

ALAN D. BARROSO
FELIPE DE C. SANTANA
VICTOR LASSANCE

**Uso de Técnicas de Coordenação entre Agentes
para Operações de Resgate em Ambientes de
Desastre Urbano**

São Paulo

2013

ALAN D. BARROSO
FELIPE DE C. SANTANA
VICTOR LASSANCE

Uso de Técnicas de Coordenação entre Agentes para Operações de Resgate em Ambientes de Desastre Urbano

Texto apresentado à Escola Politécnica da
Universidade de São Paulo como requisito
para a conclusão do curso de graduação em
Engenharia de Computação, junto ao De-
partamento de Engenharia de Computação
e Sistemas Digitais (PCS).

Orientador: Prof. Jaime S. Sichman

Co-orientadores: Profa. Anarosa A. F. Brandão e Luis G. Nardin

São Paulo

2013

ALAN D. BARROSO
FELIPE DE C. SANTANA
VICTOR LASSANCE

Uso de Técnicas de Coordenação entre Agentes para Operações de Resgate em Ambientes de Desastre Urbano

Texto apresentado à Escola Politécnica da
Universidade de São Paulo como requisito
para a conclusão do curso de graduação em
Engenharia de Computação, junto ao De-
partamento de Engenharia de Computação
e Sistemas Digitais (PCS).

Trabalho aprovado. São Paulo, 2 de dezembro de 2013:

Prof. Jaime S. Sichman
Orientador

Professora Anarosa A. F. Brandão
Co-orientadora

Luis G. Nardin
Co-orientador

São Paulo
2013

Agradecimentos

O desenvolvimento desse trabalho não se deu somente pelo esforço dos membros da equipe, mas também pela colaboração e suporte de diversas pessoas e entidades que foram fundamentais para o sucesso do projeto.

Agradecemos ao nosso orientador Prof. Jaime Simão Sichman, e aos co-orientadores Profa. Anarosa Alves Franco Brandão e Luis Gustavo Nardin, pelas instruções e direcionamentos durante todo o período do trabalho.

Agradecemos à Escola Politécnica, bem como à Pró-Reitoria de Cultura e Extensão Universitária da USP e ao Departamento de Computação e Sistemas Digitais da POLI-USP, pelo apoio e incentivo à participação nas competições.

Agradecemos à equipe do Centro de Operações do Corpo de Bombeiros, especialmente aos Capitães Yuri Moraes Bedini e Herbert Meyerhof, e ao Centro de Operações da Defesa Civil, na pessoa do Coronel Jair Paca, pela recepção e pelas informações fornecidas durante as visitas.

Agradecemos ao Centro de Estudos e Pesquisas em Desastres da USP (CEPED) pela ajuda e orientação na fase inicial do trabalho.

Por fim, agradecemos a nossos familiares pelo constante apoio e incentivo ao longo do projeto.

Resumo

Este trabalho descreve uma abordagem de alocação de tarefas que considera a existência de informações locais e globais com o objetivo de coordenar entidades de resgate para que maximizem a eficiência no resgate a vítimas e na proteção do patrimônio da cidade em caso de desastre. Supondo uma comunicação limitada e incerta, esse trabalho também apresenta uma comparação de técnicas de coordenação de forma a melhorar a alocação de tarefas. A comparação foi baseada em resultados de experimentos obtidos através do uso do simulador da *RoboCup Rescue Simulation League*.

Palavras-chaves: Coordenação, Gerenciamento de Desastres, Sistemas Multiagentes.

Abstract

This work describes a task allocation approach that takes into account local and global information with the objective of coordinating rescue agents, in order to maximize their efficiency when rescuing victims and protecting the city in a disaster occurrence. Assuming limited and uncertain communication, this work also presents a comparison between coordination techniques, enhancing task allocation mechanisms. The comparison was based on experiments results obtained by running the *RoboCup Rescue Simulation League* simulator.

Palavras-chaves: Coordination, Disaster Management, Multiagent Systems.

Lista de ilustrações

Figura 1 – Enchente causada por chuvas torrenciais na região de Santos em Março de 2013	12
Figura 2 – Deslizamento ocorrido na zona leste de São Paulo em Janeiro de 2013 .	12
Figura 3 – Simulador da competição durante uma simulação	22
Figura 4 – Arquitetura do simulador	23
Figura 5 – Diagrama de sequência de um ciclo do simulador	24
Figura 6 – Entidades de resgate que atuam em São Paulo	26
Figura 7 – Progresso da pontuação de cada versão ao longo de uma simulação no mapa <i>Test</i>	51
Figura 8 – Progresso da pontuação de cada versão ao longo de uma simulação no mapa <i>Kobe</i>	52
Figura 9 – Progresso da pontuação de cada versão ao longo de uma simulação no mapa <i>Paris</i>	53
Figura 10 – Evolução da pontuação do time <i>LTI Rescue</i>	54
Figura 11 – Comparação para o mapa <i>Kobe</i> . A esquerda, os resultados da versão v1. A direita, os resultados da versão v4	55
Figura 12 – Comparação para o mapa <i>Paris</i> . A esquerda, os resultados da versão v1. A direita, os resultados da versão v4	55
Figura 13 – Mapa da Cidade Universitária para o Simulador da RoboCup Rescue .	60

Lista de tabelas

Tabela 1 – Lista de participantes da liga Rescue Agent em 2013	36
Tabela 2 – Lista de participantes da categoria Infrastructure Competition em 2013	37
Tabela 3 – Lista de participantes da Competição Brasileira de Robótica 2013 . . .	38
Tabela 4 – Resultado final da liga Rescue Agent na <i>RoboCup Rescue</i> 2013	48
Tabela 5 – Resultado final da Competição Brasileira de Robótica 2013	49
Tabela 6 – Resultado dos testes de hipótese realizados entre versões do código. . .	56

Sumário

1	Introdução	11
1.1	Motivação	11
1.2	Objetivo	13
1.3	Organização do documento	14
2	Metodologia	15
2.1	Estudo de conceitos teóricos e ferramentas utilizadas	15
2.2	Desenvolvimento dos agentes para <i>Robocup Rescue 2013</i>	16
2.3	Visita às entidades de resgate da cidade de São Paulo	16
2.4	Análise crítica das visitas e ideias de melhorias das técnicas de simulação	16
2.5	Desenvolvimento dos agentes para a CBR 2013	16
2.6	Desenvolvimento do tutorial para criação de novos mapas no RCRS	17
3	Conceitos e ferramentas utilizadas	18
3.1	Aspectos conceituais	18
3.1.1	Sistemas Multiagentes (SMAs)	18
3.1.2	Técnicas de coordenação em SMA	19
3.1.3	Simulador da <i>RoboCup Rescue Simulation League</i>	21
3.2	Aspectos práticos	25
3.2.1	CEPED-USP	25
3.2.2	Entidades de resgate	26
3.2.2.1	Defesa Civil de São Paulo	26
3.2.2.2	SAMU	26
3.2.2.3	Corpo de Bombeiros	27
3.2.3	Integração das entidades	27
4	Visitas a entidades de resgate	28
4.1	Relato de visita à Defesa Civil	28
4.2	Relato de visita ao COBOM	28
4.2.1	Atuação do Corpo de Bombeiros	29
4.2.2	Estrutura do COBOM e atendimento	29
4.2.3	Procedimentos básicos de atendimento	30
4.2.4	Estratégias dos Bombeiros	32
4.2.4.1	Divisão e preposicionamento	32
4.2.4.2	Atuação efetiva	33
5	Competições e workshops	35
5.1	WESAAC 2013	35

5.2	<i>RoboCup Rescue</i> 2013	35
5.2.1	Times participantes	36
5.2.2	Regras da competição	37
5.2.3	<i>Infrastructure Competition</i>	37
5.3	CBR 2013	38
5.3.1	Equipes participantes	38
5.3.2	Regras da competição	38
6	Desenvolvimento dos agentes	40
6.1	<i>RoboCup Rescue</i> 2013	40
6.1.1	Correção de funcionalidades	40
6.1.2	Setorização eficiente do mapa	40
6.1.3	Tratamento especial de postos de gasolina	41
6.1.4	Desbloqueio preventivo	41
6.1.5	Escuta do pedido de ajuda de civis	42
6.1.6	Cálculo de prioridade nos resgates	42
6.1.7	Mudança na forma de envio de mensagens entre agentes	43
6.2	CBR 2013	43
6.2.1	Fecho convexo para distribuição dos bombeiros	43
6.2.2	Utilização otimizada das estruturas de dados	44
6.2.3	Caminho alternativo para o objetivo	45
6.2.4	Melhoria no desbloqueio de caminhos	45
6.2.5	Identificação prévia de caminho já desbloqueado	46
6.2.6	Pedido de resgate para agentes bloqueados	46
6.3	Estratégia adicional	46
7	Resultados obtidos	48
7.1	Participação nas competições	48
7.1.1	<i>RoboCup Rescue</i> 2013	48
7.1.2	CBR 2013	49
7.2	Comparação da evolução dos agentes ao longo do ano	49
7.2.1	Apresentação dos mapas utilizados para os testes	50
7.2.1.1	Mapa <i>Test</i>	50
7.2.1.2	Mapa <i>Kobe</i>	50
7.2.1.3	Mapa <i>Paris</i>	51
7.2.2	Análise dos resultados	51
7.2.2.1	Código Base 2012 (v1)	51
7.2.2.2	Versão <i>RoboCup Rescue</i> 2013 (v2)	52
7.2.2.3	Versão CBR 2013 (v3)	53
7.2.2.4	Versão final 2013 (v4)	54
7.2.2.5	Comparação entre as versões	54

7.3	Recomendação de melhorias para o simulador	57
7.3.1	Modelagem dos agentes	57
7.3.2	Ambiente de simulação	58
7.3.3	Funcionamento e comportamento dos agentes	58
7.4	Criação do tutorial para o desenvolvimento de novos mapas	59
8	Conclusões	61
8.1	Considerações finais	61
8.2	Trabalhos futuros	62
8.2.1	Mudanças estruturais	62
8.2.1.1	Visão de mundo adaptada	62
8.2.1.2	Máquina de Estados	63
8.2.2	Particionamento na comunicação entre agentes	64
8.2.3	Estratégias específicas em caso de comunicação instável ou inexistente	65
8.2.4	Posicionamento nas primeiras guarnições	65
	Referências	67
	Apêndices	69
	APÊNDICE A – Resumo estendido - WESAAC 2013	70
	APÊNDICE B – Descrição do time - RoboCup Rescue 2013	74
	APÊNDICE C – Descrição do time - CBR 2013	85
	APÊNDICE D – Tutorial para criação de mapas para o RoboCup Rescue Simulator	90

1 Introdução

1.1 Motivação

A expansão urbana observada no Brasil nas últimas décadas causou o crescimento acelerado e desordenado das cidades, potencializando assim a ocorrência de desastres naturais em ambientes urbanos, principalmente em áreas mais densamente povoadas. Somado a isso, o Brasil sediará em breve eventos internacionais de relevância mundial, como a Copa do Mundo em 2014 e os Jogos Olímpicos em 2016, agravando as possíveis consequências de tais desastres devido a elevada concentração de estrangeiros que desconhecem a dinâmica das cidades brasileiras. Por exemplo, a ocorrência de alagamentos e o consequente bloqueio de vias pode desorientá-los, aumentando as chances de ficarem ilhados ou serem carregados pelas águas.

Segundo (AMARAL; GUTJAHR, 2011), desastre é uma situação de crise que causa sérios danos e prejuízos a um sistema social envolvendo perdas materiais, humanas, econômicas e ambientais de grande extensão, cujos impactos excedem a capacidade dessa sociedade de se recuperar com seus próprios recursos. Desastres podem ser tratados de duas maneiras distintas e por vezes complementares: a primeira é evitando a sua ocorrência; a segunda é por meio do estabelecimento prévio de políticas e estratégias de gerenciamento de desastres.

Desastres naturais, em específico, são eventos de difícil previsibilidade e pouco pode-se fazer para evitá-los, sendo necessária, portanto, a existência prévia de políticas e estratégias de gerenciamento que auxiliem na mitigação de suas consequências. O gerenciamento de desastres é um dos instrumentos de gestão urbana que, integrado a outras políticas públicas, tem a finalidade de reduzir, prevenir e controlar o risco de desastres naturais em ambientes urbanos.

No caso específico desse tipo de desastres, pesquisas e dados científicos recentes indicam que eles estão se tornando mais frequentes, intensos, dinâmicos e complexos (GUHA-SAPIR *et al.*, 2011). No Brasil, certos tipos de desastres naturais, como enchentes e deslizamentos de terra (respectivamente Figuras 1 e 2), que poderiam ter suas consequências ao menos minimizadas têm a implantação de seus planos de mitigação postergados devido ao seu alto custo. Assim, faz-se necessário a existência de um plano de gerenciamento de desastres eficaz para minimizar os danos causados após sua ocorrência.

Mais especificamente, a etapa de resposta a desastres desse plano consiste em um conjunto de ações, que se desenvolvem durante o período de desastre ou imediatamente

Figura 1 – Enchente causada por chuvas torrenciais na região de Santos em Março de 2013



Fonte: Blog clicRBS (2013)

Figura 2 – Deslizamento ocorrido na zona leste de São Paulo em Janeiro de 2013



Fonte: Estadão Conteúdo (2013)

após a ocorrência do evento, com a finalidade de (i) buscar e resgatar as pessoas afetadas; (ii) mitigar o impacto imediato do desastre e seus efeitos de longo prazo; e (iii) garantir o funcionamento dos sistemas básicos de infraestrutura da comunidade afetada. As características contextuais em que essa etapa é realizada são: (i) ambiente complexo, dinâmico e parcialmente observável; (ii) envolvimento de um grande número de entidades de resgate heterogêneas; (iii) existência de informação incompleta e incerta; (iv) insuficiência de re-

ursos humanos e materiais para a realização simultaneamente de todas as tarefas; e (v) necessidade de tomada de decisão e realocação de recursos em tempo reduzido e limitado (KELLY, 2013).

Assim, ações eficientes de coordenação e alocação dos recursos existentes são de extrema importância na etapa de resposta a desastres. No entanto, tais ações só são possíveis quando as tarefas a serem realizadas são conhecidas, o que nem sempre ocorre, visto que o ambiente é parcialmente observável e a informação sobre ele é incerta. Dessa forma, é imperativa a existência de formas de exploração e comunicação eficiente no ambiente para identificação de tais tarefas.

No intuito de possibilitar a avaliação de técnicas de coordenação para resposta a desastres, a plataforma de simulação RoboCup Rescue Simulator (RCRS) foi proposta em 1999 (KITANO et al., 1999). Trata-se de uma plataforma de simulação multiagentes que reproduz as condições de desastre geradas como consequência da ocorrência de um terremoto em uma área urbana. Suas características intrínsecas a esse tipo de situação e sua arquitetura multiagentes possibilitam o uso de técnicas de coordenação da área de Sistemas Multiagentes (SMA) na tentativa de minimizar os impactos causados por tal desastre.

1.2 Objetivo

O objetivo geral deste trabalho é propor uma técnica de coordenação para as diferentes entidades de resposta a desastres que maximize a eficácia do resgate de vítimas e a proteção do patrimônio público, minimizando assim os impactos causados. Tal técnica propõe uma abordagem de alocação e execução de tarefas que considera a existência de informações locais e globais com o intuito de coordenar as entidades de resgate, levando em consideração não somente as técnicas conceituais de coordenação de agentes, mas também incorporando características das técnicas efetivamente utilizadas por entidades de resgate reais, tais como Corpo de Bombeiros (COBOM), Defesa Civil e Serviço de Atendimento Móvel de Urgência (SAMU).

Por fim, este trabalho também visa facilitar o uso do RCRS para simulação de áreas cujos mapas não estão disponíveis. Sendo assim, um dos objetivos desse trabalho é disponibilizar um tutorial detalhado dos processos e ferramentas envolvidos no desenvolvimento de novos mapas para o simulador, sendo fornecido como exemplo prático o mapa do campus da Cidade Universitária da USP em São Paulo.

1.3 Organização do documento

O restante deste documento é estruturado da seguinte forma. O Capítulo 2 especifica a metodologia empregada no desenvolvimento do trabalho por meio de suas etapas. No Capítulo 3, são brevemente apresentados os aspectos conceituais aplicados no projeto e uma breve descrição das entidades de resgate da cidade de São Paulo, bem como o Centro de Estudos e Pesquisas sobre Desastres (CEPED). O Capítulo 4 relata as visitas realizadas às entidades de resgate, enquanto o Capítulo 5 detalha as duas competições e o workshop que a equipe participou ao longo do ano. O Capítulo 6 descreve as técnicas de coordenação que foram implementadas nas diversas etapas do projeto para uso no RCRS. O Capítulo 7 traz os resultados obtidos, tanto aqueles obtidos nas competições, como uma análise de desempenho comparativa considerando a evolução do código. Por fim, no Capítulo 8 são apresentadas as conclusões e sugeridas ideias para trabalhos futuros.

Anexos a este documento, encontram-se três artigos científicos produzidos em conjunto com o orientador e co-orientadores. O Apêndice A apresenta o resumo estendido publicado nos anais do WESAAC 2013. Já o Apêndice B contém a descrição das técnicas de coordenação usadas na competição *RoboCup Rescue* 2013. O Apêndice C contém a descrição das técnicas de coordenação usadas na Competição Brasileira de Robótica (CBR) 2013. Por fim, o Apêndice D apresenta um tutorial desenvolvido para a criação de novos mapas para o RCRS.

2 Metodologia

Para possibilitar um desenvolvimento de forma iterativa e progressiva, a metodologia aplicada nesse projeto contemplou diversas etapas, definidas de acordo com o modelo espiral.

A primeira etapa envolveu o estudo dos conceitos teóricos e ferramentas utilizadas durante o desenvolvimento do projeto, detalhados na seção 3.1. A segunda etapa compreendeu a realização de visitas a diversas entidades de resgate da cidade São Paulo, com o intuito de identificar as técnicas de coordenação utilizadas em situações reais de desastre. Em seguida, na terceira etapa foi realizada a análise crítica dessas estratégias em contraste com as características do simulador. A quarta etapa, realizada em paralelo com as outras duas etapas anteriores, visou o desenvolvimento do código para a competição internacional de simulação de agentes, a *Robocup Rescue Simulation 2013*¹, ocorrida em Junho de 2013 em Eindhoven, na Holanda. A quinta etapa teve novamente como foco o desenvolvimento das técnicas de coordenação de entidades de resgate para desenvolver um time para a Competição Brasileira de Robótica 2013², a qual foi realizada em Fortaleza/CE em Outubro de 2013. O desempenho destas diversas técnicas foi então analisado, através de uma comparação do desempenho dos diversos times desenvolvidos. Finalmente, a última etapa teve como objetivo a criação de um tutorial para facilitar o desenvolvimento de novos mapas para uso no RCRS, cujo uso foi ilustrado através da criação do mapa do campus da Cidade Universitária da USP em São Paulo.

2.1 Estudo de conceitos teóricos e ferramentas utilizadas

Nessa fase do projeto, foram pesquisados livros e artigos científicos relacionados ao tema da programação e coordenação multiagentes. Esse estudo preliminar foi necessário para um melhor entendimento do funcionamento da competição, bem como para habilitar os integrantes da equipe na criação de novas técnicas de coordenação para ambientes de desastre.

Em seguida, a equipe se concentrou no estudo mais aprofundado do RCRS e seus diferentes tipos de agentes. Esse estudo consistiu no entendimento do funcionamento da plataforma e dos passos iniciais para o desenvolvimento de agentes operacionais para a simulação, concretizados por meio do desenvolvimento e testes de agentes muito simples.

¹ <http://roborescue.sourceforge.net/>.

² <http://www.cbrobotica.org/>.

2.2 Desenvolvimento dos agentes para *Robocup Rescue 2013*

Na segunda etapa do projeto, foram desenvolvidas as estratégias de coordenação dos agentes para a competição *Robocup Rescue 2013*. Para essa fase, primeiramente a equipe se reuniu com a antiga equipe de desenvolvimento do time *LTI Rescue* para discutir quais novas estratégias poderiam ser criadas, e quais os problemas e *bugs* conhecidos a serem corrigidos.

Terminada essa fase de reuniões, a equipe usou esses conhecimentos para determinar as estratégias que seriam implementadas para a competição. As estratégias determinadas foram reunidas da descrição do time enviada para a competição (ver Apêndice B).

As últimas ações dessa etapa foram a efetiva implementação das estratégias dos agentes e a realização dos testes de verificação e validação para garantir que essas estratégias estavam corretamente implementadas.

A equipe participou da competição *RoboCup Rescue Agent Simulation* na cidade de Eindhoven na Holanda representada por um de seus integrantes, sendo os resultados obtidos descritos na Seção 7.1.1.

2.3 Visita às entidades de resgate da cidade de São Paulo

A terceira etapa do projeto consistiu na visita a diversas entidades de resgate da cidade de São Paulo. Durante o período do projeto, a equipe teve a oportunidade de visitar duas entidades de resgate: o Centro de Operações da Defesa Civil de São Paulo e o Centro de Operações do Corpo de Bombeiros (COBOM) de São Paulo. O COBOM foi visitado duas vezes, dando a oportunidade ao grupo de conhecer detalhadamente as estratégias aplicadas por essas entidades de resgate.

2.4 Análise crítica das visitas e ideias de melhorias das técnicas de simulação

A etapa de análise crítica e recomendações de melhorias para o simulador ocorreu durante o nosso período de estágio, quando tivemos mais tempo para realizar as visitas e discuti-las. A análise e as recomendações de melhorias das estratégias de coordenação podem ser encontradas no Capítulo 4 e Seção 7.3, respectivamente.

2.5 Desenvolvimento dos agentes para a CBR 2013

Essa etapa do projeto não estava inicialmente prevista, no começo do ano. No entanto, após a participação na competição *Robocup Rescue 2013* em Junho, quando já

háviamos compreendido bem o funcionamento do simulador e dos agentes, tivemos uma série de ideias que poderiam melhorar a qualidade e eficiência dos agentes. Dessa forma, nos propusemos a participar da Competição Brasileira de Robótica (CBR) 2013 para possibilitar uma validação dessas ideias, e também como motivação para implementar as estratégias baseadas na análise das visitas realizadas às entidades de resgate e aproveitar o aprendizado adquirido com a competição anterior.

Durante essa etapa, para participar no torneio, escrevemos um documento que descrevia o time a ser implementado e na competição (ver Apêndice C). Em seguida, passamos dois meses desenvolvendo as estratégias mais importantes, tendo como resultado a participação à distância e o primeiro lugar no torneio, sendo os resultados obtidos descritos na Seção 7.1.2.

2.6 Desenvolvimento do tutorial para criação de novos mapas no RCRS

A última etapa do projeto consistiu no desenvolvimento de um tutorial para a criação de novos mapas para utilização no RCRS, facilitando assim sua aplicação para análise de cenários reais de desastre. Usando o tutorial, foi criado o mapa do Campus da Cidade Universitária da USP em São Paulo, como demonstrado na Seção 7.4.

3 Conceitos e ferramentas utilizadas

3.1 Aspectos conceituais

Nesta seção, são abordados alguns dos conceitos importantes referentes ao projeto. São enfatizados os conceitos de Sistemas Multiagentes (SMAs) e de técnicas de coordenação entre agentes, além de uma descrição do simulador usado na competição *RoboCup Rescue* 2013. Desta maneira, essa seção é dividida em três partes.

Na primeira parte, são discutidos os conceitos envolvendo a programação orientada a SMAs. Também é mostrado como a solução para o problema proposto pela competição pode ser modelado sob a forma de um SMA.

Na segunda parte, são discutidos os esquemas de coordenação existentes para um SMA, sendo descrita em mais detalhes a técnica de coordenação escolhida para o projeto. Também é abordada a técnica de coordenação usada anteriormente na competição.

Na terceira e última parte, é discutido o modo de funcionamento do simulador utilizado para a competição, através da apresentação de detalhes dos seus principais módulos e do seu ciclo de funcionamento.

3.1.1 Sistemas Multiagentes (SMAs)

SMAs são sistemas onde um conjunto de agentes inseridos em um ambiente comum tem como intuito resolver um problema que um único agente não poderia solucionar ou que seria resolvido de forma menos rápida ou eficiente. Portanto, um dos fundamentos de um sistema como esse é a noção do que é um agente.

Como nos explica ([WOOLRIDGE, 2001](#)), um agente é um sistema de computador autônomo, capaz de tomar de forma autônoma as decisões que o levarão a resolver um determinado problema. É importante citar alguns conceitos interessantes que estão por trás da noção do que é um agente. Entre eles, estão os conceitos de crença, desejos e intenções de um agente.

As crenças de um agente são um conjunto de elementos que representam aquilo que o agente acredita. Isso inclui, por exemplo, o estado do ambiente em que ele está inserido. Um agente, normalmente, não possui um modelo exato do seu ambiente, e portanto se baseia nas suas crenças correntes. Os desejos de um agente são suas motivações. Eles representam os objetivos que o agente deseja alcançar. Finalmente, as intenções de um agente representam seu estado deliberativo. As intenções de um agente indicam a maneira como um agente deve agir perante uma situação.

Outro conceito interessante é o de decisão. Um agente, para tomar uma decisão, se baseia nos três aspectos fundamentais descritos acima. O agente toma as decisões a partir de suas crenças e de suas intenções, de maneira que o estado resultante de suas ações esteja mais próximo de seus desejos. É nesse ponto que entra o conceito de autonomia: se algo que foi solicitado a um agente entra em conflito com seus desejos, o agente provavelmente não irá realizá-lo.

Além do conceito de agentes, o outro grande pilar de um SMA é a interação e a coordenação entre agentes. Não basta que os agentes ajam sobre o ambiente para que o objetivo comum seja alcançado, é necessário que eles interajam entre si e que exista uma coordenação entre os mesmos. O aspecto de coordenação entre os agentes será discutido em mais detalhes na seção seguinte.

A interação entre os agentes pode vir sob duas maneiras: interação direta ou indireta. Por interação direta, refere-se à comunicação explícita entre dois agentes com o intuito de transmitir uma informação. A outra maneira de interação se dá quando um agente percebe e/ou modifica o ambiente onde ele está inserido para poder receber e transmitir a mensagem para outros agentes, e assim a interação é feita de modo indireto.

Baseado nesses aspectos de SMAs, pode-se traçar uma comparação entre o problema apresentado pela competição *RoboCup Rescue* e esse tipo de sistema. O objetivo principal da competição é de minimizar o número de baixas humanas e de danos na simulação de uma cidade tomada por um desastre. Para realizar tal objetivo, os participantes têm a possibilidade de programar diferentes tipos de agentes, tais como policiais, ambulâncias e bombeiros, que são melhor detalhados na Seção 3.1.3. Cada um deles pode agir de maneira diferente no ambiente para poder atingir o objetivo dado. Um sistema de comunicação existente permite que eles possam se comunicar diretamente entre si.

A partir da descrição acima, pode-se ver que o problema proposto pela competição se assemelha bastante à um SMA. Os policiais, as ambulâncias e os bombeiros são os agentes do sistema, a cidade é o ambiente onde os agentes estão inseridos e o sistema de comunicação permite que os agentes troquem informações, interagindo entre si e se coordenando.

Traçada essa equivalência, pode-se observar que podem ser aplicados diversos dos conceitos e das estratégias aplicadas em SMA para resolver os desafios impostos pela competição. Entre as estratégias, estão as técnicas de coordenação descritas na seção a seguir.

3.1.2 Técnicas de coordenação em SMA

De acordo com (WOOLRIDGE, 2001), quando agentes são colocados em modo de cooperação para resolver um problema, a resolução desse problema ocorre normalmente

em três etapas: decomposição do problema em subproblemas, solução dos subproblemas e síntese da resposta. A partir dessas três etapas, podem-se definir dois tipos de esquemas de coordenação entre agentes.

O primeiro tipo é o de *compartilhamento das tarefas*. Quando um problema é dividido em subproblemas menores, os agentes do sistema podem se organizar de maneira que cada agente seja responsável por resolver um desses subproblemas. O grande desafio desse tipo de atividade está ligado à negociação e a coordenação que deve existir para que a tarefa seja executada com uma alta eficiência.

O segundo tipo é o de *compartilhamento das soluções*. Esse tipo de atividade se dá após o término da resolução de um subproblema, quando o agente informa aos outros agentes que ele terminou sua tarefa e envia aos outros o resultado obtido. O compartilhamento de resultados permite que os outros agentes do sistema possam modificar suas crenças e suas intenções, de maneira a melhor resolver o problema geral.

Quanto aos dois tipos de esquemas, o problema imposto pela competição *RoboCup Rescue* exige o uso das duas. Como existem três tipos diferentes de agentes, com ações específicas para cada um, as tarefas a serem realizadas devem ser divididas entre os agentes para que o problema possa ser solucionado. Já no caso do compartilhamento de resultados, os agentes devem informar aos outros quando uma tarefa foi realizada para que um outro agente não tente se ocupar de uma tarefa finalizada, ou que não inicie uma tarefa antes que aqueles que deveriam precedê-la terem sido finalizadas.

Um outro ponto interessante a respeito da coordenação entre agentes se refere ao modo como o planejamento das tarefas deve ser realizado. Pode-se ter três tipos principais de planejamento das tarefas: planejamento centralizado para planos distribuídos, planejamento distribuído para plano centralizado e planejamento distribuído para planos distribuídos.

O primeiro tipo necessita de uma figura central que realiza todo o planejamento das tarefas e depois comunica aos agentes suas respectivas tarefas. Já o segundo leva em conta que todos os agentes entram em cooperação para elaborar um plano comum a todos eles, mas que cada um irá cuidar dos detalhes de sua tarefa particular. Já o terceiro tipo, o mais complexo, está relacionado ao fato de que cada agente elabora um plano individual próprio de maneira a cooperar com os outros agentes, e estes devem ao final concordar com a adoção de um plano global final.

O código para a competição, sendo uma evolução de um projeto já existente no LTI (*LTI Rescue*), emprega o terceiro tipo de planejamento de tarefas. Entretanto, a primeira versão do *LTI Rescue* o planejamento centralizado, sendo que todo o planejamento era realizado nas centrais espalhadas pelo mapa. Essas centrais enviavam as ações aos outros agentes que as realizavam. Ao término da tarefa, as centrais eram informadas pelos agen-

tes e um novo plano começava a ser traçado. A centralização do planejamento devia-se principalmente ao fato de que, anteriormente, a comunicação no RCRS só acontecia entre as centrais e os agentes de mesmo tipo.

No entanto, uma evolução no simulador da competição exigiu que a estratégia de coordenação fosse modificada. A comunicação foi expandida de tal maneira que qualquer agente podia se comunicar com qualquer outro agente, incluindo o uso de broadcast. Além disso, a comunicação entre os agentes poderia ter falhas, o que fazia com que em casos onde não existisse uma comunicação efetiva, o planejamento não era possível e, portanto, a ação dos agentes era degradada. Com isso, ficou decidido que seria usada a técnica intitulada Partial Global Planning (DURFEE; LESSER, 1991), técnica essa que também será utilizada no nosso projeto.

Essa técnica baseia-se no conceito de existir um planejamento ao mesmo tempo global e local, usando a comunicação apenas para o compartilhamento de informação. O planejamento local se refere ao planejamento das ações propriamente ditas, que é feito individualmente por cada agente. Já o planejamento global se refere às informações de nível mais alto de abstração que os agentes trocam entre si, de modo que as informações provenientes dos outros agentes possam ser levadas em conta no planejamento local.

3.1.3 Simulador da *RoboCup Rescue Simulation League*

O simulador usado na competição *RoboCup Rescue* tem como objetivo reproduzir virtualmente os efeitos de uma catástrofe em uma cidade, mais particularmente um terremoto. São representados edifícios, ruas e pessoas. Por efeito do terremoto, podem ocorrer incêndios, bloqueios nas vias públicas e soterramento de civis. Um time de agentes, agindo de modo coordenado, deve tentar impedir o alastramento do fogo, visando extingui-lo, e socorrer vítimas que foram soterradas.

