

Projeto Interdisciplinar (PI) em L^AT_EX_{2 ϵ} : Um modelo de relatório para a academia

Francisco Reinaldo¹, Maria Tereza de Castro Costa², Tiago Faria Bicalho³, and Victor Vasconcelos Moreira⁴

Email 1reinaldo.opus@gmail.com, 2maryxb@gmail.com,
3tiagofariabicalho@gmail.com, 4victorvasconcelosfox@gmail.com

Resumo In this paper we present an example of technical report commonly used by students in order to present their academic research.

1 Introdução

O Projeto Interdisciplinar, do Curso de Computação - Sistemas de Informação, do Centro Universitário do Leste de Minas Gerais - UnilesteMG, consiste no desenvolvimento de uma produção acadêmica teórico-prática no formato de relatório por grupos de três alunos orientados por professores ao longo do semestre.

Preferencialmente, seu conteúdo deve ser um estudo de caso de uma organização da comunidade. Além de integrar disciplinas, este projeto aproxima o aluno da realidade do mercado de trabalho em computação.

O uso do L^AT_EX_{2 ϵ} veio para agilizar e padronizar a construção dos relatórios desse projeto, isto porque a partir do momento que se cria um “padrão de relatório” em L^AT_EX_{2 ϵ} , o usuário não precisará se preocupar com a formatação de sua produção, pois o próprio L^AT_EX_{2 ϵ} já faz isto com os seus pacotes e códigos apropriados para o tipo do documento.

Para este artigo, focalizamos no processo de como preencher o relatório com comandos em L^AT_EX_{2 ϵ} para a construção do PI dos alunos do UnilesteMG.

2 Objetivos

Esse artigo apresenta o tipo de modelo de leiaute no formato $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$ de um relatório do PI utilizado no curso de Computação - Sistemas de Informação, para facilitar a construção dos mesmos pelos alunos.

3 Desenvolvimento

Neste artigo, estaremos exibindo um modelo de relatório de PI utilizado no Curso de Computação - Sistemas de Informação e também no Laboratório de Inteligência Computacional (LIC).

Este relatório de PI é composto por vários arquivos `.tex`, e cada um com seu respectivo propósito. Para cada arquivo necessário, uma subseção é apresentada. Também explicaremos os comandos utilizados para a construção dos mesmos e as áreas que deverão ser alteradas para a customização do relatório de acordo com o PI de cada aluno.

3.1 O arquivo `principal.tex`

Para o desenvolvimento do proposto relatório, nós iniciamos com um arquivo principal chamado de `principal.tex`, Figura 1, que une todos os outros arquivos e é a origem de todo o relatório. Dentro do `principal.tex` encontramos os seguintes códigos:

```

1 \documentclass[12pt]{article}
2 \usepackage[final]{pdfpages}
3 \usepackage{preambulo}
4 \begin{document}
5 \thispagestyle{empty}
6 \begin{center}
7     \Instituicao{Centro Universitário do Leste de Minas Gerais - UnilesteMG}
8     \Curso{Curso de Computação - Sistemas de Informação}
9     \ProjetoPI{Relatório do Projeto Interdisciplinar (PI) / Pré-TCC}
10    \TituloProjeto{RoboCup Rescue}
11    \Alunos{Tiago Faria Bicalho}
12    \Periodo{6}
13    \Orientador{Francisco Reinaldo}
14    \LocalData
15 \end{center}
16 \sumarios
17 \onehalfspacing
18 \input{introducao.tex}
19 \input{hipoteses.tex}
20 \input{objetivos.tex}
21 \input{justificativa.tex}
22 \input{revisaoliteratura.tex}
23 \input{desenvolvimento.tex}
24 \input{material.tex}
25 \input{resultados.tex}
26 \input{consideracoesfinais.tex}
27 \pagebreak
28 \bibliographystyle{plain}
29 \bibliography{rescuebib}
30 \end{document}

```

Figura 1: Arquivo exemplo principal.tex.

