

Hugo Pedro Martins Carriço Proença

**MARCS - Sistema Multi-Agente para
Controlo de Tráfego Ferroviário**



**Faculdades de Ciências, Economia e Engenharia
Universidade do Porto
2003**

Hugo Pedro Martins Carriço Proença

MARCS - Sistema Multi-Agente para Controlo de Tráfego Ferroviário



*Tese submetida à Faculdade de Economia da
Universidade do Porto para obtenção do grau de Mestre
em Inteligência Artificial e Ciências da Computação*

Faculdades de Ciências, Economia e Engenharia
Universidade do Porto
2003

Agradecimentos

Este espaço é dedicado àqueles que deram a sua contribuição para que este trabalho fosse realizado. A todos eles deixo aqui o meu agradecimento sincero.

Em primeiro lugar agradeço ao Professor Doutor Eugénio Oliveira pela forma como orientou o meu trabalho, pela utilidade das suas recomendações e pela cordialidade e simpatia com que me recebeu.

Quero igualmente expressar os meus agradecimentos ao corpo docente do Mestrado em Inteligência Artificial e Ciências da Computação da Universidade do Porto pela formação transmitida durante o ano lectivo de 2002/2003.

Aos meus colegas de departamento Professor Doutor Simão de Sousa, Mestre Frutuoso Silva, Mestre Sara Madeira e Mestre Paulo Cordeiro pela ajuda na formatação deste documento e preciosas dicas na iniciação à escrita de documentos \TeX .

Finalmente agradeço aos meus familiares e amigos pelo apoio e incentivo fornecido nos momentos certos...

Resumo

A crescente aglomeração populacional em redor das zonas urbanas aumentou consideravelmente as necessidades de transporte nessas áreas. Presentemente, é necessário transportar cada vez mais pessoas de forma rápida e segura.

Por outro lado, as desvantagens resultantes do aumento do tráfego rodoviário (poluição, congestionamentos, estacionamento, . . .) levou a que cada vez mais o transporte ferroviário seja encarado como solução e encorajada a sua utilização.

Deste modo, a expansão geográfica das redes ferroviárias e o aumento do volume de tráfego levou a que os sistemas de controlo, baseados na noção de bloco fixo, deixassem de responder satisfatoriamente aos requisitos apresentados.

Paralelamente, a ocorrência de vários acidentes, maioritariamente originados por falhas humanas, esteve na génese de um outro tipo de sistemas de controlo de tráfego ferroviário: **sistemas baseados em comunicações**.

Os sistemas de controlo de tráfego ferroviário baseados em comunicações (*Communications Based Train Control* - CBTC) são sistemas automáticos que têm vindo a expandir-se de forma decisiva por várias partes do mundo, estando já difundidos pelos cinco continentes e constituindo objecto da investigação de grandes empresas na procura de melhores e mais rentáveis soluções de transporte.

Caracterizam-se pela implementação de canais de comunicação contínua e bi-direccional em toda a extensão e entre todos os componentes da rede, aumentando a segurança e capacidade. O facto de serem sistemas completamente automáticos constitui uma vantagem significativa em termos de segurança.

A crescente relevância deste tipo de sistemas levou ao esforço de standardização entre os vários fabricantes e à criação de um standard para a informação circulante no sistema, o IEEE 1494.1-1999.

Este trabalho tem por objectivo averiguar a viabilidade de um sistema multi-agente para controlo de tráfego ferroviário baseado em comunicações e estudar as potenciais vantagens que daí podem resultar.

Os sistemas de controlo CBTC actualmente existentes, apesar de possuírem arquitecturas distribuídas, têm controlo centralizado ou semi-centralizado.

Apesar da eficiência alegada pelos fabricantes será lógico pensar que o aumento do volume de tráfego e da extensão das redes abrangidas traga dificuldades acrescidas de comunicação e exija recursos computacionais centrais com cada vez maior capacidade de processamento.

Desta forma, os objectivos do **MARCS** (*Multi-Agent Railway Control System*) centram-se na implementação de um sistema completamente descentralizado preparado para o tratamento autónomo de situações excepcionais e com características de adaptabilidade ao tráfego da rede.

O estudo e a realização do sistema possibilitou o teste das vantagens resultantes da implementação de mecanismos de cooperação entre os agentes intervenientes no sistema.

Implementaram-se também mecanismos de tolerância a diferentes tipos de falhas e situações anómalas, desde avarias em troços de linha, em comboios até falhas nos agentes intervenientes no sistema de controlo, tendo-se avaliado o desempenho do sistema para esses cenários.

Paralelamente projectou-se um tipo de agentes, funcionando paralelamente ao sistema de controlo, com objectivos de aprendizagem e aquisição de conhecimento baseado na experiência entretanto acumulada. Este sistema de aprendizagem permitiu transmitir ao sistema de controlo, regras que podem otimizar os processos de controlo do tráfego.

Em resumo, a nossa tese encerra uma proposta de sistema multi-agente para o controlo de tráfego ferroviário (predominantemente urbano) com capacidades cooperativas e adaptativas.

Conteúdo

Resumo	5
Índice de Tabelas	17
Índice de Figuras	22
Índice de Algoritmos	23
1 Introdução	25
1.1 Motivação	25
1.1.1 Capacidade do Sistema Ferroviário	25
1.1.2 Sistemas de Controlo	26
1.1.3 CBTC - <i>Communications Based Train Control</i>	26
1.1.4 Agentes	27
1.2 Objectivos	27
1.2.1 Descentralização	28
1.2.1.1 Recursos Computacionais	28
1.2.2 Processos de Cooperação	28
1.2.3 Tolerância a Falhas	28
1.2.4 Tratamento de Situações de Excepção	29
1.2.5 Adaptabilidade	29
1.3 Estrutura da Tese	29
2 Agentes e Sistemas Multi-Agente	33

2.1	Definição	33
2.2	Agentes Computacionais	34
2.3	Classificação	35
2.4	Ambiente	37
2.5	Sistemas Multi-Agente	37
2.5.1	Características	38
2.5.2	Ambiente	39
2.5.3	Interacção	39
2.5.3.1	Interferência e Conflitos	40
2.5.3.2	Benefício Individual vs Global	41
2.5.4	Linguagens de Comunicação entre Agentes	42
2.5.5	Motivações para a Utilização	44
2.5.6	Domínios de Investigação	45
2.5.7	Aplicações	46
2.5.8	Aplicações no Domínio dos Transportes	48
2.5.8.1	TraMas	48
2.5.8.2	Engenharia da Mobilidade - Modelo "Co-fields"	49
2.5.8.3	<i>Smart Light</i>	51
2.5.8.4	<i>Adaptative Urban Traffic Control</i> - AUTC	52
2.5.8.5	CRASH	53
2.5.8.6	Sistema de Controlo do Fluxo de Tráfego	53
2.5.8.7	DAARTS	55
2.5.8.8	Linha Milão-La Spezia	55
2.5.8.9	Sistema Multi-Agente de Linguagem Natural - SMALN	57
2.5.9	Vantagens Associadas ao Domínio dos Transportes	59
3	Sistemas de Controlo e Sinalização Ferroviária	61
3.1	Evolução Histórica	61
3.1.1	Primórdios	61

3.1.2	Blocos	62
3.1.2.1	<i>Interlocking</i>	63
3.1.3	Desenvolvimento	64
3.1.3.1	Sinais Multi-Aspectos	64
3.1.3.2	<i>Automatic Train Protection - ATP</i>	64
3.1.3.3	<i>Distance-To-Go</i>	65
3.1.4	<i>On-Cab Signalling</i>	67
3.1.5	Blocos Móveis	67
3.1.5.1	Exemplo de Funcionamento	67
3.1.6	<i>Automatic Train Operation - ATO</i>	68
3.1.7	<i>Automatic Train Supervision - ATS</i>	69
3.1.8	<i>Automatic Train Control - ATC</i>	69
3.2	<i>CBTC-Communications Based Train Control</i>	71
3.2.1	Limitações dos Sistemas Tradicionais	71
3.2.2	Funções Básicas	71
3.2.3	Implementação	74
3.2.3.1	Informação	74
3.2.4	Utilização	74
4	Características Gerais do Sistema Proposto	77
4.1	Apresentação - Sistema MARCS	77
4.2	Requisitos	78
4.3	Arquitectura	78
4.4	Sistema de Controlo	79
4.5	Sistema de Aprendizagem	82
4.5.1	Algoritmo de Aprendizagem	83
5	Linguagens e Metodologias Utilizadas	85
5.1	Modelação	85

5.1.1	UML - <i>Unified Modelling Language</i>	85
5.1.1.1	Modelo	86
5.1.1.2	Características da UML	86
5.1.1.3	Diagramas UML	87
5.1.1.4	Diagramas Utilizados	88
5.2	Representação da Informação	89
5.2.1	XML - <i>Extensible Markup Language</i>	90
5.2.1.1	Características	90
5.2.1.2	DTD - <i>Document Type Definition</i>	90
5.2.1.3	Vantagens	92
5.3	Implementação	92
5.3.1	Java <i>versus</i> C	94
5.3.2	Linguagem C (C++)	96
5.3.2.1	<i>Standard Template Library (STL)</i>	96
5.3.2.2	Vectores Dinâmicos	96
5.3.3	<i>Open GL</i>	97
5.3.3.1	Implementações genéricas	98
5.3.3.2	Duplo <i>Buffer</i>	98
5.3.3.3	GLUT	98
5.3.3.4	GLUI	99
5.4	Comunicação	100
5.4.1	Plataforma de Comunicação	101
5.4.1.1	Plataformas Disponíveis	101
5.4.2	KQML - <i>Knowledge and Query Manipulation Language</i>	103
5.4.2.1	Performativas Utilizadas	105
6	Implementação do Sistema	107
6.1	Arquitectura Geral	107
6.2	Representação do Conhecimento	108

6.2.1	Mensagens	108
6.2.2	Grafos	109
6.2.2.1	Grafo Dual	109
6.2.2.2	Estrutura de Dados	110
6.2.2.3	Cálculo do Trajecto Ideal	112
6.2.3	Comboios Supervisionados	114
6.2.4	Plataformas de Paragem	115
6.3	Atribuição de Troços	115
6.3.1	Critério de Atribuição	116
6.3.2	Noção de Vizinhança	116
6.4	Informação Proveniente do Simulador	118
6.4.1	Inicialização	118
6.4.2	Simulação	119
6.5	Sistema de Controlo	120
6.5.1	Conflito	120
6.5.2	Resolução de Conflitos	120
6.5.3	<i>Distance-To-Go</i>	122
6.5.4	Ordens de Controlo	122
6.5.4.1	Sinalização	123
6.5.5	Função de Utilidade	123
6.5.5.1	Definição	123
6.5.5.2	Razões para a escolha da função de utilidade	124
6.5.6	Situações Especiais	125
6.5.6.1	Resolução	125
6.5.7	Alteração Temporária de Objectivos	127
6.5.8	Entrada e Saída nas Estações	129
6.5.8.1	Entrada nas Estações	129
6.5.8.2	Permanência nas Estações	131

6.5.8.3	Saída das Estações	131
6.5.9	Transferência de Supervisor	131
6.5.9.1	Falha na Transferência	134
7	Características Avançadas do Sistema	135
7.1	Processos de Cooperação entre Agentes	135
7.1.1	Troca de Planos	135
7.1.2	Troca de Informação	136
7.1.2.1	Assumpção ou Término de Supervisão	136
7.1.2.2	Tráfego em Zonas Específicas	137
7.1.2.3	Congestionamentos	137
7.1.3	Repercussões	139
7.1.3.1	Exemplo	139
7.2	Tolerância a Falhas	140
7.2.1	Falha num Agente Supervisor	141
7.2.1.1	Leilão	141
7.2.1.2	Valor da Zona	143
7.2.1.3	Exemplo	143
7.2.2	Avarias em Troços de Linha	145
7.3	Aprendizagem	146
7.3.1	Analogia	146
7.3.2	Recepção da Informação	148
7.3.2.1	Registo de Actividade	149
7.3.2.2	Transmissão da Informação	149
7.3.3	Pré-Processamento	149
7.3.3.1	Seleccção de Casos	150
7.3.3.2	Eliminação de Atributos	150
7.3.3.3	Discretização de Valores	151
7.3.3.4	Redução do Número de Casos	151

