

Centro Universitário do Leste de Minas Gerais - UnilesteMG

Curso de Computação - Sistemas de Informação

Relatório do Projeto Interdisciplinar (PI) / Pré-TCC

RoboCup Rescue

Aluno(s): **Tiago Faria Bicalho**

Período: **6**

Professor Orientador: **Francisco Reinaldo**

Cel. Fabriciano - MG

10 de outubro de 2009

Sumário

1	Introdução	1
2	Hipóteses	1
3	Objetivos	2
3.1	Gerais	2
3.2	Específicos	2
4	Justificativa	3
5	Revisão de Literatura	3
6	Desenvolvimento	6
6.1	Material e Métodos	8
7	Resultados	9
8	Considerações Finais	9

1 Introdução

A área de Sistemas Multiagentes (SMA) é um ramo de pesquisa da Inteligência Artificial (IA) dedicada a busca por métodos ou dispositivos computacionais que possuam ou simulem a capacidade humana de resolver problemas de forma colaborativa, pensar ou, de forma ampla, ser inteligente. SMA são sistemas compostos por múltiplos elementos computacionais interativos denominados agentes. Agentes são entidades computacionais com duas habilidades fundamentais tais como, decidir por si próprio o que devem fazer para satisfazer seus objetivos de projeto e interagir com outros agentes de forma social [4].

Os agentes em um sistema SMA comumente são apresentados sob a forma virtual em um ambiente de simulação. RoboCupRescue é um projeto voltado a competição de equipes heterogêneas de agentes homogêneos. Este projeto promove o desenvolvimento de soluções voltadas para a busca e salvamento em cenários de catástrofe, que tem como objetivo desenvolver agentes inteligentes e robôs capazes de interagir e atuar em cenários simulados de desastre, situações de calamidade possuem grande relevância social. O termo calamidade provém do latim calamidade, significando desgraça pública, catástrofe, flagelo [2]. Tais situações de catástrofe exigem ações rápidas e inteligentes, no intuito de minimizar seus efeitos, envolvendo equipes de resgate e salvamento compostas por seres humanos, que são expostos a estes ambientes hostis e, portanto corre risco de morte.

2 Hipóteses

Acredita-se que esta pesquisa é uma área de grande aplicação de sistemas artificiais, a qual poderá aprimorar ou solucionar problemas diversos, entre eles falhas em comunicação e melhorias em estratégias.

Afim de validar o projeto, participaremos de eventos cuja finalidade é promover o desenvolvimento de módulos de inteligência artificial agregados a simuladores onde após a participação poderemos avaliar o modelo e melhorá-lo, se for o caso.

3 Objetivos

3.1 Gerais

A Liga Rescue Simulation tem dois objetivos gerais. O primeiro visa o desenvolvimento de simuladores para formar a infra-estrutura de sistemas de simulação, e emular fenômenos reais predominantes em catástrofes. O segundo visa desenvolver agentes inteligentes capazes de interagir e atuar em cenários de desastre.

O objetivo primário desta pesquisa é construir uma ferramenta de aprendizagem para o diminuir o tempo de treinamento do agente; consequentemente atingir melhores desempenhos. Neste contexto, o projeto tem o propósito de adaptar e melhorar a arquitetura de aprendizagem RatoLIC, tornando-a robusta e com tomada de decisão autônoma, somado ao suporte para outros algoritmos de aprendizagem na elaboração de comportamentos do agente, assim resolvendo o problema manual da escolha da estratégia mais adequada.

3.2 Específicos

O objetivo do projeto visa desenvolver módulos de inteligência artificiais robustos capazes de se adaptar a qualquer ambiente.

Os objetivos secundários são:

- Escolha automática de estratégias utilizadas pelos agentes do RoboCupRescue;
- Redução do tempo de treinamento desses agentes;
- Redução da complexidade do problema;
- Desenvolvimento de uma arquitetura genérica, robusta e com tomada de decisão autônoma para um agente específico, fazendo ele atuar de forma autônoma.

4 Justificativa

O fato de buscas e salvamentos realizados em ambientes hostis por equipes de seres humanos incorrer em risco de morte a estes, e do mesmo modo, a necessidade de se desenvolver tecnologias que auxiliem ou mesmo executem toda a rotina de atendimento com autonomia, introduz uma série de temas avançados e interdisciplinares, tais como IA, comportamento estratégico(planejamento Multiagente, planejamento em tempo real, agentes heterogêneos).