Para customizar o relatório de acordo com o seu projeto, altere os seguintes campos do exemplo acima:

- Na linha 7, altere o texto que está entre { } e insira o nome da Instituição.
- Na linha 8, altere o texto que está entre { } e insira o nome do Curso.
- Na linha 9, altere o texto que está entre { } e insira o nome do Relatório.
- Na linha 10, altere o texto que está entre { } e insira o Título do Projeto.
- Na linha 11, altere o texto que está entre { } e insira o(s) Nome(s) do(s) Aluno(s).
- Na linha 12, altere o texto que está entre { } e insira o Período.

- Na linha 13, altere o texto que está entre { } e insira o nome do Professor Orientador do projeto.

Explicando o código contido na Figura 1, temos:

- A linha 1, `\documentclass[12pt]{article}`, define o tipo do documento {article} e o tamanho 12 do texto no documento.
- A linha 2, `\usepackage[final]{pdfpages}`, adiciona um pacote para inserção de PDF'S no documento quando se necessário.
- A linha 3, `\usepackage{preambulo}`, define o pacote que estamos utilizando para este documento (preambulo).
- A linha 4, `\begin{document}`, marca o início do documento.
- A linha 5, `\thispagestyle{empty}`, retira a numeração da página.
- A linha 6, `\begin{center}`, inicia a centralização do texto.
- A linha 7, `\Instituicao{Centro Universitário do Leste de Minas Gerais - UnilesteMG}`, adiciona o nome da Instituição ao documento. O nome da Instituição estará entre { }.
- A linha 8, `\Curso{Curso de Computação - Sistemas de Informação}`, adiciona o nome do Curso ao documento. O nome do Curso estará entre { }.
- A linha 9, `\ProjetoPI{Relatório do Projeto Interdisciplinar (PI) / Pré-TCC }`, adiciona o nome do Relatório ao documento. O nome do Relatório estará entre { }.
- A linha 10, `\TituloProjeto{Robocup Rescue}`, adiciona o Título do Projeto ao documento. O Título do Projeto estará entre { }.
- A linha 11, `\Alunos{Tiago Faria Bicalho}`, adiciona o(s) Nome(s) do(s) Aluno(s) ao documento. O Nome do Aluno estará entre { }.
- A linha 12, `\Periodo{6}`, adiciona o Período ao documento. O Período estará entre { }.
- A linha 13, `\Orientador{Francisco Reinaldo}`, adiciona o nome do Professor Orientador ao documento. O nome do Professor Orientador estará entre { }.

- A linha 14, `\LocalData`, adiciona a Data ao documento. Caso deseje alterar a data, localize-a no arquivo `preambulo.sty`. Para data corrente a geração do PDF, use `\today`.
- A linha 15, `\end{center}`, finaliza a centralização do texto.
- A linha 16, `\sumarios`, adiciona um Sumário ao documento.
- A linha 17, `\onehalfspacing`, implementa o espaçamento de um e meio no documento.
- A linha 18, `\input`, inclui um arquivo no documento.
- A linha 27, `\pagebreak` faz uma quebra de página.
- A linha 28, `\bibliographystyle{plain}`, define o estilo de bibliografia. O nome do estilo de bibliografia estará entre `{ }`.
- A linha 29, `\bibliography{rescuebib}`, insere a bibliografia. O nome da bibliografia estará entre `{ }`.
- A linha 30, `\end{document}`, finaliza o documento.

Quando convertermos o documento da Figura 1 para PDF, temos a versão de apresentação da capa e do sumário nas próximas duas páginas que se seguem:

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

3.2 O arquivo `introducao.tex`

O arquivo `introducao.tex`, Figura 2, contém os seguintes códigos:

```
1 \section{Introdução}
2
3 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 \cite{Wooldridge2002}.
4
5 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 \cite{Ferreira1999}. 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.
6
7 \endinput
```

Figura 2: Arquivo exemplo `introducao.tex`.