O time é composto por três principais tipos de agentes: os *bombeiros*, os *policiais*, e as *ambulâncias*. Cada um dos agentes é responsável por realizar uma tarefa específica. Os policiais removem os bloqueios das ruas. Os bombeiros apagam os incêndios, desde que eles tenham água em seu reservatório. Para recarregar a água, eles devem ir até os refúgios espalhados pelo mapa. As ambulâncias conseguem retirar civis dos escombros e levá-los até os refúgios para serem atendidos. Os agentes realizam as ações descritas acima no intuito de diminuir o número de agentes feridos e mortos e controlar os incêndios que aparecem. O desempenho dos times é calculado pelo simulador, levando em conta a eficiência do time em cumprir tais tarefas.

A versão atual do simulador é descrita em mais detalhes em (SILVA; NARDIN; SICHMAN, 2013). Há dois arquivos que são lidos como entrada, um contendo o mapa que será utilizado na simulação, e o outro contendo o cenário do mapa que será utilizado. Além

desses dois arquivos, há também uma conexão estabelecida com cada um dos agentes que representam os policiais, bombeiros, ambulâncias e centrais.

Durante a simulação, pode-se notar a presença de alguns tipos de ocorrências no ambiente que dificultam a ação (os bloqueios) ou causam danos aos agentes (os incêndios). Os bloqueios são modelados no simulador na forma de polígonos fixos espalhados pelas ruas da cidade que impedem a circulação dos agentes pelo mapa. Os incêndios são responsáveis por colocar a vida dos agentes em risco. É interessante notar que os incêndios tendem a se espalhar pela cidade. A Figura 3 mostra uma tela do simulador, contendo os possíveis elementos do mapa.

Figura 3 – Simulador da competição durante uma simulação



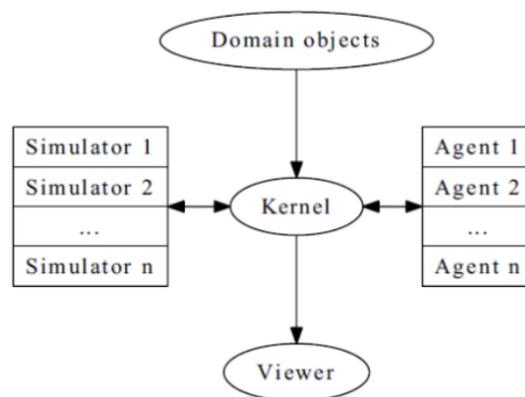
Fonte: Os autores

A arquitetura do simulador da competição pode ser dividida em cinco blocos, ilustrados na figura 4:

1. *kernel*: é o bloco principal e central da simulação. Ele é a entidade responsável por agregar todas as informações vindas dos outros blocos, além de controlar o seu funcionamento. É o kernel quem determina o andamento da simulação, lançando o funcionamento de cada um dos blocos;

2. *simuladores*: consistem de uma variedade de sistemas, sendo que cada um deles simula diferentes aspectos do ambiente em questão. Dentre eles, dois desses simuladores merecem destaque: o simulador de incêndio e o simulador de comunicação. O simulador de incêndio determina como os focos de incêndios irão se propagar, baseado num modelo que depende da temperaturas de cada um dos focos e da direção do vento. O simulador de comunicação indica se uma dada tentativa de comunicação conseguiu atingir seus alvos, e se houve algum tipo de erro durante essa transmissão;
3. *agentes*: cada agente possui um modelo interno do mundo e pode realizar ações no ambiente de simulação para atingir seus objetivos. O simulador define sete tipos de agentes: policiais, ambulâncias, bombeiros, centros de polícia, centros de ambulância, centro de bombeiros e civis. Com exceção do último tipo, todos os outros tipos de agentes são programados pelos usuários do simulador;
4. *visualizador*: é responsável por mostrar o estado atual do mapa aos usuários do simulador;
5. *objetos do domínio*: representam os objetos do mapa, como os prédios e as ruas, e guardam os seus estados.

Figura 4 – Arquitetura do simulador

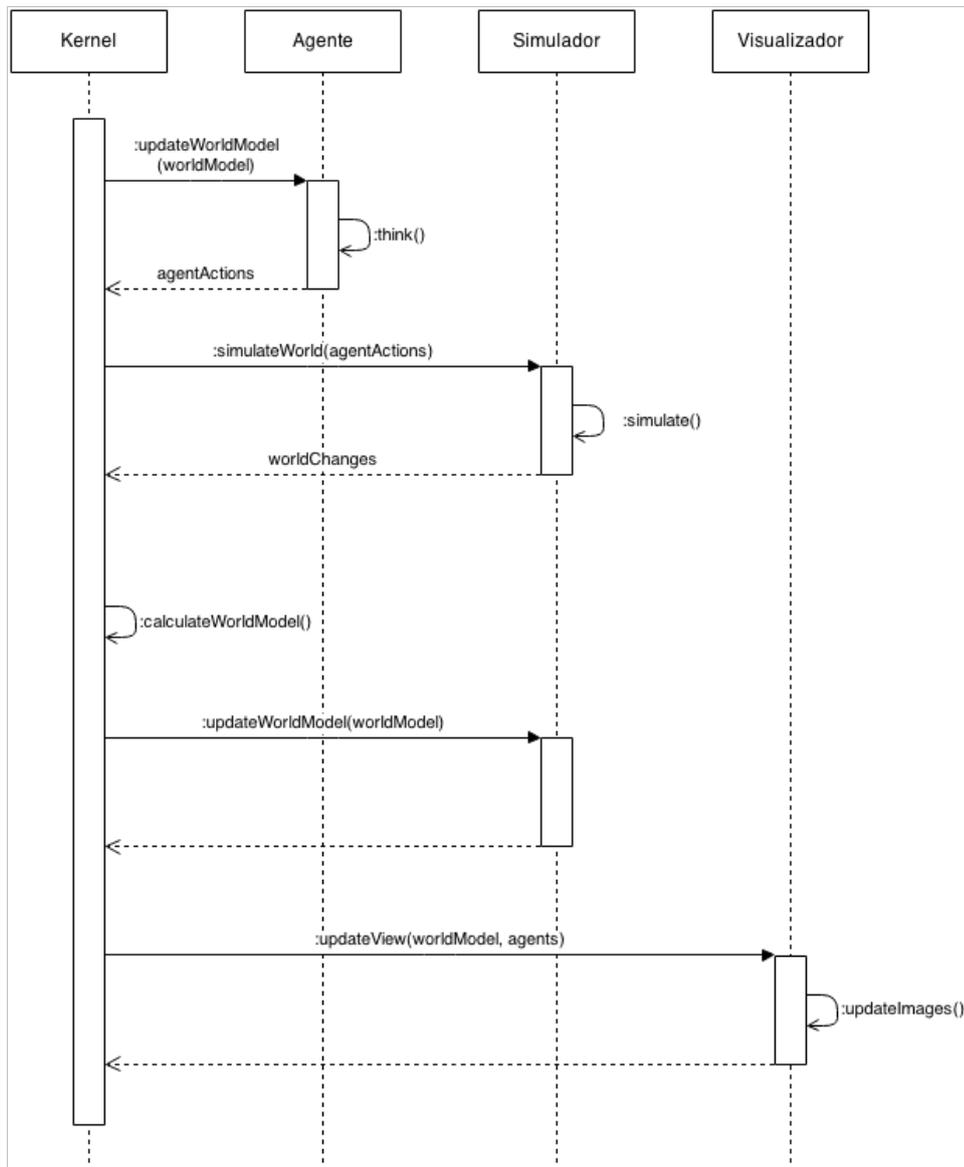


Fonte: SKINNER; RAMCHURN (2010)

O simulador funciona em um regime de ciclos. Durante cada ciclo, uma série de eventos acontecem, ilustrados no diagrama de sequências da Figura 5:

1. num primeiro momento, o kernel envia para cada um dos agentes o modelo de mundo atual acessível a cada um deles (cada agente somente percebe uma região do ambiente vizinha à sua localização);

Figura 5 – Diagrama de sequência de um ciclo do simulador



Fonte: Os autores

2. com esse modelo, os agentes então passam a calcular, segundo suas próprias estratégias, as próximas ações. Eles possuem um total de 500 milissegundos para efetuar este cálculo;
3. ao fim deste período, os agentes enviam as ações selecionadas ao kernel;
4. o kernel então envia aos simuladores o conjunto de ações tomadas pelos agentes;
5. de posse destas ações, os simuladores então calculam quais serão as mudanças decorrentes relativas ao fenômeno que de se ocupam;
6. quando terminam estes cálculo, os simuladores enviam as modificações ao kernel;

7. o kernel então reúne todas mudanças enviadas pelos diversos simuladores e calcula um novo modelo de mundo;
8. este novo modelo de mundo é então enviado para cada um dos simuladores para que, no próximo ciclo, calculem as mudanças baseando-se no modelo atual;
9. finalmente, no término o ciclo, o kernel envia o modelo de mundo para o visualizador;
10. o visualizador transforma o modelo em uma representação gráfica que será mostrada aos usuários.

3.2 Aspectos práticos

Além de ser baseado em diversos conceitos teóricos, o interesse do trabalho também reside em sua possível aplicabilidade a ambientes reais de desastres urbanos. Dessa maneira, torna-se necessário observar quais dentre as técnicas propostas podem e devem ser aplicáveis em situações reais enfrentadas pelas entidades públicas oficiais de resgate.

Nesse sentido, foi essencial conhecer as corporações que desempenham um papel fundamental nos cenários de desastres urbanos. A seguir, são apresentadas as principais entidades que foram relacionadas, o seu papel em caso de ocorrência de desastres e sua importância no projeto.

3.2.1 CEPED-USP

O CEPED, Centro de Pesquisas e Estudos em Desastres, surgiu de um acordo assinado no começo de 2013 entre a Universidade de São Paulo, o Ministério da Integração Nacional e a Defesa Civil, com o objetivo de disseminar e integrar os conhecimentos produzidos na Universidade sobre prevenção, mitigação e redução de desastres urbanos de grande escala. Esse centro de pesquisas é coordenado pela Escola Politécnica da USP, que desenvolve pesquisas sobre logística em ambientes de desastre, e conta também com a participação de diversas outras unidades da Universidade, que desenvolvem pesquisas relacionadas em diferentes áreas.

Além desse centro de pesquisas em desastres criado pela USP, existem outras entidades relacionadas, como o CEPED - UFSC, na Universidade Federal de Santa Catarina com foco em pesquisas e cursos de extensão voltados para grandes desastres, e o CEPD (Centro de Estudos e Pesquisas de Desastres), administrado pela prefeitura do Rio de Janeiro.

Através de conversas com membros do CEPED, situamos o nosso trabalho em relação às pesquisas que já foram realizadas e dos pontos de carência das mesmas. Além disso,

obtivemos mais conhecimentos sobre os aspectos práticos e gerais que estão envolvidos em eventos de desastres.

3.2.2 Entidades de resgate

Alguns dos principais alvos de estudo desse projeto são as entidades acionadas em casos de desastres urbanos, que agem na recuperação da cidade e na proteção de civis. Entre elas estão a Defesa Civil, o SAMU (Serviço de Atendimento Móvel de Urgência) e o Corpo de Bombeiros. Cada um deles possui um papel fundamental e diferente no salvamento de civis e minimização de danos ao patrimônio da cidade. Através do estudo de cada uma dessas entidades, nós pudemos modelar os métodos de coordenação por eles utilizados no ambiente de simulação, afim de avaliar o seu desempenho através de análises quantitativa.

A seguir são apresentadas cada uma dessas entidades e o seu papel em cenários de desastres (cf. Figura 6).

Figura 6 – Entidades de resgate que atuam em São Paulo



Fonte: Os autores

3.2.2.1 Defesa Civil de São Paulo

A Defesa Civil é o órgão responsável pela prevenção, mitigação e recuperação de eventos que possam levar perigo aos cidadãos de uma sociedade. A Defesa Civil de São Paulo tem como sua principal função coordenar as atividades de busca e salvamento em caso de desastres urbanos, bem como o de controlar os recursos necessários para missões e declarar casos de calamidade pública.

3.2.2.2 SAMU

O Serviço de Atendimento Médico de Urgência (SAMU), é um serviço oferecido pelo governo federal em parceria com governos estaduais e prefeituras, para o salvamento de pessoas gravemente feridas ou enfermas e que necessitam de um atendimento ágil e eficiente. Esse serviço fornece unidades móveis de salvamento, equipadas de aparelhos médicos e atendentes treinados, que podem agir no resgate e no salvamento de vidas de vítimas da maneira mais eficaz possível.

3.2.2.3 Corpo de Bombeiros

O Corpo de Bombeiros é responsável pelo salvamento de civis e recuperação urbana em casos de incêndios e acidentes graves, e possuem caminhões que possuem todos os equipamentos necessários para apagar os incêndios e prestar os primeiros socorros aos possíveis feridos no local. É um órgão fundamental em ambientes de desastres urbanos.

3.2.3 Integração das entidades

Atualmente, quando ocorre algum tipo de acidente com vítimas, alguma destas entidades é acionada pela população, através de uma chamada telefônica, por exemplo. Cada uma possui o seu mecanismo de coordenação interna independente, que visa maximizar a sua eficiência de atendimento. Em certos casos, mais de uma entidade é acionada pela população para a mesma ocorrência, o que é muito provável em caso de desastres. Nestas situações, cada uma delas tentará inicialmente agir independentemente da outra, tornando assim o atendimento desorganizado e ineficaz.

No intuito de conhecer melhor os procedimentos adotados em cada uma destas entidades, foram realizadas algumas visitas técnicas, descritas a seguir.

4 Visitas a entidades de resgate

Dentro do planejamento do nosso projeto, uma atividade prevista foi a realização de visitas às entidades de resgate da cidade de São Paulo. O objetivo destas visitas era propiciar um melhor entendimento do funcionamento desses organismos, para ser possível modelar e implementar algumas de suas práticas no time a ser desenvolvido para a *Robo-Cup Rescue Competition*. Nesse capítulo descrevemos as visitas feitas pelo grupo a essas entidades, mostrando o que foi aprendido e a relação do conhecimento adquirido com o desenvolvimento de novas estratégias nos agentes para as competições.

4.1 Relato de visita à Defesa Civil

A visita à Defesa Civil da cidade de São Paulo foi realizada no Centro de Operações da entidade no dia 3 de Maio de 2013, onde fomos recebidos pelo seu coordenador, o Coronel Jair Paca. A visita foi bem esclarecedora quanto ao papel da Defesa Civil nos salvamentos do município. A principal função da Defesa Civil é a prevenção de desastres e de recuperação após o acontecimento, e não a atuação direta em atividades de salvamento durante a ocorrência dos mesmos.

Um ponto interessante a se notar dessa visita foi o fato de que o Coronel, apesar de hoje ser o coordenador da Defesa Civil, já foi no passado o chefe dos bombeiros da cidade de São Paulo. O relato de sua experiência nos ajudou bastante na compreensão da importância dos bombeiros em casos de grandes desastres no município. De acordo com o Coronel, quando um grande desastre ocorre, a entidade que comanda a situação e que é responsável pelo resgate dos civis é o Corpo de Bombeiros.

Do ponto de vista da modelagem dos agentes, a visita à Defesa Civil não foi muito proveitosa. Como o simulador que nós utilizamos é mais voltado para as ações realizadas durante um grande desastre, as funções da Defesa Civil não podem ser modeladas no simulador e, portanto, acabam por fugir do escopo do nosso projeto.

No entanto, os conhecimentos do Coronel Jair Paca sobre o modo de ação dos bombeiros nos ajudaram bastante a compreender qual é a principal entidade a ser modelada em nosso projeto: os bombeiros.

4.2 Relato de visita ao COBOM

Foram realizadas duas visitas realizadas ao Centro de Operações do Corpo de Bombeiros (COBOM) da cidade de São Paulo, respectivamente em 28 de Junho e 5 de Julho

de 2013. Nestas visitas, fomos recebidos pelo Capitão Yuri Moraes Bedini e pelo Capitão Herbert Meyerhof. Em cada uma delas, tivemos a oportunidade de conhecer diversos aspectos da atuação dos Bombeiros, desde a estrutura e funcionamento do COBOM, até as estratégias e protocolos de salvamento utilizados nas ocorrências.

4.2.1 Atuação do Corpo de Bombeiros

Como dito anteriormente, as visitas ao COBOM foram feitas com o objetivo de analisar o funcionamento do Corpo de Bombeiros, e obter informações específicas sobre os mecanismos de atuação das brigadas de incêndio durante ocorrências reais. Cabe ressaltar, porém, que a atuação dos bombeiros não se limita ao combate a incêndios. Os mesmos tem outras áreas de atuação que incluem: (i) busca e salvamento, terrestre ou aquático; (ii) atendimento pré-hospitalar; (iii) prevenção contra incêndio e pânico; (iv) corte de árvores em risco iminente de queda. No entanto, nos restringimos apenas à atuação em incêndios, devido a pertinência do assunto com o nosso projeto.

Durante as visitas, obtivemos conhecimento sobre alguns protocolos e regras básicas que devem ser levados em conta em operações de salvamento. Além disso, pudemos identificar muitos mecanismos utilizados que foram adaptados e aplicados em forma de estratégias para a competição. A seguir, descrevemos resumidamente o funcionamento do COBOM e sua estrutura física, bem como alguns princípios e protocolos básicos utilizados durante uma operação.

4.2.2 Estrutura do COBOM e atendimento

Durante as visitas, fomos apresentados às instalações do COBOM, onde todas as chamadas de socorro são atendidas e onde é feito o acionamento das guarnições.

O COBOM (Centro de Operações do Corpo de Bombeiros) está localizado em uma única sala no prédio do Corpo de Bombeiros, no centro de São Paulo, na Praça Clóvis Beviláqua. A sua estrutura consiste de cerca de 30 cabines de atendimento, cerca de 10 cabines de coordenação e comunicação, uma cabine de integração com o SAMU, e uma cabine de comando e controle.

Em cada cabine de atendimento, um atendente recebe ligações feitas por pessoas para o número 193. Após o recebimento do relato da ocorrência, o atendente registra o ocorrido em um sistema de atendimento, onde ele insere o maior número de informações possíveis, de acordo com a riqueza de detalhes do relato.

Após o registro da ocorrência, os bombeiros das cabines de coordenação controlam a atuação das guarnições nas diferentes ocorrências. Acessando o sistema, tem-se acesso a algumas informações vitais da ocorrência, como a sua gravidade e o local do incêndio ou acidente, bem como as guarnições que encontram-se disponíveis para o resgate. O sistema

permite que o bombeiro escolha quais serão as guarnições a atuarem na ocorrência e atualize essas informações para que outros bombeiros sempre tenham a informação correta e atual para efeito de coordenação. No COBOM, as cabines de coordenação são divididas entre os diferentes grupamentos de bombeiros, ou seja, em cada cabine, o bombeiro tem acesso às guarnições disponíveis para o seu grupamento. Cada grupamento é responsável pelo atendimento a uma região específica da cidade, possui suas próprias guarnições e uma central, onde existe um estacionamento das viaturas.

No centro da sala de operações, existe ainda uma cabine utilizada para integração com o SAMU, com um televisor e canal telefônico, através dos quais o COBOM pode monitorar e se comunicar com o centro de operações de atendimentos por ambulâncias. Essa estrutura permite que os bombeiros tenham conhecimento da atuação das ambulâncias no atendimento a ocorrências da cidade, propiciando uma melhor integração e coordenação entre as duas entidades e um atendimento mais eficiente ao cidadão.

Na cabine de comando e controle, o chefe do centro de operações controla a atuação de suas equipes, através de dois sistemas principais: o primeiro deles é utilizado no monitoramento do atendimento às ocorrências, desde o momento do registro do chamado até a alocação dos recursos necessários. Essa ferramenta exibe uma lista com as ocorrências registradas, mostrando detalhes como local e status, até que cada uma seja atendida (tenha recursos alocados); o outro sistema possibilita monitorar a localização das guarnições através de um mapa interativo da cidade de São Paulo, através do qual o comandante pode analisar a disponibilidade de viaturas para eventuais ocorrências, de acordo com sua proximidade de diferentes locais.

Os elementos descritos acima mostram com mais detalhes o funcionamento do Centro de Operações do Corpo de Bombeiros de São Paulo. Esse conhecimento foi muito importante para o trabalho, para adquirir uma perspectiva mais prática no desenvolvimento do projeto, tendo a visão do que acontece nos centros de comando quando um incêndio é registrado e de como os bombeiros atuam alocando recursos e monitorando a operação.

4.2.3 Procedimentos básicos de atendimento

Toda a operação de combate a incêndio do corpo de bombeiros deve seguir um procedimento de cinco passos, representados por cada letra da sigla SICER (Salvamento, Isolamento, Confinamento, Extinção, Rescaldo). Em cada um desses passos, deve-se levar em conta uma série de fatores do ambiente, como a situação da edificação, o número de pessoas no local e as condições meteorológicas:

1. *Salvamento*: Essa etapa é a parte mais importante de qualquer operação de combate a incêndio, portanto ela é sempre a primeira missão a ser realizada pela brigada de

incêndio que estiver atuando na ocorrência. Para que o salvamento seja bem sucedido, o comandante da operação deve obter conhecimento sobre alguns fatores chaves, como o número de vítimas, o tempo de exposição ao incêndio e as condições físicas de cada uma delas. Além disso, o comandante deve sempre avaliar as possibilidades de salvamento, levando em conta a disponibilidade de recursos e reforços, bem como os perigos que o incêndio pode fornecer aos membros da operação. Assim, não se deve arriscar a perda de homens para o combate ao tentar salvar vítimas com remotas chances de sobrevivência.

2. *Isolamento*: Após retirar as vítimas dos locais e fornecer o atendimento adequado (realizado pelo próprio bombeiro ou por um médico no local), o maior risco oferecido pelo incêndio neste momento é a perda de controle do mesmo e seu alastramento. Assim, este passo consiste no combate para impedir que a área tomada pelo incêndio aumente, evitando que os danos e riscos sejam maiores que os já causados. Nessa etapa, é importante que o comandante saiba a atual extensão do incêndio e tenha consciência da área que ainda é possível proteger. Em casos de incêndios em prédios, onde o fogo se alastra para os andares de cima, o primeiro ataque deve ser no piso superior ao do local do fogo, de modo que esse possa ser resfriado e impedir o espalhamento. Por outro lado, em caso de incêndios em áreas abertas, ou em regiões onde as residências são próximas umas das outras, a primeira linha de ataque deve ser posicionada para proteger a área para onde o fogo pode se espalhar, levando em consideração fatores como a direção do vento e a proximidade do local. A tarefa de isolamento do incêndio torna-se crítica em casos em que este ocorre nas proximidades de uma planta ou objeto inflamável, que possam causar explosões caso expostos ao calor do fogo, levando a acidentes de proporções catastróficas. Esse cenário varia de casos simples como um incêndio próximo a um botijão de gás, até situações mais graves como ocorrências vizinhas a um posto de gasolina ou a uma fábrica de combustíveis. Portanto é necessário que a brigada de incêndio esteja preparada para agir rapidamente e com máxima prioridade em cada caso, podendo ser necessário evacuar a área e isolar a região de perigo.
3. *Confinamento*: Após assegurar o local e garantir que o incêndio não saia do controle, deve-se começar o combate ao fogo efetivamente. O primeiro passo para o combate é eliminar as possíveis fontes causadoras do fogo (combustível ou comburente), sendo que essa tarefa denomina-se confinamento. Na maioria dos casos, o confinamento do fogo consiste em impedir o contato com o ar (oxigênio), fechando as janelas do local e isolando completamente o ambiente, bem como evitando que o fogo entre em contato com qualquer material inflamável, como madeira por exemplo. Dessa maneira, extinguem-se as fontes de alimentação do fogo, possibilitando assim um ataque mais efetivo.

4. *Extinção*: Após realizar todos os procedimentos descritos anteriormente, deve-se realizar um ataque mais efetivo ao incêndio, utilizando os recursos de água disponíveis nas viaturas. Nessa etapa, o posicionamento dos bombeiros com as mangueiras d'água é fundamental para o sucesso da operação. Partindo do que foi dito na etapa de isolamento do incêndio, o combate ao fogo deve ser feito de maneira a circunscrevê-lo, para que não haja propagação a nenhum outro local. Portanto, os esguichos devem ser posicionados em torno do incêndio e caminhar em direção ao centro, eliminando o risco de alastramento e extinguindo o fogo com eficiência.
5. *Rescaldo*: Na grande maioria das vezes, após o fogo ter sido apagado, a presença de calor e de materiais combustíveis no local possibilita a re-ignição do incêndio. Nesse caso, os homens da patrulha poderão ser pegos despreparados e correrão sérios riscos, e o fogo poderá voltar a tomar o local. Portanto, é de grande importância que toda a tropa esteja atenta para retirar os materiais combustíveis do local e extinguir o incêndio por completo, utilizando para tal extintores para a eliminação de focos de calor.

4.2.4 Estratégias dos Bombeiros

Após as visitas realizadas, obtivemos um conhecimento mais profundo e detalhado de como é a atuação efetiva dos bombeiros. Porém, como era esperado, em muitos casos não existe aplicabilidade das técnicas e táticas utilizadas pelos bombeiros no cenário do simulador da competição. Isso se deve muitas vezes ao distanciamento que o ambiente do simulador apresenta da realidade e às suas limitações. No entanto, identificamos dois conjuntos de estratégias utilizadas em casos reais que poderiam servir de base para criarmos estratégias mais efetivas para a competição, descritos a seguir.

4.2.4.1 Divisão e repositonamento

Tais estratégias se referem a como as viaturas e equipes estão divididas e onde normalmente se encontram geograficamente:

- *Divisão em Grupamentos* O Corpo de Bombeiros da cidade de São Paulo é dividido em diversas regiões na cidade, cada uma delas chamada de Grupamento de Bombeiros (GB), que possui seus próprios recursos, sua própria central e é responsável por atender às ocorrências de sua região;
- *Viaturas nas Centrais*: Um fato que foi destacado pelos bombeiros durante as visitas é que as viaturas de bombeiros estão sempre paradas nas centrais do respectivo Grupamento de Bombeiros (GB) enquanto não há ocorrências. Isso permite que ela esteja sempre recarregada de água e preparada para atender a qualquer ocorrência;

- *Preposicionamento de viaturas*: Em alguns casos, por medidas de prevenção, o corpo de bombeiros posiciona suas viaturas em locais estratégicos da cidade, para que o possível bloqueio de alguma via não impeça que todas as viaturas de um grupamento (GB) possam chegar a um incêndio em um determinado local da cidade, comprometendo assim a operação. Um bom exemplo acontece em casos em que há previsão de chuvas fortes com possibilidades de alagamento sob viadutos. Nesses casos, as viaturas são divididas em dois grupos e cada grupo é posicionado em um dos lados do viaduto. Dessa maneira, mesmo em caso de alagamento, o atendimento a ocorrências em determinados locais da cidade não é comprometido.

4.2.4.2 Atuação efetiva

Tais estratégias se referem às diferentes ações que um grupamento de bombeiros pode tomar no combate a incêndios:

- *Abordagens de Ataque e Defesa*: No manual do Corpo de Bombeiros ([BOMBEIROS, 2003](#)), dois termos repetidos muito frequentemente são ataque e defesa. Eles definem as duas abordagens básicas de atuação dos bombeiros em situações de incêndio. Quando o bombeiro atua em modo de ataque, possui uma abordagem mais veloz e agressiva, onde o intuito é utilizar todos os recursos disponíveis, apontando os jatos de água diretamente para o foco do incêndio, até que esse se extinga por completo. Já quando ele atua em modo de defesa, o principal objetivo é preservar o patrimônio e as pessoas, estabelecendo bloqueios para o incêndio e resfriando lugares vizinhos com a água, impedindo assim que o fogo se alastre. Na maioria dos casos, essas duas abordagens são utilizadas em conjunto, pois muitas vezes as viaturas que chegam primeiro no local da ocorrência não possuem os recursos suficientes para utilizar a abordagem de ataque. Nestes casos, devem realizar a defesa do local para impedir que o fogo se alastre, até que os reforços cheguem ao local e então, com a disponibilidade de recursos, seja tomada a abordagem de ataque;
- *Circunscrição da área de incêndio*: De acordo com o manual de atuação dos bombeiros ([BOMBEIROS, 2003](#)), a técnica de posicionamento adequada para o combate ao incêndio consiste em circunscrever o fogo, fazendo com que os jatos de água estejam espalhados ao redor da área e sejam sempre direcionados ao centro do incêndio;
- *Posicionamento das primeiras guarnições*: Como visto anteriormente, a melhor maneira de posicionamento das viaturas em uma ocorrência é feita através da circunscrição do local do incêndio. Porém, em casos em que não é possível fazer a proteção de toda a redondeza ao redor da área tomada pelo fogo, deve-se realizar linhas de bloqueio, e a primeira linha deve ser posicionada contra o vento. Assim, as primeiras

viaturas que chegarem ao local devem se posicionar sempre de frente para a direção do vento, impedindo que o fogo possa se alastrar;

- *Uso de guarnição reserva:* Durante a visita ao COBOM, uma das estratégias que nos foi explicada foi o uso de uma guarnição reserva durante as ocorrências. O papel dessa viatura seria ficar estacionada perto do local do incêndio, e servir de substituta para as viaturas que ficassem indisponíveis, ou então para agregar-se a elas caso a intensidade do fogo aumentasse muito.

5 Competições e workshops

Durante o período do projeto, a equipe participou de duas competições de robótica na categoria *Robocup Rescue Simulation*, e de um workshop-escola na área de sistemas de agentes, denominado WESAAC 2013. A primeira competição, de âmbito mundial, foi a *RoboCup Rescue* 2013. A segunda foi a Competição Brasileira de Robótica 2013. Esta seção descreve as participações do grupo nesses eventos.

5.1 WESAAC 2013

No final de maio de 2013, nossa equipe participou da sétima edição do WESAAC¹ (Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações) realizada em São Paulo - SP, na USP. O principal objetivo do evento é integrar pesquisadores e estudantes de todos os níveis na área de Agentes e Sistemas de Agentes, divulgar as atividades dos diversos grupos de pesquisa do Brasil, possibilitando o intercâmbio de conhecimentos e experiências.

O evento é constituído de uma combinação de Oficinas e Palestras, proferidas por pesquisadores experientes, da academia e da indústria, e apresentações de Trabalhos Completos e Resumos Estendidos. Nossa equipe, além de assistir a algumas oficinas e palestras, teve a oportunidade de participar com um resumo estendido, aceito e publicado nos anais do WESAAC 2013 (BARROSO et al., 2013b), e de apresentar um pôster no workshop.

Essa participação agregou bastante conhecimento ao grupo, pois tivemos uma oportunidade de validar as nossas ideias para o projeto, além de poder receber um feedback construtivo dos nossos orientadores e de outros participantes do workshop que visitaram e leram nosso pôster. Por fim, a experiência acadêmica de se envolver com um evento de nível internacional foi de grande valia e nos preparou para as participações nas competições de simulação de agentes, realizada no meses seguintes, como explicado a seguir.

5.2 *RoboCup Rescue* 2013

A RoboCup é uma competição mundial anual que reúne alunos de graduação de todo o mundo para participar de desafios de robótica e computação. A RoboCup 2013² foi realizada na Holanda, na cidade de Eindhoven. A RoboCup possui uma grande variedade

¹ <http://www.wesaac2013.pcs.usp.br/>.

² <http://www.robocup2013.org/>.

de categorias, desde futebol de robôs até robôs que possam resgatar pessoas de um local perigoso.

Nosso objetivo nesse evento foi representar o time do Laboratório de Técnicas Inteligentes, denominado *LTI Rescue*, na competição da liga *Rescue Agent Simulation*, na qual o time tem regularmente participado desde 2010. Tal liga consiste de uma competição simulada, onde o cenário é uma cidade em uma situação de desastre. O objetivo dos participantes dessa liga é criar o comportamento de três tipos de agentes, Ambulâncias, Policias e Bombeiros, e tentar minimizar os danos causados pelo desastre.

5.2.1 Times participantes

A liga *Rescue Agent Simulation* teve um total de 13 times participantes em 2013, provenientes de vários locais do mundo, e em sua maioria vinda do Oriente Médio. A Tabela 1 apresenta todos os participantes da competição.