5 Revisão de Literatura

O projeto RoboCup Rescue surgiu como idéia após o terremoto ocorrido em 17 de Janeiro de 1995 na cidade de Kobe, no Japão, onde cerca de 6500 pessoas morreram, 80.000 casas foram destruídas e aproximadamente 1 milhão de pessoas foram afetadas. Com isso, o RoboCup Rescue Simulation Project visa promover a pesquisa e desenvolvimento envolvendo trabalho coordenado multiagente. A meta da competição é conseguir salvar o maior número de vítimas de um desastre virtual simulado, através da ação em conjunto de 3 tipos de agentes (bombeiros, ambulâncias e policiais). Existem muitas restrições tais como número de mensagens limitado a base, carga de água de cada caminhão bombeiro e eventos como massas de ar se deslocando espalhando o fogo além da ocorrência de novos tremores que podem mudar o cenário, isto deixa a simulação ainda mais real e desafiadora, e com isso, exigindo um maior esforço dos participantes, o que acaba contribuindo ainda mais com o desenvolvimento de técnicas de inteligência artificial, robótica e sistemas multiagente [3].

O ambiente RoboCup Rescue possibilita por sua complexidade, a utilização de suas ferramentas para um estudo específico e aprofundado. O simulador de incêndios abre portas para aplicações diretas em diversos estudos entre eles o comportamento coletivo em relação a casos onde o pânico em grandes multidões toma conta da situação, o simulador após uma reconfiguração dos agentes civis é capaz de exibir com certo grau de perfeição o comportamento de uma multidão apavorada, que pode orientar pesquisadores a encontrar as melhores rotas de fuga em caso de incêndio. Assim podemos definir que este é o projeto que desencadeará uma forma completamente nova de se estudar o comportamento humano, e através dele podemos resolver os diversos problemas encontrados nas simulações realizadas com base em dados já existentes.

O termo “agente”, ou software agente, tem sido utilizado em uma série de tecnologias, por exemplo, em inteligência artificial, bancos de dados, sistemas operacionais e redes de computadores. Embora não exista uma definição única (RUSSEL; NORVIG, 2003), todas as definições concordam que: “Um agente é um componente de software especial que possui autonomia e que provê uma interface para um sistema arbitrário e/ou comporta-se como um agente humano.”[1]. Os agentes são compostos por características próprias onde estas definem o agente sendo ele bombeiro, polícia, ambulância ou civil. O bombeiro tem como característica relevante a quantidade de água no tanque, já a polícia os sensores para percepção dos materiais diversos que fazem a obstrução nas ruas, onde a sua função é exatamente retirá-los para que estes não atrapalhem os demais agentes a realizarem suas tarefas, a ambulância tem como característica o sensor auditivo, esta é capaz de ouvir os gritos dos agentes civis e assim encontrá-los para um futuro salvamento, ver Figura 1.



Figura 1: Visualizador.

Os agentes humanóides, também chamados de agentes locais, têm a capacidade de atuar de modo direto sobre o ambiente, e podem ser do tipo: civil, bombeiro, polícia e ambulância. Qualquer humanóide tem a capacidade de se comunicar e movimentar no espaço geográfico da simulação, cada agente possui capacidades específicas. O agente civil representa uma família ou indivíduo vítima da catástrofe, que eventualmente precisará ser socorrida (pelo agente ambulância). O agente bombeiro tem a ca-

pacidade de extinguir incêndios em edifícios, manipulando as mangueiras para definir o ângulo de saída e a quantidade de água. O agente ambulância representa a capacidade de salvamento de agentes humanóides (inclusive outras ambulâncias). Apenas o agente ambulância tem a capacidade de socorrer qualquer humanóide. Um agente não humanóide, também chamado de agente global, representa uma organização de agentes humanóides. Essas organizações são o quartel de bombeiros e a central de ambulâncias (dentre outros). Os agentes não humanóides são representados por construções e não atuam diretamente no ambiente, suas ações concretizam-se através dos agentes humanóides de que são compostos. O quartel de bombeiros coleta e integra toda a informação enviada pelos agentes bombeiro e os aloca de acordo com uma política simples assim como a esquadra de polícias o faz. A central de ambulâncias coleta e integra toda a informação enviada pelos agentes Ambulância e os aloca de acordo com uma política simples, assim como todos os agentes não humanóides os fazem.

