos sociais, tais como: entrada, saída, envio e recebimento de mensagens.

Capítulo 3

Modelagem e o Ambiente Simula

O ambiente SIMULA é um ambiente desenvolvido em Java que é utilizado no desenvolvimento de aplicações com o uso de agentes reativos.

3.1 Descrição do Problema

O problema apresentado tem como objetivo simular um ambiente de guerra onde teremos quatro agentes reativos que irão se interagir, tentando simular a forma real de um combate em uma guerra. Nesta simulação teremos o soldado (agente), inimigo (agente), campo minado (agente) e a bandeira (agente). Os agentes irão se interagir e dependendo das suas posições podemos visualizar se o soldado ou o inimigo vencerá o combate. Cada agente criado possui regras definidas dependendo de várias situações que podem ocorrer durante essa simulação, nas próximas seções será mostrado cada regra determinada para cada agente, assim como as suas ações.

3.2 Objetivo

O objetivo da simulação é a interação entre os agentes criados e a visualização de todas as possibilidades de vitória tanto do agente soldado como a do inimigo. O agente soldado tem como finalidade "pegar" todas as bandeiras, com isso estará cumprindo a sua tarefa imposta nessa guerra, pois cada bandeira pode significar uma conquista de um território, destruição do campo inimigo, etc. Para o agente inimigo o objetivo é atrapalhar o agente soldado nessa conquista, pois no ambiente da simulação possuem agentes campo minado, com o objetivo de destruir a conquista do soldado na busca pela vitória.

3.3 Funcionalidades do Ambiente

Para que o usuário possa criar uma simulação utilizando o ambiente SIMULA é necessário que ele define uma situação inicial da sua aplicação e as regras de comportamentos dos agentes criados. Realizando essas definições, o ambiente se encarrega de executar a simulação definida pelo usuário.

A Figura 3.1 ilustra um esquema de como o usuário do ambiente SIMULA deve proceder para desenvolver suas aplicações em sistemas multagentes reativos e de quais são as etapas internas de execução até ser atingida uma solução (situação final) para a aplicação descrita [6].

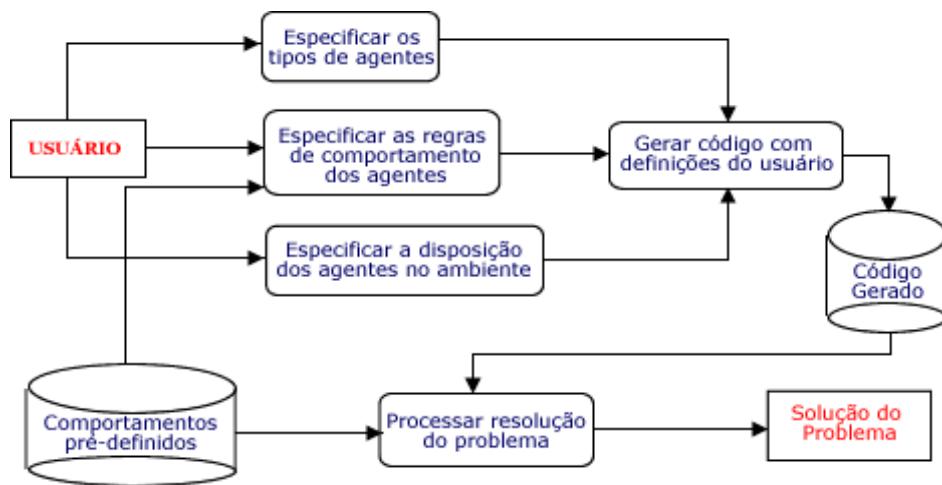


Figura 3.1: Funcionalidades do Ambiente SIMULA.

Observa-se pela Figura 3.1 que o ambiente esta dividido em dois módulos: definição e execução. Esses módulos serão descritos a seguir.

3.3.1 Módulo de Definição

Neste módulo são especificados os tipos de agentes, suas regras de comportamento e a disposição dos agentes no ambiente.

- Especificação dos tipos de agentes: Nesta especificação o usuário define quais serão os agentes que será utilizado na simulação, cada agente terá um nome, uma figura, uma quantidade e a área de percepção do agente.
- Especificação das regras de comportamento dos agentes: essas regras são especificadas para cada tipo de agente e tem como objetivo guiar as ações dos agentes na simulação. Os comportamentos dos agentes do ambiente SIMULA são limitados e divididos em três

tipos: comportamentos ativos, comportamentos passivos e comportamento de estado. Os comportamentos são pré definidos, cabendo ao usuário apenas atribuí-los a cada agente de acordo com a aplicação. A definição de comportamentos aos agentes é feita através de regras que determinarão as ações a serem executadas pelos agentes, bem como sob quais condições [7].

- Especificação da disposição dos agentes no ambiente: possibilita ao usuário especificar a posição inicial de cada agente no ambiente.

3.3.2 Módulo de Execução

Após o usuário realizar todas as definições necessárias para a simulação, pode-se acionar o módulo de execução. Este módulo divide-se em duas partes:

- Geração de Código: as definições e os comportamentos pré-definidos do ambiente feitas pelo usuário para a simulação dos agentes reativos, serão utilizados para gerar um código em Java que será responsável pelo processo de simulação;
- Execução: depois do código gerado, o processo de execução é mostrado da forma gráfica, onde os agentes são visualizados e suas regras de comportamentos são acionadas.

3.4 Comportamento Pré-definidos do Ambiente

Os comportamentos são definidos pelo usuário, se mais de um agente tiver o mesmo comportamento, eles agirão da mesma forma.

Há três classes de comportamentos pré-definidos no ambiente SIMULA e que são usados para montar as regras de comportamento dos agentes [6]:

- Comportamentos Ativos: são comportamentos que ativam uma ação do agente em seu ambiente, causando uma mudança de estado desse agente;
- Comportamento Passivo: é um comportamento associado a um comportamento de movimento;
- Comportamentos de Estado: são comportamentos relacionados à percepção do agente e ao seu estado atual, e que podem condicionar a execução de uma determinada ação pelo agente.

3.5 Regras de Comportamento

Para cada regra de comportamento deverá ser especificado [7]:

- 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>

Deve-se lembrar que em cada tipo de agente poderá ter várias regras de comportamento.

Os elementos que compõem as regras de comportamento pode ser descrito da seguinte forma:

- pré-condição: serve para em um determinado estado o agente possa decidir ou não em executar a regra de comportamento associada a ele. Os usuários podem utilizar, expressões de comparação, os comportamentos de estado pré-definidos, os conectivos lógicos AND, OR, NOT e operadores relacionais;
- ação-ativada: se a pré-condição for verdadeira a ação ativada irá representar o(s) comportamento(s) a ser(em) executado(s). O usuário poderá utilizar os comportamentos ativos pré-definidos, caso haja mais do que um comportamento a ser ativado o usuário deverá utilizar o conectivo AND.
- ação-condicional: representa uma condição dentro da execução da ação ativada. Nesta versão do SIMULA esta opção não está disponível.
- pós-condição: são os efeitos causados pela execução dos comportamentos. O usuário pode efetuar alterações ou atualizações de variáveis. São permitidos comandos de atribuição e expressões aritméticas [7];
- prioridade: o usuário define a ordem de execução das regras. É determinada por um valor numérico, a prioridade maior é 0.