Tabela 1 – Lista de participantes da liga Rescue Agent em 2013

Nome do Time	País	Instituição
S.O.S	Tehran, Iran	Amirkabir University of Technology
MRL	Qazvin, Iran	Qazvin Azad University
Poseidon	Tehran, Iran	Farzanegan 1 High School
Ri-one	Shiga, Japan	Ritsumeikan University
GUC_ArtSapience	Cairo, Egypt	German University in Cairo
ZJUBase	Hangzhou, China	Institute of Cyber-Systems and Control, Zhejiang University
Apollo-Rescue	Nanjing, China	Nanjing University of Posts and Telecommunications
CSU-YUNLU	Changsha, China	Central South University
anct_resq2013	Asahikawa, Japan	Asahikawa National College of Technology
LTI Rescue	Sao Paulo, Brazil	POLI-USP
NAITO-Rescue2013	Nagoya ,Japan & Aichi Institute of Technology	Nagoya Institute of Technology
RoboAKUT	Istanbul, Turkey	Bogazici University
MinERS	Minneapolis, United States	University of Minnesota

Fonte: *RoboCup Rescue Simulation League*

É interessante notar que só houve dois times oriundos da América: um dos Estados Unidos e o LTI Rescue Agent, que foi o único time da América Latina nesta categoria.

5.2.2 Regras da competição

A competição durou um total de sete dias, começando no dia 24 de Junho de 2013. No início, três dias foram reservados para a preparação dos times e os últimos quatro dias foram aqueles de competição propriamente dita. Os três primeiros dias de preparação são reservados para que os times testem seus projetos e verifiquem que tudo está funcionando de maneira correta. No caso da categoria que participamos, os dias de preparação foram importantes para podermos verificar se não havia nenhum *bug* restante no nosso código e prepará-lo para ser lançado automaticamente nos computadores da competição.

Terminado os dias de preparação, os quatro dias de competição eram divididos em três partes. Uma fase eliminatória, de duração de dois dias, onde todos os times competiam entre si. A partir do resultado da primeira fase, oito times eram escolhidos para a fase semifinal, de duração de somente um dia. Usando os resultados somente da semifinal, quatro times são escolhidos para a final. No último dia, os quatro times da final competem para a escolha do vencedor. A dificuldade dos mapas aumenta conforme as fases progridem.

5.2.3 Infrastructure Competition

A *Infrastructure Competition* é uma variante da *Rescue Agent Simulation*, em que os participantes propõem mudanças no simulador da competição para melhorá-lo ou adicionar novos desafios. Nesta categoria, somente três times participaram como competidores, indicados na Tabela 2.

Tabela 2 – Lista de participantes da categoria Infrastructure Competition em 2013

Nome do Time	País	Instituição
S.O.S	Tehran, Iran	Amirkabir University of Technology
MRL	Qazvin, Iran	Qazvin Azad University
Hinomiyagura	Nagoya, Japan	Meijo University

Fonte: *RoboCup Rescue Simulation League*

Os times S.O.S e MRL participaram de ambas as categorias. Enquanto o time MRL propôs a adição de um quarto tipo de agente no simulador, um agente voador com a capacidade de tirar fotos do ambiente, o time S.O.S mostrou uma versão do simulador com um ambiente totalmente em 3D para se aproximar mais da realidade. Já o time Hinomiyagura exibiu uma maneira de usar o simulador em uma placa GPU para aumentar a velocidade de processamento.

Feita a apresentação dos participantes, os times da competição tinham então o direito de votar em seu favorito de acordo com certas categorias, como a utilidade que a

modificação traria a competição e seu aporte científico à comunidade. Finalmente, com a votação realizada, o time MRL foi o vencedor dessa competição.

5.3 CBR 2013

A Competição Brasileira de Robótica (CBR) é a principal competição científica nacional na área de robótica autônoma. O evento é composto pela Competição Brasileira de Robôs, promovida em conjunto com o IEEE, e pela Competição RoboCup Aberta do Brasil (Brazilian RoboCup Open). A CBR 2013 foi realizada do dia 16 a 20 de Outubro em Fortaleza/CE no Campus da UNIFOR, com a participação de mais de 100 equipes de todo o Brasil.

Além da equipe do *LTI Rescue*, outras duas equipes participaram na categoria de *Agent Rescue Simulation*. A participação no evento foi muito importante para o grupo, pois conseguimos realizar a avaliação final das estratégias que foram desenvolvidas, tanto as que foram criadas através da experiência e dos conhecimentos adquiridos na primeira competição, quanto aquelas desenvolvidas a partir da modelagem obtidas a partir da interação com as entidades de resgate de São Paulo.

5.3.1 Equipes participantes

A Competição Brasileira de Robótica desse ano, na categoria *Agent Rescue Simulation* contou com a presença de 3 equipes, apresentados na tabela 3.

Tabela 3 – Lista de participantes da Competição Brasileira de Robótica 2013

Nome do Time	Instituição
EPICENTER	Universidade Federal do Rio Grande do Sul
<i>LTI Rescue</i>	Universidade de São Paulo
Team Rescue UFPel	Universidade Federal de Pelotas

Fonte: Site da Competição Brasileira de Robótica 2013

5.3.2 Regras da competição

A competição brasileira foi um pouco diferente da competição mundial. Primeiramente, ela teve apenas um único dia de preparação, onde podiam ser testados os agentes para verificar que se comportavam como o previsto. A duração e a organização dos jogos entre os times foi outro fator diferente. A CBR 2013 teve apenas três dias de competição

propriamente dita, onde, em cada dia todos os times competiam entre si. Em cada dia, foram simulados dois mapas diferentes, com exceção do terceiro dia, onde apenas um mapa foi simulado. Para decidir quais seriam os ganhadores, os resultados de cada mapa eram normalizados segundo o maior pontuador e então somados. Assim, o time que tivesse mais pontos ao final dos três dias ganharia a competição.

6 Desenvolvimento dos agentes

Este Capítulo descreve as estratégias desenvolvidas para os agentes durante todo o período do projeto. O desenvolvimento foi dividido em três etapas. A primeira etapa corresponde às implementações realizadas para a competição *RoboCup Rescue* 2013. A segunda etapa contempla as estratégias para a Competição Brasileira de Robótica (CBR) 2013, quando incorporamos as ideias obtidas nas visitas aos órgãos públicos de resgate. A terceira e última etapa contém uma estratégia adicional implementada pelo grupo, fora do escopo de competições.

6.1 *RoboCup Rescue* 2013

Quando realizamos a especificação das estratégias que desejávamos implementar para a competição *RoboCup Rescue* 2013, ainda no primeiro quadrimestre de 2013, não havíamos tido muito contato com o código que utilizamos como base para o nosso desenvolvimento. Além disso, ainda não estávamos familiarizados com o funcionamento do simulador da competição. A seguir, apresentamos as principais modificações realizadas no código até a realização da competição internacional em Junho de 2013, algumas delas descritas em (BARROSO et al., 2013a).

6.1.1 Correção de funcionalidades

Logo que começamos a desenvolver as estratégias especificadas, nos deparamos com diversos problemas que conseguimos mapear como sendo proveniente do código que utilizamos como base, sendo assim importante modificar algumas funções básicas e lógicas que implementavam estratégias de forma incorreta. Esses reparos tomaram um tempo maior do que o esperado, porém trouxeram benefícios de médio e longo prazo, pois pudemos confiar nas funções básicas a partir de então.

6.1.2 Setorização eficiente do mapa

Uma das tarefas mais difíceis na competição é distribuir bem os agentes de cada equipe, para que estejam sempre próximos aos focos de incêndio e às vítimas de soterramento, podendo chegar assim rapidamente a esses locais. Uma das técnicas utilizadas para a distribuição dos agentes pelo mapa é a *setorização do mapa*, de modo que cada agente ou grupo de agente fica restrito a um determinado setor pré-designado no mapa.

A ideia dessa técnica é dividir o mapa, criando um setor para cada refúgio presente no mapa; deste modo, um agente de polícia será alocado para cada setor, realizando rondas

pelo seu setor e desfazendo eventuais bloqueios existentes. Caso haja excedente, os agentes de polícia restantes serão alocados para o desbloqueio preventivo.

O time de ano anterior havia criado um algoritmo para a setorização do mapa, porém somente acessível para os agentes de polícia e com um número de divisões que não eram otimizadas (SILVA; NARDIN; SICHMAN, 2012). Visto a importância da setorização em mapas como os utilizados em competições, extraímos e criamos uma classe de setorização fora do escopo dos agentes, permitindo que cada tipo de agente utilizasse a setorização como uma ferramenta independente.

Além disso, otimizamos o número de divisões, feitas na forma de um grid $M \times N$, de forma a garantir no mínimo dois agentes por setor, para tratar o caso de um deles iniciar a simulação soterrado. Também evitamos a formação de grids $1 \times N$ ou $M \times 1$, quando possível. Para ilustrar essa lógica, suponha que haja 14 (ou mesmo 15) policiais em uma simulação. Ao invés de criarmos um grid 1×7 , optamos por criar um grid 2×3 , que pareceu ser mais eficaz quando testado.

A setorização do mapa foi parcialmente baseada na estratégia utilizada pelos bombeiros, como visto na Seção 4.2.4.1. Assim como os bombeiros dividem a cidade em regiões, o mapa foi dividido em setores. Outra similaridade é a distribuição dos agentes. Os bombeiros dividem seus agentes de maneira a deixá-los espalhados pela cidade para conseguir atacar mais rapidamente os focos de incêndio. No caso da estratégia implementada, os agentes foram divididos para ter uma varredura mais abrangente do mapa.

6.1.3 Tratamento especial de postos de gasolina

Os postos de gasolina, adicionados esse ano à competição, servem basicamente como “catalisadores” quando ocorrem incêndios ao seu redor, aumentando a velocidade e intensidade da propagação do incêndio pelo mapa. Dessa forma, ao realizarmos testes com mapas que possuíam postos de gasolina, percebemos que era importante que os bombeiros os tratassem de forma especial, lhe dedicando maior atenção. No entanto, ao implementarmos a estratégia de manter um bombeiro como vigilante de cada posto de gasolina, percebemos que isso não era suficiente para limitar o incêndio. Cabe ressaltar que durante a competição percebemos que nenhuma equipe, mesmo a campeã, conseguiu desenvolver uma boa solução para esse caso.

6.1.4 Desbloqueio preventivo

No ambiente do simulador, quando um agente de ambulância retira um ferido que está soterrado sob escombros, ele deve colocá-lo na ambulância e levá-lo até o refúgio mais próximo para que o ferido possa ser salvo. Se a ambulância não conseguir levar o ferido a tempo para o refúgio, ele morre, diminuindo assim o desempenho da equipe na

competição. Muitas vezes, durante o salvamento de um ferido soterrado a ambulância encontra um bloqueio a caminho do refúgio. Se isso acontecer, ela deve esperar um agente de polícia chegar ao local para desfazer o bloqueio, perdendo assim um tempo precioso e arriscando a vida da vítima.

O desbloqueio preventivo visa prevenir que essa situação ocorra. Quando um agente de ambulância é acionado e recebe a tarefa de resgatar uma vítima em um determinado local no mapa, a ambulância envia uma mensagem aos agentes de polícia do mapa caso o resgate esteja impedido pela presença de bloqueios. O agente de polícia mais próximo se desloca então até o local da vítima, tentando desbloquear os caminhos entre a ambulância e a vítima, e entre a vítima e o refúgio. Dessa maneira, quando a ambulância estiver a caminho do refúgio não correrá riscos de ficar presa em um bloqueio e a vítima perder a vida.

O desbloqueio preventivo nos pareceu uma ótima ideia logo que escolhemos as estratégias a serem desenvolvidas, porém percebemos que nosso desbloqueio não era muito eficiente. Dessa forma, fizemos pequenas alterações, deixando para melhorar o desbloqueio de forma mais abrangente para a competição nacional, como explicado na Seção 6.2.

6.1.5 Escuta do pedido de ajuda de civis

Os agentes civis no simulador, quando estão soterrados, podem enviar uma mensagem com o conteúdo “help” ou “ouch”, pedindo ajuda. As entidades de resgate próximas ao civil podem escutar essa mensagem, e descobrir que existe uma pessoa em perigo naquele local. Essa informação pode então ser passada aos outros agentes para que alguém vá em direção ao civil para resgatá-lo.

Na solução implementada anteriormente, os agentes não escutavam essa mensagem, e precisavam entrar nas casas para encontrar as vítimas, o que aumentava o tempo de salvamento e punha em risco a vida do civil. A implementação desse mecanismo ajuda a aumentar as chances de salvamento desses civis e, conseqüentemente, melhora o desempenho dos nossos agentes.

6.1.6 Cálculo de prioridade nos resgates

Uma parte muito importante do processo de resgate efetuado pelas ambulâncias e pela própria polícia, através do desbloqueio de rotas, é decidir qual é a ordem de prioridade de resgate dos agentes. Identificamos que a versão anterior do código não levava em conta alguns fatores importantes no cálculo das prioridades, como o tipo de agente a ser resgatado (Ambulância, Policial, Bombeiro ou Civil). Como exemplo, deixar de salvar um Bombeiro pode prejudicar muito o desempenho da equipe ao longo da simulação, causando uma perda de pontos muito maior do que a que seria causada pela perda de um

Civil.

Com isso em mente, a nossa equipe definiu uma função que calcula a importância de um resgate, baseado no tipo de agente, na distância da ambulância ao local do resgate, além de dar um peso maior as vítimas com maior grau de ferimento.

6.1.7 Mudança na forma de envio de mensagens entre agentes

Foi implementado uma maneira de maximizar o número de mensagens que é transmitido por cada canal por cada agente, respeitando o fato de que vários agentes usam o mesmo canal. Isso melhorou muito a garantia do envio de mensagens se comparado à versão anterior. Cabe ressaltar, no entanto, que essa solução não é a melhor possível, pois não prioriza as mensagens por tipo e devida importância. Também não há nenhuma forma de redundância no envio de mensagens que podem ser perdidas em alguns ciclos. Essas melhorias são propostas na Seção 8.2.

6.2 CBR 2013

Durante os meses de Julho a Outubro de 2013, estudamos e especificamos as estratégias que seriam mais importantes para o aumento da eficiência e qualidade dos nossos agentes. Para podermos analisar o resultado do nosso desenvolvimento, nos inscrevemos para participar da Competição Brasileira de Robótica (CBR) 2013. Dessa vez com mais experiência, priorizamos a implementação das estratégias que aprendemos nas visitas e estudo das entidades de resgate de São Paulo, descritas a seguir.

6.2.1 Fecho convexo para distribuição dos bombeiros

Uma estratégia muito utilizada pelos bombeiros durante os incêndios é a formação de uma espécie de arco ao redor da área dominada pelo fogo, de modo a impedir que ele se espalhe para os locais em volta.

Pensando na necessidade de definição do posicionamento correto dos bombeiros, que nos foi explicada durante as visitas, uma das estratégias adotadas foi espalhar os agentes bombeiros ao redor da área de incêndio, fazendo com que os jatos de água estejam sempre direcionados para o centro do fogo, e assim tentando impedir que este se espalhe.

Para o posicionamento correto e eficiente dos agentes bombeiros ao redor da área de incêndio, utilizamos um sistema de alocação de tarefas, onde para N agentes, definem-se N posições ao redor da área em chamadas através de um algoritmo chamado de Fecho Convexo (LIU et al., 2012). O método utilizado para a criação do Fecho Convexo foi o da Varredura Graham (*Graham Scan*).

Para definir o conjunto de pontos utilizados no algoritmo, buscamos inicialmente um prédio que já tenha sido queimado ou que ainda esteja pegando fogo e que se encontre mais próximo de um agente bombeiro. A partir desse prédio, busca-se o conjunto de prédios, que já tenha sido queimado ou esteja pegando fogo, pertencentes ao cluster do prédio inicial. A partir desse conjunto de prédios, roda-se o fecho convexo para se obter um conjunto de prédios que esteja ao redor desse cluster e que devem ser utilizados como objetivo.

A partir da lista de prédios que é resultado do algoritmo, é feita uma distribuição dos prédios aos agentes bombeiros, de forma que eles fiquem ao redor do incêndio, evitando a sua propagação. Essa maneira de agir corresponde tanto aos modos de ataque quanto aos de defesa dos bombeiros. Caso o incêndio ainda esteja no começo o algoritmo retorna um prédio e todos os bombeiros passam a ter esse prédio como objetivo, no modo ataque. Se o incêndio já estiver espalhado, os bombeiros tentarão se distribuir ao redor para evitar sua propagação, como no modo de defesa dos bombeiros explicado na Seção 4.2.4.2.

6.2.2 Utilização otimizada das estruturas de dados

Um dos problemas identificados durante nossa participação na competição *Robo-Cup Rescue* 2013 foi o fato de que alguns algoritmos demoravam muito tempo para serem calculados, levando, em alguns raros casos, à perda da rodada do agente por este não responder antes do tempo limite. Uma das causas encontradas para a demora da resposta foi o fato dos agentes estarem buscando itens específicos em listas. Como o número de elementos tratados durante uma simulação da competição é elevado, o acesso ou a busca de um elemento contido em uma lista é consideravelmente lento.

Como solução para o problema, decidimos adotar um uso generalizado de *Hashsets*, principalmente para os conjuntos de objetos que compõem o cenário. A razão para o uso de *Hashsets* é simples. O acesso ou a busca de um elemento é feita com uma complexidade $O(1)$, enquanto que o mesmo é feito em complexidade $O(n)$ em uma lista, onde n é o número de elementos da lista. O uso desta estrutura aumentou o desempenho de nossos algoritmos, além de nos permitir o uso de algoritmos mais complexos de cálculo dentro das simulações.

Apesar de termos aumentado o uso de *Hashsets*, ainda existem alguns casos em que o uso de listas é obrigatório. Em certas ocasiões, os elementos a serem analisados precisam estar ordenados, o que nem sempre é possível com o uso dos *Hashsets*. Para esses casos específicos, manteve-se o uso das listas.

6.2.3 Caminho alternativo para o objetivo

Essa importante funcionalidade foi criada baseando-se no fato de que os agentes por vezes acabavam presos em bloqueios sem tentar caminhos alternativos. A fim de permitir encontrar um caminho alternativo para o objetivo, mantivemos na memória uma lista de caminhos que estão efetivamente bloqueados por um determinado período (15 ciclos foram usados para a competição). Essa lista de caminhos bloqueados é criada pelo próprio agente que não consegue passar por uma determinada área, mas poderia futuramente ser também transmitida a todos os agentes. É importante ressaltar que esses bloqueios devem ter uma “data de validade”, pois o caminho pode ser desbloqueado por policiais alguns ciclos depois.

Uma outra forma de que esses bloqueios expirem é transformando esses bloqueios em um objetivo para os agentes policiais, que assim que o completassem se comunicariam com os outros agentes informando da liberação da via. Cabe lembrar que isso não exclui a ideia já implementada, pois caso não haja comunicação é necessário que os bloqueios expirem de alguma forma.

6.2.4 Melhoria no desbloqueio de caminhos

A melhoria na maneira de desbloquear caminhos foi uma importante mudança que afetou muito a eficiência de toda a equipe, visto que a sua liberação de forma mais eficiente faz com que os outros tipos de agentes também possam fazer seu trabalho com mais agilidade. A mudança ocorreu em dois aspectos, detalhados abaixo:

- *priorização das áreas a serem liberadas*: os policiais passaram a ter como objetivo somente o desbloqueio de áreas e de pessoas, deixando de ter os bloqueios individuais como objetivo direto. Dessa forma, passamos a realizar desbloqueios de forma mais objetiva, para salvar humanos (agentes ou civis) e abrir entradas de prédios. Com isso, o ambiente passou a ficar com bloqueios retirados parcialmente, porém mais eficiente no que diz respeito aos objetivos finais de resgate, o que passou a aumentar a pontuação final e consequente eficiência dos agentes como um todo.
- *geografia do método de desbloqueio*: houve uma mudança no método de desbloqueio oferecido pelo simulador. Antes, era somente possível enviar uma identificação do bloqueio a ser retirado. A partir de 2013, esse método foi substituído por outro no qual é necessário especificar a direção do desbloqueio, através de coordenadas. Com isso, o bloqueio deixou de ser objetivo e o caminho a ser liberado é que se tornou o foco. Fizemos essa alteração para otimizar o desbloqueio do caminho, escolhendo então a direção que estivesse mais de acordo com os nossos objetivos, visando a liberação de vias e pessoas.

6.2.5 Identificação prévia de caminho já desbloqueado

Essa estratégia consiste na tentativa de identificar que um caminho a ser desbloqueado já o foi previamente, sem que um agente necessite se locomover até o local exato do desbloqueio, economizando de um a dois ciclos por objetivo já concluído. Isso se mostrou importante principalmente quando a comunicação é instável ou inexistente, pois não se sabe ao certo quais bloqueios os outros agentes já liberaram.

Antes da mudança, em muitos momentos, os policiais precisavam ir até a entrada do prédio para conferir se o mesmo estava liberado. Com essa estratégia adicionada, os policiais chegam próximo a entrada e já conseguem calcular se há ou não bloqueios que impedem a entrada no prédio. Não havendo bloqueios, já consideram este objetivo concluído e iniciam um novo. Isso torna os agentes policiais mais eficientes durante as simulações.

6.2.6 Pedido de resgate para agentes bloqueados

Essa ideia veio do fato que em certas ocasiões alguns agentes já começavam a simulação bloqueados, ou então ficavam bloqueados depois de alguns passos. Isto impedia que pudessem auxiliar nos objetivos do time. Além disto, desbloqueá-los não era considerada uma tarefa para os policiais, pois não estavam feridos. Com a implementação dessa estratégia e dando uma prioridade alta a agentes que estivessem precisando de ajuda por estarem feridos ou realmente bloqueados, conseguimos em poucos ciclos ter a maioria dos agentes trabalhando, aumentando assim a eficiência total da equipe. Vale ressaltar que tal estratégia só funciona com a comunicação existente e confiável, sendo necessária uma outra estratégia para o caso de falha ou falta de comunicação.

6.3 Estratégia adicional

Após o término da competição da CBR 2013, nossa equipe achou necessária a implementação de uma nova estratégia, mais eficiente, de busca de caminhos. O desempenho obtido em um dos mapas (*Paris*), que tinha uma topologia mais complexa, nos levou a procurar um algoritmo de busca mais eficiente.

Desde o código base de 2012, utilizava-se uma técnica simplificada para calcular o caminho mais curto entre dois pontos no mapa, através do algoritmo *Breadth First Search (BFS)*, que realiza uma busca em largura no Grafo até encontrar o Vértice desejado. No nosso caso, como buscamos o melhor caminho e esse pode muito bem não ser o que passa pelo menor número de arestas, o BFS não é o mais indicado.

Dessa forma, foi implementado um algoritmo mais complexo para o cálculo do melhor caminho, o algoritmo A*. O algoritmo do A* se baseia na criação de uma árvore

de busca, onde cada nó possui uma função de avaliação na forma $f(x) = g(x) + h(x)$, onde $g(x)$ representa a distância percorrida desde o ponto de saída até o nó e $h(x)$ é uma função heurística que estima a distância ainda a ser percorrida desde o nó até o objetivo.

Na estratégia implementada, para $g(x)$ foi utilizada a soma das distâncias euclidianas dos pontos centrais das vias percorridas até a via representada pelo nó. Já para $h(x)$ foi utilizada como heurística a distância euclidiana do ponto central da via representada pelo nó até o ponto central da via representada pelo objetivo.

7 Resultados obtidos

7.1 Participação nas competições

7.1.1 *RoboCup Rescue* 2013

Durante a fase eliminatória, cinco mapas diferentes foram rodados em cada um dos dias. No primeiro dia, nosso time não teve um bom desempenho, e terminou a classificação em 11º lugar. No segundo dia, aplicamos algumas mudanças no código e conseguimos melhorar o desempenho do time. O fato de um dos outros times ter feito mudanças no código e ter piorado seu desempenho também nos ajudou a melhorar a classificação. Apesar disso, ainda sim terminamos a fase eliminatória em 11º lugar, e em virtude deste fato não pudemos passar para a próxima fase.

A tabela 4 apresenta o resultado final da competição, com a classificação das treze equipes participantes.

Tabela 4 – Resultado final da liga Rescue Agent na *RoboCup Rescue* 2013

Nome do Time	Pontos	Rank
anct_resq2013	52	13
Apollo-Rescue	178	7
CSU-YUNLU	182	6
GUC_ArtSapience	213	3
LTI Rescue	93	11
MinERS	202	4
MRL	237	1
NAITO-Rescue2013	71	12
Poseidon	202	5
Ri-one	165	9
RoboAKUT	132	10
S.O.S	236	2
ZJUBase	176	8

Fonte: *RoboCup Rescue Simulation League*

Os pontos eram calculados através do *score* fornecido pelo simulador ao final da simulação. Os oito primeiros times passaram para a fase de semifinal, onde competiram entre si. No resultado da semifinal, quatro times foram classificados para a fase final: GUC_ArtSapience, MRL, Poseidon e S.O.S. O time egípcio GUC_ArtSapience sagrou-se campeão da competição.

7.1.2 CBR 2013

No primeiro dia, a performance do time *LTI Rescue* foi muito boa, terminando em primeiro lugar entre os três participantes. Foram simulados dois mapas, Berlin e México, sendo que o primeiro mapa só pode ser simulado uma única vez em razão de problemas técnicos. Comparando os resultados aos da *RoboCup Rescue 2013*, o time teve um rendimento muito melhor nestes mesmos mapas.

No segundo dia da competição, os mapas utilizados foram os de Kobe e Eindhoven. Nosso time nesse dia não conseguiu um bom desempenho, terminando em terceiro lugar em ambos os cenários. Uma das razões para este fato foi alta taxa de perdas na comunicação para os dois mapas que foram simulados. Isso nos mostrou o quão frágil é a performance da equipe em situações desse tipo, mostrando como esse deve ser um ponto a ser melhorado no futuro.

Já o terceiro dia da competição foi marcado pela utilização de um novo mapa, desenvolvido especialmente para esta competição: o mapa da cidade de Fortaleza, sede do evento. Em razão de problemas operacionais, o mapa só conseguiu ser simulado uma única vez. O time do *LTI Rescue* acabou na segunda posição nesse mapa.

Apesar das performances inferiores do segundo e do terceiro dia, graças o ótimo rendimento do time no primeiro dia, a equipe do *LTI Rescue* sagrou-se campeã da Competição Brasileira de Robótica na categoria *Agent Rescue Simulation*. A tabela 5 mostra o resultado final da CBR 2013.

Tabela 5 – Resultado final da Competição Brasileira de Robótica 2013

Classif. Final	Nome do Time	Pontos
1	LTI Rescue	4,35
2	Team Rescue UFPel	3,83
3	EPICENTER	3,81

Fonte: Site da Competição Brasileira de Robótica 2013

7.2 Comparação da evolução dos agentes ao longo do ano

Nessa seção, será analisada a evolução do nosso time durante o período do projeto. Para tal, foram feitas comparações entre as diversas versões do código, desde a versão inicial do time *LTI Rescue* desenvolvida anteriormente por outra equipe até a última versão construída para a participação na CBR 2013.

Para efeito de comparação, as variações do código foram separadas em quatro versões:

- **Código Base 2012 (v1)**: Versão recebida pela equipe com as estratégias implementadas no time *LTI Rescue* até 2012;
- **Versão *RoboCup Rescue* 2013 (v2)**: Código construído para a competição da *RoboCup Rescue* 2013, onde foram realizadas as correções de *bugs* e a adição de algumas estratégias mais simples;
- **Versão CBR 2013 (v3)**: Versão usada na Competição Brasileira de Robótica 2013. Essa versão teve melhorias e novas estratégias que priorizaram principalmente o comportamento dos agentes de polícia e dos agentes bombeiros;
- **Versão final 2013 (v4)**: Versão preparada pela equipe, mas não utilizada em nenhuma competição. Essa versão inclui uma implementação do algoritmo A* e a sua utilização na busca de caminhos no lugar do algoritmo de busca em largura.

Para realização dos testes, foram utilizados três mapas: *Test*, *Kobe* e *Paris*.

7.2.1 Apresentação dos mapas utilizados para os testes

7.2.1.1 Mapa *Test*

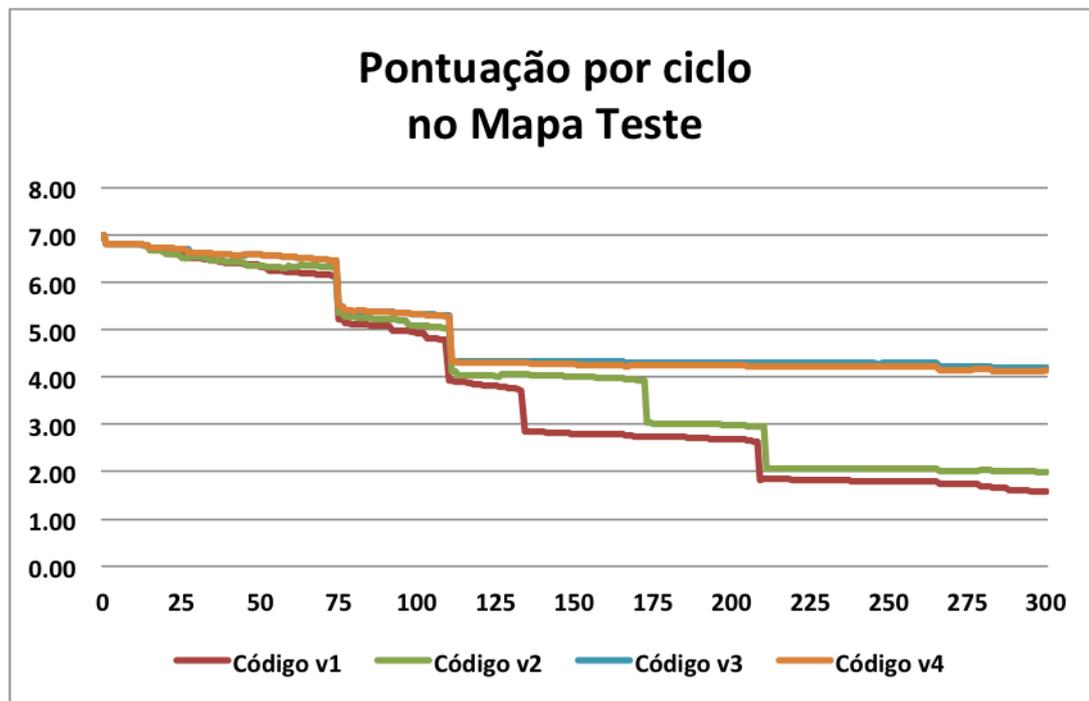
O mapa *Test* representa o menor mapa atualmente utilizado no simulador. Seu propósito é ser usado para a realização de testes rápidos; graças ao seu tamanho diminuto, sua execução é realizada de maneira rápida e não necessita de uma grande quantidade de memória. Como seu tamanho e o desafio proposto por esse mapa são limitados, ele não é utilizado oficialmente nas competições.

O gráfico da Figura 7 mostra a evolução da pontuação de cada uma das versões, simuladas uma única vez.

7.2.1.2 Mapa *Kobe*

O segundo mapa usado para os testes foi o mapa de *Kobe*, o menor entre os mapas oficialmente utilizados em competições oficiais. O mapa de *Kobe*, apesar de pequeno, apresenta grandes desafios, principalmente em alguns dos cenários existentes. A existência de grandes edifícios dificulta bastante o controle do incêndio, pois é praticamente impossível de extingui-lo sem que ele se espalhe antes. As ruas estreitas também aumentam a dificuldade desse mapa, já que a locomoção dos agentes fica restrita e, se não for bem coordenada, pode levar facilmente a caminhos sem saída.

O gráfico da Figura 8 mostra a evolução da pontuação de cada uma das versões, simuladas uma única vez.

Figura 7 – Progresso da pontuação de cada versão ao longo de uma simulação no mapa *Test*

Fonte: Os autores

7.2.1.3 Mapa *Paris*

O terceiro e último mapa usado para os testes foi o mais desafiador. O mapa de *Paris* é um mapa de grandes proporções, dificultando a ação dos agentes em casos onde eles não estão próximos do local onde o problema foi detectado. O gráfico da Figura 9 apresenta o progresso da pontuação de cada uma das versões, também simuladas uma única vez.