Atualmente o simulador possui módulos compilados na linguagem C e Java. Estes se conectam ao kernel, ver Figura 2, que é responsável por controlar o ambiente de simulação.

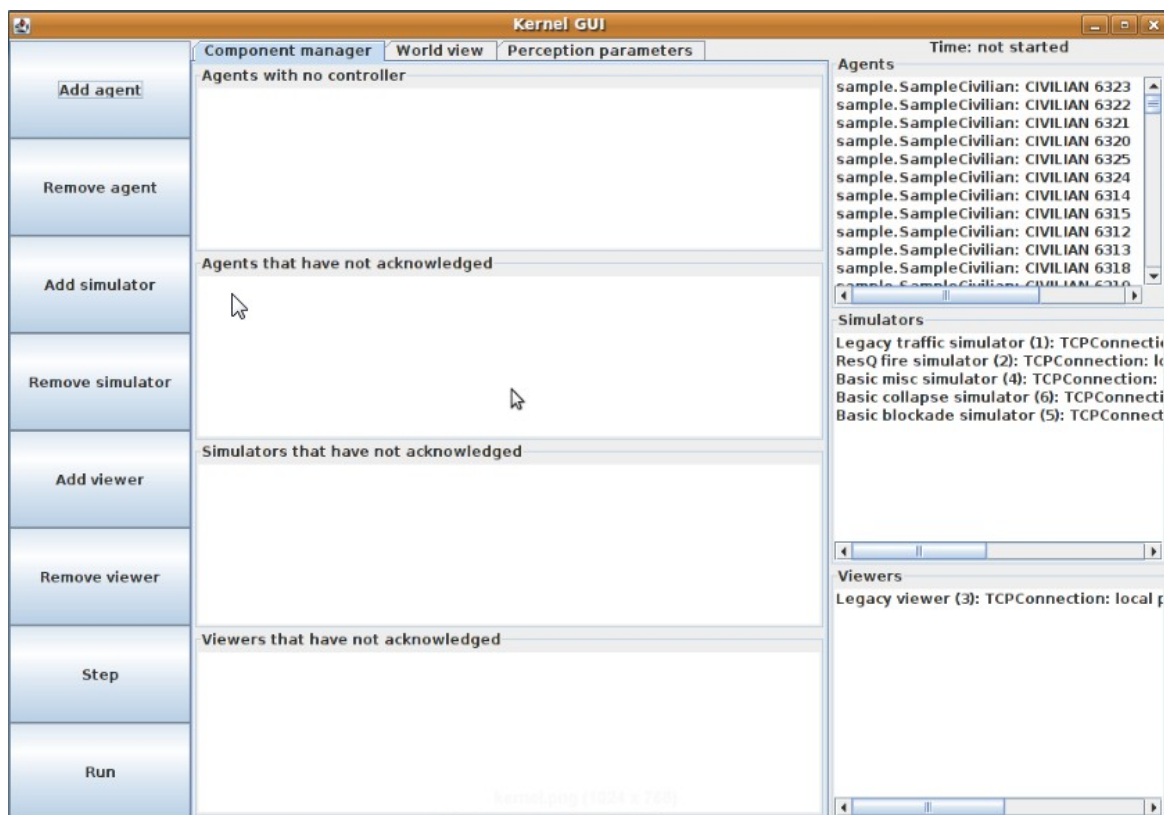


Figura 2: Kernel.

É através deste que poderemos inserir novos agentes, novos visualizadores, e alterar

padrões de sensibilidade dos agentes, cada um dos módulos é responsável por uma atividade, estas atividades são: Massa de ar em movimento que podem espalhar o fogo para outras edificações, tráfego de veículos e pessoas pelas vias, comportamento de cada uma das equipes e civis.

As regras do campeonato RoboCupRescue consistem basicamente na versão do simulador a ser utilizado, geralmente entregue pela organização da competição 4 meses antes da mesma, a quantidade de agentes em cada equipe, pontos de ignição, torres de comando de cada equipe além de refúgios, como segue tabela 1 abaixo.

Tabela 1: Configuração dos agentes.