7.3.4	Extracção de Conhecimento	152
7.3.4.1	Construção de Transacções	152
7.3.4.2	Suporte e Confiança	153
7.3.4.3	Algoritmo APRIORI	153
7.3.4.4	Geração de Regras	154
7.3.5	Envio de Regras	154
8	Ilustração de Cenários, Avaliação e Conclusão	157
8.1	Avaliação	157
8.1.1	Medidas de Avaliação Implementadas	158
8.1.2	Situações Excepcionais de Tráfego	158
8.1.3	Avarias em Troços	162
8.1.4	Falhas em Agentes Supervisores	163
8.1.5	Cooperação entre Supervisores - Troca de Informação	165
8.1.6	Cooperação entre Supervisores - Troca de Planos	167
8.1.7	Aprendizagem	169
8.2	Trabalho Futuro	172
8.2.1	Controlo	172
8.2.2	Agentes	172
8.2.3	Aprendizagem	172
8.3	Generalização do Sistema	173
8.4	Conclusões	176
A	Execução e Utilização do Sistema	179
A.1	Execução do Sistema	179
A.1.1	Simulador	180
A.1.2	Agente Supervisor	181
A.1.3	Agente Comboio	183
A.1.4	Agente Estacao	183

A.1.5	Agente Aprender	183
A.2	Manual de Utilizador - Aplicação Simulador	184
A.2.1	Simulação	186
A.2.2	Edição	189
A.2.3	Criação de um Mapa Ferroviário - Exemplo	190
A.2.4	Definição da Topologia	191
A.2.4.1	Criação de Estações e Plataformas de Paragem	191
A.2.4.2	Criação dos Comboios	192
A.2.4.3	Definição de Trajectos	192
A.2.5	Criação dos Agentes	193
A.2.6	Início da Simulação	194
A.2.7	Modos de Controlo	194
B	Detalhes de Implementação	197
B.1	Ambiente de Desenvolvimento	197
B.2	Detalhes de Implementação - Agentes	199
B.2.1	Transmissão da Informação	199
B.2.2	Tipo de <i>Sockets</i>	199
B.2.2.1	Implementação	200
B.2.2.2	<i>Non-Blocking Sockets</i>	202
B.2.3	Estrutura dos Agentes	203
B.2.4	Modos de Recepção de Mensagens	204
B.2.4.1	Modo <i>Mensagem</i>	205
B.2.4.2	Modo <i>Ficheiro</i>	205
B.3	Classes Implementadas	206
B.3.1	Aplicação de Simulação	206
B.3.2	Agente Supervisor	207
B.3.3	Agente Comboio	207
B.3.4	Agente Estação	207

B.3.5	Agente Aprender	207
B.4	Estrutura Genérica de Informação	208
C	Descrição das Mensagens	209
C.1	Constantes	220
D	Standard 1494.1-1999 IEEE	223
E	DTD	227
F	Terminologia	231

Lista de Tabelas

5.1	Diagramas e perspectivas UML.	88
5.2	Comparação entre tempos de execução.	95
5.3	Comparação entre tempos de execução (Matrizes).	96
B.1	Ambiente de desenvolvimento (Zonas)	199

Lista de Figuras

2.1	Classificação de agentes.	36
2.2	Sistema multi-agente.	38
2.3	Sistema coordenador da mobilidade.	50
2.4	Estrutura ” <i>co-field</i> ”.	51
2.5	Arquitectura proposta (AUTC).	52
2.6	<i>Traffic Flow Control System</i>	54
2.7	<i>DAARTS</i>	55
2.8	Linha ferroviária entre Milão e La Spezia.	56
2.9	Arquitectura proposta pelo sistema SMALN.	57
3.1	Sinais ferroviários.	63
3.2	<i>Interlocking</i>	63
3.3	Sistema integrado de <i>interlocking</i> (ALSTOM).	64
3.4	Sinalização de quatro aspectos.	65
3.5	Velocidade permitida pelo método tradicional de blocos fixos.	66
3.6	Velocidade permitida pelo método <i>Distance-To-Go</i>	66
3.7	Definição da área exclusiva de um comboio (MAL).	68
3.8	Actualização da área exclusiva de um comboio.	68
3.9	Sistema de controlo automático ATC (ALSTOM).	70
3.10	Esquema de um sistema de controlo automático (ATC).	70
4.1	Arquitectura geral do sistema MARCS.	79
4.2	Esquema de um sistema com um nó de supervisão.	80

4.3	Esquema geral do sistema de controlo MARCS.	81
5.1	Exemplo de um diagrama de sequência.	89
5.2	Visualização de um ficheiro XML num <i>browser</i> WEB.	93
5.3	Exemplo de uma janela de interacção GLUI.	100
6.1	Arquitectura abstracta dos agentes.	108
6.2	Vértice com restrições de direcção.	110
6.3	Criação de vértices e arestas fictícias.	110
6.4	Construção do grafo dual.	111
6.5	Estrutura para representar um grafo.	113
6.6	Grafo (Exemplo).	114
6.7	Distribuição da competência de supervisão.	116
6.8	Noção de vizinhança (Exemplo).	117
6.9	Notação das situações de tráfego (Exemplo).	120
6.10	Situação de conflito entre 3 comboios.	121
6.11	Controlo - Igualdade de circunstâncias.	126
6.12	Controlo - Análise de trajectos.	126
6.13	Controlo - Vantagem global.	127
6.14	Controlo - Igualdade de circunstâncias (Trajectos bloqueados).	128
6.15	Alteração temporária de objectivos.	128
6.16	Entrada e saída de um comboio numa estação (Diagrama de sequência). . .	130
6.17	Transferência entre agentes supervisores (Diagrama de sequência).	132
6.18	Comboio sob supervisão exclusiva.	133
6.19	Entrada na zona de transferência.	134
7.1	Troca de informação sobre o tráfego.	137
7.2	Informação de conflito.	138
7.3	Informação de conflito - Reencaminhamento.	138
7.4	Exemplo de cooperação.	139

7.5	Estado inicial do grafo.	140
7.6	Estado final do grafo.	140
7.7	Falha num agente supervisor (Diagrama de sequência).	142
7.8	Falha num agente supervisor.	143
7.9	Índices associados a cada agente vizinho.	144
7.10	Nova situação de supervisão.	145
7.11	Falha num troço de linha (Diagrama de sequência).	146
7.12	Conflito de interesses.	147
7.13	Antecipação do conflito de interesses.	148
7.14	Diagrama de sequência do processo de aprendizagem.	156
8.1	Painel de avaliação do desempenho do sistema.	158
8.2	Situação excepcional de tráfego.	159
8.3	Resolução do conflito de interesses.	160
8.4	Avaria num troço de linha (Avaliação).	162
8.5	Distribuição inicial da supervisão.	164
8.6	Distribuição final da supervisão após falha de S_1	165
8.7	Cooperação entre supervisores (Troca de informação - Avaliação).	166
8.8	Cooperação entre supervisores (Troca de planos - Avaliação).	168
8.9	Sistema de aprendizagem (Avaliação).	169
8.10	Divisão hipotética em agentes <i>Zona</i>	174
8.11	Divisão hipotética em agentes <i>Zona</i> - INEM.	175
A.1	Aplicação de simulação.	182
A.2	Definição das quatro zonas do <i>interface</i>	185
A.3	Informação disponibilizada sem objectos seleccionados.	187
A.4	Início do processo de construção.	191
A.5	Seleccção de um novo componente.	192
A.6	Definição da topologia.	193

A.7	Definição de componentes fixos.	194
A.8	Rede ferroviária definida.	195
B.1	Ambiente de desenvolvimento <i>Microsoft Visual C++</i>	198
B.2	Estrutura dos agentes.	204
B.3	Envio de informação (Modo Mensagem).	205
B.4	Criação de uma nova mensagem (Modo Mensagem).	205
B.5	Envio de informação (Modo <i>Ficheiro</i>).	206
B.6	Término do modo <i>Ficheiro</i>	206
B.7	Estrutura genérica de informação.	208

Lista de Algoritmos

6.1	Construção do grafo dual	112
6.2	Algoritmo de Dijkstra.	114
7.1	Discretização de valores	151
7.2	Algoritmo APRIORI.	154
7.3	Envio de regras	155
B.1	Recepção de mensagens	204
B.2	Tratamento de mensagens e iniciativa de actuação	205

Capítulo 1

Introdução

Neste capítulo apresentam-se as motivações que estiveram na origem deste trabalho, os objectivos que se pretendem atingir e faz-se uma breve apresentação da metodologia escolhida para tal. Na parte final descreve-se a estrutura deste documento.

1.1 Motivação

O volume de tráfego ferroviário conhecerá nas próximas duas décadas um incremento significativo. Circularão mais passageiros e mercadorias em redes de maiores dimensões (Moncaster 2002).

É necessário que os sistemas de controlo respondam aos novos requisitos e garantam a segurança em redes com crescente volume de tráfego.

O aumento do volume de tráfego terá correspondência directa com a probabilidade de ocorrência de situações excepcionais (avarias em linhas ou comboios, atrasos, . . .), tornando-se desejável que os sistemas saibam lidar autonomamente com este tipo de situações, sem necessidade de intervenção humana.

1.1.1 Capacidade do Sistema Ferroviário

Existem três formas de aumentar a capacidade de um sistema ferroviário.

1. Aumentar o número de passageiros que cada comboio pode transportar.
2. Aumentar o número de linhas disponíveis.
3. Aumentar o número de comboios em circulação, aumentando a frequência com que estes operam.

Alterar o número máximo de passageiros de um comboio tem inconvenientes de várias ordens:

- O aumento do número de passageiros para o mesmo espaço disponível tem custos inaceitáveis a nível de conforto, sendo as exigências dos consumidores cada vez maiores.
- Aumentar as dimensões dos comboios, através da incorporação de mais carruagens, diminui as suas capacidades de aceleração, travagem, manobra, estacionamento e aumenta o consumo. Acresce o facto de grande parte das estações não estar preparada para este aumento.

O aumento do número de linhas incorpora custos elevados e exige nalguns casos a desactivação das linhas em funcionamento. Para algumas situações (túneis, pontes e zonas de orografia muito irregular) o aumento do número de linhas tem custos equivalentes à construção de troços de raiz.

A única opção viável é aumentar a frequência de circulação dos comboios.

A frequência permitida equivale à definição de um espaçamento mínimo entre composições. Este valor é condicionado essencialmente pela segurança, isto é, pela garantia de que cada comboio consegue efectuar uma travagem segura antes de embater noutra. Os comboios têm que viajar com espaçamento suficiente para que, em caso de paragem abrupta (fisicamente inviável) do comboio da frente, os subsequentes possam parar de forma segura.

1.1.2 Sistemas de Controlo

A função primordial dos sistemas de controlo de tráfego ferroviário consiste em monitorar a distância entre composições e garantir a segurança através dos meios de sinalização. Um sistema será tanto mais eficiente quanto menor for o espaçamento mínimo entre composições sem perda de segurança.

A investigação na área dos sistemas de controlo de tráfego ferroviário centra-se exactamente neste ponto, o de garantir segurança a comboios que viajam separados por distâncias cada vez menores.

Variantes entretanto implementadas, quer na própria metodologia de controlo (blocos móveis, distância para percorrer, sinalização directa para a cabine,...(Secção 3.1)) quer na forma de transmitir e processar informação (sistemas baseados em comunicações (Secção 3.2)) optimizaram os sistemas de controlo.

1.1.3 CBTC - *Communications Based Train Control*

Os sistemas *CBTC - Communications Based Train Control* são sistemas automáticos de controlo de tráfego ferroviário que se baseiam na implementação de canais de comunicação

contínuos bi-direccionais entre os comboios e o sistema de controlo.

Estão em franca expansão, especialmente nos grandes aglomerados populacionais como Nova Iorque, São Francisco, Ankara, Paris ou Kuala Lumpur, estando, desde 1999, definido um standard relativo à informação a circular e respectiva periodicidade.

Comparados com os sistemas tradicionais exigem menores custos de instalação e manutenção e aumentam a capacidade e segurança das redes (Secção 3.2).

1.1.4 Agentes

A noção de agente computacional tem, devido à sua crescente utilização em diferentes domínios, suscitado alguma discussão sobre a sua definição e distinção relativamente às restantes entidades computacionais.

Essencialmente caracterizados pela capacidade de interagir autonomamente com o ambiente que os rodeia, podem apresentar características de adaptabilidade, mobilidade, cooperação ou competição.

Têm demonstrado constituir uma alternativa consistente aos sistemas computacionais tradicionais, muitas vezes com vantagens ao nível da tolerância a falhas, degradação suave do desempenho, persistência ou capacidade para traçar objectivos e elaborar planos para o seu alcance.

A noção de sistema multi-agente resulta da incorporação de vários agentes, de características similares ou distintas, em interacção num determinado ambiente por forma a atingir um objectivo.

Neste tipo de sistemas exploram-se processos de cooperação, competição ou antagonismo com vista à realização da tarefa planeada (Secção 2.5).

1.2 Objectivos

O principal objectivo deste trabalho consiste na implementação de um sistema multi-agente para controlo descentralizado de tráfego ferroviário baseado em comunicações.

Adicionalmente, pretendem-se explorar algumas das características habitualmente associadas à noção de *Agente*: adaptabilidade, autonomia, cooperação ou competição.