Além das grandes distâncias envolvidas, este mapa também apresenta dificuldades para resgate em prédios localizados fora das vias principais de locomoção. A única maneira de acessá-los é passando pelas pequenas ruas que saem das grandes vias, o que torna difícil o controle em casos em que o incêndio se inicia nesses prédios.

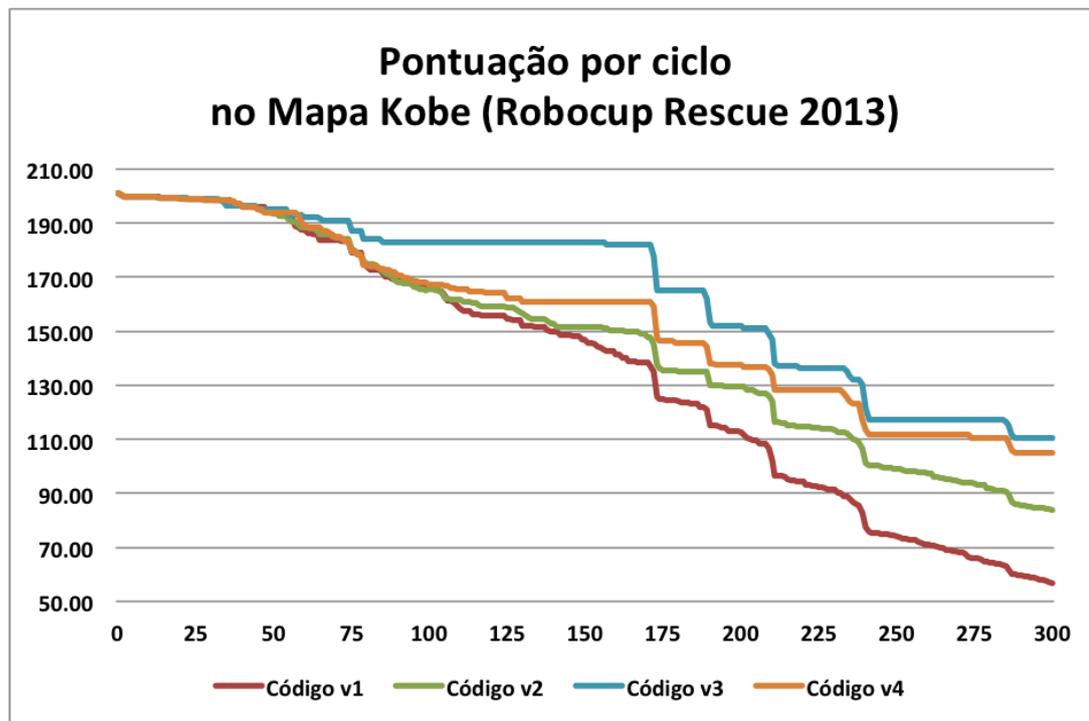
7.2.2 Análise dos resultados

7.2.2.1 Código Base 2012 (v1)

Antes de realizar a análise do impacto das evoluções de código feitas pela equipe, é interessante realizar uma comparação com o código base do time *LTI Rescue*, desenvolvido anteriormente. Quando comparados entre cada um dos mapas, o código base 2012 mostra uma pontuação bem inferior aos outros.

Estes resultados pouco satisfatórios se devem principalmente à existência de alguns

Figura 8 – Progresso da pontuação de cada versão ao longo de uma simulação no mapa *Kobe*



Fonte: Os autores

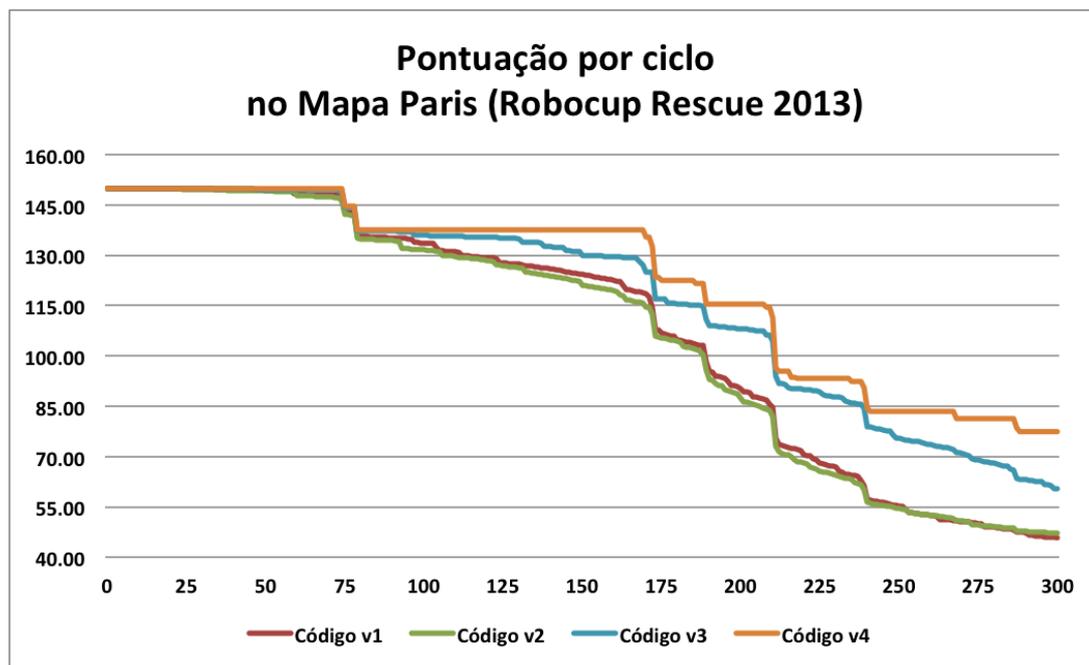
bugs. A setorização do mapa para os agentes de polícia, uma das principais estratégias desenvolvidas no código v1, não funcionava corretamente, e possuía problemas no momento do cálculo do número de setores necessários, podendo fazer com que os em alguns casos os agentes de polícia ficassem imóveis.

7.2.2.2 Versão *RoboCup Rescue* 2013 (v2)

A versão v2 do time já mostrou um grande avanço, principalmente devido à correção dos *bugs* inicialmente presentes no código da versão anterior.

A principal estratégia adicionada aos agentes foi o uso do desbloqueio preventivo. O desbloqueio preventivo mostrou bons resultados ao aumentar a mobilidade dos agentes pelo mapa, possibilitando que eles chegassem mais rapidamente às áreas de incêndio e aos civis que deveriam ser resgatados.

A setorização parcial das ambulâncias também ajudou bastante no aumento da pontuação geral. As ambulâncias, a partir desta versão, ao invés de utilizar um caminho aleatório generalizado, só caminham aleatoriamente dentro de seu setor. Esta versão diminuiu os casos em que mais de uma ambulância passa pelo mesmo local, deixando esses agentes mais espalhados pelo mapa e, conseqüentemente, aumentando as chances de encontrarem mais vítimas e de detetarem problemas.

Figura 9 – Progresso da pontuação de cada versão ao longo de uma simulação no mapa *Paris*

Fonte: Os autores

7.2.2.3 Versão CBR 2013 (v3)

Ainda que não apresente grandes mudanças na pontuação quando comparada à versão anterior, a versão v3 mostrou ser mais eficiente. A principal alteração diz respeito ao modo como os agentes de polícia realizam o desbloqueio. A rapidez com que os agentes liberam as vias principais aumentaram ainda mais a mobilidade das entidades de resgate, e fizeram com que os agentes agissem ainda mais rapidamente em resposta aos problemas que aparecem no mapa.

Essa versão do código foi a que apresentou o melhor desempenho, quando comparado às outras. O uso da técnica de circunscrição do incêndio, aprendida com as visitas realizadas às entidades de resgate públicas da cidade de São Paulo, melhorou muito a maneira como os agentes bombeiros atacam os prédios em chamas. O aumento da pontuação nessa versão se deve principalmente a esse novo comportamento dos bombeiros. O uso da técnica se mostra altamente eficiente em mapas de grande porte, como os dois últimos testados pela equipe. Como os incêndios tendem a se espalhar rapidamente, principalmente em mapas grandes, a técnica mostra uma maneira muito efetiva do controle do fogo.

Outra melhoria efetivada nesta versão foi a introdução de uma nova maneira de encontrar os caminhos no mapa. A adição da busca evitando bloqueios fez com que os agentes parassem de ficar travados em bloqueios, e começassem a procurar novas rotas que não passassem por caminhos bloqueados.

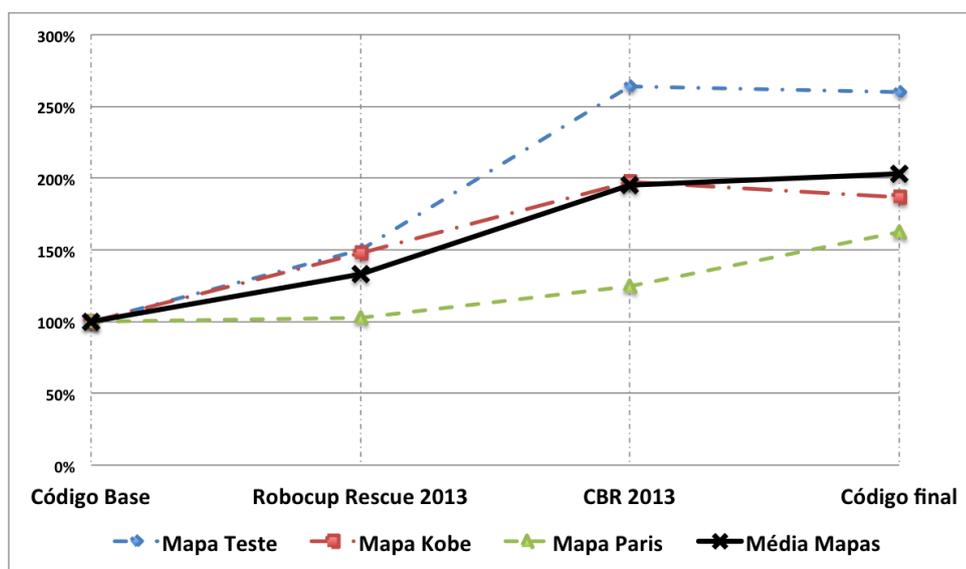
7.2.2.4 Versão final 2013 (v4)

A última versão do código, apesar de não ter melhorado a pontuação em todos os mapas, mostrou um aumento considerável na velocidade em que o código funciona. Ao trocar o uso da técnica de busca em largura para o A*, o número de ciclos necessários para encontrar o caminho até o objetivo diminui em vários ciclos. O uso da estratégia, apesar de não trazer melhorias diretas ao resultado final, abre a oportunidade de adicionar novas estratégias ao código pela reduzir o tempo de execução geral dos agentes.

7.2.2.5 Comparação entre as versões

O gráfico da Figura 10 mostra a evolução da pontuação obtida pelas versões através da média de 10 simulações para cada par (versão, mapa), comparadas com a versão v1. A tabela considera as pontuações da versão v1 como 100%, e as pontuações das outras versões divididos pelos resultado da v1, mostradas em termos de porcentagem.

Figura 10 – Evolução da pontuação do time *LTI Rescue*



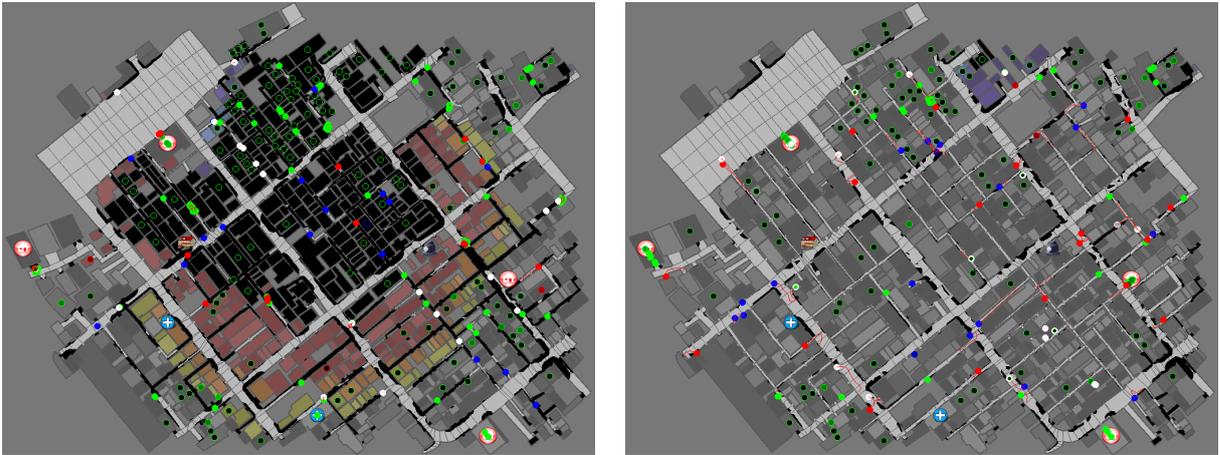
Fonte: Os autores

A tabela mostra como o código conseguiu evoluir durante o ano, chegando a ter quase o dobro da pontuação da versão inicial do código utilizada em 2012.

A figura 11 mostra uma comparação entre os resultados finais dos testes realizados no mapa de *Kobe*, comparando as versões v1 e v4. Já a figura 12, mostra a mesma comparação para os testes realizados no mapa de *Paris*.

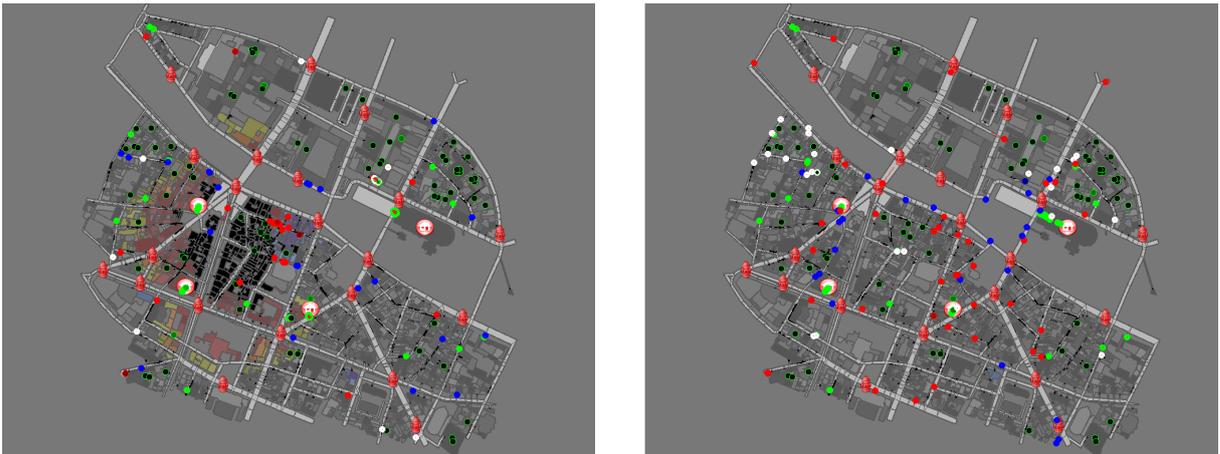
Apesar de calculadas as médias da evolução do código durante o tempo do projeto, os resultados não mostram realmente que o desempenho aumentou. As pontuações resultantes das simulações variavam de maneira expressiva, marcando assim um comportamento estocástico dos eventos capturados. Dessa maneira, para provar que uma versão

Figura 11 – Comparação para o mapa *Kobe*. A esquerda, os resultados da versão v1. A direita, os resultados da versão v4



Fonte: Os autores

Figura 12 – Comparação para o mapa *Paris*. A esquerda, os resultados da versão v1. A direita, os resultados da versão v4



Fonte: Os autores

do código foi melhor que a outra, não bastou comparar os resultados finais, foi necessário realizar um estudo estatístico em cima das pontuações.

A prova foi feita então baseada em um teste de hipótese onde se comparava as versões duas a duas, de maneira a provar que uma versão é melhor que as anteriores. Para realizar tal prova, foram coletados os resultados das 10 simulações feitas nos três mapas descritos acima, para cada uma das versões produzidas pelo projeto. Devido ao baixo número de simulações, o teste de hipótese utilizado foi não paramétrico.

Como exemplo, o seguinte teste de hipótese foi utilizado para o mapa Kobe com as versões v1 e v4:

- H_0 : A média dos scores do código v1 executado no mapa Kobe é maior ou igual que

a média dos scores do código v4 executado no mesmo mapa;

- H1: A média dos scores do código v1 executado no mapa Kobe é menor que a média dos scores do código v4 executado no mesmo mapa.

Em três dos dezoito casos estudados, concluímos que H0 não foi rejeitado e, portanto, precisamos efetuar um outro teste de hipótese para saber se as versões eram iguais ou diferentes. Como exemplo, o seguinte teste de hipótese foi utilizado nessa segunda etapa para o mapa Kobe com as versões v3 e v4:

- H0: A média dos scores do código v3 executado no mapa Kobe é igual à média dos scores do código v4 executado no mesmo mapa;
- H1: A média dos scores do código v3 executado no mapa Kobe é diferente da média dos scores do código v4 executado no mesmo mapa.

A Tabela 6 mostra o resultado dos testes de hipótese entre as versões indicadas para cada um dos mapas testados. O ✓ indica houve uma melhora entre a nova versão (maior numeração) em relação à versão antiga (menor numeração). O ✗ mostra que a versão mais antiga teve um desempenho melhor do que a versão mais recente. Finalmente, o - indica que a comparação entre as versões foi inconclusiva, não sendo possível determinar com significância estatística se houve uma efetiva melhora de desempenho entre elas.

Tabela 6 – Resultado dos testes de hipótese realizados entre versões do código.

Versões	Test	Kobe	Paris
v1 e v2	✓	✓	-
v1 e v3	✓	✓	✓
v1 e v4	✓	✓	✓
v2 e v3	✓	✓	✓
v2 e v4	✓	✓	✓
v3 e v4	-	✗	✓

Fonte: Os autores

A tabela indica que, para a maioria dos casos, houve uma melhoria no desempenho durante a evolução do projeto. É importante ressaltar que a versão final do código teve um desempenho melhor do que a versão base de 2012 para todos os mapas simulados.

A falta de melhora entre as versões v1 e v2 para o mapa de *Paris* se deve ao fato do mapa ter sido criado para a competição *RoboCup Rescue* 2013, ter uma geografia bem particular e só ter ficado disponível para testes após a competição, quando a versão v2 já estava pronta.

Observando os resultados da comparação entre as versões v3 com a v4, é possível notar que a última versão do código não possui melhorias significativas para a maioria dos mapas. No entanto, se compararmos as versões no mapa de *Paris*, a versão mais recente do código teve um desempenho muito superior àquele da versão para a CBR 2013, pois a equipe se concentrou na melhoria da pontuação dos agentes especificamente para este mapa.

7.3 Recomendação de melhorias para o simulador

As informações obtidas com os agentes de resgate da cidade de São Paulo nos inspiraram a implementar novas estratégias para os agentes do time *LTI Rescue*. No entanto, muitas dessas novas estratégias não chegaram a ser implementadas devido às limitações existentes no simulador usado pelo projeto. Na tentativa de superar as limitações impostas, o grupo sugeriu algumas mudanças ao simulador que abririam uma nova gama de possibilidades aos agentes.

As sugestões propostas abaixo também refletem outra barreira encontrada pelo projeto. De acordo com os especialistas das entidades de resgate, o ambiente simulado está longe de condizer à realidade, principalmente no que diz respeito ao alastramento da área de incêndio. Nosso grupo então propôs novas ideias ao simulador na tentativa de diminuir a distância entre o mundo real e o mundo virtual simulado. Além de promover a aproximação, essas mudanças podem ajudar os futuros projetos a aplicarem estratégias mais próximas daquelas usadas pelos agentes de resgate reais.

7.3.1 Modelagem dos agentes

Um aspecto muito importante do simulador, que possui discrepância com a realidade, é a modelagem dos agentes, ou seja, a maneira como cada tipo de agente é representado no sistema e as suas funções nas operações de resgate. O simulador representa três tipos de agentes: os bombeiros, que possuem um certo volume de água inicialmente e são responsáveis por apagar os incêndios no mapa; os policiais, que devem desbloquear as ruas com bloqueios, liberando a passagem para outros agentes; e finalmente as ambulâncias, que são responsáveis por desenterrar e resgatar os civis. Porém, na realidade, as funções desempenhadas pelos agentes de polícia e de ambulância no simulador são realizadas pelos próprios bombeiros, que possuem os equipamentos necessários para fazer cada uma dessas operações.

Além disso, em situações reais, cada viatura dos bombeiros possui uma série de capacidades diferentes, podendo haver uma viatura que pode desenterrar feridos e apagar incêndios. Dessa forma, a modelagem atual dos agentes do simulador não condiz com os papéis de cada uma das entidades de resgate públicas de São Paulo. Uma possível

melhoria seria tornar a modelagem dos agentes mais flexível, possibilitando extensões ou mudanças para adequá-los a situações mais reais.

7.3.2 Ambiente de simulação

Durante os estudos e pesquisas sobre as estratégias utilizadas pelo Corpo de Bombeiros (COBOM), dois fatores de discrepância ficaram evidentes: o fogo se alastra muito rapidamente no simulador, e o alastramento parece levar em conta que dois prédios vizinhos sejam adjacentes. No primeiro caso, foi apontado pelo Capitão Herbert Meyerhoff (Coordenador do COBOM) que a velocidade com que o fogo se alastra no simulador é muito maior do que aquela da realidade, tornando muitas estratégias inviáveis. No segundo caso, foi destacado que na cidade de São Paulo, assim como em outras grandes cidades, os prédios não são localizados tão próximos uns dos outros, fazendo com que o alastramento do fogo na realidade demore mais do que na simulação. Portanto, uma possível melhoria do simulador seria levar em conta um possível afastamento entre os prédios, tornando o alastramento do incêndio mais lento e portanto mais condizente com o mundo real.

7.3.3 Funcionamento e comportamento dos agentes

Além de fatores relacionados à simulação do ambiente, foram identificadas outras discrepâncias que o simulador apresenta em relação ao mundo real. Muitas dessas diferenças estão relacionadas com o comportamento de cada agente: em muitos casos, os agentes do simulador não agem de maneira condizente com o que as entidades reais fariam. Isso acontece principalmente pelo fato de que, por ser simplificado, o modelo dos agentes não possibilita certas ações que seriam necessárias no mundo real.

Em situações reais de incêndios de grande porte, onde a necessidade do abastecimento de água é constante, as viaturas dos bombeiros abastecem umas às outras, havendo muitas vezes um rodízio entre as viaturas que estão sendo abastecidas e as que estão combatendo o incêndio. Essa técnica é muito importante no combate, pois permite que um conjunto de agentes que estiver atacando o incêndio não fique desabastecido.

Além disso, uma outra diferença encontrada é que no simulador os bombeiros são as viaturas propriamente ditas, ou seja, ao apagar um incêndio, os bombeiros devem ficar próximo à área de incêndio, permitindo um combate mais rápido e eficaz. Porém, essa proximidade do incêndio pode trazer perigos aos agentes, de modo que em situações reais a viatura dos bombeiros para a uma distância segura do prédio em chamas, e um dos agentes do Corpo de Bombeiros deve sair e realizar o ataque ao incêndio.

Portanto, para as situações acima, uma possível melhoria seria a criação de modelos mais genéricos e flexíveis para os agentes, de modo que suas ações fossem menos limitadas, permitindo que os agentes do simulador pudessem ter um comportamento mais próximo

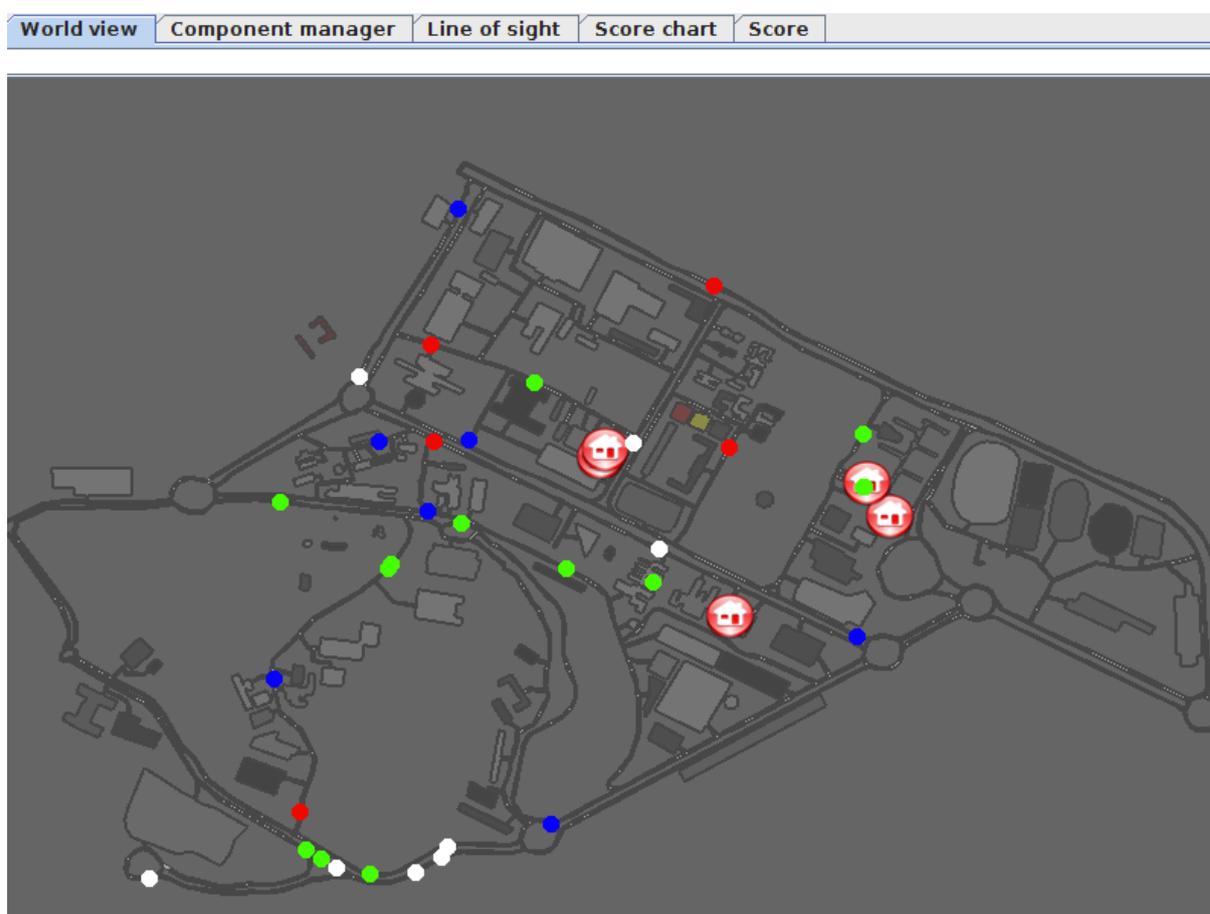
da realidade.

7.4 Criação do tutorial para o desenvolvimento de novos mapas

Um dos objetivos do nosso projeto também foi o de facilitar a realização de simulações em áreas que atualmente não têm mapas disponíveis para uso no RCRS. Para tal, desenvolvemos um tutorial (ver Apêndice D) que contém uma explicação detalhada de como criar um mapa qualquer, detalhando os processos, as ferramentas e os conceitos envolvidas no processo de desenvolvimento de novos mapas para o simulador. São descritos no tutorial conceitos e ferramentas como o OpenStreetMap, JOSM e GML, além dos processos de captura da imagem de satélite usada como base, do processo de conversão para o formato GML e da criação de um cenário com o posicionamento dos elementos no mapa.

A título de ilustração do processo completo de criação de um mapa, decidimos escolher a região do Campus da Cidade Universitária da USP em São Paulo como exemplo realizado pela equipe. A Figura 13 mostra o mapa criado com base na geografia das ruas, avenidas e prédios da Cidade Universitária. Esse mapa, entretanto, acabou não sendo utilizado para a realização de testes de desempenho por ter sido finalizado no último mês do projeto. No entanto, futuras equipes que trabalhem com o simulador poderão fazer uso deste mapa para testes e mesmo oficialmente em uma competição.

Figura 13 – Mapa da Cidade Universitária para o Simulador da RoboCup Rescue



Fonte: Os autores

8 Conclusões

8.1 Considerações finais

O primeiro objetivo deste trabalho foi o de incorporar novas estratégias e novas técnicas de coordenação aos agentes do time *LTI Rescue*, com o intuito de participar das diversas competições existentes dentro da liga *Robocup Rescue Simulation*. Segundo esse propósito, o projeto obteve sucesso, tanto na melhoria de desempenho dos agentes, quanto nos resultados nas competições.

Analisando os dados provenientes dos testes realizados com a versão inicial da equipe e com as mais variadas versões desenvolvidas para o projeto, pode-se verificar uma melhoria significativa dos agentes. A melhoria é mais ainda acentuada quando são analisados os resultados em cenários de médio a pequeno porte, como o mapa de *Kobe*, onde se obteve um aumento de 198% na pontuação. É possível observar, a partir da média dos resultados, uma melhora de aproximadamente 203% na pontuação em relação à versão inicial do time.

O segundo objetivo do projeto durante seu desenvolvimento foi o de se aproximar dos entidades de resgate reais da cidade de São Paulo, para poder aprender mais as técnicas utilizadas no mundo real. A equipe teve a oportunidade de realizar duas visitas: uma ao Centro de Operações da Defesa Civil e outra ao Centro de Operações dos Bombeiros (COBOM). Analisando os resultados obtidos das visitas, o grupo pode aprimorar as técnicas aplicadas nos agentes do time *LTI Rescue* de modo a melhorar seu desempenho.

A aplicação de técnicas reais usadas pelos agentes de resgate da Cidade de São Paulo se mostrou eficiente, principalmente se analisados os resultados referentes ao desempenho do time durante a *RoboCup Rescue 2013* e a CBR 2013. Para a CBR 2013, a implementação da técnica de circunscrição da área de incêndio conseguiu melhorar o resultado dos agentes em 195% em relação aos agentes do código base de 2012.

A evolução observada nos agentes mostra que o uso de técnicas reais pode ajudar muito na melhora do desempenho dos agentes do time *LTI Rescue*. Entre os trabalhos futuros aconselhados para o time, recomenda-se um estudo mais aprofundado da coordenação dos agentes reais. A aplicação desse tipo de técnica pode levar a uma evolução ainda maior dos agentes do time do *LTI Rescue*.

Apesar das evoluções mostradas pelo projeto referentes ao desempenho dos agentes, não foi possível implementar todas as técnicas inicialmente planejadas para o projeto. A familiarização da equipe com o ambiente de desenvolvimento também pode proporcionar ideias de evoluções durante o desenrolar do projeto que, por terem vindo tardiamente,

não chegaram a ser implementadas. As técnicas imaginadas pela equipe que podem levar uma melhora não só no desempenho do time, mas também facilitar o desenvolvimento futuro de técnicas de coordenação estão descritas na seção 8.2.

As visitas aos agentes reais de resgate da cidade de São Paulo também renderam ideias de como melhorar e tornar o simulador da competição mais próximo da realidade. O simulador, apesar de ser interessante como uma ferramenta para o uso na competição, está ainda muito longe da realidade, principalmente em relação à maneira como o incêndio se espalha. A equipe pode então traçar algumas mudanças que poderiam ser incorporadas ao simulador, de maneira a torná-lo mais próximo da realidade.

O terceiro e último objetivo do projeto foi o de construir um mapa que refletisse uma realidade mais próxima da cidade de São Paulo. O grupo então desenvolveu um novo mapa baseado na estrutura real das ruas e dos prédios do Campus da Cidade Universitária em São Paulo. Durante esse processo de criação do mapa, a equipe pode criar um tutorial para facilitar o desenvolvimento de novos mapas. Apesar de ainda não ter sido publicado oficialmente na página da competição, o tutorial pode ajudar a comunidade participante da competição a desenvolver uma nova série de mapas.

Como explicado no Capítulo 2, acreditamos que cada etapa do projeto foi fundamental para o seu sucesso, desde o estudo teórico sobre sistemas multiagentes, até a participação nas competições, realizadas paralelamente às visitas às entidades de resgate públicas de São Paulo. Em cada uma dessas etapas, adquirimos conhecimentos e desenvolvemos técnicas que foram fundamentais para o desempenho nas competições de simulação, culminando com o primeiro lugar na competição CBR 2013.