3.6 Interface do Ambiente SIMULA

A interface do SIMULA á princípio é bem simples, é constituída de um menu com quatro opções (Arquivo, Configurações, Executar, Ajuda), através dos quais o usuário poderá utilizar todo o recurso disponível nesse ambiente para a simulação de agentes reativos, a interface do SIMULA pode ser observada na Figura 3.2.

Estas opções são descritas da seguinte forma:

- Arquivo: Possui funções como Novo, Abrir, Salvar, Salvar Como e Sair. Serve para manipulação de arquivos.

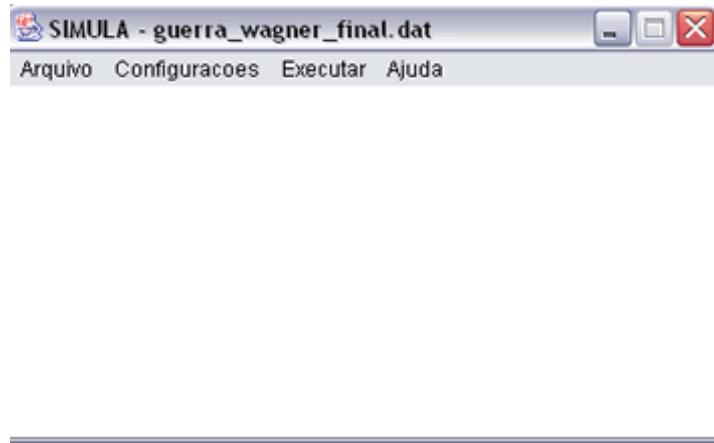


Figura 3.2: Interface do Ambiente SIMULA

- Configurações: Possui funções como Definições de Agentes, Definição de Variáveis, Definição de Comportamento, Definição de Critério de Parada, Dimensão do Ambiente e Distribuição de Agentes. Nessas opções do menu o usuário define todas as características de cada agente criado.
- Executar: Possui duas funções como Geração de Código e Executar. Estas opções do menu são utilizadas para geração do código dos agentes criados e possibilita a execução da modelagem definida pelo usuário.
- Ajuda: Não está disponível nesta versão do ambiente SIMULA.

3.6.1 Restrições do Ambiente

Como se trata de uma primeira versão do ambiente SIMULA é natural que se encontre algumas restrições, podemos relatar os seguintes problemas encontrados durante a simulação proposta:

- Podemos observar que quando estivermos com um arquivo do SIMULA com extensão .dat aberto, se selecionarmos a opção Abrir do menu de Arquivos fará com que esse arquivo aberto seja sobreescrito por um outro arquivo novo. Se não salvar as alterações, todas as alterações feitas, serão perdidas. Uma solução seria a criação de uma janela avisando ao usuário se ele gostaria de salvar as alterações antes de abrir um novo arquivo.
- A opção gerar código no menu Executar em algumas vezes gera o código mesmo tendo erros na simulação, ou em alguns casos o código

não é gerado travando o SIMULA, obrigando o usuário a reiniciar o software.

3.6.2 Janela de Definição de Agentes: Primeira Etapa

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 [7]. Nesta modelagem serão definidos quatro agentes (soldado, inimigo, campo minado e bandeira) como mostra as Figuras 3.3, 3.4, 3.5 e 3.6.

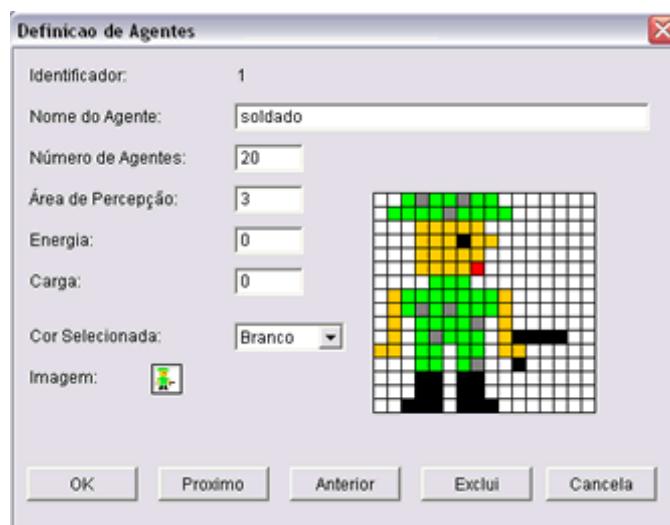


Figura 3.3: Definição do Agente Soldado.

Como foi mostrado nessas figuras a definição de cada agente depende do nome do agente, número de agentes, área de percepção, energia e carga.

A área de percepção é um valor limitado pelas dimensões do ambiente, que neste caso é 25. Se o agente tiver uma área de percepção igual a zero, significa que não tem percepção alguma no ambiente e se tiver uma área de percepção igual a 25, significa que percebe todas as posições no ambiente. Além disso, é possível editar uma figura e selecionar uma cor para o agente. Isto é útil para identificar os diferentes tipos de agentes durante o processo de execução da simulação. O agente também pode possuir uma energia e uma carga associada, que poderão ser alteradas pelo usuário nas regras de comportamento do agente em questão [7].

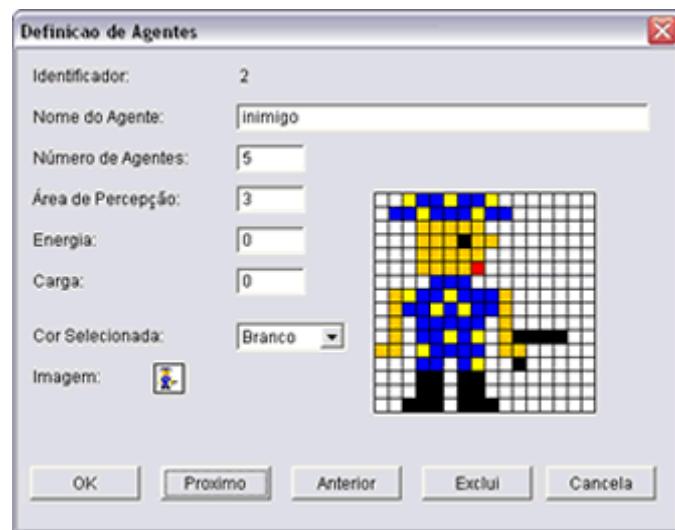


Figura 3.4: Definição do Agente Inimigo.