Explicando o código contido na Figura 2, temos:

- A linha 1, `\section{}`, divide o documento em Seções. O nome da Seção estará entre `{ }`.
- A linha 3, `\cite{}`, cria uma marca para citar um livro ou artigo descrito na bibliografia. O nome da Citação estará entre `{ }`.
- A linha 7, `\endinput`, marca o fim da inclusão de um arquivo no documento.

Para customizar o relatório, altere o texto do exemplo acima de acordo com a introdução do seu projeto.

3.3 O arquivo hipoteses.tex

O arquivo hipoteses.tex, Figura 3, contém os seguintes códigos:

```
1 \section{Hipóteses}
2
3 Acredita-se que esta pesquisa é uma área de grande aplicação de sistemas artificiais, a qual poderá aprimorar
4 ou solucionar problemas diversos, entre eles falhas em comunicação e melhorias em estratégias.
5
6 Afim de validar o projeto, participaremos de eventos cuja finalidade é promover o desenvolvimento de módulos de
7 inteligência artificial agregados a simuladores onde após a participação poderemos avaliar o modelo e melhorá-
8 lo, se for o caso.
9
10 \endinput
```

Figura 3: Arquivo exemplo hipoteses.tex.

Para customizar o relatório, altere o texto do exemplo acima de acordo com as hipóteses do seu projeto.

Ao converter os documentos acima (introducao.tex e hipoteses.tex) para PDF, temos a versão de apresentação mostrada na próxima página:

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.

3.4 O arquivo objetivos.tex

O arquivo objetivos.tex, Figura 4, contém os seguintes códigos:

```
1 \section{Objetivos}
2
3 \subsection{Gerais}
4
5 À Liga Rescue Simulation tem dois objetivos gerais. O primeiro visa o desenvolvimento de simuladores para
6 formar a infra-estrutura de sistemas de simulação, e emular fenômenos reais predominantes em catástrofes. O
7 segundo visa desenvolver agentes inteligentes capazes de interagir e atuar em cenários de desastre.
8
9 O objetivo primário desta pesquisa é construir uma ferramenta de aprendizagem para o diminuir o tempo de
10 treinamento do agente; conseqüentemente atingir melhores desempenhos. Neste contexto, o projeto tem o propósito
11 de adaptar e melhorar a arquitetura de aprendizagem RatoLIC, tornando-a robusta e com tomada de decisão
12 autônoma, somado ao suporte para outros algoritmos de aprendizagem na elaboração de comportamentos do agente,
13 assim resolvendo o problema manual da escolha da estratégia mais adequada.
14
15 \subsection{Específicos}
16
17 O objetivo do projeto visa desenvolver módulos de inteligência artificiais robustos capazes de se adaptar a
18 qualquer ambiente.
19
20 Os objetivos secundários são:
21 \begin{itemize}
22   \item Escolha automática de estratégias utilizadas pelos agentes do RoboCupRescue;
23   \item Redução do tempo de treinamento desses agentes;
24   \item Redução da complexidade do problema;
25   \item Desenvolvimento de uma arquitetura genérica, robusta e com tomada de decisão autônoma para um agente
26 específico, fazendo ele atuar de forma autônoma.
27 \end{itemize}
28
29 \endinput
```

Figura 4: Arquivo exemplo objetivos.tex.

Explicando o código contido na Figura 4, temos:

- A linha 3, `\subsection{}`, divide o documento em Subseções. O nome da Subseção estará entre `{ }`.
- A linha 14, `\begin{itemize}`, marca o início do ambiente de listas em um documento.
- A linha 15, `\item`, adiciona um hífen antes de uma frase ou palavra.
- A linha 19, `\end{itemize}`, marca o fim do ambiente de listas em um documento.

Para customizar o relatório, altere o texto do exemplo acima de acordo com o objetivo do seu projeto.

Quando convertermos o documento acima para PDF, temos a versão de apresentação mostrada na próxima página:

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; conseqüentemente 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.