8.2 Trabalhos futuros

Apesar do bom resultado obtido no desenvolvimento das versões do time *LTI Rescue*, sabemos que há muito ainda a ser melhorado. Dessa forma, discutimos nesta seção o que acreditamos que possa ser adicionado e modificado no código final do time para um melhor desempenho em futuras competições e trabalhos relacionados.

8.2.1 Mudanças estruturais

8.2.1.1 Visão de mundo adaptada

Para explicar a motivação da criação desta estratégia, é necessário lembrar o funcionamento de um ciclo do simulador utilizado nesta competição. De forma simplificada, a cada rodada os agentes recebem informações sobre o ambiente, dentro do seu campo limitado de visão, e mensagens enviadas que estejam no seu campo de audição. O agente

então delibera, e pode decidir ou não executar uma ação no ambiente e enviar mensagens aos demais.

Quando executamos um algoritmo, queremos utilizar o cenário mais completo possível, devendo processar os dados recebidos em todo o ciclo, para formar um modelo mais completo e consolidado. Esse modelo deve, portanto, incluir todas as listas de bloqueios conhecidos, vítimas encontradas e prédios em chamas, além das posições correntes dos agentes e do cenário que o *World Model* já fornece. A ideia dessa alteração é de criar uma classe Estado, que represente todas essas informações, deixando os dados mais consistentes e permitindo que se armazene uma lista de Estados que representará a memória do agente.

Essa estrutura permite que qualquer variável que seja considerada importante para a tomada de decisão, como por exemplo a posição dos agentes e civis, estado dos prédios e bloqueios das vias, passe a ser disponibilizada de forma organizada. Além disso, os algoritmos implementados podem utilizar tais informações para realizar cálculos mais complexos e precisos.

8.2.1.2 Máquina de Estados

Esta ideia consiste em transformar a estrutura do código em uma máquina de estados. O modo como nossas estratégias tinham sido criadas se aproximava bastante da lógica de funcionamento de uma máquina de estados; no entanto, a implementação era feita através do uso de códigos sequenciais, usando comandos *if-then-else*.

Apesar de funcionar corretamente, a implementação de novas estratégias era dificultada: (i) ao criar uma nova estratégia, encaixá-la no fluxo de raciocínio do agente era uma tarefa complicada, sendo necessário em alguns casos repetir partes do código em vários comandos *if-then-else*; (ii) adicionar uma nova condição ao comportamento de um agente também se tornava complicado, pois as prioridades das ações dos agentes se baseavam na ordem em que as comparações eram feitas, e tais adições poderiam, como efeito colateral, influir em outros comportamentos.

Dessa maneira, decidimos criar uma estrutura baseada em uma máquina de estados para podermos melhor organizar nosso código e aumentar a velocidade de implementação de novas estratégias. A máquina de estados, como nós a imaginamos, parte da estrutura de um agente que é modelado com objetivos e um modelo de mundo (WOOLRIDGE, 2001). Cada estado indica quais são os objetivos atuais dos agentes, definindo quais serão suas ações em cada turno. As transições entre estados são calculadas a partir do modelo de mundo do agente e dos objetivos atuais.

A implementação da máquina foi baseada em uma das maneiras como a plataforma de desenvolvimento multiagente *Jade* lida com o comportamento dos agentes (BELLIFE-

MINE; POGGI; RIMASSA, 2001). A cada ciclo, no momento em que o simulador dispara a função *think()* dos agentes, cada agente usa seu conhecimento do mundo e o estado corrente para calcular a ação que deve ser executada e qual será o próximo estado. A partir deste próximo estado, o agente pode então considerar diversas alternativas de ação futuras.

8.2.2 Particionamento na comunicação entre agentes

Até a última competição, a comunicação entre os agentes era feita de maneira idêntica para todos os tipos de agentes, fortemente baseada no uso da estratégia de coordenação *Partial Global Planning* (PEREIRA; NARDIN; SICHMAN, 2012). Essa estratégia tem como um de seus fundamentos o compartilhamento das informações, dos problemas e do conhecimento do mundo de cada agente (DURFEE; LESSER, 1991). Até então, os agentes enviavam, a cada turno, as mudanças que ocorreram no seu modelo de mundo, usando o canal que possuía a maior banda no mapa.

Apesar de interessante, o modo como a estratégia foi implementada causou problemas em alguns mapas. Durante uma simulação, como havia um grande número de elementos em constante mudança, o tráfego de informações do canal de comunicação acabou sendo caótico e sobrecarregado, causando sérios problemas na transmissão das informações. Como resultado da sobrecarga do canal, os bombeiros não foram informados de um foco de incêndio que estava ocorrendo, ocasionando a queima de boa parte do mapa.

Como solução para esse tipo de problema, houve tentativas de usar um método de revezamento dos agentes dentro do mesmo canal. Cada agente só poderia se comunicar a cada dois turnos, ou seja, metade dos agentes se comunicavam durante os turnos pares e a outra metade durante os turnos ímpares. Mesmo com a aplicação dessa estratégia, o resultado indesejado continuou acontecendo.

Decidimos então criar uma nova estratégia de comunicação para os agentes. Ao invés de todos os agentes se concentrarem no uso de um único canal para o envio da informação, decidimos usar um método de particionamento da informação, para que os agentes enviem as diversas partes da mensagem através dos vários canais disponíveis no mapa. Isso tende a acabar com a sobrecarga de um único canal, além de permitir uma eficiência maior no uso dos canais de um cenário. Além do uso da divisão das mensagens, decidimos usar um modo de compressão das mensagens para que elas tenham o menor tamanho possível dentro de cada um dos canais disponíveis.

Uma outra estratégia de comunicação que também foi estudada para implementação foi uma mudança na informação que é transmitida entre os agentes. Como explicado acima, os agentes enviam entre si as mudanças ocorridas em seus respectivos modelos de

mundo. Acreditamos que um modo mais efetivo seria a transmissão de dados já processados pelos agentes; eles calculariam previamente, baseados em seus modelos de mundo, as tarefas que precisariam ser realizadas, e enviariam aos outros essa lista de tarefas já calculada. O uso dessa estratégia deve permitir uma coerência maior entre as tarefas que devem ser realizadas por cada agente, aumentando sua eficiência e, como consequência secundária, diminuindo o tamanho das mensagens enviadas.

8.2.3 Estratégias específicas em caso de comunicação instável ou inexistente

Foi observado durante a competição nacional que nossa equipe estava com um desempenho bom em mapas com comunicação de boa qualidade, porém com um desempenho ruim em mapas com comunicação instável ou inexistente. Esse comportamento pode ser explicado pelo fato de nossa estratégia depender da comunicação dos agentes para poderem se coordenar e trabalharem juntos. É importante, no entanto, que eles sejam capazes de realizar a mesma tarefa quando só puderem contar com os conhecimentos próprios.

No nosso caso, os agentes não consideravam uma ação de percorrer o mapa em busca de possíveis civis pedindo resgate ou de focos de incêndio, pois recebiam mensagens dos outros agentes relatando ocorrências. Uma possível solução para esse problema seria criar uma estratégia de patrulhamento, que deveria ser usada por todos caso não houvesse comunicação, ou somente por alguns caso houvesse comunicação. Para o caso de comunicação com altas taxas de perdas de mensagens, uma comunicação com garantia de recebimento ou redundância no envio poderia identificar ou sanar tais casos.

8.2.4 Posicionamento nas primeiras guarnições

Essa estratégia foi baseada no modo de coordenação utilizado pelos bombeiros, que posicionam seus caminhões de maneira a bloquear o incêndio de forma mais eficiente (Seção 4.2.4.1). No mundo real, nos casos em que a circunscrição é difícil, os bombeiros se posicionam contra o vento. Para o caso da simulação dos agentes, deve ser feita uma adaptação dessa estratégia, já que o vento não parece ser um fator importante no alastramento do fogo.

A nova estratégia seria baseada na análise dos prédios localizados na borda do incêndio que estejam em chamas. Os bombeiros então encontrariam o conjunto de prédios cuja temperatura está acima da média dos outros prédios, e esses prédios então passariam a ser considerados como a região crítica da zona de incêndio.

Encontrada a zona crítica, uma parte dos agentes então se posicionaria perto dessa região e, usando os jatos de água, diminuiriam a temperatura dessa zona. A diminuição da temperatura teria como consequência a diminuição da velocidade de alastramento do

incêndio.

Referências

- AMARAL, R. do; GUTJAHR, M. R. *Desastres Naturais*. v. 8, 2011.
- BARROSO, A. D. et al. RoboCup Rescue 2013 - LTI Agent Rescue Team Description. Maio 2013.
- BARROSO, A. D. et al. Using agent coordination techniques to support rescue operations in urban disaster environments. In: *Anais do VII Workshop-Escola de Sistemas de Agentes, seus Ambientes e apliCacções*. [S.l.: s.n.], 2013. p. 189–191.
- BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. Developing multi-agent systems with jade. In: CASTELFRANCHI, C.; LESPÉRANCE, Y. (Ed.). *Intelligent Agents VII Agent Theories Architectures and Languages*. [S.l.]: Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 1986). p. 89–103. ISBN 978-3-540-42422-2.
- BOMBEIROS, C. de. *O Comando Tático de Operações de Incêndio*. [S.l.], 2003.
- DURFEE, E.; LESSER, V. Partial global planning: a coordination framework for distributed hypothesis formation. *Systems, Man and Cybernetics, IEEE Transactions on*, v. 21, n. 5, p. 1167–1183, 1991.
- GUHA-SAPIR, D. et al. *Annual Disaster Statistical Review 2010: The Numbers and Trends*. [S.l.], 2011. Disponível em: <http://www.cred.be/sites/default/files/ADSR_2010.pdf>.
- KELLY, C. H. *Drones: The future of disaster response*. 2013. Disponível em: <<http://whatsnext.blogs.cnn.com/2013/05/23/drones-the-future-of-disaster-response/>>.
- KITANO, H. et al. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Tokyo, JP: IEEE Press, 1999. v. 6, p. 739–743.
- LIU, R. et al. A fast convex hull algorithm with maximum inscribed circle affine transformation. *Neurocomputing*, v. 77, n. 1, p. 212 – 221, 2012. ISSN 0925-2312.
- PEREIRA, A. H.; NARDIN, L. G.; SICHTMAN, J. S. LTI Agent Rescue: A partial global approach for task allocation in the robocup rescue. *Revista de Informática Teórica e Aplicada*, v. 19, n. 1, p. 71–92, 2012.
- SILVA, A. B. M. da; NARDIN, L.; SICHTMAN, J. S. Um método baseado em particionamento para exploração de ambientes de desastre. In: *Anais do ENIA 2012*. Curitiba, PR: [s.n.], 2012.
- SILVA, A. B. M. da; NARDIN, L. G.; SICHTMAN, J. S. *RoboCup Rescue Simulator Tutorial*. São Paulo/SP, Brazil, 2013. Disponível em: <http://roborescue.sourceforge.net/2013/robocup_rescue_simulator-tutorial.pdf>.

SKINNER, C.; RAMCHURN, S. D. The robocup rescue simulation platform. In: HOEK, W. van der et al. (Ed.). *AAMAS*. IFAAMAS, 2010. p. 1647–1648. ISBN 978-0-9826571-1-9. Disponível em: <<http://dblp.uni-trier.de/db/conf/atal/aamas2010.html#SkinnerR10>>.

WOOLRIDGE, M. *Introduction to Multiagent Systems*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 047149691X.

Apêndices

APÊNDICE A – Resumo estendido - WESAAC 2013

Using Agent Coordination Techniques to Support Rescue Operations in Urban Disaster Environments

Alan D. Barroso, Felipe de C. Santana, Victor Lassance, Luis G. Nardin, Anarosa A. F. Brandão, Jaime S. Sichman*
Laboratório de Técnicas Inteligentes (LTI) – Escola Politécnica (EP) – Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto, 158 – trav. 3 – 05508-970 – São Paulo – SP – Brasil
{alan.barroso, fesantana, victor.lassance.silva, luis.nardin}@usp.br, {anarosa.brandao, jaime.sichman}@poli.usp.br

Abstract—This extended abstract describes a task allocation and coordination policy that aims to maximize the efficiency of teams when they are rescuing the victims and protecting the city’s heritage in the case of an urban disaster. Our approach considers the existence of local and global information that could help this coordination. Supposing that communication is limited and unreliable, we also present a comparison of those coordination techniques, aiming to improve the task allocation process. This abstract comprises our motivation, the main goal and the key conceptual aspects of the research, as well as the development steps and specification of the whole project.

I. INTRODUÇÃO

A expansão urbana observada no Brasil nas últimas décadas causou o crescimento acelerado e desordenado das cidades, potencializando assim a ocorrência de desastres, principalmente em áreas mais densamente povoadas. Além disso, o Brasil irá sediar eventos de dimensão internacional, como a Copa do Mundo de 2014 e os Jogos Olímpicos de 2016, aumentando as consequências de um possível desastre urbano.

No caso específico de desastres naturais, pesquisas e dados científicos recentes indicam que eles vêm se tornando mais frequentes, intensos, dinâmicos e complexos [1]. No Brasil, alguns dos desastres que têm ocorrido com maior frequência e que poderiam ter um trabalho de prevenção esbarram em custos altos, o que faz com que os responsáveis posterguem a implementação da solução do problema. Dessa forma, como ocorre com o caso de enchentes e deslizamento de terras, o gerenciamento de desastres deve ser efetivo para minimizar os danos causados, através de ações eficientes de coordenação e alocação dos reduzidos recursos existentes em um ambiente parcialmente observável.

Nesse trabalho será proposta uma abordagem de alocação de tarefas que considera a existência de informações locais e globais com o objetivo de coordenar as entidades de salvamento para que maximizem a eficiência no resgate a vítimas e na proteção do patrimônio da cidade em caso de desastre. Supondo uma comunicação limitada e incerta, esse trabalho também apresentará uma comparação de técnicas de coordenação de forma a melhorar a alocação de tarefas.

Na seção II são apresentados os principais tópicos conceituais utilizados nesse trabalho, que são Sistemas Multiagentes e Técnicas de Coordenação como o *Partial Global Planning*. Já a seção III descreve a estrutura de fases de implementação des-

te trabalho, enquanto na seção IV detalha-se sua especificação por meio da descrição dos quatro algoritmos de coordenação que serão implementados. A seção V traz uma breve conclusão, descrevendo nossos próximos passos.

II. ASPECTOS CONCEITUAIS

Para melhor entendimento da abordagem proposta nesse trabalho, nesta seção são apresentados os aspectos conceituais relacionados às escolhas quanto às estratégias de coordenação de tarefas. O primeiro aspecto é referente ao modo como o projeto pode ser visto como um sistema multiagente. O segundo é referente ao tipo de técnica de coordenação escolhida.

A. A competição Robocup Rescue e seu simulador

A adoção do paradigma multiagente na abordagem apresentada para alocação de tarefas é fortemente dependente do domínio do problema em que estamos trabalhando, em particular do simulador da competição *Robocup Rescue*¹. O simulador possui três blocos principais: *kernel*, simuladores e agentes. O *kernel* serve como bloco central e junta todas as informações pertinentes aos problema. Os simuladores têm como função principal simular os diversos fatores que influenciam o andamento de uma rodada da competição. O terceiro bloco é composto por agentes programáveis, que são responsáveis por reduzir as perdas dentro do ambiente de desastre.

Os agentes programáveis possuem um certo grau de autonomia: a decisão de realizar ou não uma ação depende somente da vontade do próprio agente. Esses agentes autônomos devem então se organizar e se coordenar para poder resolver o problema proposto na competição. De acordo com [2], a descrição do problema acima se aproxima bastante à um sistema multiagente. Esse fator possibilita ao projeto o uso de técnicas de coordenação e comunicação já estudadas em sistemas desse tipo, como as descritas a seguir.

B. Técnicas de Coordenação

De acordo com [2], existem três abordagens principais em técnicas de coordenação para sistemas multiagentes: planejamento centralizado para planos distribuídos, planejamento distribuído para plano centralizado e planejamento distribuído para planos distribuídos. A primeira abordagem diz respeito a uma figura central que faz todo planejamento das tarefas e

(*) Jaime S. Sichman is partially supported by CNPq and FAPESP, Brazil.

¹<http://www.robocup2013.org/?lang=en>.

comunica aos respectivos agentes suas tarefas. A segunda leva em conta que todos os agentes entram em cooperação para elaborar um plano comum a todos. Na terceira abordagem, a mais complexa, cada agente elabora individualmente um plano próprio, tendo um objetivo comum de cooperar com os outros agentes.

Escolhemos a terceira abordagem por duas razões principais, relacionadas ao modo como a comunicação entre agentes do simulador é feita. Para um agente se comunicar com os outros, ele leva um ciclo de simulação e, para receber a resposta, um segundo ciclo, caso a mensagem chegue corretamente ao destinatário. Isso dificulta o uso da primeira técnica de forma competitiva, pois o tempo envolvido para distribuir os planos entre os agentes seria muito grande. Além disso, a comunicação entre agentes chega até mesmo a ser inexistente em alguns cenários, fazendo com que a segunda abordagem também não possa ser selecionada.

Por essas razões, foi escolhida a técnica de planejamento distribuído para planos distribuídos, mais especificamente através do uso da técnica denominada *Partial Global Planning*. A técnica, descrita em [3], baseia-se na existência de dois níveis de planejamento simultâneos, um local e outro global. No primeiro, o agente cria o seu próprio plano; já no segundo, os agentes trocam informações para que os planejamentos locais levem em conta os conhecimentos dos outros agentes.

III. FASES DO PROJETO

Esse trabalho visa analisar os métodos de coordenação atualmente utilizados pelas entidades de proteção, para poder sugerir a integração destas diferentes entidades, através de um mecanismo de comunicação criado com forte embasamento teórico e prático. Para atingir esse objetivo, o trabalho deve ser dividido em três fases, cada uma essencial para o resultado final.

A. Fase 1 - Competição Robocup Rescue

Essa fase do projeto está voltada para a participação do grupo na competição *Robocup Rescue*, que ocorrerá em Eindhoven, Holanda, em junho de 2013. A *Robocup Rescue* é uma competição internacional que tem como objetivo colocar em prática técnicas de coordenação de agentes para diminuir os danos e as perdas humanas causadas em um ambiente de desastre. No ambiente de simulação, são utilizados agentes com diferentes papéis, como do Corpo de Bombeiros, Agentes de Polícia e Ambulâncias, cada um deles com funcionalidades específicas. Um dos membros do grupo, Luis Gustavo Nardin, participou desta competição nos anos 2011 e 2012, como detalhado em *LTI Agent Rescue Team Description* [4] [5].

O desenvolvimento das técnicas de coordenação envolve o estudo dos aspectos teóricos e práticos que envolvem a competição como um todo, desde a teoria sobre sistemas multiagentes até as regras da competição e o funcionamento do simulador. Portanto, a participação na competição será essencial para o andamento do nosso trabalho, pois com ela poderemos adquirir os conhecimentos teóricos sobre sistemas multiagentes e técnicas de coordenação, bem como conhecer o funcionamento do simulador da *Robocup Rescue*, que utilizaremos em fases posteriores do projeto.

B. Fase 2 - Modelagem e análise de estratégias atuais de coordenação

Está será uma das fases essenciais do projeto, pois a partir dela o trabalho tomará um viés mais prático. Tentaremos aplicar os conhecimentos adquiridos anteriormente para poder modelar e analisar quantitativamente os métodos de coordenação atualmente utilizados pelas entidades de proteção em cenários de desastre.

Para analisar os métodos de salvamento atuais, serão pesquisadas as técnicas de coordenação das equipes de resgate (Bombeiros, SAMU e Defesa Civil) da cidade de São Paulo, através de visitas aos seus centros de operações e contatos com seus responsáveis. Quando essas informações tiverem sido coletadas, as entidades de resgate serão modeladas no ambiente do simulador e, posteriormente simuladas. A partir destes resultados, uma análise do funcionamento desses agentes será feita para determinar a melhor forma de integrá-los.

C. Fase 3 - Aplicação das técnicas de coordenação na integração das entidades de salvamento modeladas e análise dos resultados

Essa será a última e conclusiva fase do projeto, na qual serão integradas as estratégias de coordenação e comunicação criadas na fase 1 com o comportamento modelado na fase 2. Após realizada a junção das estratégias criadas e dos métodos de coordenação aplicados, o desempenho da solução final será avaliado quantitativamente através do simulador, analisando então quais foram os ganhos trazidos pela integração entre as diferentes equipes, e pelo uso de algumas técnicas adicionais de coordenação. Para realizar essa análise de desempenho, serão definidos parâmetros quantitativos referentes à simulação, como por exemplo o número de civis resgatados e o número de prédios que foram destruídos pelo fogo. Dessa forma, o resultado poderá ser expresso em porcentagem de melhora ou piora de cada um dos parâmetros selecionados.

Esta terceira fase é fundamental para o projeto, demonstrando seus resultados conclusivos: mostrar como o uso de algumas técnicas específicas de coordenação e a integração entre as diferentes entidades pode maximizar a eficiência no resgate a vítimas e na proteção do patrimônio da cidade em caso de desastre.

IV. ESPECIFICAÇÃO

Esse trabalho utiliza como base o código do time implementado pela equipe do LTI, participante da competição de 2012. Adicionalmente, apresentamos quatro principais algoritmos de coordenação para melhorar o projeto existente: (i) o particionamento baseado em refúgios para os policiais, (ii) o desbloqueio preventivo, (iii) a inclusão de ações nos agentes de resgate ao escutar, na simulação, o pedido de ajuda dos civis, e (iv) a determinação da eficiência de um resgate de acordo com parâmetros pré-definidos. Outras técnicas serão estudadas para uma implementação futura; no entanto, devido ao curto tempo restante até a competição, estas não serão implementadas por completo para a competição desse ano.

É interessante ressaltar que como esses algoritmos são voltados para a competição *Robocup Rescue 2013*, estes nem sempre irão refletir de maneira idêntica a forma como os

agentes de resgate se comportariam em um ambiente real. Os algoritmos descritos abaixo visam exclusivamente aumentar o *score* final do time na competição.

A. Particionamento do mapa baseado em refúgios

Uma das tarefas mais difíceis na competição é bem distribuir os agentes de cada equipe, para que estes estejam sempre próximos aos focos de incêndio e às vítimas de soterramento, podendo chegar assim rapidamente a esses locais. Uma das técnicas utilizadas para a distribuição dos agentes pelo mapa é o particionamento do mapa, de modo que cada agente ou grupo de agentes fique restrito a um setor designado no mapa.

A ideia dessa técnica consiste em reservar áreas próximas aos refúgios para que um policial fique em constante ronda em torno de um refúgio. Os policiais alocados nessa tarefa terão como objetivo retirar os bloqueios encontrados nessa região, na tentativa de limpar a entrada e os principais caminhos próximos ao refúgio. Essa técnica será utilizada somente em mapas que contém mais do que um número mínimo definido de policiais disponíveis, pois deverá haver um número mínimo de policiais para efetuar um desbloqueio preventivo.

B. Desbloqueio preventivo

Dentro do ambiente do simulador, quando uma ambulância decide salvar um civil preso nos escombros, ela deve se locomover até o local e tirá-lo dos destroços. Para finalizar o processo de salvamento, a ambulância deve ainda transportar o civil machucado do local do acidente até um refúgio para que ele possa ser atendido pelos médicos. Um dos grandes problemas encontrados durante esse processo é a presença de bloqueios entre a ambulância e o civil, e entre o civil e o refúgio. A presença desses bloqueios impede temporariamente a ambulância de alcançar o civil e/ou levá-lo até um refúgio, causando a morte do mesmo e consequentemente a diminuição do *score*.

Uma solução para o problema, utilizada atualmente, é chamar um agente de polícia no momento em que a ambulância encontra um bloqueio. No entanto, o tempo levado para o agente de polícia atingir o bloqueio e retirá-lo pode aumentar o tempo de chegada no local do acidente.

Dessa forma, é proposta a utilização de uma técnica de desbloqueio preventivo: uma ambulância, ao determinar que irá salvar um civil, envia uma mensagem de *broadcast* para todos os agentes de polícia. O agente de polícia mais próximo do civil começa então um processo de busca e remoção de bloqueios entre o civil e a ambulância e entre o civil e o refúgio. Esse processo tem por objetivo principal aumentar as chances de sobrevivência do civil e, consequentemente, do nosso *score*.

C. Inclusão de ações ao escutar pedido de ajuda dos civis

No simulador, os agentes civis, quando soterrados, podem enviar uma mensagem pedindo ajuda para serem socorridos. Os agentes de salvamento próximos ao civil podem escutar essa mensagem e descobrir que existe uma pessoa em perigo naquele local. Essa informação pode então ser repassada aos

outros agentes para que alguém vá em direção ao civil para resgatá-lo. Na solução implementada atualmente, os agentes não estão tomando ações baseados nessa mensagem, e precisam entrar nas casas para encontrar as vítimas, o que aumenta o tempo de salvamento. A implementação desse mecanismo pode ajudar a aumentar as chances de resgate e, consequentemente, melhorar o *score* do nosso time.

D. Determinação da eficiência de um resgate

Nem sempre é possível chegar no local do acidente a tempo de salvar um civil. Em certas situações, devido à presença de um incêndio, por exemplo, os civis morrem antes mesmo que a ambulância chegue ao local. Para diminuirmos o desperdício de recursos ao deslocar uma ambulância até o local do incêndio e não ser capaz de salvar o civil, uma solução é de ponderar a ida de uma ambulância a um local ao invés de outro pela eficiência calculada do resgate. Essa eficiência pode levar em conta, por exemplo, o tempo de deslocamento até o local da vítima, a presença de bloqueios conhecidos no caminho e a presença de incêndios nas proximidades do local onde o civil se encontra.

Como dito anteriormente, essa técnica visa somente aumentar o *score* da competição e não reflete necessariamente o comportamento real de uma ambulância. Em uma situação real, esse cálculo poderia levar em conta outros fatores que não estão presentes no simulador e, portanto, não podem ser incluídos na competição.

V. CONCLUSÕES

Neste trabalho, foi brevemente apresentado nosso projeto de um time de agentes para resgate de civis em situações de desastre. Descrevemos as suas etapas e as técnicas que ainda serão implementadas. Atualmente, o projeto encontra-se na primeira fase, onde foram realizados os estudos teóricos necessários e as principais estratégias a serem implementadas para a competição já foram definidas. Os próximos passos incluem a implementação e testes das estratégias acima citadas no simulador da competição, além da participação efetiva na competição em junho de 2013. Posteriormente, ao iniciar a segunda fase do projeto, serão realizadas visitas aos centros de coordenação e estudos sobre os protocolos de atuação das entidades de salvamento do município de São Paulo, com o intuito de modelar e analisar esses protocolos no simulador.

REFERENCES

- [1] D. Guha-Sapir, F. Vos, R. Below, and S. Ponsérre, "Annual disaster statistical review 2010: The numbers and trends," Centre for Research on Epidemiology of Disasters, Tech. Rep., 2011. [Online]. Available: http://www.cred.be/sites/default/files/ADSR_2010.pdf
- [2] M. Wooldridge, *An Introduction to Multiagent Systems*, 2nd ed. Chichester, UK: John Wiley & Sons Ltd., 2009.
- [3] E. Durfee and V. Lesser, "Partial global planning: a coordination framework for distributed hypothesis formation," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 5, pp. 1167–1183, 1991.
- [4] A. H. Pereira, L. G. Nardin, and J. S. Sichman, "LTI agent rescue: A partial global approach for task allocation in the robocup rescue," *Revista de Informática Teórica e Aplicada*, vol. 19, no. 1, pp. 71–92, 2012.
- [5] A. B. M. da Silva, L. G. Nardin, and J. S. Sichman, "Um método baseado em particionamento para exploração de ambientes de desastre," in *Anais do 9o. Encontro Nacional de Inteligência Artificial*. Curitiba, BR: Sociedade Brasileira de Computação, 2012.

APÊNDICE B – Descrição do time -
RoboCup Rescue 2013

RoboCup Rescue 2013

LTI Agent Rescue Team Description

Alan D. Barroso, Felipe de C. Santana, Victor Lassance,
Annibal B. M. da Silva, Luis G. Nardin,
Anarosa A. F. Brandão, Jaime S. Sichman

Laboratório de Técnicas Inteligentes - EP/USP
Av. Prof. Luciano Gualberto, 158 – trav. 3
05508-970 – São Paulo – SP – Brasil
{alan.barroso, fesantana, victor.lassance.silva,
annibal.silva, luis.nardin}@usp.br,
{anarosa.brandao, jaime.sichman}@poli.usp.br

Abstract. In this paper, we describe the strategy implemented by the LTI Agent Rescue team for the RoboCup Rescue Agent Simulation 2013 competition. We mainly focus in some aspects identified as the bottleneck of the team's performance in the 2011 competition: the map exploration and the coordination strategy. In order to improve the map exploration, we use a map partitioning technique to allocate Police Forces to specific regions of the map, while changing the task allocation approach from a hybrid to a distributed one in order to improve the coordination strategy. Moreover, we implemented some techniques to avoid communication channel overload such as data compression and interleaved communication.

1 Introduction

Natural disasters are a source of concern for virtually every country in the world as they usually are difficult to predict and cause huge human and material losses. Although some disasters may be predicted with a considerable advance, most of them cannot, which leaves a short period of time for any preventive action. Thus, it requires the pre-existence of efficient policies to help handling and responding to such situations, which makes disaster management an important discipline.

According to [1], disaster management is comprised of four phases: mitigation, preparedness, response and recovery. In order to promote research on techniques focused on the response phase of disaster management, the RoboCup Rescue Agent Simulation League [5] was founded in 2000. The competition approaches the problem from a multiagent system perspective, in which teams of heterogeneous agents act on simulated post-earthquake scenarios. This kind of scenario imposes great challenges to the development of teams of agents since the environment is dynamic, information and communication are limited and unreliable, and the time to make a decision is limited.

The LTI Agent Rescue team is applying for its second participation on the competition. For RoboCup 2013, we focus on some aspects identified as the bottleneck of the team's performance on the 2011 RoboCup competition: the map exploration and the coordination strategy.

Last year, the map exploration was performed based on the random walking strategy, however it caused a concentration of the agents in specific regions of the map leaving many parts of the map unexplored, victims undetected and fires unnoticed. Besides, the channel communication characteristics were not explored since the team used a strategy that concentrated all gathered information and communication on the *center* agents.

Therefore, some changes and improvements have been made for the RoboCup 2013 competition. In order to mitigate the map exploration problem, we developed a method of partitioning, used by the Police Forces to divide the map into sectors and reduce the patrolling area of each one of them (see Section 3). Regarding the coordination approach problem, we changed how the agents exchange information among themselves in order to allow them to communicate directly without requiring the *center* intermediation. Thus, it changed the task allocation approach from a hybrid to a completely distributed one. Moreover, some techniques to avoid channel overload, such as the Adaptive Huffman encoding [4,7] and interleaved information exchange, were implemented (see Section 4).

The rest of the document is structured as follows. Section 2 describes the distributed strategy employed in the development of each kind of agent, as well as their specific behaviors. Next, the map partitioning technique developed to subdivide the map for Police Forces' surveillance are described in Section 3. In Section 4, we present the communication protocol used to enable the information exchange among the agents. We present some preliminary results in Section 5. Finally, the following activities in 6 and the main conclusions in Section 7.

2 Strategy

In RoboCup 2011 [6], we implemented a strategy that followed a hybrid task allocation approach to control and coordinate agents in the RoboCup Rescue Simulator (RCRS) platform. Such approach was inspired on the *Partial Global Planning* (PGP) approach [3] in which the agents exchange information in order to reach common conclusions about the problem to be solved. It is considered *partial* because agents have limited information of the environment available in order to make the best decision; on the other hand, it is *global* because the agents can exchange and obtain relevant missing information from each other when necessary.