Averiguam-se as vantagens que uma abordagem baseada em agentes poderá trazer para a resolução de um problema que actualmente envolve bastantes recursos no meio empresarial e académico.

1.2.1 Descentralização

Tendo verificado que os sistemas existentes no mercado apresentam soluções de controlo e planeamento centralizadas ou semi-centralizadas, pretende-se implementar um sistema completamente descentralizado que se constitua como alternativa fiável.

A identificação dos actores fundamentais no sistema (estações, comboios e sistema de controlo), levou a que cada um esteja nele representado pelo respectivo agente, de forma a que a necessidade de obtenção de objectivos próprios conduza à realização do objectivo comum a todo o sistema, o controlo do tráfego na rede ferroviária.

1.2.1.1 Recursos Computacionais

O previsível aumento da dimensão das redes ferroviárias e do número de composições que nela circulam levará a que sistemas centralizados de controlo necessitem de computadores com maior capacidade de processamento.

A opção por um modelo completamente descentralizado permitirá a independência entre a capacidade dos recursos computacionais exigidos e o tamanho da rede. Para redes de maiores dimensões bastará incorporar mais agentes na resolução do problema. Cada agente, por ter visão parcial e limitada do ambiente, não necessita de *hardware* de elevada capacidade computacional.

1.2.2 Processos de Cooperação

A existência de várias entidades autónomas no sistema propicia a implementação de processos de cooperação com vista à aquisição dos objectivos propostos.

A partir da visão parcial de cada agente, pretende-se que estes partilhem conhecimento e objectivos com os restantes por forma a facilitar os processos de controlo da rede ferroviária e potenciar a fluidez do tráfego.

Desta interacção deverá resultar uma visão mais abrangente do sistema com inerentes benefícios em termos de eficiência.

1.2.3 Tolerância a Falhas

Pretende-se que esta abordagem traga vantagens em termos de tolerância a falhas. A ocorrência de falhas em algum dos intervenientes no processo de controlo deverá ser autonomamente colmatada pelos restantes sem que o desempenho global do sistema se deteriore.

Avárias ou impedimentos temporários em troços de linha ou composições deverão também ser tratados de forma autónoma pelo sistema sem perda substancial da sua funcionalidade.

1.2.4 Tratamento de Situações de Excepção

O aumento do volume de tráfego terá correspondência directa no número de situações imprevistas, como atrasos, avarias em comboios ou troços de linha.

Estas situações originam situações de tráfego não planeado, desde vários comboios a competirem pela passagem nos mesmos troços até ao bloqueamento mútuo de trajectos.

A flexibilização do tempo de paragem dos comboios nas estações pode também contribuir para a ocorrência de conflitos de tráfego não planeados. A simples variação deste tempo para uma composição pode obviamente alterar completamente o escalonamento de tráfego previsto para a rede ferroviária.

É importante que os novos sistemas saibam lidar com este tipo de situações, evitando o mais possível a necessidade de intervenção humana para a sua resolução.

1.2.5 Adaptabilidade

Porventura numa perspectiva mais académica, pretende-se que o sistema seja adaptativo, isto é, possua a capacidade de alterar os processos de controlo durante a sua execução.

A partir da análise ao historial de situações acumuladas pelo sistema, pretende-se que sejam inferidas regras que permitam evitar as ordens de paragem em resultados de conflitos de tráfego, contribuindo para a melhoria na fluidez do tráfego.

No entanto, é importante que a adaptabilidade não interfira com a segurança, isto é, a inexistência de qualquer regra inferida não condicione a segurança e operabilidade do sistema.

Tendo referido a motivação e os objectivos da tese, passa-se agora a sumariar a sua estrutura.

1.3 Estrutura da Tese

A parte principal desta tese está dividida em oito capítulos com a seguinte estrutura:

Capítulo 1 - Introdução O primeiro capítulo, descreve as motivações que originaram a tese e os objectivos que nortearam a implementação do sistema proposto: **Multi-Agent Railway Control System - MARCS**.

Capítulo 2 - Agentes e Sistemas Multi-Agente (SMA) Este capítulo apresenta a noção de agente computacional, os sistemas compostos por agentes e a sua crescente relevância no domínio dos sistemas informáticos. Referem-se algumas das características normalmente associadas aos agentes e, mediante estas, estabelece-se uma possível classificação. Finalmente são apresentadas algumas aplicações de sistemas multi-agente, especialmente ligadas ao domínio dos sistemas de controlo e sinalização.

Capítulo 3 - Sistemas de Controlo e Sinalização Ferroviária No terceiro capítulo abordam-se os sistemas de controlo de tráfego ferroviário, os seus primórdios e evolução histórica até ao estado-da-arte dos sistemas modernos. Dá-se particular relevância aos sistemas de controlo baseados em comunicações.

Capítulo 4 - Características Gerais do Sistema Aqui procede-se à apresentação geral do sistema proposto. A partir de uma visão global, apresenta-se a sua divisão em dois sistemas: Controlo e Aprendizagem. Descrevem-se as principais características de cada um deles, e os aspectos que mais os podem distinguir das abordagens tradicionais ao problema.

Capítulo 5 - Linguagens e Metodologias Utilizadas Nesta parte descreve-se a metodologia seguida para a implementação do sistema e as linguagens nele utilizadas. Apresentam-se possíveis alternativas, efectua-se comparações e justificam-se as opções tomadas.

Capítulo 6 - Implementação do Sistema Este é o capítulo onde se detalha a implementação do sistema, especialmente o comportamento esperado para as situações previstas. Referem-se as estruturas de representação da informação e os processos implementados para a manusear.

Capítulo 7 - Características Avançadas do Sistema Nesta parte destacam-se os aspectos mais relevantes do sistema, nomeadamente os mecanismos de cooperação entre agentes, os processos de tolerância a falhas e situações excepcionais. Detalha-se o processo de aprendizagem, os algoritmos utilizados e o conhecimento a adquirir.

Capítulo 8 - Ilustração de Cenários, Avaliação e Conclusão No último capítulo apresentam-se alguns cenários para os quais o sistema pode apresentar utilidade relevante, explorando-se casos potencialmente vantajosos relativamente aos tradicionais. Apresentam-se possíveis medidas de avaliação do seu desempenho e retiram-se as respectivas conclusões.

Adicionalmente são incluídos sob a forma de apêndice, como complemento à informação especificada nos capítulos anteriores:

Apêndice A - Execução e Utilização do Sistema Nesta parte pretende-se fornecer ajuda acerca da correcta forma de execução e utilização do sistema, nomeadamente através da interacção com a aplicação de simulação. Mostram-se os principais comandos e ilustra-se a respectiva funcionalidade.

Apêndice B - Detalhes de Implementação Esta parte funciona como complemento ao capítulo 6. Pormenoriza a implementação do sistema, as classes, estruturas e algoritmos implementados.

Apêndice C - Descrição das Mensagens Descrição das mensagens trocadas pelos intervenientes do sistema. Para cada uma delas especificam-se os agentes emissores e receptores, bem como os parâmetros necessários à sua correcta utilização.

Apêndice D - Standard 1474.1-1999 Extracto do standard definido pelo *Institute of Electrical and Electronic Engineers* para a informação que deve circular num sistema de controlo baseado em comunicações.

Apêndice E - Document Type Definition's Contém a especificação do formato a que os ficheiros XML utilizados para guardar conhecimento devem obedecer.

Apêndice F - Terminologia Esta última parte é referente à terminologia mais comum no domínio dos transportes ferroviários. Além dos termos utilizados neste documento aparecem também referidas algumas das noções historicamente mais importantes nos sistemas controlo de tráfego ferroviário.

Capítulo 2

Agentes e Sistemas Multi-Agente

A crescente relevância e utilização de agentes no domínio informático tem suscitado bastante discussão sobre as características em comum e os aspectos que os distinguem dos demais programas.

Neste capítulo apresentam-se as características normalmente associadas aos agentes computacionais e, com base nestas, elabora-se uma classificação. Seguidamente focam-se os sistemas onde interagem múltiplos agentes na resolução de um problema, os sistemas multi-agente. Apresentam-se razões justificativas para a utilização deste tipo de sistemas e descrevem-se algumas das suas aplicações, especialmente no domínio dos sistemas de controlo e sinalização.

2.1 Definição

A pesquisa num dicionário linguístico¹ resulta em diversos significados atribuídos à palavra "Agente":

- Alguém que actua ou tem o poder ou capacidade de actuar.
- Alguém empossado por outro para a sua representação.
- Uma forma através da qual algo é feito ou causado (instrumento).
- Uma força ou substância que provoca uma mudança (agente químico, agente infeccioso).
- Um representante ou oficial de um governo ou departamento administrativo de um governo.
- Um espião.

¹GuruNet - www.atmica.com

- Linguística. O nome que especifica a pessoa que influencia uma acção.

Este conjunto de definições é bastante curioso uma vez que, como se verá de seguida, incorpora algumas características que identificam um agente enquanto entidade computacional. Fornece uma ideia das propriedades que um agente deverá apresentar para estar em conformidade com os significados tradicionais da palavra.

2.2 Agentes Computacionais

Também no domínio informático existem bastantes definições para *Agente*, com óbvios pontos em comum mas também com suficientes diferenças para dar a noção de um limite por vezes ténue, entre um agente e outra aplicação computacional.

De facto, e dependendo em grande parte do sistema em que se inserem, o comportamento e características associadas a cada agente pode variar bastante.

Vejam-se algumas das definições propostas para *Agente*:

(Oliveira 2002) "Entidade controlada por um programa que está situada num ambiente e é capaz de agir com autonomia nesse ambiente enquanto persegue os seus objectivos."

(Russell Norvig 1995) "Qualquer programa que possa ser visto como percebendo o ambiente através de sensores e nele actuando através de actuadores é um agente."

(Laboratory 1992) "O termo agente é usado para representar dois conceitos ortogonais. O primeiro é a sua capacidade para executar de forma autónoma. O segundo consiste na capacidade de realizar raciocínio orientado por um objectivo."

(Maes 1995) "Agentes são sistemas computacionais que habitam um ambiente dinâmico, percebem e actuam sobre esse ambiente, mediante uma série de objectivos ou realização de tarefas para as quais foram concebidos."

(Wooldridge Jennings 1995) "Os agentes continuamente realizam três tarefas: percepção dinâmica do ambiente, acções sobre o ambiente e raciocínio para interpretar as percepções, resolver problemas, inferir e determinar acções."

(Hayes-Roth 1995) "Um agente é um sistema computacional com as seguintes propriedades:

- Autonomia. Devem funcionar sem intervenção directa do ser humano, contendo simultaneamente algum controlo sobre as suas acções e sobre o seu estado interno.
- Percepção social. Interagem com outros agentes através de alguma linguagem específica.

- **Reactividade.** Apercebem-se do ambiente que os rodeia e respondem a mudanças aqui ocorridas.
- **Proactividade.** Não se limitam a responder ao ambiente, podendo tomar a iniciativa em função de objectivos.”

(**Technologies 2001**) ”Um *software* de alto nível de abstracção que pode ser caracterizado pelo seu comportamento em vez dos seus métodos e atributos.”

(**Nwana 1996**) ”Um componente de *software* e/ou *hardware* que actua num ambiente com o objectivo de cumprir as tarefas para que foi concebido.”

Da análise das anteriores definições emergem algumas características que melhor podem identificar um agente:

Capacidade de percepção O agente deve possuir percepção do ambiente que o rodeia.

Capacidade de actuação Tem que ser permitida ao agente a execução de acções no ambiente onde está inserido.

Autonomia O agente deve executar as acções de forma autónoma, para realizar os seus próprios objectivos sem intervenção do ser humano.

Estado interno A informação contida pelo agente permite a definição do seu estado interno e determina a sua forma de actuar sobre o ambiente.

O ambiente onde o agente se insere e a acção que sobre ele deve efectuar são factores de extrema importância para as suas características específicas. Os requisitos do sistema determinam o comportamento e estrutura do agente, originando a divisão em diferentes categorias.

2.3 Classificação

Tal como foi dito atrás, é o ambiente em que o agente se insere que vai determinar as suas características, induzindo à criação de famílias consoante os aspectos em comum. A figura 2.1 representa uma possível classificação para agentes segundo as suas características de autonomia, cooperação e aprendizagem (Nwana 1996).

Podem-se classificar os agentes por outras propriedades entre as quais:

Tempo de Vida Existem agentes com tempo de vida efémero, cuja tarefa é atingir um determinado objectivo e terminar de seguida. Pelo contrário, existem agentes permanentes, com ciclo de vida infinito.