3.5 O arquivo justificativa.tex

O arquivo justificativa.tex, Figura 5, contém os seguintes códigos:

```
1 \section{Justificativa}
2
3 O fato de buscas e salvamentos realizados em ambientes hostis por equipes de seres humanos incorrer em risco de
4 morte a estes, e do mesmo modo, a necessidade de se desenvolver tecnologias que auxiliem ou mesmo executem toda
5 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).
\endinput
```

Figura 5: Arquivo exemplo 'justificativa.tex.

Para customizar o relatório, altere o texto do exemplo acima de acordo com a justificativa do seu projeto.

3.6 O arquivo revisaoliteratura.tex

Logo após criarmos o arquivo revisaoliteratura.tex, Figura 6 e 7, onde encontramos os seguintes códigos:

```

1 \section{Revisão de Literatura}
2
3 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
\cite{RoboCupRescueHome2009}.
4
5 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 a 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.
6
7 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
um 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." \cite{Bellifemine2007}. 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-\ref{fig:imagem1}.
8 \begin{figure}[ht]
9 \centering
10 \includegraphics[width=0.96\textwidth]{imagens/viewer.jpg}
11 \caption{Visualizador.}\label{fig:imagem1}
12 \end{figure}
13
14 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 capacidade 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
15 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 policias 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.
16
17 Atualmente o simulador possui módulos compilados na linguagem C e Java. Estes se conectam ao kernel, ver
Figura-\ref{fig:imagem2}, que é responsável por controlar o ambiente de simulação.
18
19 \begin{figure}[ht]
20 \centering
21 \includegraphics[width=0.96\textwidth]{imagens/kernel.jpg} \caption{Kernel.}\label{fig:imagem2}
22 \end{figure}

```

Figura 6: Parte 1/2 do arquivo revisaoliteratura.tex.

```

23
24 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.
25
26 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.
27
28
29 \begin{table}[h]
30 \begin{center}
31 \caption{Configuração dos agentes.}
32 \begin{tabular}{|c||c||c|} \hline
33 Entidade & ε & Mínimo & ε & Máximo \\ \hline
34 \hline
35 Bombeiros & ε & 0 & ε & 15 \\ \hline
36 Polícia & ε & 0 & ε & 15 \\ \hline
37 Ambulância & ε & 0 & ε & 8 \\ \hline
38 Civil & ε & 70 & ε & 90 \\ \hline
39 Torre Central dos Bombeiros & ε & 0 & ε & 1 \\ \hline
40 Torre Central da Polícia & ε & 0 & ε & 1 \\ \hline
41 Torre Central das Ambulâncias & ε & 0 & ε & 1 \\ \hline
42 Refúgios & ε & 0 & ε & 5 \\ \hline
43 Pontos de Ignição & ε & 2 & ε & 8 \\ \hline
44 \hline
45 \end{tabular}
46 \end{center}
47 \end{table}
48
49
50 \endinput

```

Figura 7: Parte 2/2 do arquivo revisaoliteratura.tex.

Explicando o código contido nas Figuras 6 e 7, temos:

- A linha 7, `\ref{}`, cria uma referência cruzada em um documento. O nome da Referência estará entre `{ }`.
- A linha 8, `\begin{figure}[]`, cria um ambiente para inserir uma imagem em um documento. Qualquer coisa que seja incluída num ambiente *figure* ou *table* será tratado como matéria flutuante. Ambos ambientes flutuantes proporcionam um parâmetro opcional cujo os parâmetros de posicionamento estão entre `[]`. O designador de colocação `[!hbp]` permite ao L^AT_EX₂_ε posicionar a tabela ou imagem justamente aqui (h) ou ao final (b) de alguma página ou em alguma página especial para elementos flutuantes (p), e em qualquer parte senão ficar bem (!). Se não fornecer nenhum designador de posição, então as classes normalizadas assumem `[hbp]`.
- A linha 9, `\centering`, centraliza a imagem.
- A linha 10, `\includegraphics[]{}{}`, inclui uma imagem em seu documento,

cujo o tamanho da imagem é definido entre [] e o nome do arquivo estará entre { }.