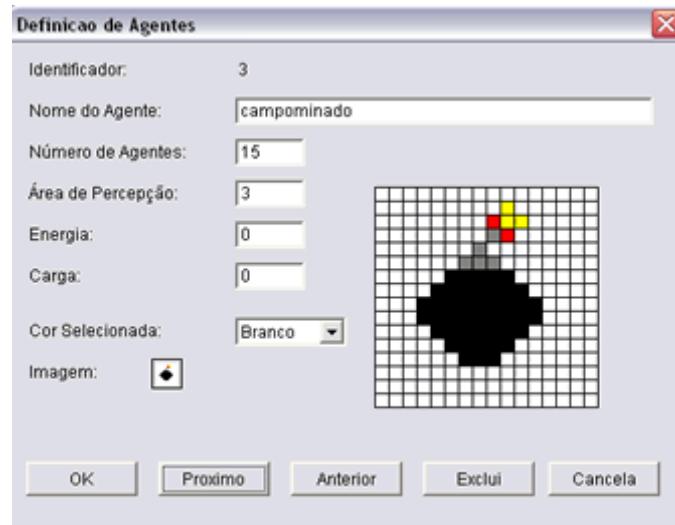


Figura 3.5: Definição do Agente Campo Minado.

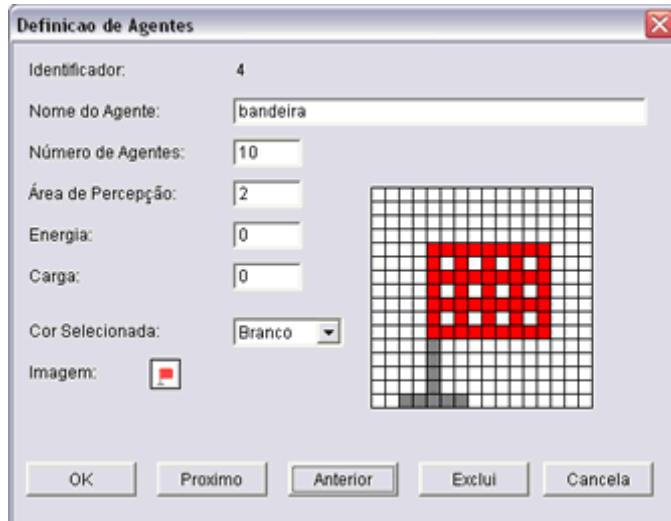


Figura 3.6: Definição do Agente Bandeira.

3.6.3 Janela de Definição de Variáveis: Segunda Etapa

A próxima etapa de definição da aplicação é a definição de variáveis que podem ser utilizadas nas regras de comportamento ou no critério de parada. Na simulação apresentada nesse relatório a variável é utilizada como critério de parada. A Figura 3.12 mostra a janela de definição de variáveis.

3.6.4 Janela de Definição dos Comportamentos dos Agentes: Terceira Etapa

O usuário poderá definir as regras de comportamentos de cada agente criado. Essas regras determinarão as ações a serem executadas pelos agentes e as suas condições. A janela de definição de comportamentos é mostrada na Figura 3.7. O usuário seleciona cada tipo de agente, definido anteriormente, a especificação da prioridade da regra e a criação da regra de comportamento. As opções das regras de comportamento foram descritas na seção 3.5.

Para criar uma regra, o usuário deverá selecionar o botão de alterar para cada um dos elementos que compõem uma regra. A Figura 3.8 mostra a segunda janela na definição dos comportamentos. Podemos observar que existem várias opções a serem definidas. Na opção comportamento o usuário seleciona o comportamento desejado para o agente, podemos citar algumas regras:

- percebe_agente
- percebe_pista

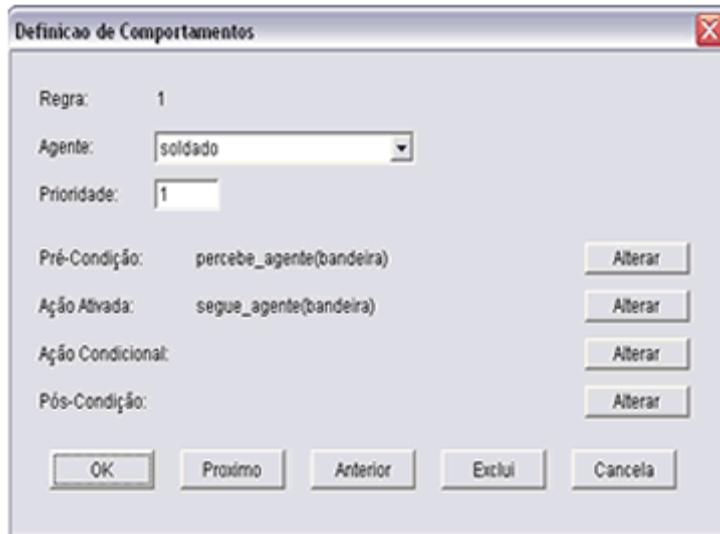


Figura 3.7: Janela de Definição dos Comportamentos dos Agentes.

- atinge_agente
- atinge_pista

Na opção agente objetivo corresponde ao nome do tipo de agente que será utilizado como parâmetro do comportamento [7]. Pode-se ainda utilizar as variáveis que foram definidas para serem utilizadas, por exemplo, no critério de parada.

Quando for escolhida a regra o usuário deverá selecionar o botão adicionar para que a regra seja atualizada na tela.

Os botões início de bloco e fim de bloco são utilizados, respectivamente, para a abertura e fechamento de parênteses na definição das regras. O botão remover anterior é utilizado para apagar os caracteres da regra que esta sendo criada.

Existe uma lista de comportamentos para que usuário possa saber a sintaxe das regras e também para que ele saiba os comportamentos disponíveis no ambiente.

3.6.5 Janela de Definição do Critério de Parada: Quarta Etapa

Para se ter o final da execução da simulação é necessário definir um critério de parada. Para a sua definição podem ser utilizadas as variáveis definidas pelo usuário, que nesta simulação as variáveis criadas são band e sold. Pode-se ainda montar um critério de parada composto, unindo as condições por AND (&&). Na Figura 3.13 mostra a janela de definição do critério de parada.