The proposed hybrid approach intended to exploit the advantages of the *centralized* and *distributed* approaches, considering and handling the existence of both local and global information, making the *platoon* agents not completely dependent of the *center* agents, being even capable of functioning on a envi-

ronment without communication. Despite being inspired on PGP approach, the proposed approach had four key differences to the PGP:

- The agents exchanged tasks instead of plans;
- There was no direct communication among *platoon* agents, which required the performance of such tasks by the *center* agents;
- The global information was stored in the *center* agents;
- The global information was only requested and used when the agent's had no local tasks to act upon.

Based on the experience obtained in RoboCup 2011, we have decided to adopt a new strategy for RoboCup 2013, in which the information storage and decision making are completely local, and the agents exchange obtained informations directly among themselves. The new strategy follows a distributed approach.

The main reason for such change is that, since the release of the new RCRS platform, in 2010, the *platoon* agents can communicate through channels directly among themselves, canceling any imperative need of the *center* agents to exchange information. In fact, using center-intermediation would generate a delay of one cycle in the process of information exchange, worsening the situation instead of bringing any real advantage in the case of a distributed approach. Hence, the *center* agents play no role in this new strategy.

Moreover, using a distributed approach maintains some advantages of the previous one, such as the nonexistence of a single point of failure and the elimination of a possible bottleneck. Besides, task allocation can still be performed even if the agents have little or no interaction with other agents, which is interesting for an environment like the RCRS, where the communication is subject to failure or not available at all.

The disadvantage of this approach is that the communication channel may be overloaded, for example, in the case when all the agents are connected to the same channel. Thus, in order to overcome this problem, a set of techniques such as interleaved information exchange, data compression, and transmission of only required data is applied and detailed in Section 4.

In order to implement the new strategy, each agent maintains two databases, one of *tasks* and another of *world information*. In our context, *world information* are data about the state of the environment's objects; *tasks* are world information upon which the agents may act, such as buildings on fire for Fire Brigades, blocked roads for Police Forces, and victims for Ambulance Teams. The task database will also indicates which agents are currently acting on each task.

Using these information, the following general agent behavior was designed:

1. The agent percepts the environment and receives messages from other agents;
2. The agent updates its world information database;
3. The agent generates a list of new identified tasks from the world information database;
4. The agent adds the new identified tasks to its tasks database; it also updates the list of the agents that are acting on each task;

5. The agent calculates the payoff of each task, taking into account the possibility of cooperation with other agents, and chooses the task with the highest payoff for act upon; if the agent has no task it can act upon, it moves randomly on the map;
6. The agent sends information to the other agents, comprised of:
 - the world information whose state has changed;
 - the task it has chosen to work on (if any);

Next, the specific behavior of each kind of agent is described:

2.1 Police Force

The Police Forces have an important role in the new strategy developed in order to overcome the problem of map exploration faced in RoboCup 2011. Each one is given the task of performing the surveillance of a specific partition of the map. Other teams such as MRL, HfutEngineRescue, IAMRescue, and RMAS_ArtSapience have already reported the use of mapping partitioning and we think that it will help us too. Hence, our intents in partitioning the map were (i) to be able to better explore the map and identify faster possible tasks to act upon, and (ii) reduce the number of entities that each Police Force has to check fastening the exploration process.

Besides, since Fire Brigades need to get to refuges to refill their water tank and Ambulance Teams need to have fast ways to reach the refuge when transporting civilians, clearing paths to refuges is considered crucial in our strategy. Thus, in order to make refuges accessible to the other kinds of agents, the first task of the Police Forces is to move to the closest refuge in its sector, if there is any, and clear the blockades found in its path.

Once it reaches the refuge, the Police Force starts patrolling its sector by visiting each road and building contained in it. During such activity, it may find new blockades or receive information from other agents about blockades, requiring it to choose one to act upon. To choose the right blockade to act, the *police* agent rules out the blockades that have other agent acting upon, since there is no advantage of more than one Police Force acting in the same blockade, and calculates the payoff of the remaining blockades. The payoff calculus (Eq. 1) takes into account the blockade repair cost and its distance, giving higher payoffs to blockades with small repair cost and that are closer to the agent.

$$\frac{1}{repairCost} * \frac{1}{\sqrt{distance}} \quad (1)$$

In order to clear more roads and cover a broader area, the Police Force clears the blockades in a way it can transpose it. In some cases, this does not require the complete clearing of the blockade, which saves time, improving the overall Police Force results.

2.2 Ambulance Team

The aim of the Ambulance Team is to rescue the largest number of civilians trapped inside buildings, called victims. In order to succeed in its goal, if the Ambulance Team has no task assigned to act upon, it checks its tasks database and calculates the payoff value related to each task in it. The payoff calculus (Eq. 2) is the difference between the remaining life time of the victim and the expected time to rescue it, which is calculated based on the distance from the ambulance to the victim ($distanceAV$), the time to load the victim ($loadTime$), the distance from the victim location to the nearest refuge ($distanceVR$) and the time to unload the victim ($unloadTime$). Then, the Ambulance Team chooses the task with the highest payoff value to act upon. Once a task is selected, the Ambulance Team moves to the victims location, performs the rescue procedure, loads the victim, transports it to the closest unblocked refuge, and unloads the victim.

$$victimLifeTime - \left(\frac{1}{distanceAV} + loadTime + \frac{1}{distanceVR} + unloadTime \right) \quad (2)$$

Usually, the Ambulance Team acts on the same task until the rescue is complete (victim in the refuge), however, some situations may cause the Ambulance Team to give up on the victim and look for other task to act upon, for example if the Ambulance Team cannot reach the victim because all the possible paths are blocked or if the Ambulance Team identifies that the building in which the victim is trapped is on fire.

2.3 Fire Brigade

The Fire Brigade has the most difficult coordination of the three kinds of agents, since, as opposed to victims and blockades, fires are highly dynamic. They can appear randomly, spread during the course of the simulation and also reemerge after being extinguished.

The goal of the Fire Brigades is to find fires quickly and extinguish them, preventing it from spreading. Map exploration is important for that, so Police Forces play an important role in the fire fighting activity. In case the Fire Brigade needs to choose which burning building it will try to extinguish the fire, it first calculates the payoff (Eq. 3) of all the fire it knows and chooses the one with the highest payoff value

$$\left(\frac{1}{buildingArea} * fieryness \right) + \frac{1}{\sqrt{distance}} + numberOfUnburningNeighbors \quad (3)$$

Using Eq. 3, it is expected to the Fire Brigade to choose the closest building with the smallest area, highest fire intensity and with the largest number of neighbors buildings that are not burning. Therefore, it should choose a small building that is closer to it and it is located at the border of a fire cluster.

3 Map Partitioning

The solution we propose to the problem of map exploration is the partitioning of the map into sectors, in which each Police Force is assigned to patrol one of the sectors. Police Forces are the obvious choice to explore the map because they have the ability to remove blockades, granting access to any place in the map. Additionally, the division of Police Forces in sectors ceases their encounters, reducing the number of idle agents.

The maps of the RCRS are graphs, such as the neighbors entities shown in the viewer of the simulator. Then, the goal is to obtain sectors in which their set of entities (buildings and roads) form a connected subgraph. Each entity will belong to an unique sector, so that Police Forces that patrol different sectors do not visit the same location.

The algorithm divides the map into n sectors, where n is the number of Police Forces in the map. The number of sectors was defined arbitrarily and may be changed according to the map size and configurations.

The proposed map partitioning method divides the map into sectors comprised of:

- An index that identifies it;
- The two sets of X and Y coordinates that delimit its area;
- The set of entities it contains forming a single connected subgraph.

The pseudo-algorithm that performs the map partitioning is presented in Pseudo-Algorithm 1.

Pseudo-Algorithm 1 Map partitioning

Require: $n \Leftarrow numberOfSectors$

```
mapLength  $\Leftarrow$  Length of the map
mapHeight  $\Leftarrow$  Height of the map
term[2] = Factorization(n) {Factorize n into 2 numbers}
if mapLength > mapHeight then
    length = mapLength / term[2]
    height = mapHeight / term[1]
else
    length = mapLength / term[1]
    height = mapHeight / term[2]
end if
sectors  $\Leftarrow$  createSectors(length, height)
sectors  $\Leftarrow$  allocateEntities(sectors)
return sectors
```

First, it obtains the length (*mapLength*) and height (*mapHeight*) dimensions of the map in pixels. Then, it factorizes the number of sector (n) passed as parameter into two terms, where those terms are the closest factors of n , e.g. if

$n = 20$ then $term[1] = 4$ and $term[2] = 5$. If n is a prime number greater than 3, then the first step would be to get the biggest non-prime number that is smaller than n ($n - 1$) and use this value in the factorization, then the remaining agent would be allocated randomly in one of the sectors. This case has not been covered by the Pseudo-Algorithm 1 for the sake of legibility.

After this, the largest dimension is divided by the largest term and the smallest dimension is divided by the smallest term. Based on those values, the sectors are created and their set of X and Y coordinates are adjusted.

In the sequence, for each sector, the entities geographically contained in it are identified and grouped into connected subgraphs (Pseudo-Algorithm 2). Note that there may be more than one connected subgraph in each sector. Thus, the largest connected subgraph of each sector is chosen as the main connected subgraph of that sector and the remaining subgraphs are relinked to one of the main subgraphs. The relink is performed between the subgraph entities closest to the largest main subgraph that may form a new connected subgraph.

Pseudo-Algorithm 2 Allocates Entities

Require: $sectors \Leftarrow$ Created Sectors

Identify the entities within each sector

Identify the largest connected subgraph within each sector

Allocate the remaining subgraphs of each sector to a sector's main graph

4 Communication

Communication is usually a limited resource in a disaster situation, nevertheless such resource is extremely useful for coordination purposes and sharing of information. Those characteristics are very well reflected in the RCRS platform with the limited bandwidth and the probability of message loss and channel failure.

In order to overcome those limitations, we implemented several techniques that allow a better use of the resources available and circumvent any problem that can be experienced during the simulation.

The first improvement incorporated is the use of Adaptive Huffman coding [7], which allows the data compression and transmission of more data over the same amount of bandwidth. The Adaptive Huffman coding is an adaptive technique based on Huffman coding [4] used for data compression in real time, since it permits building the code as the symbols are being transmitted, having no initial knowledge of source distribution, that allows one-pass encoding and adaptation to changing conditions in data.

Another improvement is the interleaved use of the channel by different agents depending on their IDs, where agents with even IDs transmit on even cycle, while the others transmit on odd cycles. Despite of the benefit of not overload the

channel, such technique may cause a delay on information transmission, because of that we apply such technique only when the channel bandwidth is considered insufficient for the data transmission, which is still in the analysis phase.

5 Preliminary Results

With the purpose of understanding the impact of the new features for the competitions, we have analyzed the map partitioning strategy by running tests using some of the 2011 RoboCup Rescue Agent Simulation League¹ maps [2].

Characteristics and Results	Berlin	Istanbul	Kobe	Paris	VC
Dimension (in MPx)	2.2 x 1.6	1.3 x 1.0	0.5 x 0.3	1.0 x 1.0	0.4 x 0.4
Entities (in hundreds)	48	46	22.5	46.5	32
Police agents	6	11	20	30	25
Number of Sectors	6	11	20	30	25
AVG of entities / sector	802	417	113	155	129
STDDEV of entities / sector	140	299	73	115	47
AVG/STDDEV ratio	17.46%	71.70%	64.60%	74.19%	36.43%

Table 1. Characteristics and Results of the Map Partitioning

Given the characteristics of the 5 maps used in the tests and using the number of police agents as the number of sectors, the average number and the standard deviation of entities per sector was calculated for each map. We can see the details in Table 1. Therefore, it is possible to infer from the results that VC and Berlin maps have shapes which are more likely to fit rectangular sectors.

A number of simulations using a team with Map Partitioning and another one without it served as input to the Wilcoxon's Rank Sum Test, in order to compare some key performance indicators, like the number of civilians found during the whole execution and the step in the simulation the first fire was found.

Even though it was not possible to prove that the improvement was statistically significant, the Map Partitioning showed better results in most cases for all metrics. More details can be found in [2].

6 Following Activities

Additionally to the changes explained before, the 2013 Team intend to add some new features to this year's competition. We've decided to focus on only four new strategies, mainly because of the time left to implement this new changes.

¹ <http://roborescue.sourceforge.net/results2011/index.html>

6.1 Partitioning based on refuges

The current version of the partitioning algorithm simply divides the map on rectangles based only on the number of Police Forces the map disposes. The new approach consists of changing the partitioning to make the divisions based on the distribution of refuges on the map. On the new version, one *police* agent shall be assigned to each refuge partition. As before, the Police Forces left out of the partitioning shall be used to help other agents on their actions, by removing blockades, for example.

6.2 Preventive blockade removal

One of the main problems we face on the simulation are the blockades that stop the Ambulances of reaching the civilians who need to be rescued. To solve this problem, we intend to use the Police Forces to perform preventive blockade removals, so that Ambulances could easily save their targets. When an Ambulance decides to save a civilian, it will communicate to all agents the location of its target. The closest available Police Force will then move to the target location and, after that, to the closest refuge, removing all the blockages on the way and increasing the chances of survival of the civilian. Since part of the Police Forces are used on the partitioning of the map, only the remaining agents will do the preventive removal.

6.3 Civilian asking for help

The 2011 Team doesn't take into account the cry for help sent by the trapped civilians. By missing this opportunity, we lose the chance of knowing the position of civilians in danger and, by consequence, have a smaller score because of the deceased people. Therefore, this year's team intend to implement, in all the agents, the act of hearing in the environment to capture messages and react.

6.4 Rescue payoff

Until now, the payoff calculation done by the ambulances are very simple and don't take into account most of the variables envolving the rescue of a civilian. We pretend to evolve the calculation of the payoff so it may base on factors such as: distance to target, life of the target, presence of fire near or on the target area, presence of blockades on the way to the target. Using a new payoff, the ambulances shall choose more wisely the civilians to be rescued and, therefore, increase the chances of survival and the score of the team.

7 Conclusion

This paper presented a description of the strategy implemented in the LTI Agent Rescue team for the RoboCup Rescue 2013. The new strategy has some differences from the one implemented for RoboCup 2011 mainly related to the map

exploration, the coordination approach, and the communication efficiency. The map exploration proposed is based in a map partitioning technique used to avoid that specific regions of the map stays unexplored. The coordination approach was changed from a hybrid to a distributed one, making the agents completely autonomous on their decisions and actions. The increase on the communication efficiency was obtained by the application of several techniques to avoid channel overload and to allow a better use of the resources available.

Therefore, we expect to have an increase in the teams performance by applying those changes. However, we are still implementing some of those changes and we could not perform a complete comparison analysis with RoboCup 2011 performance. Nevertheless, we surely will have all those changes in place for RoboCup 2013.

Acknowledgments

Annibal B. M. da Silva is supported by AEP/Poli/USP. Jaime S. Sichman is partially supported by CNPq/Brazil.

References

1. W. Blanchard, L. G. Canton, C. L. Cwiak, K. C. Goss, D. A. McEntire, L. Newsome, M. D. Selvers, E. A. Sorchik, K. Stenson, J. E. Turner, III, W. L. Waugh, Jr., and D. West. Principles of emergency management supplement. Technical report, International Association of Emergency Managers, 2007.
2. A. B. M. da Silva, L. Nardin, and J. S. Sichman. Um método baseado em particionamento para exploração de ambientes de desastre. *Encontro Nacional de Inteligência Artificial*, 2012.
3. E. H. Durfee and V. R. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, sep/oct 1991.
4. D. A. Huffman. A method for the construction of Minimum-Redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.
5. H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 6, pages 739–743, Tokyo, Japan, October 1999. IEEE.
6. A. H. Pereira, L. G. Nardin, A. A. F. Brandão, and J. S. Sichman. LTI Agent Rescue team: A BDI-based approach for Robocup Rescue. In *Proceedings of the RoboCup 2011*, Istanbul, Turkey, 2011.
7. J. S. Vitter. Design and analysis of dynamic huffman codes. *J. ACM*, 34:825–845, 1987.

APÊNDICE C – Descrição do time - CBR 2013

DESCRIÇÃO DA EQUIPE LTI AGENTS

ALAN D. BARROSO*, FELIPE DE C. SANTANA*, VICTOR LASSANCE*, LUIS G. NARDIN*, ANAROSA A. F. BRANDÃO*, JAIME S. SICHMAN*

**Laboratório de Técnicas Inteligentes - EP/USP
Av. Prof. Luciano Gualberto, 158 – trav. 3
05508-970 – São Paulo – SP – Brasil*

Emails: alan.barroso@usp.br, fesantana@usp.br, victor.lassance.silva@usp.br, luis.nardin@usp.br, anarosa.brandao@poli.usp.br, jaime.sichman@poli.usp.br

Resumo— A equipe LTI Agents, iniciada em 2009, decidiu participar mais uma vez da Competição de Robótica Brasileira na categoria Agent Rescue Simulation. Prosseguindo com suas evoluções iniciadas no início do ano na tentativa de aproximar suas estratégias das utilizadas pelas entidades de resgate, a equipe utiliza técnicas aplicadas pelos bombeiros como a circunscrição dos focos de incêndio. Para melhorar o desempenho dos agentes, a equipe usa um novo modelo de mundo mais completo, técnicas de particionamento ao enviar mensagens e *Hashsets* para aumentar a velocidade de acesso aos objetos. Para facilitar a criação de novas estratégias, a equipe desenvolveu uma estrutura de máquinas de estado. Por último, tentando resolver um antigo problema, a equipe mudou seu modo de busca para ter uma melhor eficiência.

Palavras-chave— Técnicas de Coordenação, Gestão de Desastres Urbanos, Sistemas MultiAgentes.

1 Introdução

A forte expansão urbana observada durante as últimas décadas no Brasil, assim como em outros países, tem causado o crescimento acelerado e desordenado das cidades, impulsionando assim a ocorrência de desastres, especialmente em regiões mais densamente povoadas. Soma-se a isso o futuro próximo do país, que organizará alguns dos eventos esportivos mais importantes do mundo, como a Copa do Mundo em 2014 e os Jogos Olímpicos do Rio de Janeiro em 2016, aumentando a importância de se preparar para lidar com possíveis desastres urbanos.

No caso específico de desastres naturais, pesquisas e dados científicos recentes indicam que eles vêm se tornando mais frequentes, intensos, dinâmicos e complexos (Guha-Sapir et al., 2011). Alguns dos desastres que têm ocorrido com maior frequência e que poderiam ter um trabalho de prevenção esbarram em elevados custos, o que faz com que os governantes posterguem a implementação da solução definitiva do problema. Dessa forma, como ocorre com o caso de enchentes e deslizamento de terras, o gerenciamento de desastres deve ser eficaz e objetivo para minimizar os danos causados, através de ações eficientes de coordenação e alocação dos reduzidos recursos existentes em um ambiente parcialmente observável, como é o propósito desta competição.

Este ano a equipe LTI Agents está aplicando mais uma vez para a competição, no entanto com uma equipe de estudantes diferente das anteriores. Para a CBR 2013, na seção RoboCup Rescue Simulation Agents, focaremos em alguns aspectos identificados como o gargalo das últimas participações nacionais e internacionais da RoboCup. Além disso, algumas das ideias propostas e implementadas foram baseadas em visitas às entidades

de resgate na cidade de São Paulo, como o Corpo de Bombeiros e a Defesa Civil.

O resto do documento está estruturado da seguinte forma. A seção 2 explica as principais estratégias visando a performance e a estruturação do código, o que inclui: a criação de uma visão de mundo adaptada (2.1), a criação de uma máquina de estados (2.2), o uso de hashsets (2.3) e a criação de um algoritmo para decidir o melhor caminho (2.4). A seção 3 trata sobre a comunicação entre agentes, enquanto a seção 4 traz as principais ideias que surgiram das visitas às entidades. Por fim, a seção 5 aponta as principais conclusões do trabalho até então realizado.

2 Mudanças estruturais

2.1 Criação de uma visão de mundo adaptada

Para explicar a motivação da criação desta estratégia, é necessário lembrar o funcionamento de um ciclo do simulador utilizado nesta competição. De forma simplificada, a cada rodada os agentes recebem os parâmetros que representam tudo o que está dentro do campo de visão e o que foi escutado pelo agente. O agente então executa sua função de pensar e pode ou não executar uma ação e enviar mensagens aos demais.

Quando rodamos um algoritmo, queremos utilizar o cenário o mais completo possível, devendo processar a visão e a comunicação recebidas todo ciclo para formar um modelo mais completo e consolidado. Esse modelo deve, portanto, incluir todas as listas de bloqueios conhecidos, vítimas encontradas e prédios em chamas, além das posições dos agentes e do cenário que o *World Model* já fornece. A ideia dessa alteração é de criar uma classe Estado que represente todas essas informações, deixando os dados mais consistentes e per-

mitindo que se armazene uma lista de Estados que representará a memória do agente.

Essa estrutura permitirá que qualquer variável que seja considerada importante para a tomada de decisão, como a posição dos agentes e civis, estado dos prédios e bloqueios das vias, passe a ser disponibilizada de forma organizada. Além disso, os algoritmos implementados poderão utilizar-se de memória para realizar cálculos mais complexos e precisos.

2.2 Criação de uma máquina de estados

Esta ideia consiste em transformar a estrutura do código em uma máquina de estados. O modo como nossas estratégias tinham sido criadas se aproximava bastante à lógica de funcionamento de uma máquina de estados, no entanto tudo era implementado através do uso de códigos sequenciais, usando *ifs* e *elses*.

Apesar de funcionar corretamente, a implementação de novas estratégias muitas vezes era confusa e difícil. Quando criávamos uma nova estratégia, encaixá-la no fluxo de pensamento do agente era uma tarefa complicada havendo a necessidade, em alguns casos, de repetirmos partes do código dentro de mais de um *if*. Além disso, adicionar uma nova condição no comportamento de um agente se tornava complicado pois, com a estrutura original, as prioridades das ações dos agentes poderiam ser facilmente quebradas já que se baseavam na ordem em que as comparações eram feitas.

Dessa maneira, decidimos criar uma estrutura sob a forma de uma máquina de estados para podermos melhor organizar nosso código e aumentar a velocidade de implementação de novas estratégias. A máquina de estados, como nós a imaginamos, parte da estrutura de um agente que é modelado com objetivos e um modelo de mundo (Wooldridge, 2009). Cada estado indica quais são os objetivos atuais dos agentes, definindo quais serão suas ações em cada turno. As transições entre estados são calculadas a partir do modelo de mundo do agente e dos objetivos atuais.

A implementação da máquina foi baseada em uma das maneiras como a plataforma de desenvolvimento multiagente *Jade* lida com o comportamento dos agentes (Bellifemine et al., 2001). A cada turno, no momento em que o simulador dispara a função *think()* dos agentes, cada agente usa seu conhecimento do mundo e o estado atual para calcular qual o estado seguinte que seu modelo comportamental deve seguir. Uma vez calculado o novo estado do agente, o mesmo executa sua ação sob o mundo e calcula o novo modelo do mundo de acordo com o estado em que ele se encontra no momento.

A figura 1 ilustra uma pequena parte do comportamento da estratégia utilizada pelos bombei-

ros. Um bombeiro que se encontrava no turno anterior apagando o fogo de um dos prédios pode, se seu tanque não possui mais água, passar a ter como objetivo recarregar seu tanque ou, no caso em que o fogo já esteja apagado, iniciar o modo de patrulhamento, buscando novos focos de incêndio.

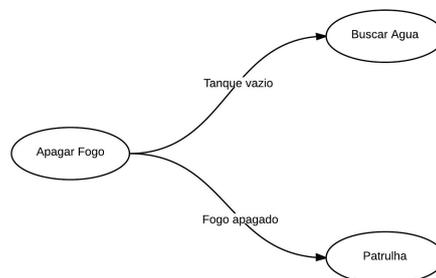


Figura 1: Exemplo de comportamento dos bombeiros

2.3 Uso de hashsets

Um dos problemas identificados durante nossa participação na competição *Robocup Rescue 2013* foi o fato de que alguns algoritmos demoravam muito tempo para serem calculados, levando, em alguns raros casos, perda da rodada do agente por não responder antes do tempo limite. Uma das causas encontradas para a demora da resposta foi o fato dos agentes estarem buscando itens específicos em listas. Como o número de elementos tratados durante uma simulação da competição é de ordem de grandeza elevada, o acesso ou a busca de um elemento contido em uma lista é consideravelmente lento.

Como solução para o problema, decidimos adotar um uso generalizado de *Hashsets*, principalmente para os conjuntos de objetos que compõem o cenário. A razão para o uso de *Hashsets* é simples. O acesso ou a busca de um elemento é feita com uma complexidade $O(1)$, enquanto que o mesmo é feito em complexidade $O(n)$ em uma lista, onde n é o número de elementos da lista. O uso acabou por aumentar a performance de nossos algoritmos além de nos permitir o uso de algoritmos mais complexos de cálculo dentro das simulações.

Apesar de termos aumentado o uso de *Hashsets*, ainda existem alguns casos em que o uso de listas é obrigatório. Em certas ocasiões, os elementos a serem analisados precisam estar ordenados, o que nem sempre é possível com o uso dos *Hashsets*. Para esses casos específicos, manteve-se o uso das listas.

2.4 Busca do melhor caminho

Em sua última competição, o time utilizou uma técnica simplificada para calcular o caminho mais

curto entre dois pontos no mapa, através do algoritmo *Breadth First Search (BFS)*, que realiza uma busca em largura no Grafo até encontrar o Vértice desejado. No nosso caso, como buscamos o melhor caminho e esse pode muito bem não ser o que passa pelo menor número de arestas, o BFS não é o mais indicado.

Dessa forma, estamos implementando algoritmos mais complexos para o cálculo do melhor caminho, nos baseando em conhecidos algoritmos como o D^* e Dijkstra, que levam em consideração o peso calculado para cada via do mapa. Apesar do D^* normalmente ser mais rápido, a utilização do cálculo de heurísticas para os pesos torna a sua implementação mais complexa que a do Dijkstra, portanto, estamos realizando testes com os dois.

Para a utilização dos algoritmos citados acima, devemos definir o cálculo do peso das arestas. Consideramos fatores importantes o tamanho da via, o número de bloqueios e o conjunto de prédios em incêndio no caminho. Para cada tipo de agente, o peso terá influência maior dos parâmetros que são mais ligados aos objetivos de cada agente. Como exemplo, bloqueios não são um problema para os agentes de polícia, assim como caminhos com prédios em chamas podem ser mais recomendados para bombeiros.

3 Comunicação entre agentes

Até a última competição, a comunicação entre os agentes era feita de maneira idêntica para todos os tipos de agentes se baseando fortemente no uso da estratégia de coordenação *Partial Global Planning* (Pereira et al., 2012). Essa estratégia tem como um de seus fundamentos o compartilhamento das informações, dos problemas e do conhecimento do mundo de cada agente (Durfee and Lesser, 1991). Até então, os agentes enviavam, a cada turno, as mudanças que ocorreram no seu modelo de mundo, usando o canal que possuía a maior banda no mapa.

Apesar de interessante, o modo como a estratégia foi implementada causou problemas em alguns mapas. Durante um dos mapas, como havia um grande número de elementos em constante mudança, o tráfego de informações do canal de comunicação acabou sendo caótico e sobrecarregado, causando sérios problemas na transmissão das informações. Como resultado da sobrecarga do canal, os bombeiros não foram informados de um foco de incêndio que estava ocorrendo, ocasionando a queima de boa parte do mapa.

Como solução para esse tipo de problema, houve tentativas de usar um método de reversamento dos agentes dentro canal. Cada agente só poderia se comunicar a cada dois turnos, ou seja, metade dos agentes se comunicavam durante os turnos pares e a outra metade durante os turnos ímpares. Mesmo com a aplicação dessa estratégia,

o resultado indesejado continuou acontecendo.

Decidimos então criar uma nova estratégia de comunicação para os agentes. Ao invés de todos os agentes se concentrarem no uso de um único canal para o envio da informação, decidimos usar um método de particionamento da informação para que os agentes enviem as diversas partes da mensagem através dos vários canais disponíveis no mapa. Isso tende a acabar com a sobrecarga de um único canal, além de permitir uma eficiência maior no uso dos canais de um cenário. Além do uso da divisão das mensagens, decidimos usar um modo de compressão das mensagens para que elas tenham o menor tamanho possível dentro de cada um dos canais disponíveis.

Uma outra estratégia de comunicação ainda sendo estudada para implementação é uma mudança na informação que é passada entre os agentes. Como explicado acima, atualmente os agentes enviam entre si as mudanças ocorridas em seus respectivos modelos de mundo. O modo sendo estudado propõe a transmissão de dados já processados pelos agentes, realizando um cálculo prévio das tarefas que precisam ser realizadas, baseadas no modelo de mundo do agente, e enviando essa lista de tarefas calculadas. O uso dessa estratégia deve permitir uma coerência maior entre as tarefas que devem ser realizadas por cada agente, aumentando sua eficiência e, como consequência secundária, diminuindo o tamanho das mensagens enviadas.

4 Aprendizado com as visitas

Como dito anteriormente, um dos assuntos nos quais as pesquisas do grupo se concentraram para o desenvolvimento de estratégias foi a atuação das entidades de resgate da cidade de São Paulo em situações reais. Para isso, durante o período de pesquisas, o grupo realizou visitas aos centros de operações da Defesa Civil e dos Bombeiros (COBOM), onde nesse último puderam conhecer as estratégias utilizadas durante ocorrências reais de incêndio.

Após as visitas e o levantamento das estratégias utilizadas, o grupo analisou e selecionou aquelas que pudessem ser aplicadas no cenário do simulador, de modo a aumentar a performance dos bombeiros. A seguir serão explicadas com mais detalhes as estratégias selecionadas.

4.1 Ataque x Defesa

Durante as visitas, dois termos repetidos muito frequentemente foram ataque e defesa. Eles definem as duas maneiras básicas de atuação dos bombeiros em situações de incêndio. Quando um bombeiro atua em modo de ataque, possui uma abordagem mais veloz e agressiva, apontando os jatos de água diretamente para o foco do incên-

dio, até que o mesmo seja apagado por completo. Quando ele atua em modo de defesa, o principal objetivo é preservar as pessoas e o patrimônio, estabelecendo bloqueios para o incêndio e resfriando lugares vizinhos com a água, impedindo que o fogo se alastre.

Na maioria dos casos, essas duas abordagens são utilizadas em conjunto, pois muitas vezes as viaturas que chegam primeiro ao local da ocorrência não possuem os recursos suficientes para utilizar a abordagem de ataque.

Essas duas abordagens serão adaptadas para o cenário da competição. Os agentes que chegarem primeiro ao local de incêndio, não possuindo os recursos necessários (em termos de volume total de água) para realizar uma abordagem de ataque, vão agir de forma defensiva, direcionando os jatos de água para os prédios vizinhos, além de chamar reforço. Com a chegada de reforço, ocorrerá a mudança para o modo de ataque, direcionando o fogo para o seu foco.

Para utilizarmos essa técnica, diferenciando quando realizar ataque e defesa, optaremos primeiramente por uma abordagem mais simplificada, levando em conta a área do incêndio e o número de agentes no local.

4.2 Circunscrição da área de incêndio

Uma outra estratégia muito utilizada pelos bombeiros durante os incêndios é a formação de uma espécie de arco ao redor da área dominada pelo fogo, de modo que se impeça que o mesmo se espalhe para os locais em volta.

Pensando na necessidade de definição do posicionamento correto dos bombeiros, explicada durante as visitas, uma das estratégias que adotaremos é fazer com que os agentes de bombeiros se espalhem ao redor da área de incêndio, fazendo com que os jatos de água estejam sempre direcionados para o centro do fogo, impedindo que este se espalhe e extinguindo-o com maior eficiência.

Para o posicionamento correto e eficiente dos agentes ao redor da área de incêndio, utilizaremos um sistema de alocação de tarefas, onde se tivermos N agentes, teremos N posições ao redor da área que serão definidas através de um algoritmo chamado de Fecho Convexo (Liu et al., 2012). Cada posição será uma tarefa, o agente escolherá qual o ponto vago (não alocado) mais próximo e tomará essa tarefa. Quando houver poucos agentes, as alocações se darão primeiramente nas áreas com mais perigo de mortes, onde há mais civis.