- A linha 11, `\caption{}`, define a legenda de um objeto. O nome da Legenda estará entre { }. `\label{}`, cria uma marca para a referência cruzada em um documento. O nome da Marca estará entre { }.
- A linha 12, `\end{figure}`, marca o fim do ambiente de inserção de imagens.
- A linha 29, `\begin{table}[]`, cria um ambiente para inserir uma tabela em um documento. Os procedimentos de posicionamento de uma tabela no $\text{\LaTeX}2_{\epsilon}$ são iguais aos de inserção de figura como vimos anteriormente.
- A linha 30, `\begin{center}`, cria um ambiente para centralizar objetos.
- A linha 32, `\begin{tabular}{}`, cria uma tabela cuja quantidade de colunas e a largura das mesmas são definidas entre { }. `\hline`, cria uma linha horizontal. Para facilitar o processo de criação de uma tabela, utilizaremos o software *LaTable* cuja especialidade é criar tabelas para serem inseridas em um documento em $\text{\LaTeX}2_{\epsilon}$.
- A linha 33, `&`, cria uma divisão entre as células das colunas. `\\`, salta uma linha.
- A linha 45, `\end{tabular}`, marca o fim da tabela.
- A linha 46, `\end{center}`, marca o fim do ambiente de centralização de objetos.
- A linha 47, `\end{table}`, marca o fim do ambiente da tabela.

Para customizar o relatório, altere o texto do exemplo acima de acordo com a revisão de literatura do seu projeto.

3.7 O arquivo desenvolvimento.tex

Logo após criarmos o arquivo desenvolvimento.tex, Figura 8, onde encontramos os seguintes códigos:

```

1 \section{Desenvolvimento}
2 Atualmente o simulador utilizado na competição se encontra na versão 1.0, este pode ser encontrado no site
\begin{verbatim}http://sourceforge.net/projects/roborescue/\end{verbatim} gratuitamente, o aplicativo é
desenvolvido na linguagem Java e contém em seu corpo estrutural funções em C as quais são reflexos de versões
passadas totalmente programadas em C.
3
4 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.
5 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.
6 O simulador obteve bons resultados durante a execução da aplicação grafica apenas em dois sistemas
operacionais da distribuição Linux, estes são Kurumin Linux 8.0 e Ubuntu.
7 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.
8
9 A instalação procede da seguinte maneira.
10 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.
11
12 Para isso primeiramente logue como root, então abra o shell e digite o comando:\begin{verbatim}
13 export http_proxy="http://AD6usuario-da-rede:senha-da-rede@10.2.0.1:3128",\end{verbatim} caso essa
configuração não funcione, você pode tentar também escrever a seguinte linha no arquivo
\begin{verbatim}\verb!/etc/apt/apt.conf:\end{verbatim}
14 \begin{verbatim}
15 Acquire{
16 HTTP::proxy "http://USusuario-da-rede:senha-da-rede@10.2.0.1:3128";
17 }
18 \end{verbatim}
19 Então poderemos finalmente atualizar a lista do apt-get utilizando o seguinte comando.
20 \begin{verbatim}apt-get update\end{verbatim}
21 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
22 Para isso digite o seguinte comando.
23 \begin{verbatim}apt-get install xterm\end{verbatim}
24 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
25 xterm
26 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:
27 \begin{verbatim}apt-get install ant\end{verbatim}
28 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.
29 Com o terminal X apontando internamente a pasta boot digite:
30 \begin{verbatim}demo.sh\end{verbatim}
31 Uma apresentação básica de funcionamento do simulador ocorrerá, note que este não tem inteligência artificial
e os agentes executam as atividades de forma desorganizada.
32
33 \endinput

```

Figura 8: Arquivo exemplo desenvolvimento.tex.