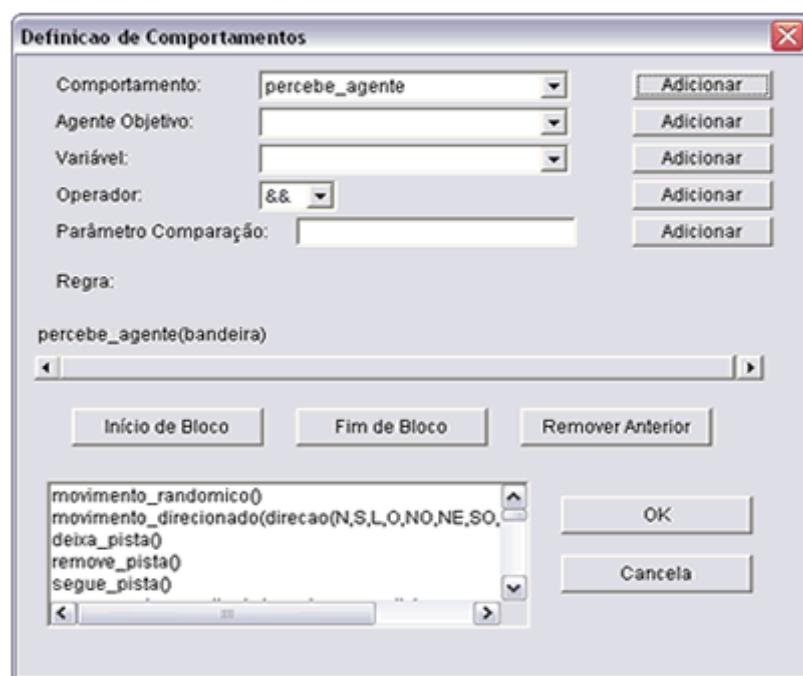


Figura 3.8: Janela de Definição das Regras de Comportamento dos Agentes.

Para que os critérios de parada sejam atualizados na tela é necessário pressionar o botão adicionar.

3.6.6 Janela de Definição das Dimensões do Ambiente: Quinta Etapa

Nesta etapa o usuário poderá definir o número de linhas e de colunas da matriz do ambiente de atuação dos agentes. O valor padrão da janela é 25x25. Na Figura 3.9 podemos visualizar a janela de definição das dimensões do ambiente.

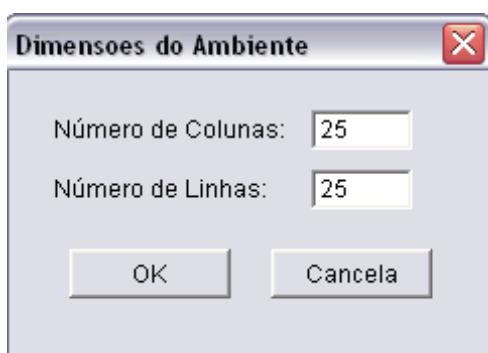


Figura 3.9: Janela de Definição da Dimensão do Ambiente.

3.6.7 Janela de Definição da Distribuição dos Agentes no Ambiente: Sexta Etapa

O usuário poderá definir uma posição inicial para cada um dos agentes criados. Essa posição deve respeitar os limites da matriz do ambiente dos agentes. Para os agentes que não tiverem uma posição inicial definida pelo usuário, será gerado posições aleatórias para os mesmos. Essa janela pode ser observada na Figura 3.10.

3.6.8 Janela de Execução

Após o usuário realizar todas as definições para a criação da simulação com os agentes reativos, é necessário gerar o código através do menu executar. Gerado o código o usuário poderá executar a simulação através do menu executar. Na Figura 3.11 ilustra a simulação proposta nesse trabalho, nesta figura pode-se verificar as variáveis e seus valores, o número de agentes de cada tipo e o número de ciclos executados (corresponde a execução de uma regra de comportamento por cada tipo de agente).

Existem alguns botões que possuem os seguintes significados:

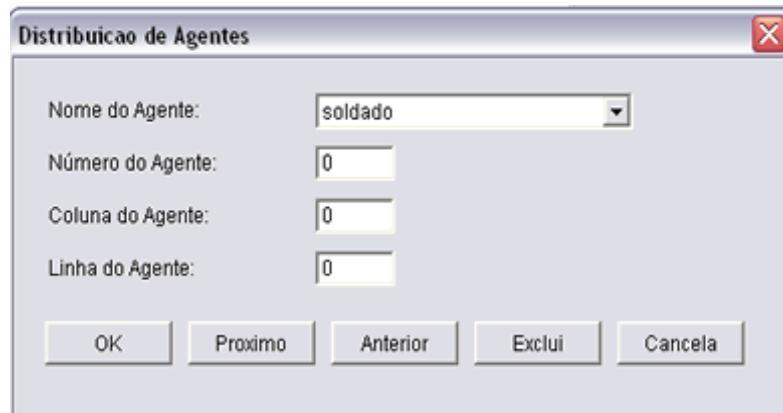


Figura 3.10: Janela de Definição da Distribuição dos Agentes no Ambiente.

- Iniciar: inicia a execução da simulação.
- Parar: encerra a execução da simulação.
- Reiniciar: reinicializa a execução da simulação.
- Passo: executa passo-a-passo a simulação
- Sair: volta para o ambiente

3.7 Definição dos Agentes para a Modelagem

A seguir apresenta-se a definição dos agentes, segundo especificações apresentadas no ambiente SIMULA.

Nome do Agente: Soldado

Número de Agentes: 20

Área de Percepção: 3

Energia: 0

Carga: 0

Nome do Agente: Bandeira

Número de Agentes: 10

Área de Percepção: 3

Energia: 0

Carga: 0

Nome do Agente: Inimigo

Número de Agentes: 10

Área de Percepção: 3

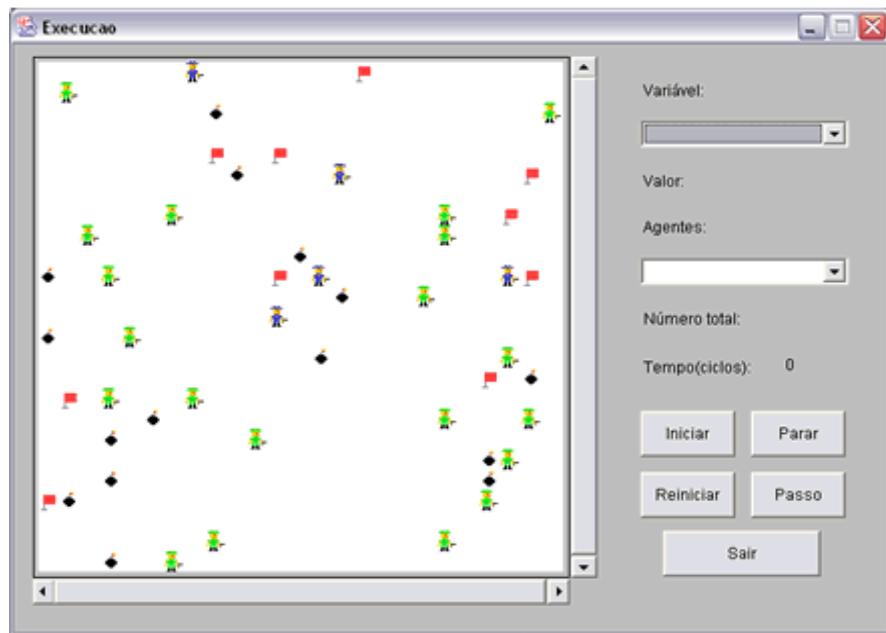


Figura 3.11: Janela de Execução da Simulação Proposta neste Trabalho.