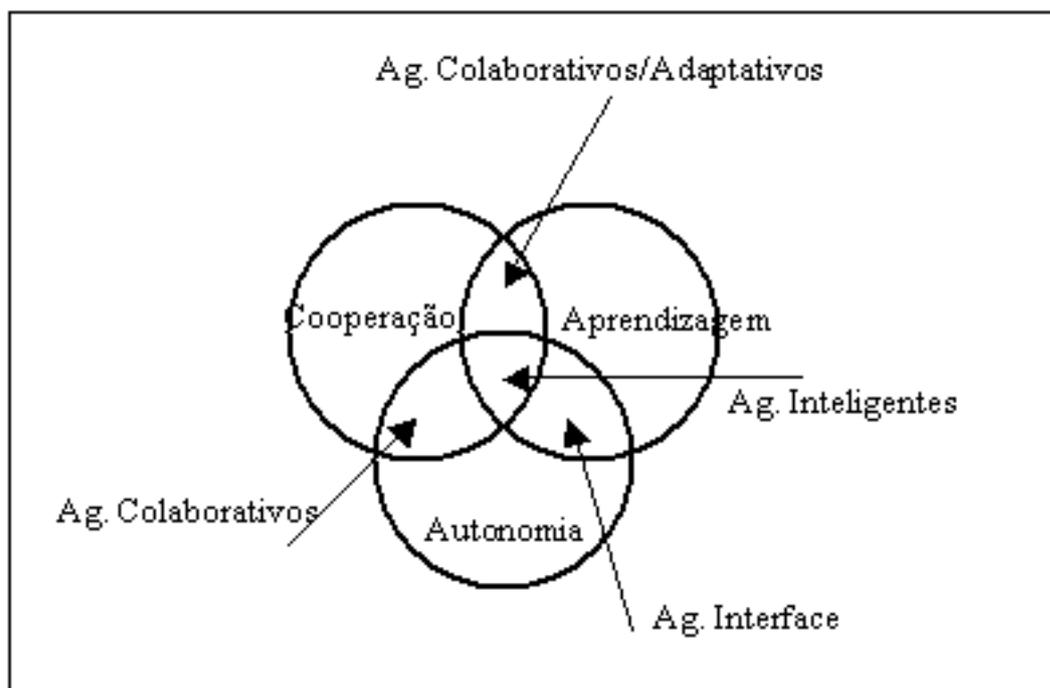


Figura 2.1: Classificação de agentes.

Adaptabilidade Podem-se caracterizar pela capacidade com que modificam o seu estado interno em função do ambiente ou das acções executadas.

Mobilidade Capacidade de se copiar para diferentes localizações.

Comunicabilidade Característica que lhes permite comunicar com outros existentes no mesmo ambiente e daí retirar proveitos.

Proactividade Capacidade para tomar a iniciativa, em vez de responder apenas a acções ocorridas no ambiente (Agentes reactivos).

Determinismo Um agente pode funcionar de forma determinística ou não, isto é, para uma dada situação poderá ou não saber-se antecipadamente qual a acção correspondente do agente.

Crença Alguns agentes podem modificar o seu estado interno e forma de actuar no ambiente segundo as suas crenças, desejos e intenções. Nestas arquitecturas (*Believes, Desires and Intention- BDI*) é mais clara a distinção entre um agente e outro tipo de programa.

Sociabilidade Característica que define a forma de relacionamento entre agentes. Podem ser estabelecidas relações de competição, de colaboração, de antagonismo ou autismo.

Estas propriedades mostram quão diferentes podem ser os agentes computacionais. Ao implementar agentes cujas características são combinações de propriedades atrás referidas

alargam-se sobremaneira a gama de comportamentos possíveis para cada um. Pode até suceder que dois agentes, por terem finalidades e se inserirem em ambientes distintos, apresentem muito poucos pontos em comum sem que por isso deixem de ser classificados de agentes.

2.4 Ambiente

Qualquer agente interage num ambiente, isto é, está inserido num meio que o influencia e é influenciado por ele.

Este condiciona obrigatoriamente as propriedades de cada um. Tal como um animal está preparado para o habitat onde vive, também os agentes devem apresentar características de acordo com o ambiente onde vão interagir, seja a internet, uma rede de área local ou um único computador.

O ambiente pode ser caracterizado mediante vários aspectos (Honavar 1999):

Observável Que informação está acessível ao agente através dos seus sensores?

Controlável Até que ponto as acções tomadas pelo agente influenciam o ambiente?

Predictabilidade O ambiente é determinístico, estocástico ou caótico?

Dinamismo O ambiente é estático ou dinâmico?

Discreto Podem-se discretizar os estados do ambiente ou apenas se pode distinguir um único estado com alterações contínuas?

Relacionamento Existem outros agentes no ambiente? De que forma o podem influenciar?

Abertura O ambiente é aberto ou fechado? Durante o ciclo de vida do agente podem aí interagir outros? De que tipo?

2.5 Sistemas Multi-Agente

A ideia primária de sistema multi-agente resulta da incorporação de vários agentes para resolver um problema de complexa resolução por parte de um só.

Tirando partido da capacidade e características de cada um implementam-se processos de interacção que podem conduzir mais facilmente à realização de um objectivo.

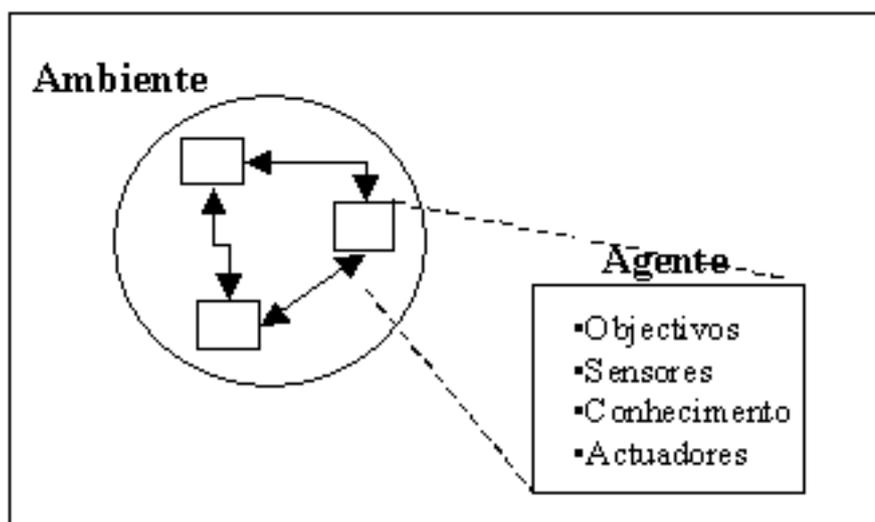


Figura 2.2: Sistema multi-agente.

2.5.1 Características

Da interacção entre os vários elementos resultam as principais características de um sistema multi-agente (Corporation 2003):

Um agente não pode resolver o problema Por falta de informação ou capacidade, um único elemento não consegue isoladamente resolver o problema. Caso contrário não seria necessário um sistema multi-agente, podendo a solução ser encontrada sem desvantagens por um só.

Sistema descentralizado de controlo. Este é um factor essencial de robustez. Não existe nenhuma entidade responsável pelo controlo directo do sistema, não estando este dependente da consistência de um único elemento.

A informação está distribuída. Mais um factor de fiabilidade de um sistema multi-agente. Ao distribuir e replicar a informação por vários agentes diminui-se a probabilidade de perda de informação.

A computação é assíncrona. O facto da computação poder estar fisicamente distribuída por vários nós aumenta a capacidade de processamento de informação, tendo no entanto acrescidas necessidades de sincronização entre agentes no momento da comunicação.

Auto-organização. A inexistência de um sistema centralizado de controlo exige que cada agente se auto-organize com vista ao cumprimento do seu papel no sistema. Cada agente deve reagir e adaptar-se autonomamente às alterações do ambiente.

Competição e Cooperação São formas usuais de relacionamento entre agentes integrantes do mesmo sistema. Processos de competição e cooperação podem conduzir mais facilmente à obtenção do objectivo comum.

Pagamentos e compensações internas Não será uma noção tão intuitiva quanto as anteriores mas está frequentemente presente na organização de um sistema multi-agente. Existe normalmente um sistema de *moeda virtual* para compensar os agentes pelas transacções efectuadas. A escassez dos recursos, além de modelar a realidade, induz o sistema a atingir estados de equilíbrio. Evita que, por exemplo, num processo de leilão o mesmo agente apresente sucessiva e repetidamente as propostas mais altas.

Organização em rede Os agentes estão normalmente distribuídos num ambiente de rede, onde adquirem significado noções de paralelismo, dinamismo ou concorrência.

Inteligência emergente Mesmo nos casos em que os agentes implementados não podem ser considerados inteligentes, da sua interacção podem emergir processos inteligentes, se for analisado o comportamento global do sistema.

2.5.2 Ambiente

Tal como nos sistemas compostos por um único agente, também aqui os agentes vão estar integrados num ambiente (Figura 2.2).

Além das características de incerteza associadas e aplicáveis a domínios habitados por um único agente acresce o facto de cada um dos restantes poder alterar o estado do ambiente, de acordo ou contra as expectativas dos outros.

Este facto aumenta o dinamismo do ambiente e reforça as suas características de impredictabilidade.

No caso dos sistemas multi-agente é particularmente relevante a propriedade relativa à abertura do sistema. Todos os agentes que aí interagem são conhecidos? O seu número permanece fixo durante o tempo de vida do agente? Estes são aspectos que podem alterar completamente a forma de actuação do agente.

2.5.3 Interacção

Características como o paralelismo, a distribuição ou a descentralização normalmente associadas aos sistemas multi-agente acarretam questões acerca do grau e tipo de interacção estabelecidas entre os intervenientes. Em particular, podem surgir dúvidas acerca de (Mataric 1994):

- Quais as propriedades e princípios de organização partilhados entre os integrantes de um sistema multi-agente?
- De que forma a interacção entre os seus elementos afecta o comportamento global?
- Qual a forma de atingir o objectivo global?

- Quais as necessidades de comunicação por forma a atingir o objectivo?
- Quais os comportamentos e regras mais simples a implementar por forma a emergirem comportamentos complexos?

Em (Mataric 1994) propõe-se como definição para interacção a "influência mútua no comportamento". Consequentemente, uma vez que o estado de cada agente depende dos estados e respectivas acções de cada um dos restantes, o número de estados possíveis cresce exponencialmente num sistema multi-agente.

O comportamento global do sistema é determinado pela interacção estabelecida localmente entre os seus componentes, originando a noção de **comportamento emergente**.

Este manifesta-se a partir de uma visão global do sistema ou após longos espaços temporais sendo resultado da interacção em vez da programação explícita.

Os sistemas emergentes têm constituído um relevante domínio de investigação, uma vez que aparentam criar "comportamentos a partir do nada" (Mataric 1994), sendo também designados por auto-organizativos.

2.5.3.1 Interferência e Conflitos

Nem sempre os intervenientes num sistema multi-agente estabelecem relações de cooperação com vista à aquisição do objectivo global. Noutras situações podem-se implementar comportamentos de oposição entre agentes, designadamente de *Interferência* ou *Conflito*.

Designa-se por *Interferência* qualquer influência que oponha ou bloqueie a aquisição de um objectivo por parte de um conjunto de agentes.

Pode-se classificar as relações de conflito entre agentes em:

Competição por recursos Nesta categoria estão englobados todos os tipos de interferência causados pela competição pelos mesmos recursos, sempre que estes são limitados (espaço, informação ou objectos).

Competição por objectivos Manifesta-se normalmente em agentes funcionalmente heterogéneos, isto é, que podendo eventualmente ter arquitectura e objectivo final similar, contêm sub-objectivos distintos. Esta heterogeneidade pode ter como consequência o bloqueio mútuo entre agentes.

Métodos de Resolução de Conflitos

Após a ocorrência de um conflito entre agentes é necessário um processo que permita a sua resolução e possibilite a continuação da actividade de cada um. Como métodos de resolução de conflitos podem referir-se (Sycara 1998):

- **Actualização do conhecimento.** Em sistemas de arquitecturas BDI é por vezes necessário eliminar proposições que estejam em conflito com as crenças de outros agentes e que, por este motivo, impossibilitam a aquisição dos objectivos propostos. Nestes casos os agentes podem simplesmente eliminar essas assumpções.
- **Argumentação.** Em situações específicas é possível aos agentes a construção de argumentos que suportem uma perspectiva particular em detrimento de outra. Os agentes tentam neste caso persuadir aqueles com que estão em conflito a alterar as suas convicções ou desejos na tentativa de resolução do conflito.
- **Relaxamento de restrições.** Sempre que existirem conflitos relacionados com restrições específicas de cada agente podem simplesmente ser relaxadas essas restrições por forma a possibilitar a resolução.
- **Normas sociais.** A implementação de normas sociais no relacionamento entre agentes destina-se precisamente a evitar situações de conflito, quer seja pela proibição de comportamentos ou pela implementação de hierarquias e prioridades entre os elementos integrantes do sistema.
- **Negociação.** É a noção de significado mais amplo, podendo ser vista como um método de coordenação, alocação de recursos ou de tarefas, informação de planos ou desejos. Todas estas tarefas servem também para evitar conflitos entre os integrantes de um sistema, razão pela qual a negociação também pode ser considerada, embora com significado mais vasto, como um método de resolução de conflitos entre agentes.