5 Conclusão

Neste trabalho, foi brevemente apresentado nosso projeto de um time de agentes para a participação na *RoboCup Rescue 2013*. A equipe utilizará técnicas aplicadas pelos bombeiros como a circuns-

crição dos focos de incêndio, além de um novo modelo de mundo mais completo, técnicas de particionamento ao enviar mensagens e *Hashsets* para aumentar a velocidade de acesso aos objetos. Para facilitar a criação de novas estratégias, a equipe desenvolveu uma estrutura de máquinas de estado e visando a performance, a equipe mudou seu algoritmo de busca em grafos.

Por fim, esperamos obter uma melhora considerável na performance da equipe aplicando as mudanças descritas acima. No entanto, nós estamos em fase de implementação de algumas dessas estratégias e, portanto, ainda não pudemos executar uma análise completa para comparação com o código da última competição. De qualquer forma, garantimos que teremos todas as mudanças que se mostrarem eficazes implementadas para a CBR 2013.

Agradecimentos

Jaime Simão Sichman é parcialmente financiado pelo CNPq e FAPESP/Brasil.

Referências

- Bellifemine, F., Poggi, A. and Rimassa, G. (2001). Developing multi-agent systems with jade, in C. Castelfranchi and Y. Lespérance (eds), *Intelligent Agents VII Agent Theories Architectures and Languages*, Vol. 1986 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 89–103.
- Durfee, E. and Lesser, V. (1991). Partial global planning: a coordination framework for distributed hypothesis formation, *Systems, Man and Cybernetics, IEEE Transactions on* **21**(5): 1167–1183.
- Guha-Sapir, D., Vos, F., Below, R. and Ponslerre, S. (2011). Annual disaster statistical review 2010: The numbers and trends, *Technical report*, Centre for Research on Epidemiology of Disasters.
- Liu, R., Fang, B., Tang, Y. Y., Wen, J. and Qian, J. (2012). A fast convex hull algorithm with maximum inscribed circle affine transformation, *Neurocomputing* **77**(1): 212 – 221.
- Pereira, A. H., Nardin, L. G. and Sichman, J. S. (2012). LTI agent rescue: A partial global approach for task allocation in the robocup rescue, *Revista de Informática Teórica e Aplicada* **19**(1): 71–92.
- Wooldridge, M. (2009). *An Introduction to Multiagent Systems*, second edn, John Wiley & Sons Ltd., Chichester, UK.

APÊNDICE D – Tutorial para criação de mapas para o *RoboCup Rescue Simulator*



Escola Politécnica da USP

Tutorial for the Creation of a Robocup Rescue Simulator Map

Table of contents

[1 - Introduction](#)

[2 - Concepts](#)

[OpenStreetMap](#)

[JOSM](#)

[Geographic Markup Language](#)

[osm2gml Library](#)

[3 - Overview](#)

[4 - Map Capturing](#)

[Download and run JOSM](#)

[Select and download the region of the map](#)

[Save the OSM file](#)

[5 - Convert the map into GML format](#)

[Compiling the library code](#)

[Importing the library source code into an eclipse project](#)

[Adding external libraries](#)

[Adapting the OSM map](#)

[Buildings](#)

[Remove Buildings from outermost shapes](#)

[Marking Buildings as Building](#)

[Separate overlapping buildings](#)

[Roads](#)

[Setup roads as both ways](#)

[Remove roads from inside buildings](#)

[Running the Library](#)

[A few tweaks in the library](#)

[Pinpointing an error location](#)

[6 - Setting up scenario for the map](#)

[Map directory](#)

[GML file layout](#)

[Building List](#)

[Road List](#)

[Creating a scenario](#)

[7 - Conclusion](#)

1 - Introduction

The Robocup Rescue Simulator comes with some example maps that can be used to test and analyze the agents' performance. Although useful, these different test maps do not provide all necessary scenarios to evaluate a specific strategy or overall performance of the agents in different situations. Furthermore, a team may need to assess their performance picturing a more realistic and familiar scenario, in order to bring a more concrete perspective to the solution.

Therefore, it would really be useful for the different teams to be able to create maps, based on their particular needs and on real world scenarios. In this tutorial, we will provide a step by step procedure one should follow to create a map compatible with the Robocup Rescue Simulator, using a desired area provided in the OpenStreetMap server. We will illustrate each step based on what we did to create the map of University of São Paulo.

2 - Concepts

OpenStreetMap

OpenStreetMap (OSM) is a free worldwide map developed collaboratively by the community, it provides free to use geographic information from many places around the world. It was created under the project Planet.osm, that aims to provide geographic information in one XML file, containing descriptions of Nodes, Ways and Relations. A new version of the project is released every week in a big XML file (around 30GB compressed). One can download the entire map, or the latest changeset, on the following url: <http://planet.openstreetmap.org/>. The image below shows the layout of a reduced .osm file.

```

<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGIMap 0.0.2">
  <bounds minlat="54.0889580" minlon="12.2487570" maxlat="54.0913900" maxlon="12.2524800"/>
  <node id="298884269" lat="54.0901746" lon="12.2482632" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636"
timestamp="2008-09-21T21:37:45Z"/>
  <node id="261728686" lat="54.0906309" lon="12.2441924" user="PikoWinter" uid="36744" visible="true" version="1" changeset="323878"
timestamp="2008-05-03T13:39:23Z"/>
  <tag k="name" v="Neu Broderstorf"/>
  <tag k="traffic_sign" v="city_limit"/>
</node>
...
<node id="298884272" lat="54.0901447" lon="12.2516513" user="SvenHRO" uid="46882" visible="true" version="1" changeset="676636"
timestamp="2008-09-21T21:37:45Z"/>
<way id="26659127" user="Masch" uid="55988" visible="true" version="5" changeset="4142606" timestamp="2010-03-16T11:47:08Z">
  <nd ref="292403538"/>
  <nd ref="298884289"/>
  ...
  <nd ref="261728686"/>
  <tag k="highway" v="unclassified"/>
  <tag k="name" v="Pastower Straße"/>
</way>
<relation id="56688" user="kmvar" uid="56190" visible="true" version="28" changeset="6947637" timestamp="2011-01-12T14:23:49Z">
  <member type="node" ref="294942404" role=""/>
  <member type="node" ref="364933006" role=""/>
  <member type="way" ref="4579143" role=""/>
  ...
  <member type="node" ref="249673494" role=""/>
  <tag k="name" v="Küstenbus Linie 123"/>
  <tag k="network" v="VWV"/>
  <tag k="operator" v="Regionalverkehr Küste"/>
  <tag k="ref" v="123"/>
  <tag k="route" v="bus"/>
  <tag k="type" v="route"/>
</relation>
...
</osm>

```

Figure 1 - OSM Layout

JOSM

JOSM is an open Java-based OpenStreetMap editor, an application originally developed by Immanuel Scholz and currently maintained by Dirk Stöcker. In this tutorial, we will use JOSM to download a part of the OSM map, as well as to edit the original map to enable conversion.

Geographic Markup Language

Geographic Markup Language (GML) is a XML-based grammar used to describe interchangeable geographic information. It was defined by the Open Geospatial Consortium, and it is largely used for providing a rich set of markup primitives that allow the creation of application specific schemas, such as the definition of Buildings and Roads in our case. The current version of the Robocup Rescue Simulator accepts gml files as map representation.

osm2gml Library

The osm2gml library was created for enabling conversion from OSM to GML standard, transforming the XML file from one format to the other. The conversion process will change the features in the original map, in order to make it compatible with the GML

representation for the maps in the Simulator. It is based on 9 different steps, including Splitting Intersecting Edges and Merging Adjacent Shapes, and its source code is available on the Simulator project folder under `/modules/maps/src`.

3 - Overview

The process of creating the map for the Robocup Rescue Simulator, in a nutshell, is comprised of 3 basic steps: Capture the map, convert the .osm file into gml format, create a valid robocup rescue scenario for the map on the simulator.

In the first step, we will use the JOSM editor to browse through the OSM worldwide map, select the desired area and download the map information for that area into a .osm file. Section 4 contains a more detailed and illustrated explanation on how to proceed on the first step. Then, in the second step we will begin an iterative process, using JOSM to edit the original OSM map, making it convertible to GML, and using the `osm2gml` library for converting the map into GML format (if the conversion fails, one should retry editing it on JOSM). Section 5 shows more details on how to use the library, as well as some recurrent changes that must be made on the OSM map before converting it. Finally, the last step comprises of the creation of the scenario file for the created GML map, setting the positions for the agents, buildings (Central Agents and Fires) and other features (Gas Stations), according to the entities represented in the .gml file. Section 6 will provide a more detailed explanation on how to setup the scenario.

4 - Map Capturing

The first step to create a map for the Robocup Rescue Simulator is the selection and capture of a desired region on the world map. For that purpose, we are going to use the tool JOSM, that allows us to browse through the OpenStreetMap (see previous section) world map, select a region and download it as a .osm file. The following procedure will show the series of steps to follow in order to capture the map region, using JOSM.

Download and run JOSM

In order to download the tool, go to <http://josm.openstreetmap.de/> and select one of the alternatives for using it (Installer, Launcher or Download .jar file). The recommended version is using the Launcher (`josm.jnlp`), which is compatible with any environment. The image below illustrates the home screen of JOSM tool.

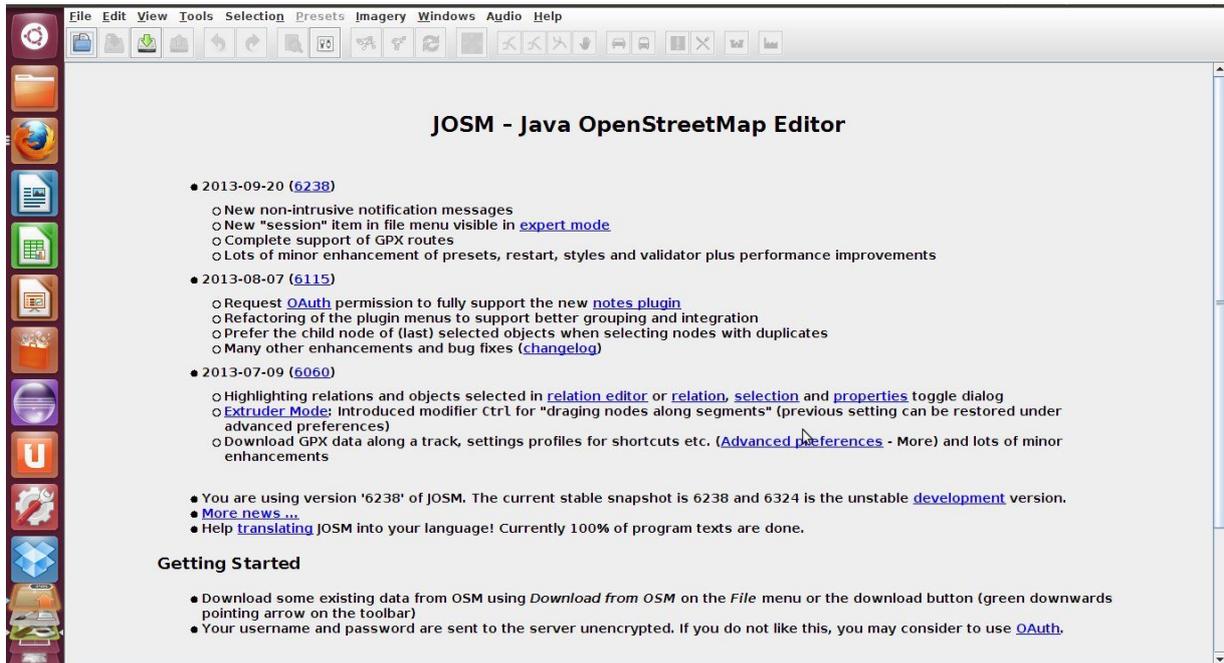


Figure 2 - JOSM Home Screen

Select and download the region of the map

On the left side of the toolbar, you will find a button with a green arrow "Download Map from OSM Server", click it to start browsing the map or press Ctrl+Shift+Down. The start screen on the map is zoomed out (see figure 3), so you will have to manually zoom in until you find the desired region. After finding the region of the map you need, select the area you want and press enter to start downloading it. Figure 4 shows the selected area corresponding to University Of São Paulo map.

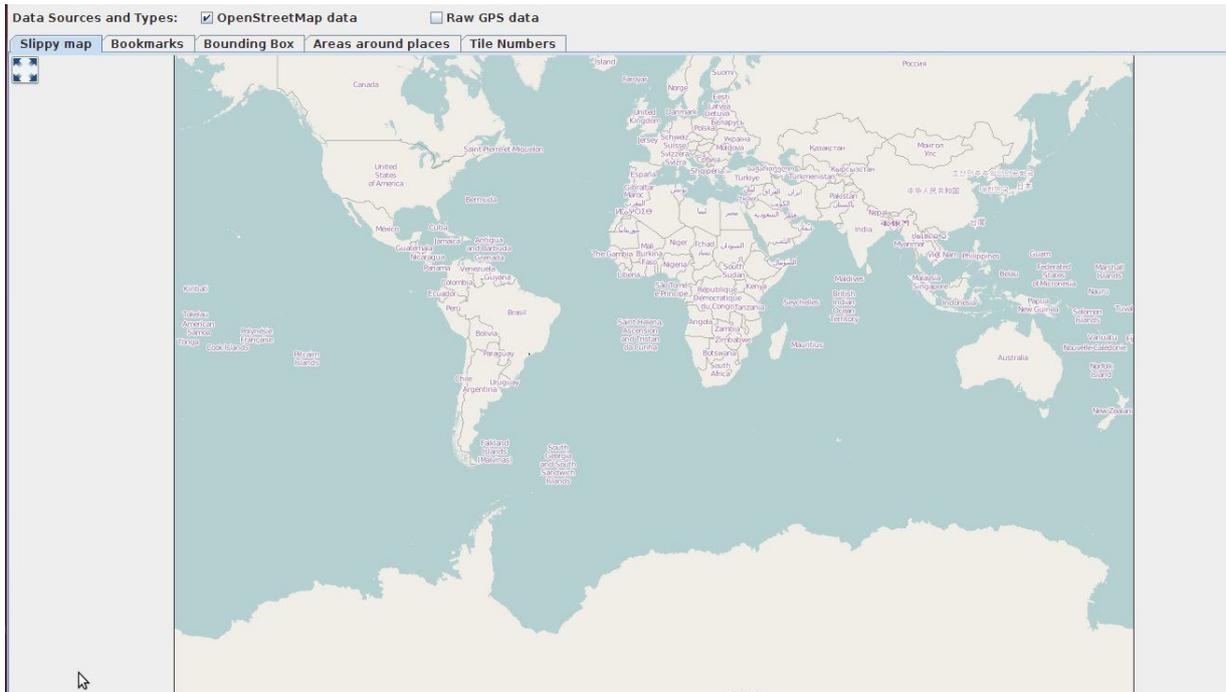


Figure 3 - JOSM Zoomed Out

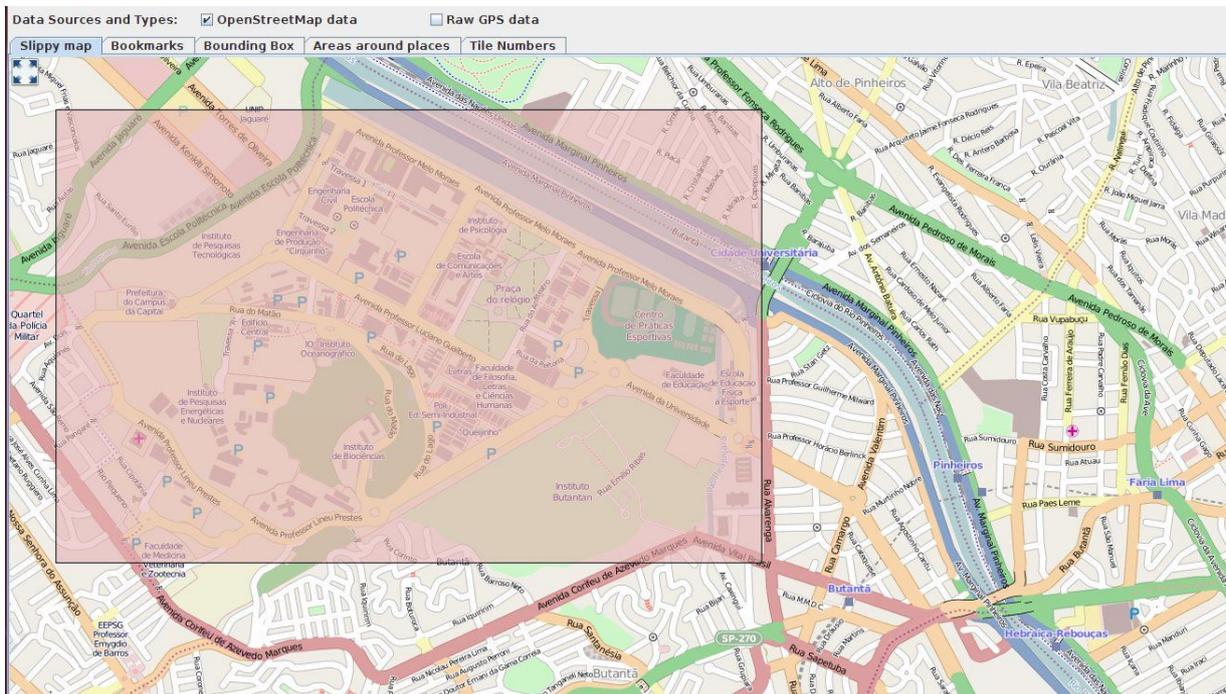


Figure 4 - JOSM Zoomed In USP

Save the OSM file

After JOSM finishes downloading the area of the map from the OpenStreetMap server, it will open the edit screen with the downloaded map on display. Before starting editing it, save the map in a separate folder (File->save as..), see figure below.

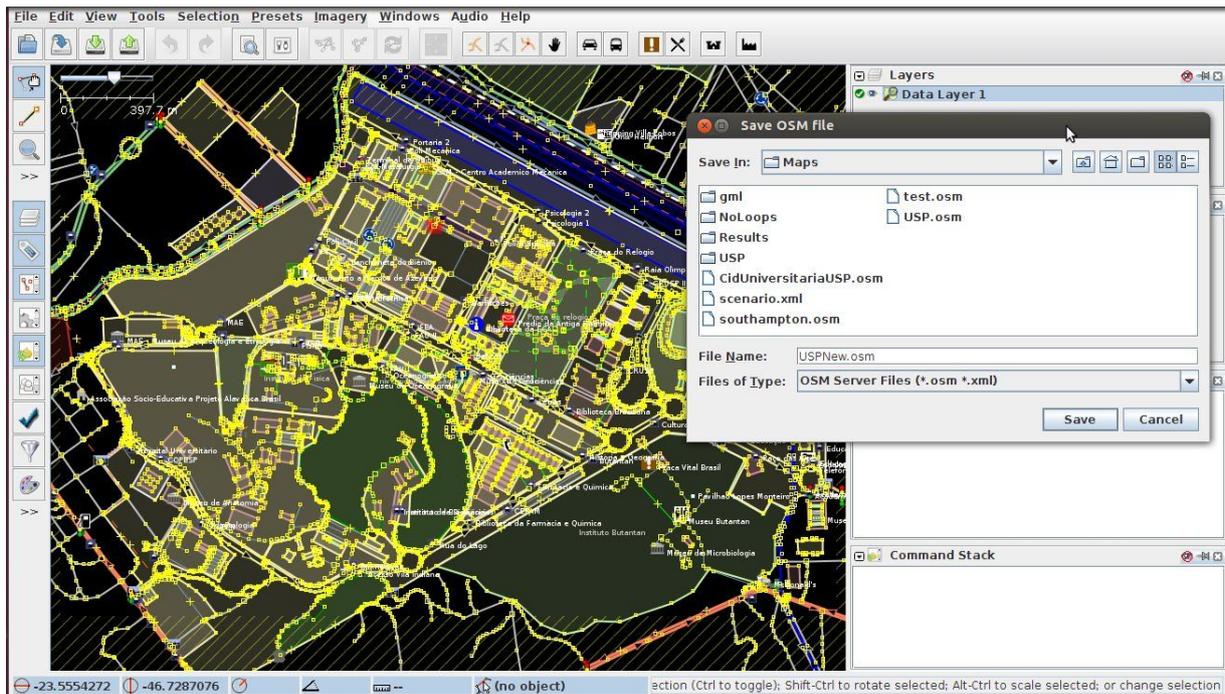


Figure 5 - JOSM Downloaded USP

In these first steps you have used the JOSM tool to browse the world map provided by OpenStreetMap servers, select the desired area and download it to your computer as a .osm file. In the next sections we will show how to edit the map in order to make it convertible to gml format.

5 - Convert the map into GML format

After downloading the map in OSM format, you will now need to convert it into a format that is compatible with the current version of the simulator (GML). In order to do that, you will use a library provided in the code of the Simulator itself, called `osm2gml`, that will convert your map following a series of steps, such as merging shapes and separating overlapping streets.

However, the original map downloaded from the OSM server will normally contain some kinds of shapes and streets that will cause the conversion to fail, due to an exception risen in the library code. Also, some of the buildings and streets in the original map are not

marked as buildings and streets, which makes them disappear in the conversion process. These are some examples of problems that may prevent the proper conversion of the OSM map; for overcoming them, some tweaks will need to be made, using JOSM. Note that some changes on the library code may be necessary as well.

Compiling the library code

The osm2gml library contains classes that will be used to convert our map (OSM format) into the GML format, compatible with the current version of the simulator. Its source code is located under the folder **/modules/maps/src**, in this tutorial we will compile the library using eclipse, in order to make it easier to modify some pieces of code, if necessary, as well as to be able to debug the application to pinpoint errors on the map. Note that it is also possible to build the library by using ant build in the source folder.

Importing the library source code into an eclipse project

Open eclipse, create a new Java Project, then right click it and select Import->File System. Browse until the src folder of the library code, and click ok. The following image illustrates how to import the library source code into a project on the workspace.

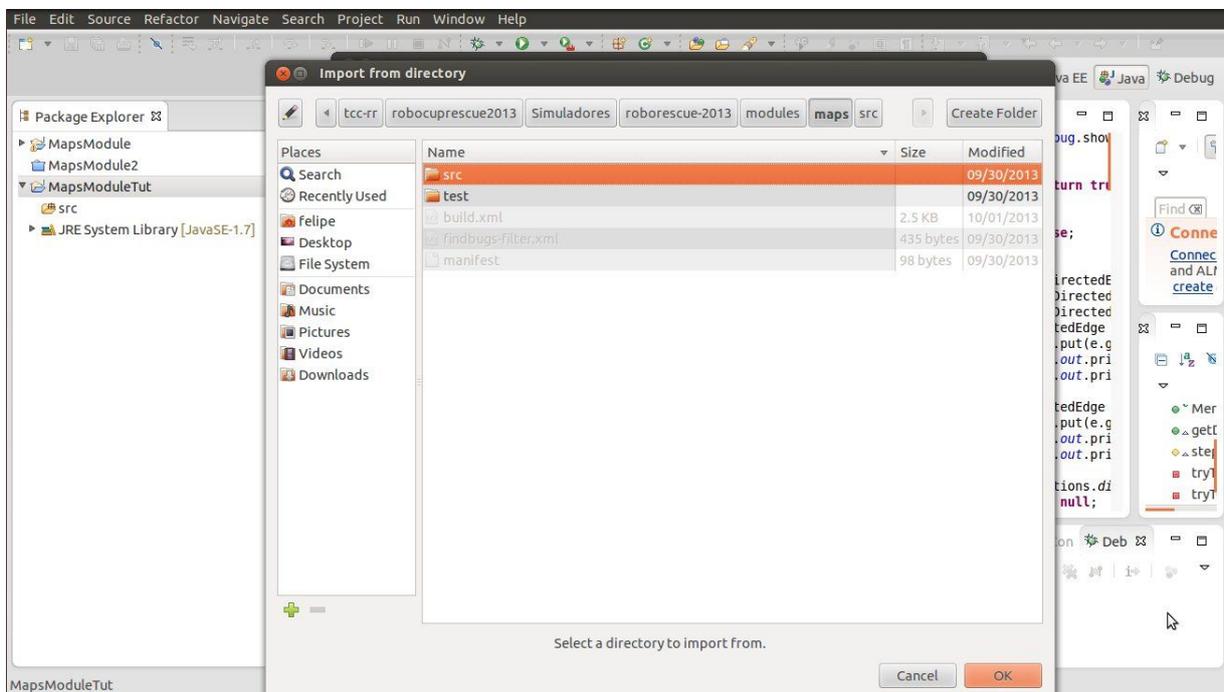


Figure 6 - Importing Library code into Eclipse Workspace

Adding external libraries

The osm2gml library depends on other external libraries provided on the Simulator project folder in two different paths: /jars and /lib. In order to compile the library we will have to add these external libraries to the project. Right click on the project, Build Path -> Configure Build Path -> Libraries -> Add External jars. The following list shows the libraries that need to be added into the project.

- **jars/**
 - rescuecore2.jar
 - gis2.jar
- **lib/**
 - jscience-4.3.jar
 - dom4j.jar
 - uncommons-maths-1.2.jar
 - jaxen-1.1.1.jar
 - jts-1.11.jar
 - common-logging-1.1.1.jar
 - jcommon-1.0.16.jar
 - jfreechart-1.0.13.jar
 - jsi-1.0b2p1.jar
 - log4j-1.2.15.jar
 - resq-fire.jar
 - trove-0.1.8.jar
 - xml-0.0.6.jar

Adapting the OSM map

In order to make the map convertible to the GML format by the Library, some changes will have to be made on the original map, downloaded from the OSM Server, using the JOSM editor. This is because the original maps do not originally mark all the buildings as buildings, which makes them disappear on the conversion process. Also, some roads that are really close to each other may cause conversion problems as well. These are some examples of the changes we will need to make on the original map, in order to make the conversion possible.

Note that the discovery of these problems was purely based on attempts and errors made during the creation of our map for University of Sao Paulo, as some of these issues may not exist, depending on the layout of the original map. The following sections will expose the most recurrent changes we have had to make in order to enable the conversion of our map.

Buildings

Remove Buildings from outermost shapes

In many cases some of the buildings will be located inside territories in the original map, which means that the shapes representing the buildings will be located inside the shape of the territory. In most of the cases, the osm2gml library interprets this set of shapes as only one, the outermost one (i.e the shape of the territory), eliminating all the buildings inside it. In order to avoid that, you will need to remove the outermost shape, allowing the buildings to be processed separately. In order to delete the territory, click in one of its edges, then press “Delete”. The image below illustrates the case of different buildings inside a territorial shape.

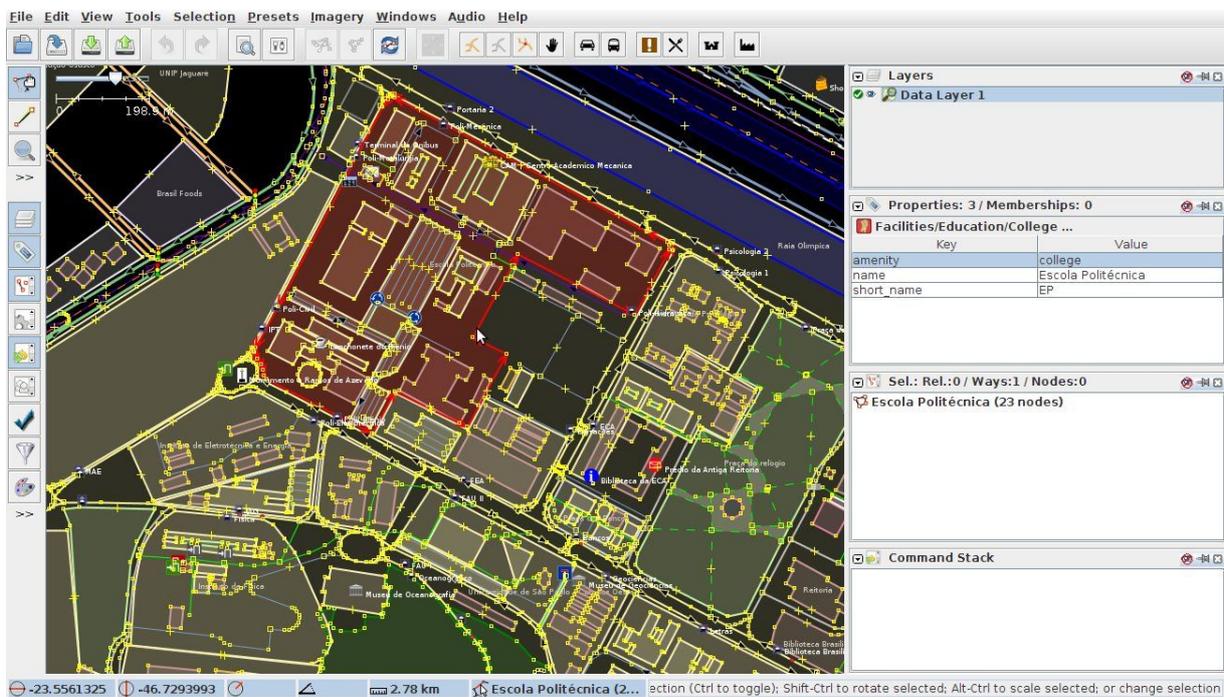


Figure 7 - Outermost shape

Marking Buildings as Building

This is a very important step, because most of the times the shapes corresponding to the buildings in the original OSM map are not marked as buildings, which means that the shape will not be listed as building on the .osm file, and the library won't be able to identify it as such, eliminating it from the resulting map. In order to configure one shape as a building, select it then go to Presets->Man Made->Man Made->Building, and select one of the types

of building (default is “yes”). The following figure shows the setup of a shape as building.

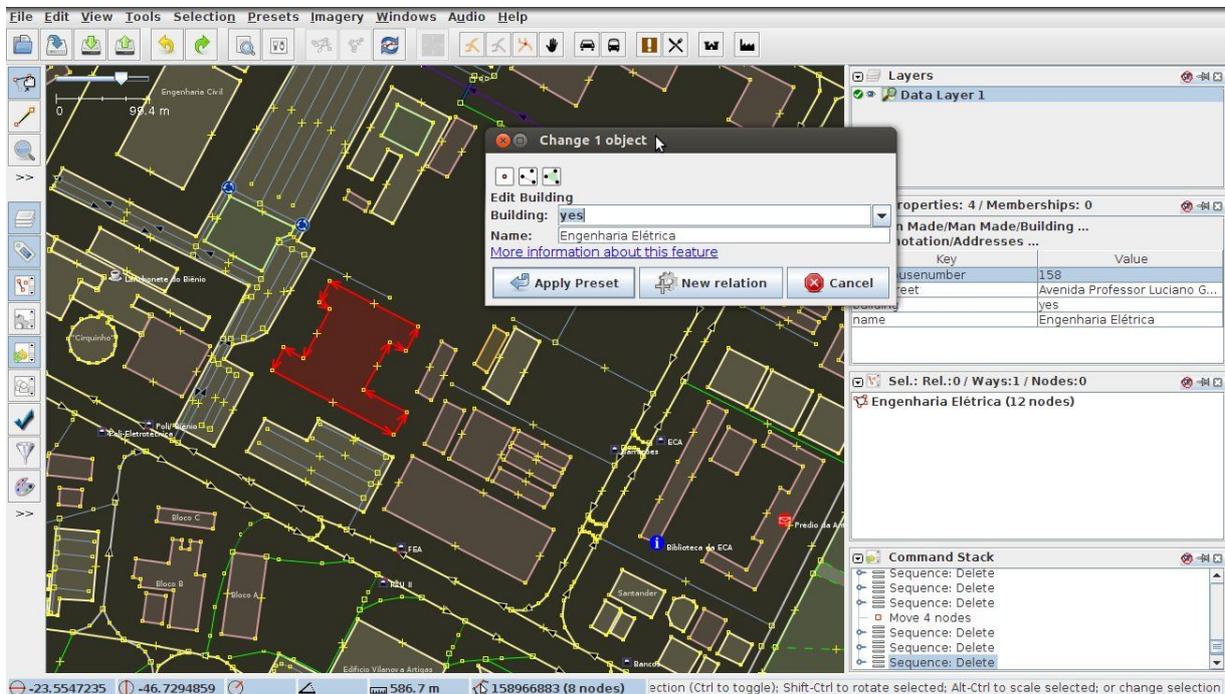


Figure 8 - Building Setup

Separate overlapping buildings

In many occasions, your original map will contain shapes that overlap each other, either two buildings, or one building and a road, or two roads. The converter will process these overlappings (merging or separating) during the conversion process, and sometimes, they may cause it to stop working. So as a safe practice, it is better to separate the overlapping shapes in your map before the conversion. Select one of the shapes and drag and drop it to separate from the other. See the next image as an example of overlapping buildings.