Energia: 0

Carga: 0

Nome do Agente: Campo Minado

Número de Agentes: 8

Área de Percepção: 3

Energia: 0

Carga: 0

3.7.1 Variável Global

A criação e atribuição inicial de valores das variáveis globais são feitas na "Janela de definição de variáveis", como mostrado na Figura 3.12. Nesta simulação foram utilizadas duas variáveis globais para controle de parada da simulação.

As variáveis criadas são:

- band: Esta variável representa o número de bandeiras que já foram capturadas pelo agente soldado durante a execução da simulação. No início o seu valor é 10 pois a medida que o soldado vai capturando essa bandeira o seu valor vai sendo decrementado até chegar a zero que é uma das condições de parada da simulação.

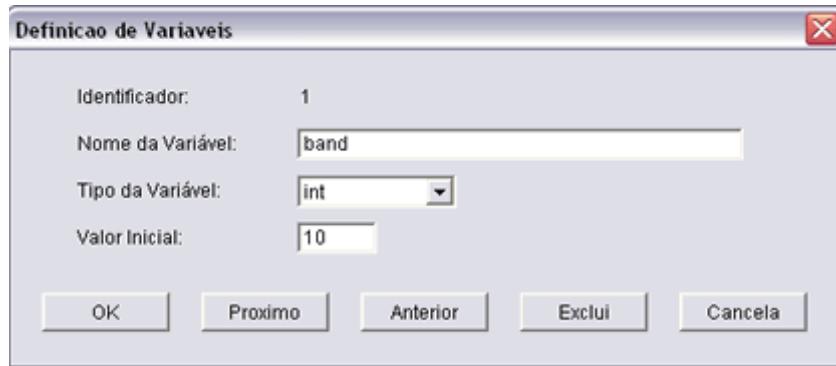


Figura 3.12: Janela de Definição de Variáveis

- sold: Esta variável representa o número de soldados que já foram mortos pelo agente campo minado durante a execução da simulação. No início o seu valor é definido dependendo do número de soldados, pois a medida que o agente campo minado mata um soldado o seu valor vai sendo decrementado até chegar a zero que é uma das condições de parada da simulação.

3.7.2 Regras de Comportamento

As regras de comportamento definem os comportamentos ativos dos agentes.

Neste trabalho utilizaremos uma abordagem baseada em comportamentos, cada comportamento e pré-definidos pelo usuário que pretende fazer a simulação dos agentes reativos, não será utilizado uma abordagem lógica, não serão levados em consideração o conhecimento sobre onde os soldados e os inimigos estão, sistemas de visão completos, etc.

Como foi mencionado foram definidas cinco regras de comportamento.

Regra 1 - Soldado Percebe Agente Bandeira

Descrição: Esta regra é utilizada quando um soldado percebe o agente bandeira, fazendo com que siga o agente que ele percebeu.

Prioridade: 1

Pré-Condição: pergebe_agente(bandeira)

Ação Ativada: segue_agente(bandeira)

Regra 2 - Soldado não Percebe Agente Bandeira

Descrição: Esta regra é utilizada quando um soldado não percebe o agente bandeira, fazendo com que ele possa se movimentar de forma randômica em busca de uma bandeira.

Prioridade: 0

Pré-Condição: (!percebe_agente(bandeira))

Ação Ativada: movimento_randômico()

Regra 3 - Bandeira Atinge Agente Soldado

Descrição: Esta regra é utilizada quando um soldado atinge o agente bandeira fazendo com que a bandeira seja capturada pelo soldado.

Prioridade:0

Pré-Condição: atinge_agente(soldado)

Ação Ativada: morte()

Pós-Condição: band=band-1

Regra 4 - Inimigo Percebe Agente Soldado

Descrição: Esta regra é utilizada quando um inimigo percebe o agente soldado, fazendo com que ele siga o agente soldado.

Prioridade:1

Pré-Condição: percebe_agente(soldado)

Ação Ativada: segue_agente(soldado)

Regra 5 - Inimigo Atinge Agente Soldado

Descrição: Esta regra é utilizada quando um inimigo atinge agente soldado, fazendo com que o inimigo seja morto.

Prioridade:0

Pré-Condição: atinge_agente(soldado)

Ação Ativada: morte()

Regra 6 - Soldado Atinge Agente Campo Minado

Descrição: Esta regra é utilizada quando o campo minado é atingido pelo agente soldado, fazendo com que o soldado seja morto.

Prioridade:0

Pré-Condição: atinge_agente(soldado)

Ação Ativada: mata_agente(soldado)

Pós-Condição: sold=sold-1

3.7.3 Critério de Parada

A simulação terminará quando todas as bandeiras forem capturadas ou os soldados serem mortos pelo campo minado. O critério de parada é definido na janela de definição do critério de parada no SIMULA, mostrado na Figura 3.13:

O critério é definido da seguinte forma:

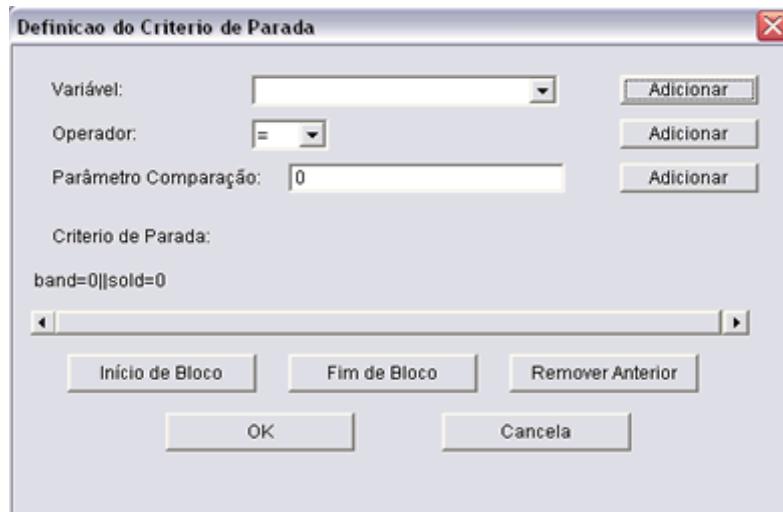


Figura 3.13: Janela de Definição do Critério de Parada

`band=0||sold=0`

Como ficou definido na regra de comportamento onde o agente bandeira é atingido pelo agente soldado ocasionando a captura da bandeira, nesta definição foi criada uma pós-condição que vai decrementando o número de bandeiras (`band=band-1`) esse número é definido dependendo da quantidade de bandeiras, ou seja, a medida que o soldado vai capturando a bandeira, poderá chegar em um momento onde o número de bandeiras chegará a zero, neste caso a simulação irá parar. O outro critério de parada foi definido na regra de comportamento onde o agente campo minado é atingido pelo agente soldado ocasionando a morte do soldado, nessa regra foi definido uma pós-condição que vai decrementando o número de soldados (`sold=sold-1`) que também é definida dependendo do número de soldados, ou seja, a medida que o soldado vai sendo morto pelo campo minado, chegará um momento onde o número de soldados chegará a zero, neste caso a simulação irá parar. Se o número de bandeiras ou (||) de soldados chegarem a zero a simulação será finalizada.