2.5.3.2 Benefício Individual vs Global

As regras de interacção estabelecidas num sistema multi-agente tendem a evitar ou minimizar a existência de competição simultânea por recursos e por objectivos. Tenta-se fazer com que a satisfação dos objectivos individuais de cada agente induza à obtenção do objectivo global, também chamado "bem comum".

Em situações específicas, os agentes podem comprometer a satisfação dos objectivos individuais a favor da obtenção do bem comum do sistema.

Depois de definido o comportamento individual de cada agente é também necessária a implementação de um conjunto de "regras sociais" (regras de interacção) que prevejam a prioridade na obtenção do objectivo global. Caso a obtenção dos objectivos individuais e globais seja disjunta, estas regras devem levar o agente a sacrificar os seus próprios interesses em nome de outros de maior prioridade.

A teoria do jogo concentra-se no estudo dos efeitos que uma potencial acção do agente causará no seu estado interno e no estado dos elementos que o rodeiam. Lida com agentes capazes de avaliar a utilidade das suas acções ou estratégias, constituindo um domínio de investigação relativo à interacção entre entidades computacionais.

2.5.4 Linguagens de Comunicação entre Agentes

Os sistemas multi-agente, pelas suas características de dinamismo, descentralização, distribuição ou proactividade, originam novos requisitos de comunicação entre as entidades nele presentes.

A arquitectura predominante na Internet, o modelo cliente-servidor limita bastante as características proactivas de alguns dos seus elementos, sendo necessário igualmente lidar com vários níveis de heterogeneidade tais como diferentes plataformas, sistemas operativos ou linguagens de implementação.

Na caracterização dos agentes como entidades inteligentes, está implícita a capacidade de comunicar com os restantes através de uma linguagem de entendimento comum, por forma a possibilitar a troca de informação e conhecimento. Este é exactamente o objectivo de uma "Linguagem de Comunicação entre Agentes".

Em (Mayfield, Labrou Finin 1995) apresentam-se os requisitos aos quais deve obedecer uma linguagem de comunicação entre agentes:

Forma Uma boa linguagem de comunicação entre agentes deve ser declarativa, sintácticamente simples e facilmente legível por seres humanos. Deve ser concisa, fácil de gerar e de identificar campos (*parsing*).

Conteúdo A linguagem deve ser estratificada de forma a adaptar-se a diferentes tipos de sistemas. Em particular deve ser feita a distinção entre a linguagem de comunicação (expressa em actos de comunicação) e a linguagem de conteúdo (expressa em factos acerca do domínio). A estratificação facilita a aplicação da linguagem em diferentes domínios, providenciando simultaneamente esquemas para a sua compreensão.

Deve disponibilizar um conjunto de actos comunicativos (performativas) que, apesar de extensível, deve possuir um núcleo que abranja grande parte das noções intuitivas de comunicação.

Semântica É importante que a linguagem não permita interpretações dúbias na transmissão de informação. Uma vez que os sistemas multi-agente podem funcionar em grandes intervalos temporais e espaciais, é importante o tratamento dado às noções de "Espaço" e "Tempo".

Implementação A implementação da linguagem centra-se essencialmente nas vertentes de eficiência e largura de banda ocupada e disponível. Mediante as condições disponíveis é importante a facilidade de criação, transmissão e leitura de mensagens, bem como a quantidade de recursos usada na transmissão. É também importante que seja possível a implementação parcial, evitando que agentes com comportamentos bastante específicos tenham que saber lidar com elevada quantidade de informação.

Rede Uma vez que a generalidade dos sistemas multi-agente funciona em ambiente de rede é importante a facilidade da linguagem se adaptar às modernas tecnologias de

comunicação. Deve suportar os tipos básicos de conexão - ponto-para-ponto, *multicast*, *broadcast*, bem como comunicação assíncrona/assíncrona. Deve disponibilizar um conjunto de primitivas que se adaptem facilmente a estes requisitos.

Ambiente O ambiente onde os agentes interagem tem normalmente características de extrema descentralização e dinamismo, onde podem coexistir diferentes linguagens ou protocolos. É também relevante a capacidade de interoperabilidade com diferentes linguagens e protocolos.

Fiabilidade A fiabilidade na comunicação é também outro aspecto no qual a linguagem de comunicação tem papel relevante. Esta pode disponibilizar mecanismos para facilmente serem detectados erros na transmissão de mensagens ou mensagens incompletas.

A *ARPA Knowledge Sharing Effort* (KSE) foi um consórcio criado com o objectivo de desenvolver convenções que facilitem a partilha e reutilização de conhecimento. Tinha como principais tarefas definir, o desenvolvimento e teste de infra-estruturas para a criação de sistemas cuja funcionalidade é mais relevante em conjunção com outros que funcionando isoladamente.

Estava organizado em quatro grupos, cada um lidando com problemas complementares relativos à tecnologia de representação do conhecimento:

Interlingua Tem por objectivo a definição de uma linguagem comum para partilha de conhecimento tendo publicado a especificação da linguagem KIF (*Knowledge Interchange Formalism*)

Knowledge Representation Systems Foca-se na implementação de estruturas comuns dentro das famílias de linguagens.

Shared and Reusable Knowledge Bases O objectivo deste grupo é facilitar a obtenção de consensos sobre o conteúdo das bases de dados partilhadas. Disponibiliza um repositório de ontologias partilhadas².

External Interfaces Neste caso o objecto da investigação é a interligação dinâmica entre os sistemas baseados em conhecimento e outros módulos, nomeadamente bases de dados convencionais.

Existem várias linguagens para comunicação entre agentes. Algumas das mais usadas são a *FIPA ACL-FIPA Agent Communication Language*³, a *KQML-Knowledge and Query manipulation Language*⁴ ou o *KIF-Knowledge Interchange Format*⁵.

²<http://www.cs.umbc.edu/kse/srkb/>

³<http://drogo.cselt.stet.it/fipa>

⁴<http://www.cs.umbc.edu/kqml/>

⁵<http://logic.stanford.edu/kif/kif.html>

Apesar de todas possibilitarem a comunicação entre agentes, nem todas têm igual finalidade. Algumas estão vocacionadas para a partilha de objectos (Corba e RMI), de conhecimento (KQML, FIPA ACL) ou de intenções, exigindo estas arquitecturas do tipo **Believes-Desires-Intention** (BDI).

2.5.5 Motivações para a Utilização

Existem várias razões justificativas para o crescente interesse na investigação e desenvolvimento de sistemas multi-agente (Sycara 1998):

- Capacidade de resolver problemas de dimensão demasiadamente elevada para um sistema centralizado devido à limitação de recursos.
- Possibilidade de resolução de problemas onde a concentração de informação numa só entidade é inviável, ou represente uma desvantagem em termos de segurança.
- Permitem a interligação e interoperabilidade entre múltiplos sistemas privados de informação. Mesmo que seja viável a implementação de um sistema que englobe toda a informação, a alteração dos requisitos obrigaria à alteração do sistema central. No caso dos sistemas multi-agente a alteração pode resumir-se ao comportamento de um único agente.
- Disponibilizam soluções para problemas que intuitivamente podem ser tratados por uma sociedade de elementos autónomos em interacção. Existem problemas cuja abordagem mais clara passa pela implementação de um sistema com estas características. O agendamento de reuniões entre diversos elementos pode constituir um bom exemplo para este tipo de sistemas.
- Providenciam soluções que utilizam de forma eficiente recursos distribuídos. Em diversos domínios (pesquisa de informação na WEB, análise sismográfica,...) a informação encontra-se distribuída espacialmente sendo este tipo de sistemas uma abordagem viável.
- Disponibilizam sistemas onde o conhecimento esteja distribuído, como por exemplo sistemas de apoio médico, de produção ou engenharia.
- Podem simplesmente melhorar o desempenho em relação a sistemas centralizados nas vertentes de eficiência computacional, fiabilidade, escalabilidade, facilidade de manutenção, flexibilidade ou reutilização.
- Facilitam a implementação de comportamentos inteligentes. Tem sido proposto que a forma mais fácil de modelar comportamentos inteligentes consiste em começar por criar "máquinas sociais" (Dautenhahn 1995). Resulta da teoria sócio-biológica que a inteligência emerge primariamente da necessidade de interacção com outros.

2.5.6 Domínios de Investigação

A investigação associada aos sistemas multi-agente abrange fundamentalmente os seguintes domínios: (Oliveira, Fisher Stepankova 1999):

Arquitecturas e organização A arquitectura física proposta por (Newell Simon 1976) constituiu durante mais de uma década um paradigma segundo o qual apenas os agentes com representação interna do estado do mundo e com capacidades de executar algum tipo de raciocínio podiam ser considerados inteligentes. (Brooks 1986) lançou posteriormente outra ideia, negando a necessidade de representações internas do mundo. Subjacente está o facto de que a inteligência não tem necessariamente que residir em cada agente, podendo alternativamente emergir da interacção entre estes.

As arquitecturas "mentalistas" permitem uma descrição de mais alto nível dos agentes. Como exemplo tem-se a arquitectura do tipo BDI.

Um novo campo de investigação de arquitecturas é o dos "Agentes Emocionais". Pretende-se que a capacidade de decisão seja não só função de uma utilidade (como na teoria do jogo), mas também influenciada pelo estado emocional dos agentes (Sarmiento Oliveira 2003).

Estratégias de coordenação e interacção Podem ser estabelecidos processos de decomposição e distribuição de tarefas com o objectivo de:

- Evitar a sobrecarga de recursos críticos.
- Atribuir tarefas aos elementos mais aptos.
- Possibilitar a sub-decomposição de tarefas.

Existe simultaneamente uma variedade de mecanismos de coordenação de agentes entre os quais:

- Protocolo *Contract Net* (Smith 1980). Propõe uma série de actos negociais divididos em rondas (anúncios, propostas, ...)
- *Partial Global Planning* (Durfée Lesser 1995) e TAEMS (K.Decker V.Lesser 1995). Protocolo de elaboração conjunta de planos entre agentes. Permite a construção de um plano distribuído reconhecido por todos os intervenientes.
- Estratégias de coordenação baseadas no mercado. Modelando o funcionamento de um mercado cada agente pode disponibilizar um conjunto de serviços ou necessidades. Vários tipos de leilão podem ser usados como protocolos eficientes.

Comunicação Tenta responder a questões como: "Como pode um agente comunicar com outros?" ou "Como se entendem?". A entrada e saída de agentes em ambientes abertos, acessível via Internet leva à necessidade de criação de facilitadores e linguagens específicas entendidas por todos os elementos.

Adaptabilidade e aprendizagem Na grande maioria dos casos é difícil determinar o comportamento pretendido para um sistema no momento da implementação. O dinamismo do ambiente obriga à implementação de sistemas adaptativos, com capacidades de aprendizagem. Para tal podem-se utilizar métodos de aprendizagem:

- Supervisionada. É especificado o comportamento desejado para cada situação.
- Por reforço. Apenas é indicado o valor de utilidade de cada acção e o objectivo a atingir.
- Não supervisionada. Não é especificada o comportamento pretendido. A aprendizagem baseia-se num processo de tentativa e retrocesso na procura de melhores soluções. Os algoritmos de agrupamento ("clustering") são normalmente usados em agentes, por exemplo para classificar e agrupar preferências ou características de utilizadores.

2.5.7 Aplicações

Os sistemas multi-agente são actualmente aplicados em diversos domínios, sendo alguns dos mais relevantes (Jennings J.Wooldridge 1998):

Controlo de processos Constitui uma das aplicações mais intuitivas de um sistema multi-agente. Um dos mais conhecidos é o ARCHON, uma plataforma de criação de agentes, que serviu para a aplicação em redes de transporte de electricidade (em uso no norte da Espanha) e controlo de aceleração de partículas.

Produção O YAMS (*Yet Another Manufacturer System*) (Parunak 1987) utiliza o protocolo *Contract Net* para modelar um sistema de produção como um conjunto de "células de trabalho". Estas são agrupadas em sistemas flexíveis de manufacturação (SFM), cada um disponibilizando funcionalidades como pintura, montagem, acabamentos, . . . Cada SFM é representado pelo respectivo agente com o objectivo de administrar de forma eficiente o processo de produção.

Controlo de tráfego aéreo Proposto em 1996 e em funcionamento experimental no aeroporto de Sydney, o sistema OASIS (Ljungberg Lucas 1992) está descrito como um "s sofisticado sistema de controlo de tráfego aéreo". Neste caso os agentes são utilizados para representar os aviões e as várias entidades de controlo. À entrada de um avião no sistema corresponde a criação de um agente com informação e objectivos correspondentes, sendo este o responsável pela interacção com os agentes integrantes do sistema de controlo. Os agentes são implementados numa arquitectura BDI.