Entidade	Mínimo	Máximo
Bombeiros	0	15
Polícia	0	15
Ambulância	0	8
Civil	70	90
Torre Central dos Bombeiros	0	1
Torre Central da Polícia	0	1
Torre Central das Ambulâncias	0	1
Refúgios	0	5
Pontos de Ignição	2	8

6 Desenvolvimento

Atualmente o simulador utilizado na competição se encontra na versão 1.0, este pode ser encontrado no site

<http://sourceforge.net/projects/roborescue/>

O simulador é capaz de inicializar todos os agentes em sua versão básica, mas estes não receberam nenhum núcleo de inteligência artificial para que estes possam executar de maneira satisfatória a atividade a ele referida. O desafio do Laboratório de Inteligência Computacional (LIC) do UnilesteMG é realizar o desenvolvimento e estudo de um núcleo de inteligência artificial capaz de superar os limites já estabelecidos em outras competições. O simulador obteve bons resultados durante a execução da aplicação gráfica apenas em dois sistemas operacionais da distribuição Linux, estes são Kurumin Linux 8.0 e Ubuntu. Informações sobre a instalação e execução do mesmo

são de grande valor, assim o acesso a estas informações é restrito uma vez que se trata de uma competição mundial.

A instalação procede da seguinte maneira. Abra o shell do Linux, precisaremos utilizar o apt-get para baixar alguns pacotes da internet, para que o apt-get possa funcionar no Unileste você precisa passar seu usuário, senha, url do proxy e porta a conectar.

Para isso primeiramente logue como root, então abra o shell e digite o comando:

```
export http_proxy="http://A06usuario-da-rede:senha-da-rede@10.2.0.1:3128",
```

```
Acquire{  
HTTP::proxy "http://USusuario-da-rede:senha-da-rede@10.2.0.1:3128";  
}
```

Então poderemos finalmente atualizar a lista do apt-get utilizando o seguinte comando.

```
apt-get update
```

Precisaremos instalar o Terminal X para que o simulador funcione corretamente, sendo que toda a programação é voltada a este tipo de terminal sendo impossível a execução em outro terminal sem que se altere a programação original da versão em questão Para isso digite o seguinte comando.

```
apt-get install xterm
```

Durante alguns minutos o sistema operacional irá acessar a internet a procura do pacote de instalação do Terminal X, ao encontrar este executará automaticamente a instalação, após o termino digite xterm Isso fará com que se abra outro shell de comando, feche o outro e utilizaremos a partir deste momento apenas o Terminal X, para dar continuidade ao processo de preparação do sistema para a execução do simulador precisaremos instalar um outro pacote denominado ANT, para isso digite:

```
apt-get install ant
```

Realizando esta atividade o Java6 será automaticamente incorporado ao sistema junto ao pacote ANT, após esta instalação acesse a pasta boot dentro da pasta descompactada da versão 1.0 do simulador. Com o terminal X apontando internamente a pasta boot digite:

```
demo.sh
```

Uma apresentação básica de funcionamento do simulador ocorrerá, note que este não tem inteligência artificial e os agentes executam as atividades de forma desorganizada.

6.1 Material e Métodos

O simulador foi instalado nas seguintes condições nas quais obtivemos sucesso.

Sistema Operacional: Kurumin Linux 8.0 ou Ubuntu

Processador: Core 2 Duo 2.2

Memória Ram: 1GB

HD: 20GB's

Monitor de 17"

Conexão com Internet

7 Resultados

O simulador é instalado e executado perfeitamente nas versões de sistemas operacionais Linux mencionadas, já em sistemas operacionais Windows a execução não ocorre corretamente, pois diversos problemas ocorrem com a comunicação do kernel do simulador com o sistema.

8 Considerações Finais

O Simulador é a melhor forma de se executar um treinamento das redes neurais embutidas em cada agente, uma vez que este software consegue efetuar simular uma catástrofe urbana com certa precisão.

A continuação do projeto se deve a implantação do módulos de inteligência artificial, o que ocorrerá nos próximos meses.

Referências

- [1] F.L Bellifemine. Developing multi-agent systems with jade. *Springer*, 2007. 4
- [2] A.B.H Ferreira. Novo aurélio século xxi: O dicionário da língua portuguesa. 1999. 1
- [3] RoboCupRescueHome. Building rescue systems of the future, 2009. 3
- [4] Mike Wooldridge. Agent-oriented software engineering ii. *Springer-Verlag Lecture Notes in Computer Science*, 2222, 2002. 1