3.7.4 Algoritmo

```
se soldado percebe_agente (bandeira)
    então soldado segue_agente (bandeira)
    senão
        se soldado (!percebe_agente (bandeira))
            então movimento_radomico()
        se bandeira atinge_agente (soldado)
```

```

então morte(bandeira) band=band-1
se soldado atinge_agente(campo_minado)
então campo_minado mata_agente(soldado) sold=sold-1
se inimigo percebe_agente (soldado)
então segue_agente (soldado)
se inimigo (atinge_agente (soldado))
então morte(inimigo)

```

3.8 Modelagem por Autômatos

Utilizando a teoria dos autômatos e as técnicas de Sistemas a Eventos Discretos pode-se modelar todos os agentes criados na forma de autômatos para um melhor entendimento da simulação proposta.

3.8.1 Agente Soldado

O autômato criado para o agente soldado pode ser visualizado na Figura 3.14.

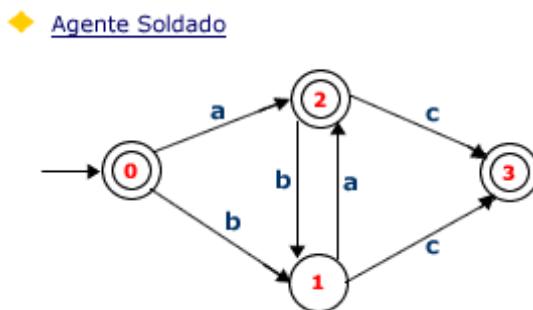


Figura 3.14: Autômato para o Agente Soldado

Para este agente foram criados os seguintes estados:

- 0: Parado.
- 1: Movimento Randômico.
- 2: Segue agente bandeira.
- 3: Morre agente soldado.

Os eventos são:

- a: agente soldado percebe agente bandeira.
- b: agente soldado não percebe agente bandeira.
- c: agente soldado atinge agente campo minado.

3.8.2 Agente Inimigo

O autômato criado para o agente inimigo é mostrado na Figura 3.15.

◆ [Agente Inimigo](#)

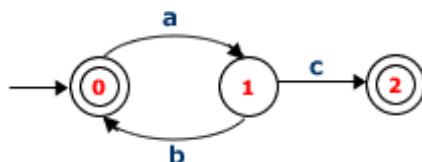


Figura 3.15: Autômato para o Agente Inimigo

Para este agente foram criados os seguintes estados:

- 0: Parado.
- 1: Segue agente soldado.
- 2: Morre agente inimigo.

Os eventos são:

- a: agente inimigo percebe agente soldado.
- b: agente inimigo não percebe agente soldado.
- c: agente inimigo atinge agente soldado.

3.8.3 Agente Bandeira

O autômato criado para o agente bandeira é mostrado na Figura 3.16.

◆ [Agente Bandeira](#)



Figura 3.16: Autômato para Agente Bandeira.

Para este agente foram criados os seguintes estados:

- 0: Parada.

- 1: Some agente bandeira

Os eventos são:

- a: o agente soldado atinge o agente bandeira.

Capítulo 4

Resultados das Simulações

Depois da definição de todas as regras mostradas no capítulo anterior para a simulação da "guerra" no ambiente SIMULA pode-se simular algumas possibilidades de vitória ou derrota dos soldados. Essa simulação possibilitará uma análise do tempo gasto (em ciclos), a quantidade de soldados para que ele consiga cumprir a sua missão, ou seja, capturar todas as bandeiras. Para cada cenário serão realizados 10 simulações. Nessa análise serão verificadas algumas possibilidades de vitória do soldado, aumentando ou diminuindo o seu número. A quantidade de bandeiras, inimigos, campo minado será fixo.

A Figura 4.1 mostra a execução da simulação padrão que tem como características:

- Número de Soldados: 10 (variável)
- Número de Bandeiras: 10 (fixo)
- Número de Inimigos: 5 (fixo)
- Número de Campo Minados: 8 (fixo)
- Dimensão do Ambiente: 25 x 25.

Será simulado 6 cenários, onde somente o número de soldados irá variar.

4.1 Cenários de Simulação

4.1.1 Primeiro Cenário

Neste cenário terá as seguintes características:

- Número de Soldados: 10 (variável)
- Número de Bandeiras: 10 (fixo)
- Número de Inimigos: 5 (fixo)

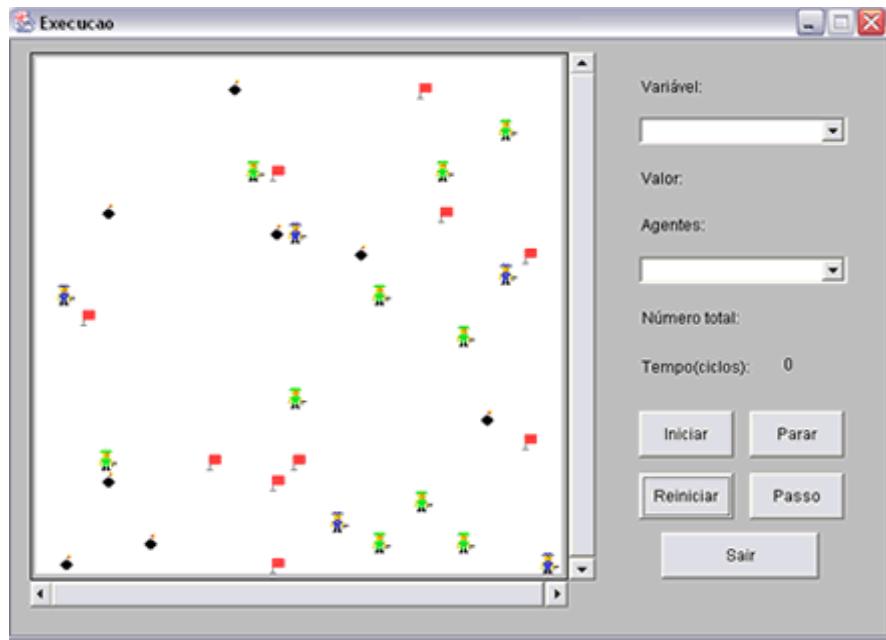


Figura 4.1: Simulação Padrão Definida.

- Número de Campo Minados: 8 (fixo)
- Dimensão do Ambiente: 25 x 25.

Os resultados obtidos neste cenário pode ser visualizado na Tabela 4.1 :

Neste cenário observa-se que existem apenas 2 soldados a mais do que o número de campos minados, existe uma dificuldade para que o soldado consiga cumprir a sua missão. O soldado conseguiu cumpri-la somente uma vez no conjunto de 10 simulações.