Manuseamento de informação A disseminação da Internet e o aumento exponencial da informação aí armazenada propiciou a criação de agentes com o objectivo de adquirir informação relevante e conhecimento na WEB.

A enorme quantidade de informação funciona por vezes como obstáculo à próxima transição pretendida para a sociedade actual: da "Sociedade da informação" para a

”Sociedade do conhecimento”. Neste sentido podem ser atribuídas a agentes tarefas de:

- Filtragem de informação. Neste caso os agentes têm por função eliminar informação potencialmente inútil para um determinado utilizador. Um exemplo bastante utilizado são os agentes encarregues de filtrar as mensagens ou notícias electrónicas destinadas a um utilizador. Com base nas preferências ou contactos, eliminam-se as mensagens potencialmente inúteis para o utilizador representado. Os sistemas *Maxims* e *Newt* (Maes 1994) constituem exemplos de aplicações em funcionamento neste domínio.
- Aquisição de informação. Neste domínio os agentes têm por função, com base em informação existente (normalmente de elevado volume) inferir conhecimento potencialmente útil para o utilizador. O sistema *Zuno* (Ltd 1997) constitui um exemplo de um sistema multi-agente com capacidades de pesquisar diferentes conjuntos de dados desorganizados ou incoerentes e obter alguma informação coerente e relevante.

Comércio electrónico Actualmente, grande parte do comércio electrónico ainda é feita exclusivamente entre seres humanos. No entanto não haverá razão para que uma parte desse comércio não possa futuramente ser efectuada por agentes que os representem. Desde que salvaguardadas as questões de segurança, a disseminação dos mercados electrónicos pode constituir uma vantagem relevante em relação aos processos actuais. Um dos sistemas propostos com essa finalidade, no âmbito *Business-to-Consumer* (B2C) é o *Kasbah* (Chavez Maes 1996).

Por outro lado, o uso de sistemas multi-agente em *Business-to-Business*(B2B), na formação de empresas virtuais permite a criação das chamadas *Instituições Electrónicas*, entidades que resultam da conveniência temporária de associação entre elementos (Rocha 2003).

Monitorização de pacientes O sistema *Guardian* (Hayes-Roth 1995) foi proposto como ajuda ao cuidado dos doentes nas unidades de cuidados intensivos. A motivação residiu na constatação de que o estado de um paciente numa unidade de cuidados intensivos é resultado do trabalho de uma equipa que se organiza e coopera em função do objectivo comum. Esta equipa é formada por elementos com capacidades distintas que devem partilhar informação com vista à aplicação dos cuidados indicados. Na arquitectura proposta intervêm três tipos de agentes:

- Agentes perceptivos e de acção. São responsáveis pelo *interface* entre o sistema e o paciente. Podem monitorizar determinadas funções vitais do paciente e aplicar as acções determinadas pelo sistema.
- Agentes de decisão. São responsáveis pelo processo de decisão, com base na informação recebida dos anteriores.
- Agente de controlo. Devido à delicadeza da área em questão implementou-se um agente com conhecimento detalhado de todos os restantes, e que serve para garantir que o sistema permanece em estado coerente.

Neste caso os agentes partilham a informação através de um sistema de memória partilhada (*blackboard*).

Jogos O domínio da diversão é um dos que intuitivamente pode beneficiar da utilização de sistemas multi-agente, especialmente nos jogos de acção, onde várias personagens com interesses opostos ou similares devem estabelecer relações de cooperação ou antagonismo com vista à obtenção de um objectivo. Em (Wavish Graham 1996) descrevem-se vários sistemas com base na tecnologia de agentes utilizados na construção de jogos de computador.

Teatro e cinema interactivos A interactividade cada vez mais presente nos domínios do teatro e cinema pode incentivar a implementação de agentes com características o mais similares possíveis a personagens reais. (Hayes-Roth, Hewett, Waashington, Hewett Seiver 1995) descrevem um sistema onde serem humanos interagem em narrativas dinâmicas. O comportamento automático e adaptativo das personagens fictícias é neste caso efectuado através de um sistema multi-agente.

2.5.8 Aplicações no Domínio dos Transportes

No domínio dos transportes a grande maioria dos sistemas multi-agente implementados é respeitante aos rodoviários. Provavelmente originado pelo congestionamento de tráfego nas grandes metrópoles e pelos crescentes problemas de ordenação e controlo do tráfego nas cidades, têm sido este o domínio preferencial de investigação na área.

Grande parte dos sistemas estudados optaram por abordagens de características similares, dividindo a área a abranger pelo sistema por vários agentes com competência de decisão local. Alguns desses sistemas optaram por fazer representar simultaneamente os veículos em circulação na rede ou as próprias vias de comunicação.

Alguns dos sistemas apresentados centram-se também na aplicação de métodos de aprendizagem como tentativa de optimizar o controlo de tráfego.

2.5.8.1 TraMas

Em (Oliveira Fernandes 1999), através do sistema TraMas, procurou-se averiguar a viabilidade de um sistema multi-agente para o domínio dos transportes rodoviários, bem como estudar diferentes modelos e estratégias de cooperação em sistemas multi-agente.

Na abordagem prevista cada agente representa um cruzamento e é responsável pela regulação do tráfego nessa área. Tem independência suficiente para decidir localmente, mas está também preparado para a comunicação com outros agentes.

Na arquitectura proposta cada agente contém três níveis:

Nível Cooperativo Tem por objectivo requisitar ajuda sempre que o agente for incapaz de

agir localmente, partilhar informação com outros agentes e efectuar pedidos a agentes vizinhos.

Nível de Decisão Neste nível decidem-se as acções a executar com o objectivo de evitar tempos de espera superiores a um limite e evitar a concentração de tráfego.

Nível de Controlo Serve para garantir a coerência dos estados de cada um dos semáforos.

Numa fase posterior foi acrescentado um nível extra (Nível de Aprendizagem) com o objectivo de lhes proporcionar mecanismos de aprendizagem. Este nível foi sub-dividido em dois módulos:

Módulo de Aprendizagem Responsável por monitorizar a eficiência das acções e decisões tomadas pelo agente.

Módulo de Decisão Responsável pelas decisões a tomar tendo por base métodos de aprendizagem por reforço.

2.5.8.2 Engenharia da Mobilidade - Modelo "Co-fields"

Em (Mamei Mahan 2003) propõe-se um sistema de controlo de movimento bastante peculiar, aplicado ao tráfego rodoviário através de um sistema multi-agente.

Constitui objectivo básico do sistema a coordenação de movimentos entre um grupo de utilizadores que conduzem os seus veículos dentro de uma cidade. Partindo da hipótese de que toda a cidade está abrangida por uma rede computacional com terminais em cada cruzamento e que todos os veículos estão equipados com um sistema computacional onde está instalado o agente representante do veículo. Deve também estar disponível um sistema de localização preciso (GPS) (Figura 2.3).

Mediante os pressupostos referidos anteriormente, o trabalho centra-se no processo de coordenação dos utilizadores por forma a evitar as filas de tráfego e o tempo de espera associado.

Neste sistema introduz-se a noção de "co-field" com o objectivo de delegar nos agentes controladores da infra-estrutura a responsabilidade de coordenar o tráfego circulante.

Um "co-field" (Figura 2.4) consiste numa representação tridimensional de uma parcela da rede a controlar. Cada agente é responsável pela definição dinâmica da orografia do respectivo "co-field" por forma a evitar congestionamentos de tráfego.

Com base no volume momentâneo de tráfego em cada parcela do "co-field" o agente controlador vai alterar a respectiva forma segundo as regras:

- Zonas com maior volume de tráfego são representadas por zonas de maior altitude.
- Zonas sem tráfego são representadas por depressões na respectiva orografia.

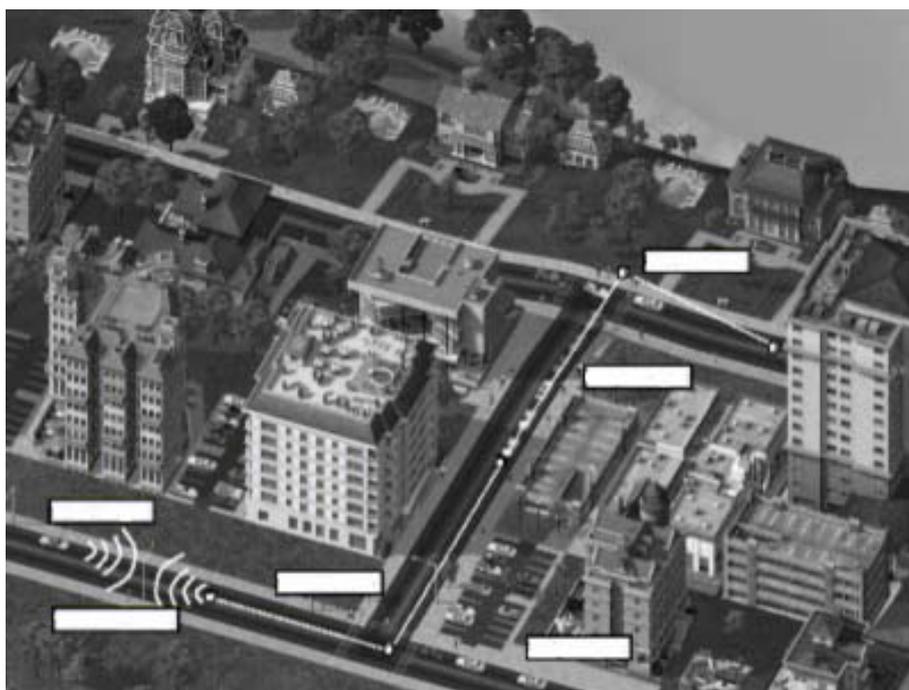


Figura 2.3: Sistema coordenador da mobilidade.

Ao transmitir a forma momentânea do "co-field" aos agentes representantes dos veículos incumbe-se a cada um a tarefa de "escorregar" através da estrutura tal como uma bola através de uma colina.

O modelo de funcionamento "co-fields" pode ser resumido através dos seguintes pontos:

- O ambiente é representado por agentes computacionais, com informação sobre o volume de tráfego em cada parcela da sua zona e com capacidades de comunicação e coordenação com agentes vizinhos.
- A coordenação entre os veículos circulantes é realizada na movimentação dos veículos através das "ondas" de cada "co-field".
- A movimentação de cada veículo e a dinâmica do ambiente induzem alterações na estrutura do respectivo "co-field", originando um ciclo recursivo que influencia o movimento de outros.
- O *feed-back* proveniente de cada integrante do sistema (veículos e controladores) permitem que o sistema se auto-organize por forma a otimizar a coordenação do tráfego.
- Cada agente representante de um veículo decide a sua próxima direcção com base na orografia recebida e no próximo destino desejado. Este facto torna possível que um veículo alcance locais de maior altitude em relação à posição actual.

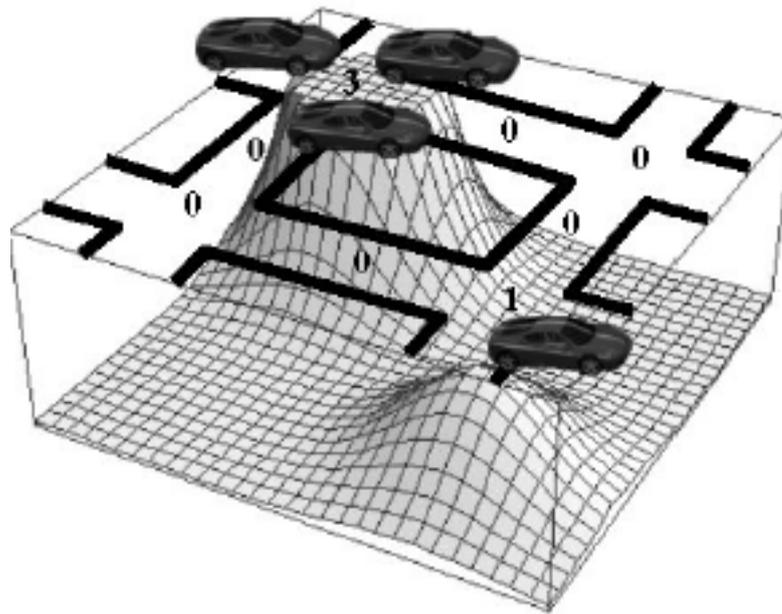


Figura 2.4: Estrutura "co-field".

2.5.8.3 Smart Light

(Gieseler, Graves Maher n.d.) propõe um sistema multi-agente baseado em métodos de aprendizagem por reforço para o controlo dos semáforos de uma rede rodoviária. O objetivo consiste, tal como na generalidade dos casos, em maximizar o fluxo de tráfego na área coberta pelo sistema.

É esperança dos autores que o sistema multi-agente seja capaz de aprender uma política global óptima de controlo que minimize os tempos médios de espera para cada veículo.