Explicando o código contido na Figura 8, temos:

- A linha 2, `\begin{verbatim}`, marca o início do ambiente onde os textos escritos nesse ambiente serão passados diretamente para o ficheiro de resultado, como se o tivesse escrito numa máquina de escrever, com todas as quebras de linha e espaços, sem que qualquer comando $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{\varepsilon}$ seja executado. Assim, `\end{verbatim}`, marca o fim do ambiente citado anteriormente.

- A linha 13, `\verb!`, escreve o texto exatamente como foi digitado, incluindo espaços e caracteres especiais.

Para customizar o relatório, altere o texto do exemplo acima de acordo com o desenvolvimento do seu projeto.

3.8 O arquivo `material.tex`

Logo após criarmos o arquivo `material.tex`, Figura 9, onde encontramos os seguintes códigos:

```
1 \subsection{Material e Métodos}
2
3 O simulador foi instalado nas seguintes condições nas quais obtivemos sucesso.\\
4 Sistema Operacional: Kurumin Linux 8.0 ou Ubuntu\\
5 Processador: Core 2 Duo 2.2\\
6 Memória Ram: 1GB\\
7 HD: 20GB's\\
8 Monitor de 17''\\
9 Conexão com Internet\\
10 \pagebreak
```

Figura 9: Arquivo exemplo `material.tex`.

Para customizar o relatório, altere o texto do exemplo acima de acordo com o material utilizado no seu projeto.

Ao converter os documentos acima (`justificativa.tex`, `revisaoliteratura.tex` e `desenvolvimento.tex`) para PDF, temos a versão de apresentação mostrada nas seis páginas que se seguem neste artigo:

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 a 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.

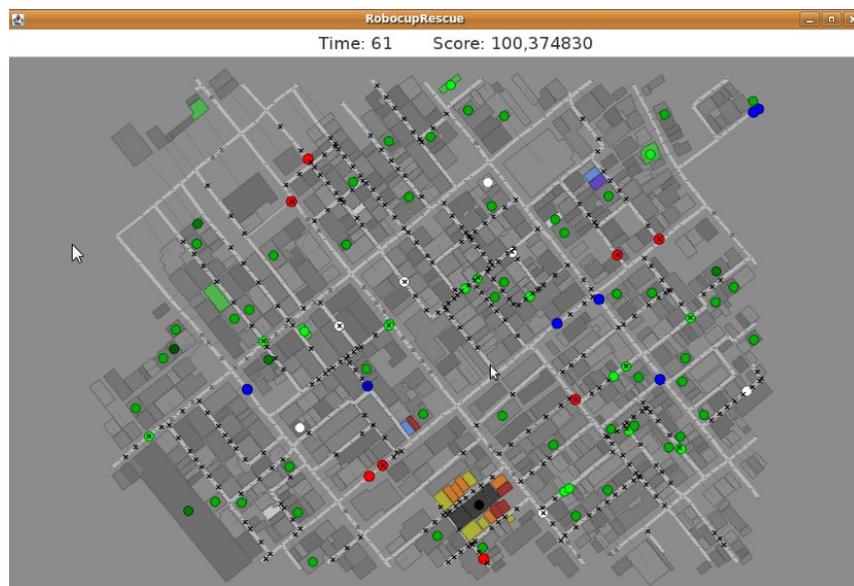


Figura 1: Visualizador.

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.