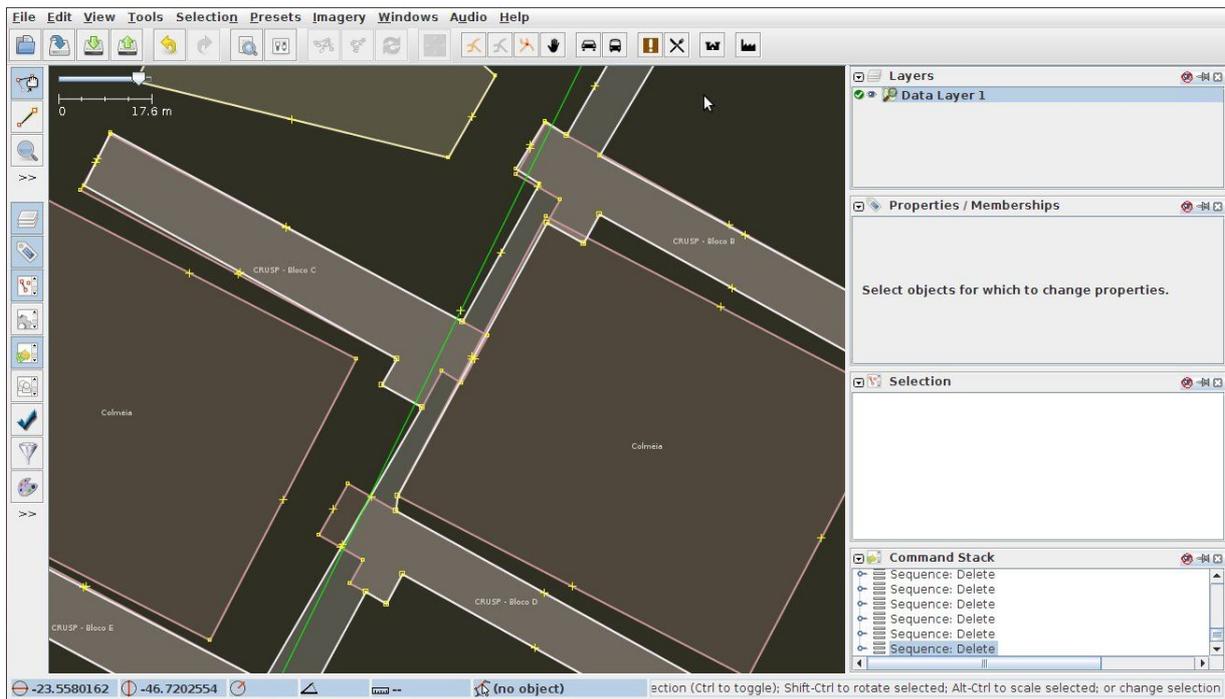


Figure 9 - Overlapping buildings

Roads

Setup roads as both ways

Most of the roads on the original map are set as only one way road, although some of them should be both ways roads, because they connect other roads in different ways (see one example of that on figure 10). In order to set the road as both ways, select it then go to Presets->Highways->Streets and select the desired street type. A dialog will appear, before clicking ok, make sure the check box "Oneway" is not selected. Figure 11 illustrates the dialog for the option of "Unclassified" street.

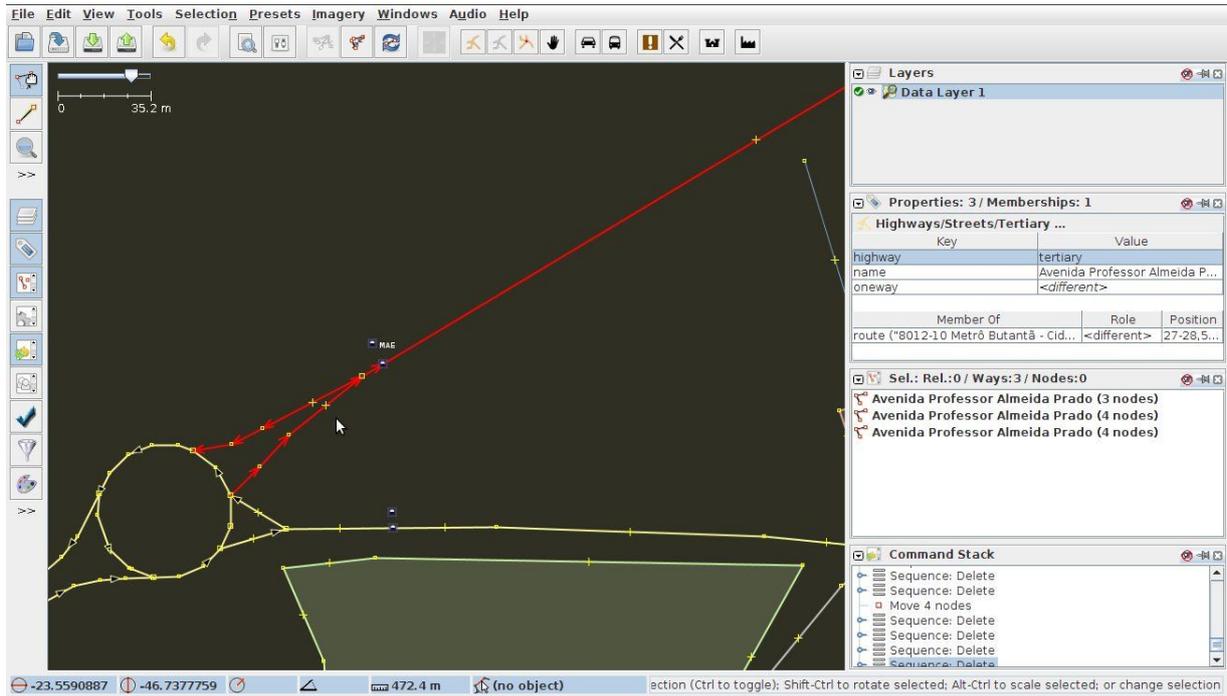


Figure 10 - Both Way Road

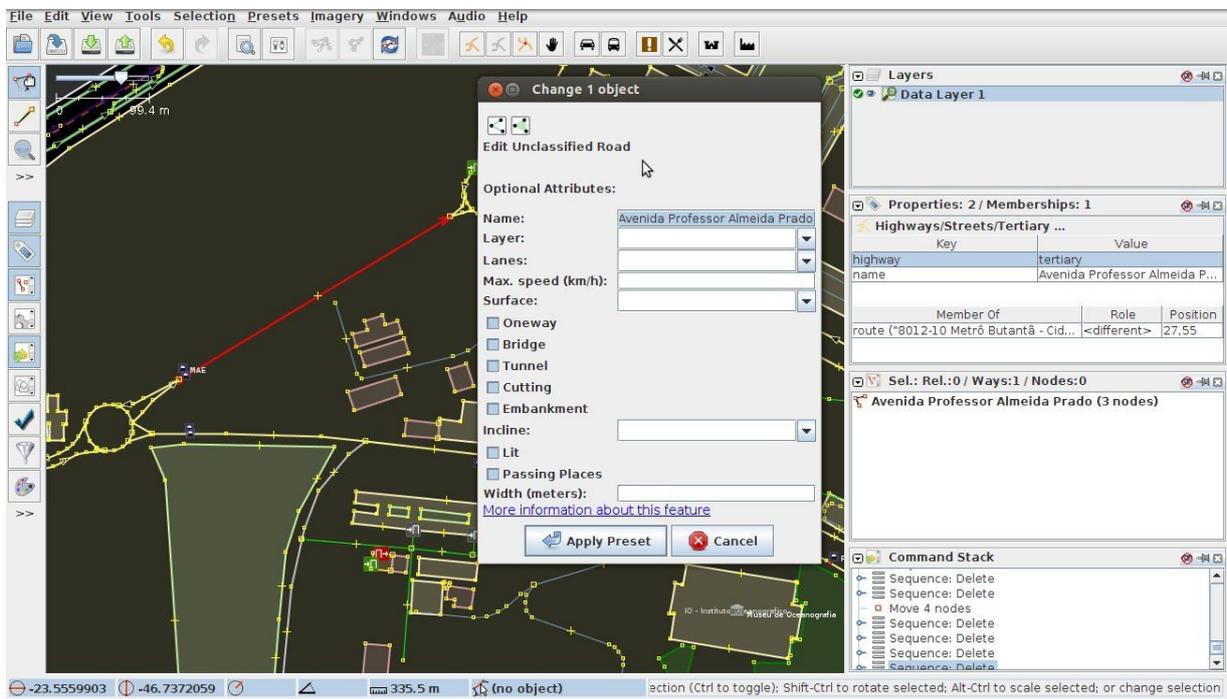


Figure 11 - Setup Dialog

Remove roads from inside buildings

The original OSM maps display some lines inside the buildings, which represent the path one can walk inside them. But sometimes these lines may be interpreted as roads that go inside the building by the osm2gml library, and this can cause the conversion process to stop working. In order to avoid problems like that, you will need to remove these lines from inside the buildings. The image below shows one example of lines displayed inside a building.

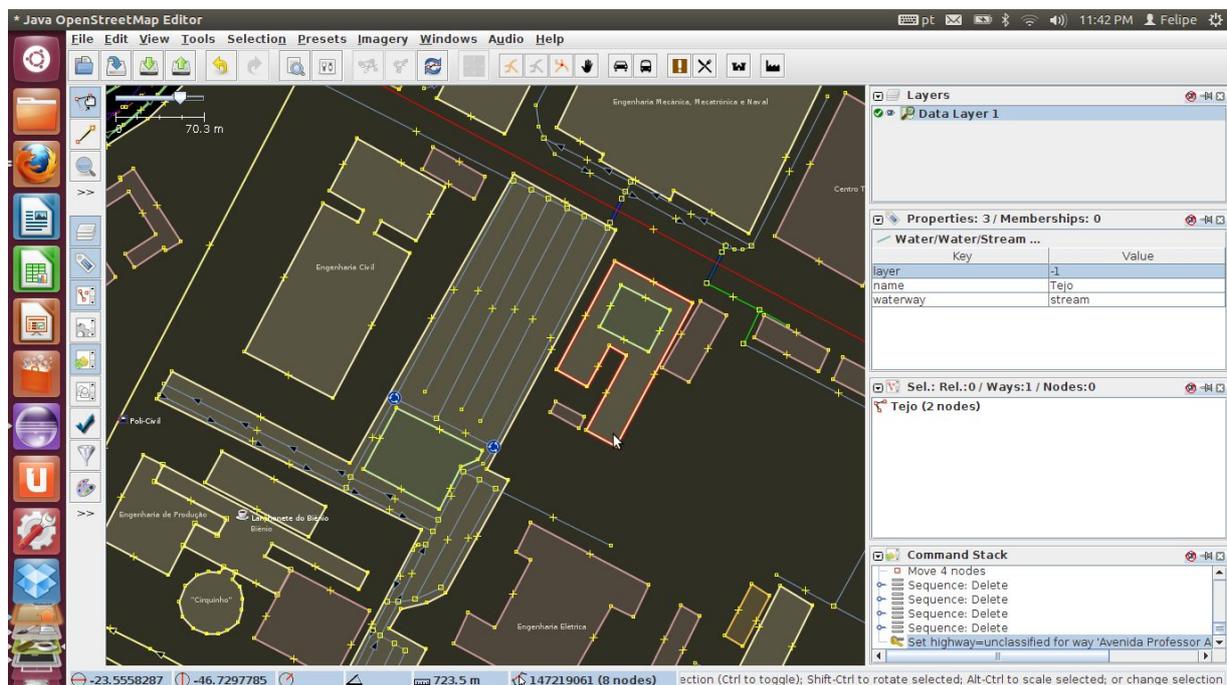


Figure 12 - Line Inside Building

Running the Library

After doing all the necessary changes in the original map, using JOSM, it is now time to run the library in order to convert the map into gml format. In Eclipse, set the run configurations to put the maps.convert.Convert class to launch, using as parameter the path and name of the original .osm file and the destination gml file, in that order. Note that even after following all the procedures above there still may be some invalid pieces in the map that will cause the conversion not to work, and you will have to use JOSM to keep doing the changes described or pinpointing the error location (debugging), therefore this is an iterative process. The following sub-sections will show you how to deal better with errors as well.

The following image illustrates the converter application running, figure 14 shows the resulting map after the application ran successfully:

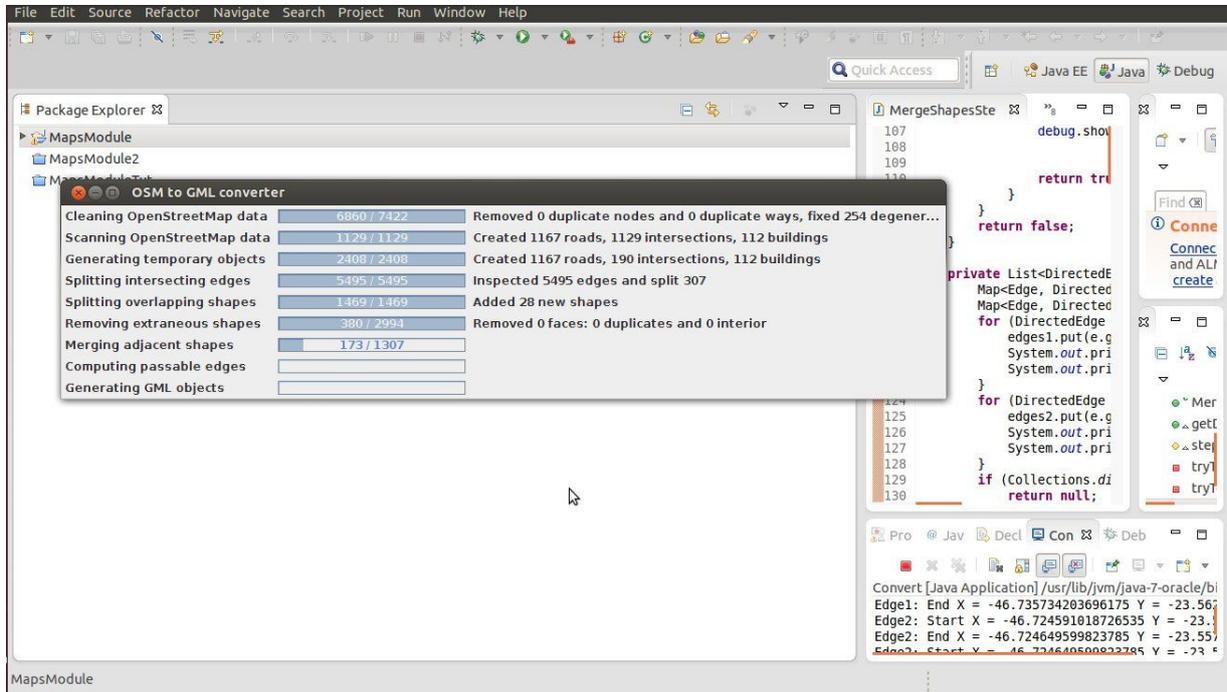


Figure 13 - osm2gml Running

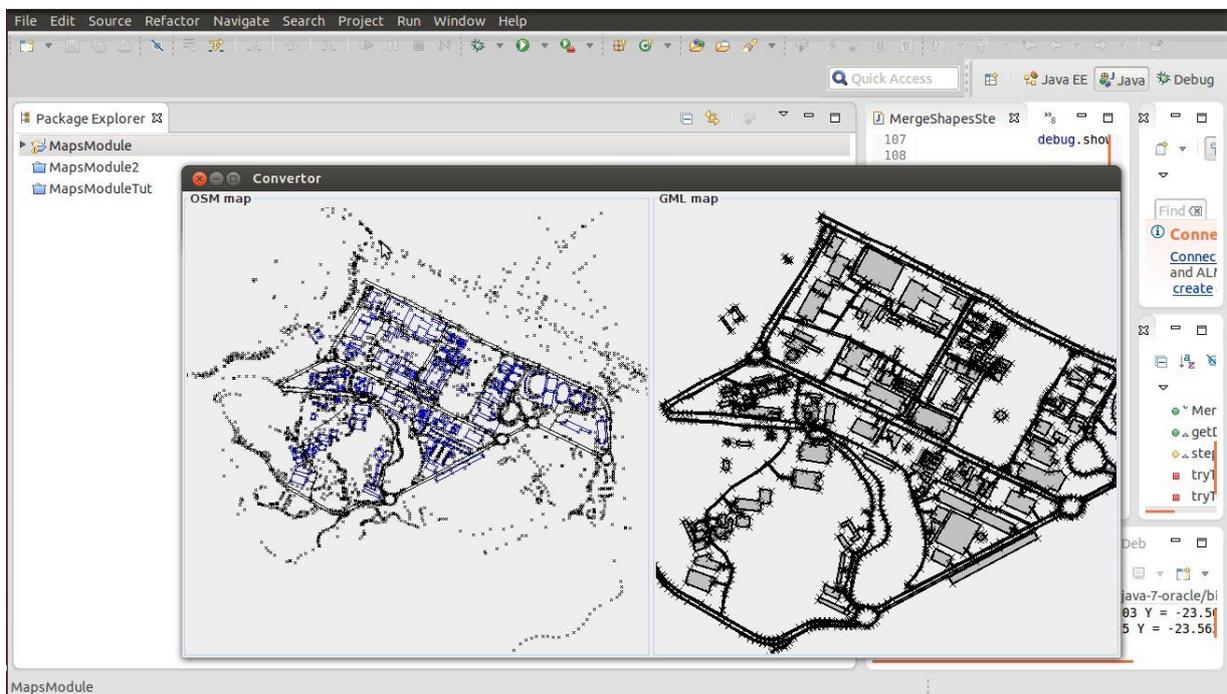


Figure 14 - osm2gml Result

A few tweaks in the library

During the construction of our University of Sao Paulo map, we have realized that many of the buildings and roads were being cut off the resulting map, during the conversion process. Also, we have noticed that most of the parts were being removed in the Step “Remove Shapes Step”, which deletes all objects (roads and buildings) that are duplicates of others, as well as interior faces. To prevent this from happening, we have made a small change on the library code, removing the shape removal part of the code for Buildings and Roads (lines 44,48,51 and 53 commented). Note that normally, changes on the library code are not recommended, but as these changes do not affect the overall behavior of the code, it is a safe strategy to make them. The following figure shows a snapshot of the commented code:

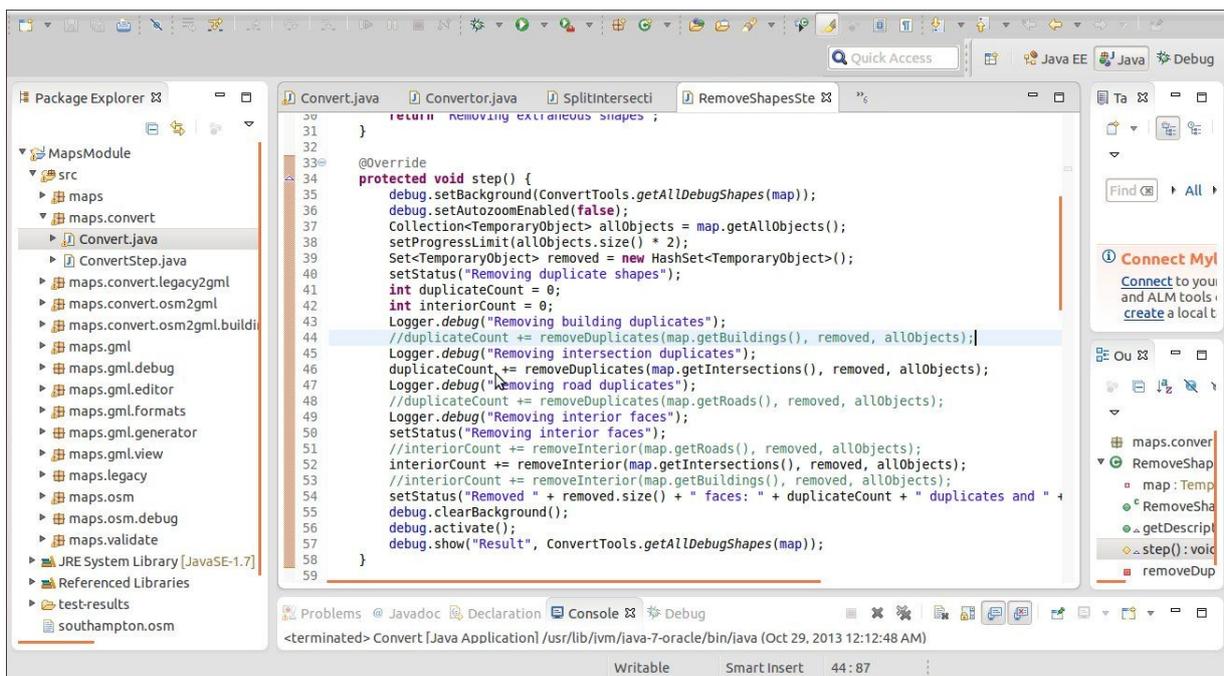


Figure 15 - Library Code commented

Pinpointing an error location

As it was told before, the process of converting the osm file into gml will probably fail for several times before it works out. which means that the conversion process is iterative and based on attempts and errors: after running the library, one can find the error that happened, fix it and run it again. However, sometimes it is really difficult to pinpoint where the error was exactly, in order to do that, we recommend debugging to track which nodes

are being processed at each time to see which one is causing the problem.

6 - Setting up scenario for the map

After creating the map for the simulator, the next step is creating the simulation scenario for the map, which means configuring all the initial simulation setup on your map, such as the initial position of each patrol agent, the locations of the central agents, first buildings on fire, etc. In this section we will show you how to set these configurations for your scenario, using the GML file created last section.

Map directory

Each map/scenario created for the current version of the Simulator is stored inside a specific folder under **/maps/gml/**. Inside the map folder there will be two files: `map.gml`, created on section 5, as well as `scenario.xml` with all the scenario configurations that we will be creating on this section. The following image illustrates the directory structure for the map we created in our example case (`/maps/gml/USP/`).

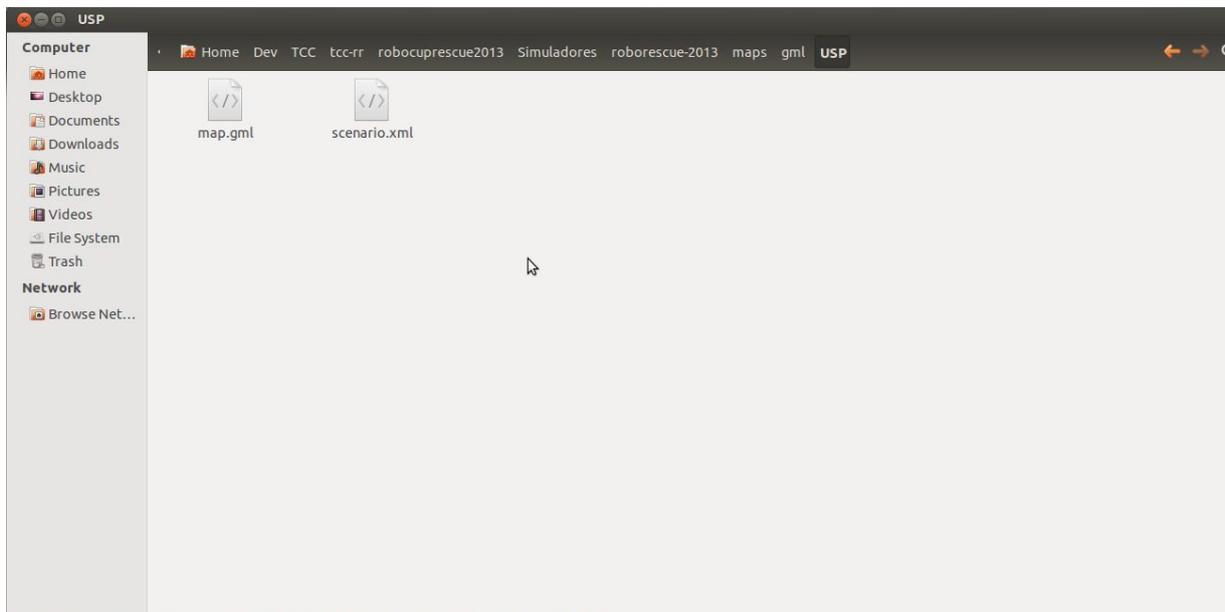


Figure 16 - GML Map Directory Structure

GML file layout

In order to create the scenario, you will need to have an understanding of the layout of the GML file created by the `osm2gml` library in the previous section. The GML file for the

Robocup Rescue Simulator is basically separated into 4 most important parts: the node list, the edge list, the building list and the road list. The edge list contains description of all the edges, that are composed by the nodes in the node list, while the buildings and roads are defined based on the edges, these last two parts will be really useful on the creation of the scenarios. The following images show a minimized representation of the building and road lists on the GML file generated in the previous section.

Building List

```
<rcr:buildinglist>
  <rcr:building gml:id="8508">
    <gml:Face rcr:floors="1" rcr:buildingcode="0" rcr:importance="1">
      <gml:directedEdge orientation="+" xlink:href="#7724"/>
      <gml:directedEdge orientation="+" xlink:href="#7723"/>
      <gml:directedEdge orientation="+" xlink:href="#3652"/>
      <gml:directedEdge orientation="-" xlink:href="#3654"/>
      <gml:directedEdge orientation="-" xlink:href="#3655"/>
      <gml:directedEdge orientation="+" xlink:href="#3651"/>
      <gml:directedEdge orientation="+" xlink:href="#7720"/>
    </gml:Face>
  </rcr:building>
  <rcr:building gml:id="8509">
    <gml:Face rcr:floors="1" rcr:buildingcode="0" rcr:importance="1">
      <gml:directedEdge orientation="+" xlink:href="#8189"/>
      <gml:directedEdge orientation="+" xlink:href="#8196"/>
      <gml:directedEdge orientation="+" xlink:href="#8195"/>
      <gml:directedEdge orientation="+" xlink:href="#4483"/>
      <gml:directedEdge orientation="-" xlink:href="#4476"/>
      <gml:directedEdge orientation="+" xlink:href="#4479"/>
      <gml:directedEdge orientation="+" xlink:href="#4474"/>
      <gml:directedEdge orientation="+" xlink:href="#8193"/>
      <gml:directedEdge orientation="+" xlink:href="#8192"/>
    </gml:Face>
  </rcr:building>
  ...
  <rcr:building gml:id="8619">
    <gml:Face rcr:floors="1" rcr:buildingcode="0" rcr:importance="1">
      <gml:directedEdge orientation="+" xlink:href="#4280"/>
      <gml:directedEdge orientation="-" xlink:href="#4282"/>
      <gml:directedEdge orientation="+" xlink:href="#4285"/>
      <gml:directedEdge orientation="+" xlink:href="#4284"/>
    </gml:Face>
  </rcr:building>
</rcr:buildinglist>
```

Figure 17 - GML Building List

Road List

```
<rcr:roadlist>
  <rcr:road gml:id="8620">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#5285" rcr:neighbour="9341"/>
      <gml:directedEdge orientation="+" xlink:href="#6940"/>
      <gml:directedEdge orientation="-" xlink:href="#5655" rcr:neighbour="9515"/>
      <gml:directedEdge orientation="+" xlink:href="#6939"/>
    </gml:Face>
  </rcr:road>
  <rcr:road gml:id="8621">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#6575" rcr:neighbour="9630"/>
      <gml:directedEdge orientation="+" xlink:href="#7372"/>
      <gml:directedEdge orientation="+" xlink:href="#7371" rcr:neighbour="9958"/>
      <gml:directedEdge orientation="+" xlink:href="#7370"/>
    </gml:Face>
  </rcr:road>
  <rcr:road gml:id="8622">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#5126" rcr:neighbour="9716"/>
      <gml:directedEdge orientation="+" xlink:href="#5296"/>
      <gml:directedEdge orientation="+" xlink:href="#5295" rcr:neighbour="9827"/>
      <gml:directedEdge orientation="+" xlink:href="#5294"/>
    </gml:Face>
  </rcr:road>
  ...
  <rcr:road gml:id="9984">
    <gml:Face>
      <gml:directedEdge orientation="-" xlink:href="#6149" rcr:neighbour="9351"/>
      <gml:directedEdge orientation="-" xlink:href="#6355" rcr:neighbour="8952"/>
      <gml:directedEdge orientation="-" xlink:href="#4484" rcr:neighbour="8960"/>
    </gml:Face>
  </rcr:road>
</rcr:roadlist>
```

Figure 18 - GML Road List

Creating a scenario

As the GML file, the scenario file is also XML-based, and contains a list of the elements that compose the initial setup for the simulation, including starting fires, refuges, civilians, patrol agents, etc. Each element of the xml file has two attributes. The first of them determines what kind of elements you are creating for the initial configuration (*fire*, *refuge*, *ambulance*, *ambulancecentre*, etc.). The second one is called "Location" and determines where the element will appear in the beginning of the simulation, the value of this attribute is a number that refers to the id of one of the elements in the GML file (either a Building or a Road). The figure below shows a reduced representation of scenario file created for University Of Sao Paulo.

```
<?xml version="1.0" encoding="UTF-8"?>
<scenario:scenario xmlns:scenario="urn:roborescue:map:scenario">
  <scenario:refuge scenario:location="8509"/>
  <scenario:refuge scenario:location="8600"/>
  <scenario:refuge scenario:location="8567"/>
  <scenario:fire scenario:location="8544"/>
  <scenario:fire scenario:location="8601"/>
  <scenario:fire scenario:location="8612"/>
  <scenario:fire scenario:location="8513"/>
  <scenario:fire scenario:location="8577"/>
  <scenario:firestation scenario:location="8598"/>
  <scenario:firestation scenario:location="8589"/>
  <scenario:firestation scenario:location="8590"/>
  <scenario:ambulancecentre scenario:location="8560"/>
  <scenario:ambulancecentre scenario:location="8553"/>
  <scenario:gasstation scenario:location="8533"/>
  <scenario:gasstation scenario:location="8575"/>
  <scenario:civilian scenario:location="8892"/>
  <scenario:civilian scenario:location="9234"/>
  <scenario:civilian scenario:location="9834"/>
  <scenario:civilian scenario:location="8723"/>
  <scenario:civilian scenario:location="9030"/>
  <scenario:firebrigade scenario:location="8999"/>
  <scenario:firebrigade scenario:location="9090"/>
  <scenario:firebrigade scenario:location="8934"/>
  <scenario:firebrigade scenario:location="9645"/>
  <scenario:policeforce scenario:location="9045"/>
  <scenario:policeforce scenario:location="9239"/>
  <scenario:policeforce scenario:location="9139"/>
  <scenario:ambulanceteam scenario:location="9211"/>
  <scenario:ambulanceteam scenario:location="9639"/>
  <scenario:ambulanceteam scenario:location="9268"/>
  <scenario:ambulanceteam scenario:location="9812"/>
  <scenario:hydrant scenario:location="9011"/>
</scenario:scenario>
```

Figure 19 - Scenario Layout

Some types of elements can be located only on Buildings, while the other part can be located only on roads. Note that in the scenario file above the elements are located in the respective type of location (defined in the GML file). The following list shows the types of elements corresponding to Building or Road.

- Building
 - *fire*

- *refuge*
- *firestation*
- *ambulancecentre*
- *firestation*
- *gasstation*
- Road
 - *ambulanceteam*
 - *policeforce*
 - *firebrigade*
 - *civilian*
 - *hydrant*

7 - Conclusion

In this tutorial we have shown a couple of basic steps that should be followed in order to create a valid map for the Robocup Rescue Simulator, based on real geographic information. Although the steps described cover basically all the cases and problems we have faced during the creation of our map, there is no official standard procedure for this, as there might be some other issues that were not covered here, in these cases, as was told before, a proper debugging of the conversion library will be really useful. Nevertheless, the previous sections provided a straightforward overview of all the basic concepts required for completing the process, turning this document on a useful reference for any general issue regarding the creation of customized maps and scenarios for the competition Simulator.