A abordagem proposta prevê a representação de cada cruzamento por um agente. Cada um destes recebe informação através dos seus sensores (processo de simulação) e dos agentes vizinhos. Consideram-se agentes vizinhos os quatro agentes que controlam a extremidade oposta de cada estrada adjacente ao cruzamento sob responsabilidade do agente.

O estado interno de cada agente divide-se em duas partes:

- Um contador para cada semáforo controlado pelo agente. Cada um destes conta o tempo que um veículo esteve consecutivamente à espera do sinal de passagem.
- Um conjunto de variáveis booleanas que especificam quais os agentes vizinhos que permitiram a passagem de tráfego em direcção ao cruzamento do próprio agente.

A componente de aprendizagem do sistema usa métodos de aprendizagem por reforço na tentativa de aprender uma política óptima de controlo em função do estado do tráfego.

Cada agente tenta individualmente aprender a política óptima de controlo para o seu cruzamento. Ao permitir a comunicação e interacção entre agentes espera-se que da aprendizagem resulte uma política óptima global.

Através do algoritmo *Aprendizagem-Q* pretende-se encontrar a acção óptima para cada estado, sendo cada um constituído por um par $\langle \text{estado } s, \text{acção } a \rangle$.

Os estados apresentados ao sistema de aprendizagem são construídos a partir das percepções do agente. Este recebe o número de passos que o primeiro carro esperou pela cedência de passagem em cada uma das direcções pertencentes ao cruzamento e informação acerca do número de q passos atrás em que cada agente vizinho permitiu a passagem de veículos em direcção ao próprio cruzamento.

A recompensa de cada acção é calculada segundo os tempos de activação do sinal de passagem no cruzamento e das transferências de recompensas efectuadas pelos agentes vizinhos. Quanto menor for o tempo de paragem no cruzamento maior será a recompensa (bem individual), o mesmo se passando pelo tempo de resposta a pedidos efectuados pelos agentes vizinhos (bem global). O valor atribuído a cada estado é reflexo dos objectivos de cada agente e da utilidade global do sistema.

2.5.8.4 *Adaptative Urban Traffic Control - AUTC*

Uma abordagem com uma estrutura ligeiramente distinta foi proposta por (Roozmond Rogier 2000) na implementação do sistema multi-agente. Tal como ilustrado na figura 2.5 aí interagem os seguintes tipos de agentes:

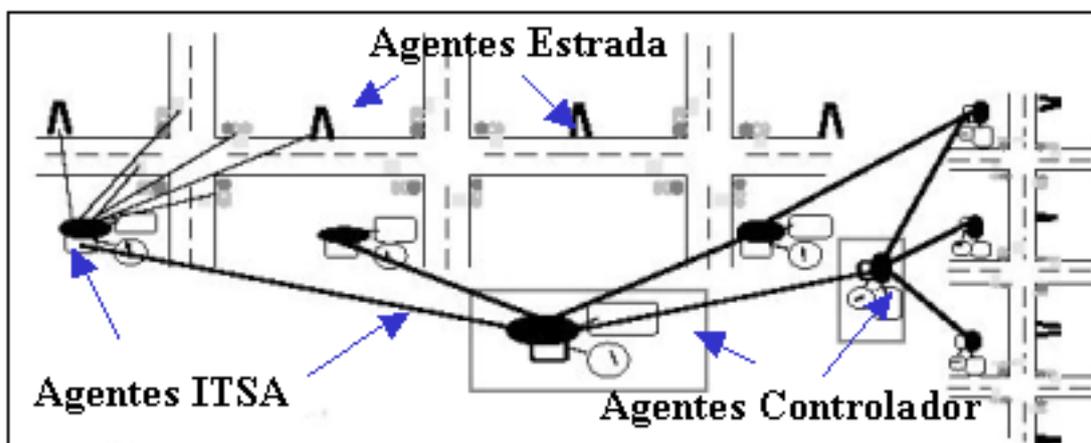


Figura 2.5: Arquitectura proposta (AUTC).

- Agentes "Controlador". Responsáveis por áreas específicas da rede rodoviária, recebem informação sobre o estado do tráfego em cada via aí localizada e
- Agentes de sinalização (ITSA). Hierarquicamente subjugados aos anteriores, controlam um cruzamento de estradas,

- Agentes de estrada. Representam uma ligação entre dois cruzamentos e guardam informação acerca do volume de trânsito associado.

Este sistema tem simultaneamente uma vertente predictiva. Implementam-se capacidades de previsão das futuras condições de tráfego. Com base no historial, os agentes reguladores dos cruzamentos tentam prever as condições num futuro bastante próximo. Simultaneamente um modelo de meta-aprendizagem compara as previsões com as condições efectivamente verificadas e efectua as correspondentes actualizações ao sistema de aprendizagem. .

2.5.8.5 CRASH

É um sistema com características bastante peculiares e foi proposto por (Carlos 2003). Contém uma componente multi-agente e destina-se igualmente ao controlo de tráfego rodoviário.

No sistema proposto intervêm dois tipos de agentes: "Cruzamento" e "Carro".

Cada elemento de "Carro" representa uma viatura em circulação na rede e baseia-se no modelo proposto por (Antoniotti Göllü 1997) no projecto *Smart Automated Highway Simulation* (Smart-AHS). Contém um estado (posição, velocidade e aceleração) e as transições aplicáveis a cada estado.

O agente "Cruzamento" tem capacidades de detecção de aproximação de carros ao respectivo cruzamento. Ao detectar um elemento, simula todas as transições de estado aplicáveis até o carro atingir o cruzamento. Se no momento em que for prevista a passagem do carro pelo cruzamento este estiver ocupado, o "Cruzamento" pede ao "Carro" para diminuir ligeiramente a aceleração. O processo é iterativo até encontrar um momento livre que permita a passagem do carro.

No sistema CRASH intervêm ainda mais três entidades, não classificadas como agentes:

Controlo Objecto a que todos os agentes "Carro" comunicam a sua posição e recebem informação sobre os elementos existentes na sua vizinhança.

Rua Previne a saída dos agentes de cada uma das ruas, validando as transições propostas pelos agentes "Carro".

Estatística Objecto encarregue de efectuar toda a componente estatística da simulação.

2.5.8.6 Sistema de Controlo do Fluxo de Tráfego

Proposto em 1994 e implementado sobre o sistema dMARS, (Gabri, Norling, Tidhar, Sonenberg Howden 1994) propõem um sistema multi-agente onde interagem apenas dois tipos de agentes: "Intercessão" e "Rua". Tal como ilustrado na figura 2.6, cada rua (Rua_i) e

cruzamento de ruas ($Cruzamento_i$) são representados pelo respectivo agente procurando que através da comunicação com os seus vizinhos se crie um comportamento emergente que conduza à obtenção de uma solução para o problema.

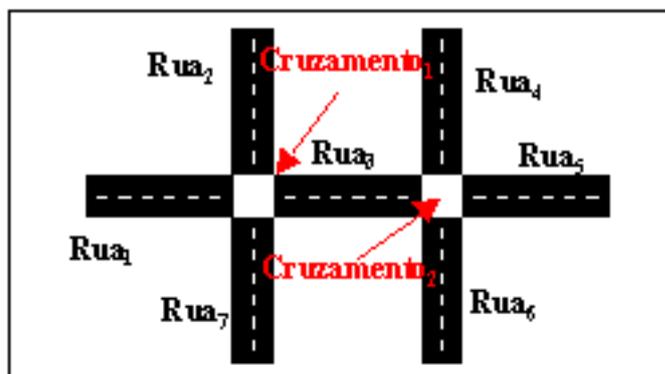


Figura 2.6: *Traffic Flow Control System.*

Todos os agentes do sistema recebem como entrada o volume de tráfego na sua área, sendo responsáveis por manter essa informação e efectuar a respectiva comunicação aos vizinhos.

Dentro dos agentes "Rua", o estado interno é definido pelos seguintes parâmetros:

- Prioridade. Varia de acordo com a hora do dia e serve como factor de adaptabilidade ao volume de tráfego.
- Capacidade. Serve para definir a capacidade de retenção de tráfego existente dentro da rua, como por exemplo um parque de estacionamento, centro comercial ou instalação desportiva.
- Escoamento de tráfego. Define, com base na análise do historial, a percentagem de tráfego que sai em direcção a cada uma das ruas adjacentes.
- Recepção de tráfego. É transmitido pelos agentes vizinhos e contém informação sobre a proporção relativa de tráfego que vem em direcção à própria rua.
- Contagem. Total de veículos em circulação na rua representada pelo agente.

Os agentes "Intercessão", ao serem responsáveis pelos cruzamentos entre ruas, constituem a componente activa do sistema, isto é, são eles que ajustam os tempos de passagem de cada direcção por forma a propiciar a maior fluidez possível. A sua base de funcionamento consiste num algoritmo que, com base no volume de tráfego em cada direcção, coloca os sinais luminosos no estado respectivo.

O estado interno de cada agente é definido pelos seguintes atributos:

- Estado dos semáforos. Guarda informação sobre o estado actual de cada semáforo que compõe o cruzamento.

- Tempo de análise. Representa o intervalo mínimo em que os agentes reavaliam a situação de tráfego. Este um valor é autonomamente ajustado pelo agente.
- Volume de tráfego. Valor calculado com base no histórico e que representa o volume de tráfego proveniente de cada rua adjacente ao cruzamento.

2.5.8.7 DAARTS

O sistema designado por DAARTS - *Decentralized Adaptive Agents for control of Traffic Signals* (Manikonda, Levy, Satapathy, Lovell, Chang Teittinen 2003) tem também por objectivo a implementação de um sistema multi-agente de controle do tráfego rodoviário.

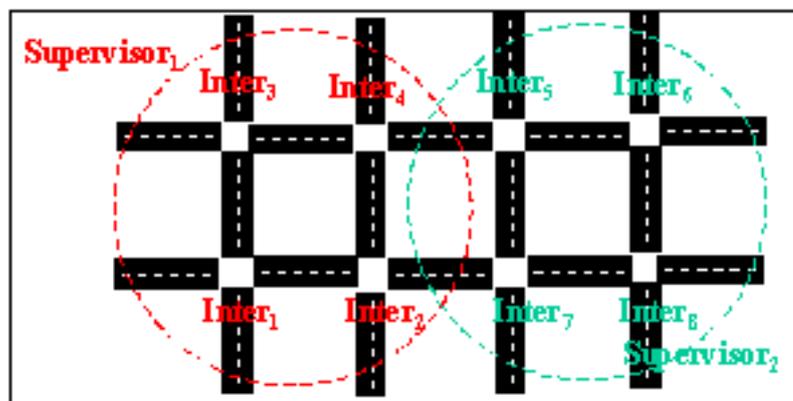


Figura 2.7: DAARTS.

Tal como ilustrado na figura 2.7, possui uma arquitectura hierarquizada em dois níveis:

- No nível inferior é composta pelos agentes "Intercessão" representantes de um cruzamento de ruas e que, baseados em informação local e proveniente dos agentes vizinhos controlam os tempos de passagem para cada via.
- Num nível superior existem os agentes "Supervisor" que abrangem conjuntos de cruzamentos e têm em conta a dinâmica do tráfego com o objectivo de prever as condições de tráfego próximas.

2.5.8.8 Linha Milão-La Spezia

Comparativamente ao domínio dos transportes rodoviários, existem bastantes menos sistemas multi-agente propostos para os ferroviários. Um deles foi o proposto por (Cuppari Guida n.d.) cujo objectivo consistiu na implementação de um sistema multi-agente destinado à gestão do tráfego ferroviário de mercadorias entre as estações de Milão e La Spezia. Esta é uma das linhas com maior volume de tráfego na Itália, sendo caracterizada pelo mínimo espaçamento entre composições. O porto de LA Spezia tem ligação directa com

o terminal ferroviário. Ao chegar à estação o comboio tem que obter uma autorização de carregamento dependendo esta de vários factores, como o número de comboios em espera, o número de operações já efectuadas ou as dimensões do comboio e local de carga ou descarga.

A linha ferroviária entre Milão e La Spezia é composta por três secções principais (S_1 , S_2 e S_3) delimitadas respectivamente pelos nós de Voghera, Arquata e Génova. Cada um destes tem ligação com outras linhas de menor capacidade (Figura 2.8).