4.1.2 Segundo Cenário

Neste cenário terá as seguintes características:

- Número de Soldados: 15 (variável)
- Número de Bandeiras: 10 (fixo)
- Número de Inimigos: 5 (fixo)
- Número de Campo Minados: 8 (fixo)

Simulação	Nº Soldados	Tempo (Ciclos)	Cumpriu?
1	10	124	Não
2	10	268	Não
3	10	51	Não
4	10	217	Não
5	10	572	Não
6	10	35	Não
7	10	133	Não
8	10	127	Não
9	10	219	Não
10	10	17	Sim

Tabela 4.1: Resultado da Simulação com 10 Soldados.

Simulação	Nº Soldados	Tempo (Ciclos)	Cumpriu?
1	15	174	Não
2	15	100	Não
3	15	263	Não
4	15	13	Sim
5	15	106	Não
6	15	10	Sim
7	15	104	Não
8	15	112	Não
9	15	6	Sim
10	15	42	Sim

Tabela 4.2: Resultado da Simulação com 15 Soldados.

- Dimensão do Ambiente: 25 x 25.

Os resultados obtidos neste cenário podem ser visualizados na Tabela 4.2 :

Neste cenário a quantidade de soldados é sete vezes maior do que o número de campos minados, em comparação com a Tabela 4.1 a quantidade de soldados que conseguiram cumprir a sua missão aumentaram para quatro, devido ao aumento do número de soldados.

4.1.3 Terceiro Cenário

Neste cenário terá as seguintes características:

- Número de Soldados: 20 (variável)

Simulação	Nº Soldados	Tempo (Ciclos)	Cumpriu?
1	20	5	Sim
2	20	190	Não
3	20	149	Não
4	20	10	Sim
5	20	13	Sim
6	20	18	Sim
7	20	43	Sim
8	20	32	Sim
9	20	86	Não
10	20	295	Não

Tabela 4.3: Resultado da Simulação com 20 Soldados.

- Número de Bandeiras: 10 (fixo)
- Número de Inimigos: 5 (fixo)
- Número de Campo Minados: 8 (fixo)
- Dimensão do Ambiente: 25 x 25.

Os resultados obtidos neste cenário podem ser visualizados na Tabela 4.3 :

Em comparação com a Tabela 4.2 a quantidade de soldados que conseguiram cumprir a sua missão aumentaram para seis, devido ao aumento do número de soldados.

4.1.4 Quarto Cenário

Neste cenário terá as seguintes características:

- Número de Soldados: 25 (variável)
- Número de Bandeiras: 10 (fixo)
- Número de Inimigos: 5 (fixo)
- Número de Campo Minados: 8 (fixo)
- Dimensão do Ambiente: 25 x 25.

Os resultados obtidos neste cenário pode ser visualizado na Tabela 4.4:

Simulação	Nº Soldados	Tempo (Ciclos)	Cumpriu?
1	25	15	Sim
2	25	8	Sim
3	25	96	Sim
4	25	64	Não
5	25	183	Não
6	25	7	Sim
7	25	362	Não
8	25	12	Sim
9	25	39	Sim
10	25	3	Sim

Tabela 4.4: Resultado da Simulação com 25 Soldados.

Cenário	Número de Soldados	Tempo Médio (em Ciclos)
1	10	177,3
2	15	83
3	20	84,1
4	25	78,9

Tabela 4.5: Valores do Tempo Médio (em ciclos) de 10 simulações.

Em comparação com a Tabela 4.3 a quantidade de soldados que conseguiram cumprir a sua missão aumentaram para sete, devido ao aumento do número de soldados.

4.1.5 Tempo Médio

A partir dos resultados apresentados nas Tabelas 4.1, 4.2, 4.3, 4.4, 4.5 pode-se realizar o cálculo do tempo médio das simulações para cada cenário. Tais cálculos permitem determinar quanto tempo, em média, os soldados gastaram para realizar ou não uma missão. Pode-se calcular o tempo médio nos seguintes casos:

- **Tempo Médio Total:** Calcula-se o tempo médio para cada cenário. Na Tabela 4.5 verifica-se que a quantidade de soldados influenciaram no tempo médio para que o soldado consiga ou não cumprir a missão. Pode-se verificar que no cenário dois foi encontrado um tempo menor do que no cenário três, isso ocorre porque a posição inicial dos agentes também interfere na solução do problema, ou seja, a quantidade maior de soldados não garante que o tempo médio final da simulação seja menor. A Figura 4.2 mostra graficamente os resultados.

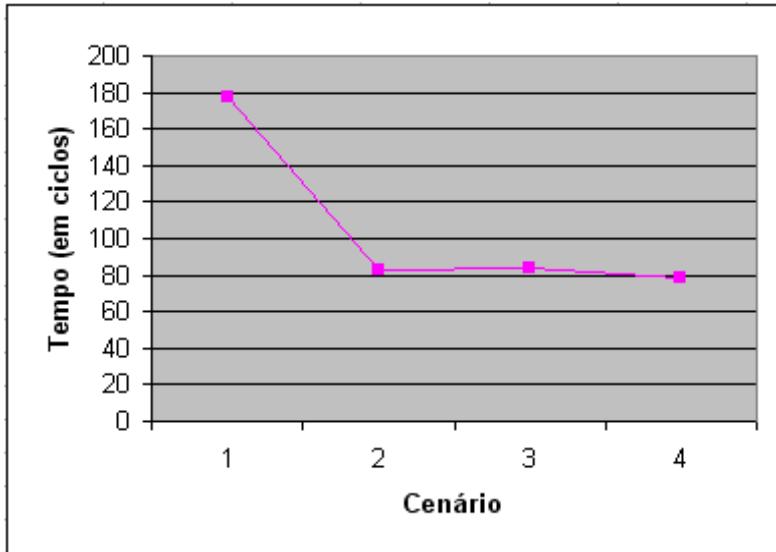


Figura 4.2: Resultado do Tempo Médio Total (em ciclos) de 10 Simulações nos 4 Cenários.

Cenário	Quantidade de Simulações que os Soldados Conseguiram Cumprir a Missão	Tempo Médio (em ciclos)
1	1	17
2	4	7,1
3	6	12,1
4	7	18

Tabela 4.6: Valores do Tempo Médio (em ciclos) para a Quantidade de Simulações que os Soldados Conseguiram Cumprir a sua Missão.

- **Tempo Médio para os Soldados que Cumpriram a Missão:** Observa-se na Figura 4.3 e na Tabela 4.6 a quantidade de simulações que os soldados conseguiram cumprir a sua missão está aumentando devido ao aumento da quantidade de soldados definidos em cada cenário.
- **Tempo Médio para os Soldados que não Cumpriram a Missão:** Observa-se na Figura 4.4 e na Tabela 4.7 a quantidade de simulações que os soldados não conseguiram cumprir a sua missão está diminuindo devido ao aumento da quantidade de soldados definidos em cada cenário.