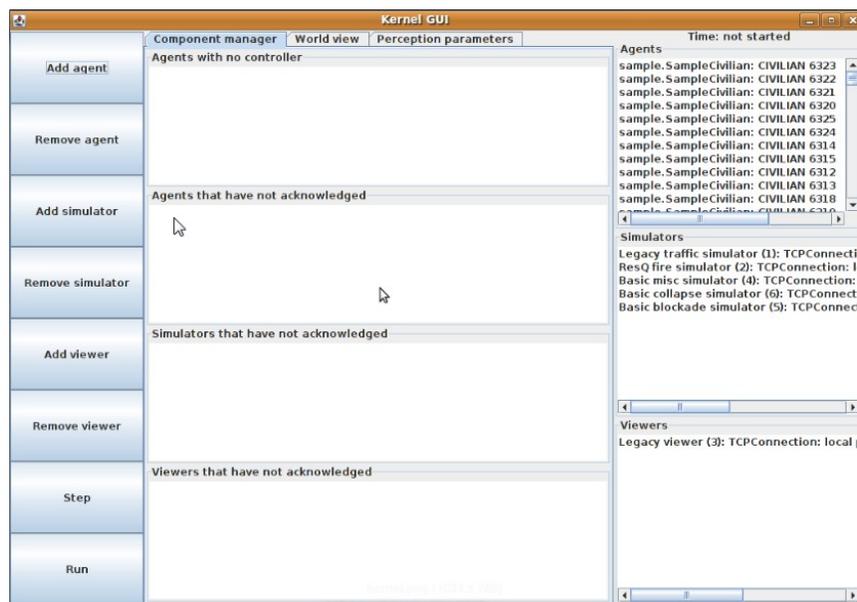


Figura 2: Kernel.

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

3.9 O arquivo resultados.tex

Logo após criarmos o arquivo resultados.tex, Figura 10, onde encontramos os seguintes códigos:

```
1 \section{Resultados}
2
3 O simulador é instalado e executado perfeitamente nas versões de sistemas operacionais Linux mencionadas, já em
4 sistemas operacionais Windows a execução não ocorre corretamente, pois diversos problemas ocorrem com a
5 comunicação do kernel do simulador com o sistema.
6
7 \endinput
```

Figura 10: Arquivo exemplo resultados.tex.

Para customizar o relatório, altere o texto do exemplo acima de acordo com os resultados obtidos do seu projeto.

3.10 O arquivo consideracoesfinais.tex

Logo após criarmos o arquivo consideracoesfinais.tex, Figura 11, onde encontramos os seguintes códigos:

```
1 \section{Considerações Finais}
2
3 O Simulador é a melhor forma de se executar um treinamento das redes neurais embutidas em cada agente, uma vez
4 que este software consegue efetuar simular uma catástrofe urbana com certa precisão.
5
6 À continuação do projeto se deve a implantação do módulos de inteligência artificial, o que ocorrerá nos
7 próximos meses.
8
9 \endinput
```

Figura 11: Arquivo exemplo consideracoesfinais.tex.

Para customizar o relatório, altere o texto do exemplo acima de acordo com as considerações finais do seu projeto.

Ao converter os documentos acima (resultados.tex e consideracoesfinais.tex) para PDF, temos a versão de apresentação mostrada na página que se segue:

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.

3.11 Referências Bibliográficas

Por último teremos as Referências Bibliográficas, que não é um arquivo e será incluída no final do relatório do Projeto.

O código para inserção das Referências se encontra no arquivo `introducao.tex`, previamente visto. Para gerar as Referências Bibliográficas você deverá compilar o arquivo utilizando o comando *BibTex*, geralmente esse comando já está incluso na maioria dos editores de arquivos em $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{\varepsilon}$. O capítulo Referências será exibido na página que se segue:

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

4 Conclusão

O sucesso na elaboração de um relatório do PI em $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}}$ utilizando o leiaute apresentado neste artigo, proporciona uma maior facilidade da construção do mesmo. Pois o aluno não terá que se preocupar com a formatação de seu relatório, conseqüentemente evitando possíveis erros e distrações durante a produção do relatório.

Ao contrário de editores de texto que utilizam a tecnologia WYSIWYG (What You See Is What You Get) onde o autor pode cometer um erro durante a elaboração de seu documento que acabe comprometendo a estrutura do seu trabalho por completo.

5 Agradecimentos

Os autores agradecem aos professores Delaine Vasconcelos Rosa e José Geraldo Costa pela revisão gramatical final.