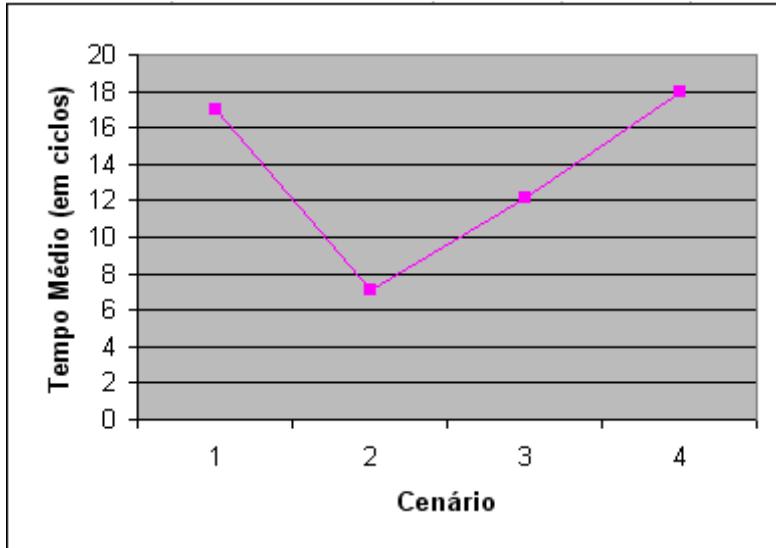


Figura 4.3: Resultados do Tempo Médio (em ciclos) para a Quantidade de Simulações que os Soldados Cumpriram a Missão.

Cenário	Quantidade de Simulações que os Soldados não Conseguiram Cumprir a Missão	Tempo Médio (em Ciclos)
1	9	175,6
2	6	85,9
3	4	72
4	3	60,9

Tabela 4.7: Valores do Tempo Médio (em ciclos) para a quantidade de Simulações que os Soldados não Conseguiram Cumprir a sua Missão.

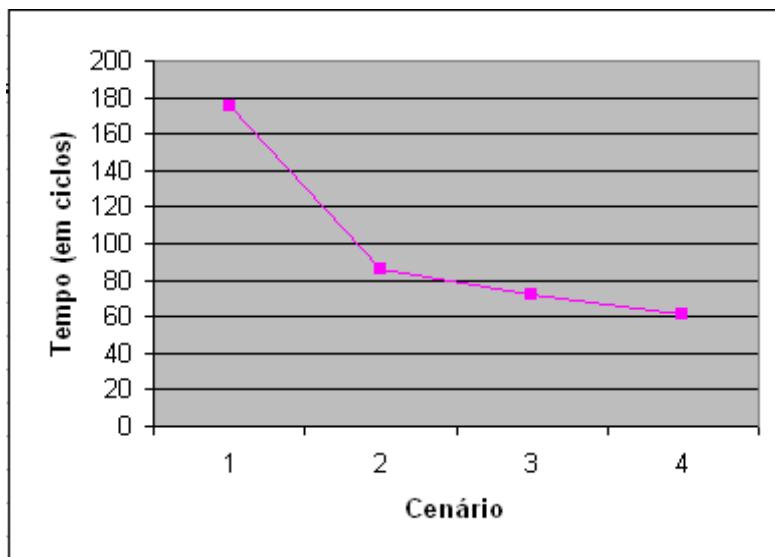


Figura 4.4: Resultados do Tempo Médio (em ciclos) para a Quantidade de Simulações que os Soldados não Cumpriram a Missão.

Capítulo 5

Conclusões

Nesta simulação utilizou-se o conceito de Sistemas a Eventos Discretos (SED) na modelagem de um Sistema Multiagente Reativo (SMA) utilizando como ferramenta o Ambiente SIMULA. A modelagem de cada agente foi feita por autômatos que na realidade é um dos assuntos abordados na teoria de SED. Foi modelado um ambiente de guerra onde quatro agentes reativos (soldado, inimigo, bandeira e campo minado) se interagem, tentando simular um combate em uma guerra.

Foram testados vários cenários para verificar as possibilidades de vitória do agente soldado e do inimigo. Para cada cenário foram aumentados os números de soldados para verificar o tempo gasto na simulação não importando se a vitória foi do soldado ou do inimigo. Podemos concluir que a quantidade de soldados não garante a sua vitória, pois depende da distribuições dos agentes definidos na simulação, que é feita de forma aleatória para cada cenário.

Tentou-se levantar os dados estatísticos com base nos autômatos criados, mas não obtivemos bons resultados. Como trabalho futuro pode-se tentar utilizar as cadeias de Markov na modelagem do problema apresentado.

Bibliografia

- [1] R. Brooks. A Robust layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, São Francisco, USA, 1986.
- [2] C. C. Conte, R. Mind is not enough: precognitive bases of social interactions. Proc. of the 1st. Symposium of Simulating Societies, Guildford, UK, 1992.
- [3] L. V. R. C. D. D. Durfee, E. H. Trends in Cooperative Distributed Problem Solving. IEEE Transaction Knowledge Data Eng., 1989.
- [4] W. Etizioni. Intelligent Agents on the Internet: Fact, Fiction, and Forecast. IEEE Expert, 1995.
- [5] G. L. Ferber, J. Intelligence artificielle distribuée. XI International Workshop on Expert Systems and their applications, Avignon, France, 1991.
- [6] R. Fozza. SIMULA - um Ambiente para o Desenvolvimento de Sistemas Multiagentes reativos. CPGCC/UFRGS, RS/Brasil, 1997.
- [7] S. F. Fozza, R. Simula - Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos. Não Publicado, 1998.
- [8] L. Magnin. SIEME - Simulateur d'environnement pour systèmes multi-agents. <http://www-laforia.ibp.fr/magnin/these/sieme>, 1996.
- [9] M. G. B. Marietto. Definição Dinâmica de Estratégias Instrucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW. Tese de Doutorado, ITA, Brasil, 2000.
- [10] B. R. L. C. A. M. Minar, N. The SWARM Simulation System: A Toolkit for Building Multi-Agent Simulations. <http://www.santafe.edu/projects/swarm/overview/overview>, 1996.
- [11] N. P. Russel, S. Artificial Intelligence - a modern approach. Cours, 1995.
- [12] J. Searle. Speech acts. Cambridge, USA, 1969.

- [13] A. L. O. Sichman, J. S. Introdução aos Sistemas Muitagentes. UFRGS, Instituto de Informática and USP Escola Politécnica, Porto Alegre, RS and São Paulo, SP, 2001.
- [14] J. N. Wooldridge, M. Agents Theories, Application and Languages. 1st International Conference on MultiAgent Systems, São Francisco, USA, 1995.