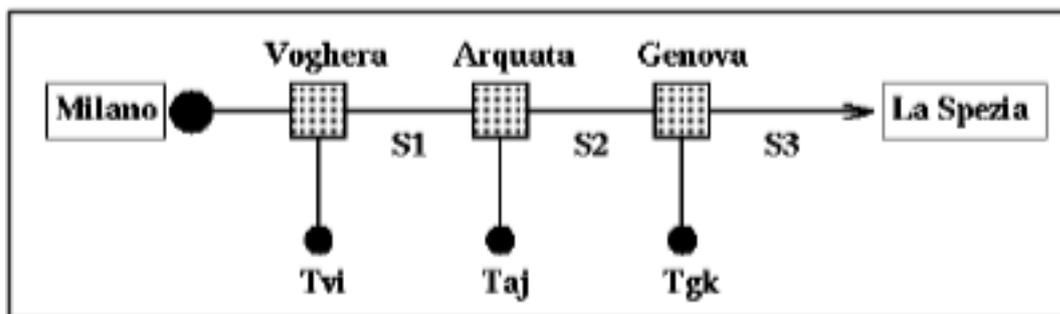


Figura 2.8: Linha ferroviária entre Milão e La Spezia.

Na arquitectura idealizada intervêm os seguintes agentes, cada um deles representantes da respectiva entidade da rede:

- T_{vi} , T_{aj} e T_{gk} são os representantes respectivamente dos nós de Voghera, Arquata e Génova.
- COIMM é o gestor de tráfego, responsável pela supervisão do tráfego em toda a extensão da linha.
- Coordenadores de tráfego STC_1 , STC_2 e STC_3 responsáveis pela gestão do tráfego nas respectivas secções S_1 , S_2 e S_3 .
- Agente responsável pelas chegadas e partidas de comboios à estação de La Spezia (PC).

O objectivo é produzir um sistema integrado que efectue a gestão autónoma das cargas e descargas dos comboios na estação de La Spezia e do respectivo tráfego pela rede ferroviária até à estação de Milão

O sistema foi idealizado para actuar fundamentalmente ao nível dos seguintes processos:

Administração dos pedidos de carga e descarga No terminal ferroviário e portuário de La Spezia todos os pedidos para carga ou descarga de mercadorias têm de obter autorização junto da administração. O terminal envia o pedido ao agente COIMM, este actualiza o índice de congestionamento da secção de rede (número de comboios por quilómetro). Consoante o valor esteja abaixo ou acima de um limiar assim o pedido é aceite ou recusado, sendo o resultado comunicado ao agente PC.

Autorização de circulação Após efectuar a carga ou descarga na estação de La Spezia é necessário efectuar um pedido de circulação na linha com direcção a Milão. Este pedido é efectuado ao agente STC específico, devendo este com base na concentração de tráfego momentânea deferir ou indeferir o pedido. Caso o pedido seja aceite ele é transmitido ao agente COIMM (conhecedor do estado de toda a rede) que terá que o confirmar. Só após este processo uma composição tem autorização para circular na rede ferroviária.

2.5.8.9 Sistema Multi-Agente de Linguagem Natural - SMALN

Outra abordagem de características bastante peculiares foi proposta em (Huber Ludwig 2002). Os autores propõem a implementação de um sistema multi-agente para controlo de uma rede ferroviária, com uma componente de interacção com o utilizador através de um módulo de processamento de linguagem natural.

Esta proposta centra-se essencialmente no processamento de linguagem natural, não tendo sido privilegiada a verosimilhança com os métodos actuais de controlo e sinalização (As redes ferroviárias estão divididas em blocos fixos e a sinalização é fixada lateralmente aos troços ferroviários).

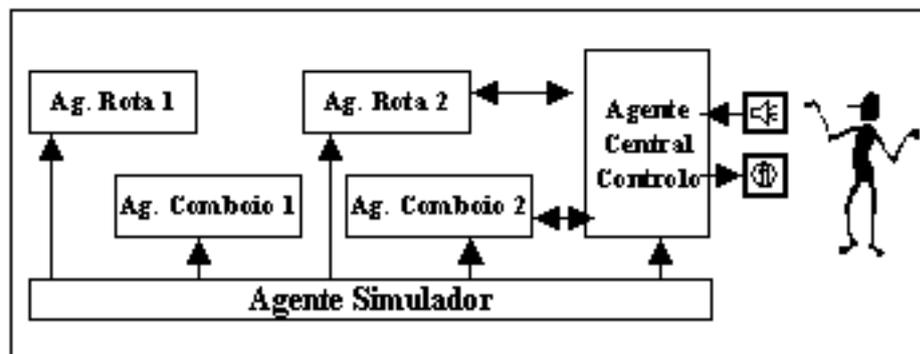


Figura 2.9: Arquitectura proposta pelo sistema SMALN.

Tal como a figura 2.9 ilustra, propõe-se uma arquitectura com quatro tipos de agentes:

Agente Central de Controlo Este é o agente responsável pelo *interface* entre o módulo de processamento de linguagem natural e o sistema multi-agente. Serve para receber regras e restrições provenientes do utilizador, bem como para lhe apresentar propostas de solução de conflitos.

Agentes de Rota Cada um destes elementos é responsável pelo controlo do tráfego numa determinada parcela da rede ferroviária, devendo accionar as agulhas de selecção de trajecto de acordo com os planos estabelecidos.

Agentes Comboios Representam os comboios que circulam na rede ferroviária.

Agente Simulador Simula todas as operações ferroviárias e controla o *interface* com o utilizador.

O agente central de controlo começa por receber, através do módulo de linguagem natural, um pedido da forma: "Mover o comboio X da estação E_1 para a E_2 ".

O agente torna-se responsável por averiguar se o estado é consistente, isto é, se o comboio X está efectivamente na estação E_1 , e seguidamente por encontrar o melhor trajecto possível para satisfazer o pedido. Para tal entra em contacto com os agentes de rota, informando-os dos momentos previstos para entrada e saída do comboio na respectiva área.

Aos agentes de rota cabe a manutenção de um estado consistente por forma a satisfazer o plano recebido. Sempre que dois planos entrarem em conflito e lhes for impossível conciliá-los, efectuam a respectiva comunicação ao agente central de controlo.

O agente central tenta primeiramente arranjar planos alternativos que não comprometam nenhum dos objectivos traçados. Caso tal não seja possível o agente vai entrar num processo de negociação com o utilizador.

É aqui que reside o aspecto especialmente característico do sistema. O agente pode entrar em contacto (através do módulo de processamento de linguagem natural) com o utilizador para lhe comunicar a impossibilidade de execução de alguns dos objectivos anteriormente traçados.

Nestes casos o utilizador deve especificar quais os objectivos prioritários, por forma que, ao estabelecer uma hierarquia, possibilite automaticamente a resolução dos conflitos.

O utilizador acaba por ter um papel essencial na resolução dos conflitos de tráfego, o que apesar de poder proporcionar soluções mais de acordo com as suas expectativas, retira aos agentes características de autonomia.

Módulo de Linguagem Natural

O módulo de linguagem natural funciona como *interface* entre o utilizador e o sistema multi-agente, e pode subdividir-se em dois módulos:

Módulo de análise sintáctica e composição semântica A partir de informação recebida através de um microfone, este começa por efectuar a análise sintáctica da informação recebida. seguidamente é elaborada uma árvore com as possíveis interpretações semânticas.

Módulo de diálogo Recebendo a árvore construída pelo módulo anterior, tem por função disponibilizar informação perceptível para o utilizador do sistema.

2.5.9 Vantagens Associadas ao Domínio dos Transportes

Existem várias razões justificativas para a adoção de um modelo baseado em agentes em problemas similares ao do tráfego ferroviário (Cuppari Guida n.d.):

- Muitos dos problemas de planeamento, escalonamento ou ordenamento são caracterizados pela dificuldade em obter uma visão global do problema. A única opção viável corresponde à decomposição do problema em vários e procura de soluções óptimas locais para cada um, sendo este comportamento propício a sistemas multi-agente.
- São normalmente problemas com ambientes extremamente dinâmicos, muitas vezes com informação incompleta ou com elevado volume de ruído. Esta é também uma das características normalmente associadas a agentes, a capacidade de lidar com informação incompleta, sendo capaz de encontrar uma boa solução na impossibilidade de atingir a solução óptima.
- O domínio dos transportes é inerentemente distribuído e descentralizado, sendo composto por uma série de entidades em comunicação. Um modelo computacional onde cada agente representa uma entidade e troque informação, planos ou desejos com outros é mais intuitivo que outro centralizado para onde deve fluir toda a informação.
- Escalabilidade e expansibilidade do sistema. Modelos centralizados limitam inevitavelmente a escalabilidade de um sistema. Pelo contrário, à alteração dos requisitos de um sistema multi-agente pode corresponder simplesmente a introdução de um novo tipo, incorporação de mais agentes ou modificação do comportamento dos existentes.
- Segurança e robustez. Este aspecto resulta dos anteriormente descritos e é também de importância decisiva para um sistema do qual dependem vidas humanas. A descentralização do controlo e da informação torna bastante menos provável a ocorrência de falhas com deterioração total do sistema, ou quebra das condições de segurança. Os agentes estão normalmente preparados para lidar com situações de excepção, possuindo mecanismos de adaptabilidade por forma a garantir patamares mínimos de segurança e funcionalidade.

Conclusão

Embora sendo uma tecnologia relativamente recente, os exemplos sumariamente descritos ilustram a aplicabilidade da tecnologia em domínios com as características enunciadas. O incremento de interesse e desenvolvimento da investigação neste domínio induzirá previsivelmente à sua crescente relevância na implementação de sistemas computacionais.

Capítulo 3

Sistemas de Controlo e Sinalização Ferroviária

Este capítulo é referente ao domínio onde se insere o sistema implementado. Resumem-se os processos de controlo de tráfego ferroviário, desde os primórdios, no início do século XIX, até aos actuais métodos e sistemas de sinalização e controlo.

”Two trains that are not in the same place at the same time cannot collide.” (Bej 1998)

A frase acima transcrita, apesar de óbvia, descreve na forma mais simples o objectivo de qualquer sistema de controlo e sinalização, aplicando-se igualmente a quaisquer corpos em movimento.

Quanto mais complexo, veloz, dispendioso, ou mais vidas humanas envolva o movimento maior será a necessidade de existência de mecanismos de controlo.

Os principais objectivos de um sistema de controlo e sinalização podem resumir-se em:

Segurança Prevenir colisões e garantir espaçamento mínimo entre elementos.

Controlo Atribuir prioridades, minimizar tempos de espera e maximizar fluxos.

Informação Providenciar informação às pessoas envolvidas.

3.1 Evolução Histórica

3.1.1 Primórdios

Em 1814, aquando da entrada em funcionamento da locomotiva George Stephenson (Ganguly 2002), a sinalização não foi um aspecto considerado relevante. Cada comboio era

controlado apenas pela visão do seu maquinista, sendo iniciado um processo de travagem após o vislumbre de outro.

Anteriormente, por volta de 1806, já a circulação de vagões puxados por tracção animal nas minas de ouro e carvão era dirigida por meio de sinais efectuados com a mão ou braço por pessoal específico para esse fim.

O aumento da velocidade dos comboios e da consequente distância de travagem rapidamente levou à conclusão de que não era seguro iniciar a paragem de um comboio apenas quando outro era avistado.

Surgiu então o primeiro método de controlo. Consistia na definição de um intervalo para a partida de comboios das estações. Inicialmente de dez minutos, este valor era controlado por alguém que, munido de bandeiras coloridas, apresentava a correspondente sinalização.

Imediatamente após a partida de um comboio era mostrada uma bandeira vermelha (Perigo), após cinco minutos passava para a cor amarela (Atenção) e no final de mais cinco minutos apresentava a bandeira verde (Livre) permitindo a partida da próxima composição.

Caso os comboios e as linhas fossem completamente fiáveis, e viajando todos a velocidade constante, este seria um sistema fiável, o que não se verificou provocando inúmeros acidentes de consequências trágicas (Moncaster 2002).

Por outro lado, o facto de permitir a passagem de apenas seis composições por hora contribuiu para que deixasse de responder às solicitações geradas pelo crescente volume de tráfego.

Como se verá de seguida, a introdução do telégrafo está na génese da forma encontrada para solucionar o problema.

3.1.2 Blocos

Foi idealizado um sistema que efectuava a divisão da rede em blocos (segmentos de linha). Cada bloco estava sob o controlo de um sinaleiro, que tendo um comboio no respectivo bloco accionava um sinal como o indicado na figura 3.1a.

Quando o sinaleiro recebia informação de que comboio tinha saído do bloco o sinal passava a ter o aspecto da figura 3.1b. A fiabilidade dessa informação era o factor crítico do sistema, e apenas foi resolvido através do uso do telégrafo como forma de transmissão de informação.

Em 1872 o Dr. William Robinson inventou um circuito fechado que respondia de forma mais eficiente ao problema em questão. Consistia na implementação de dois sensores mecânicos, um em cada extremidade do bloco, que accionavam autonomamente o respectivo sinal. Ainda hoje os actuais sensores de monitorização do movimento dos comboios permanecem fiéis a este princípio (Ganguly 2002).

A noção de bloco representou um avanço significativo na metodologia de controlo do tráfego de redes ferroviárias, sendo ainda utilizada por grande parte dos sistemas em funcionamento.



(a) Sinal de perigo.

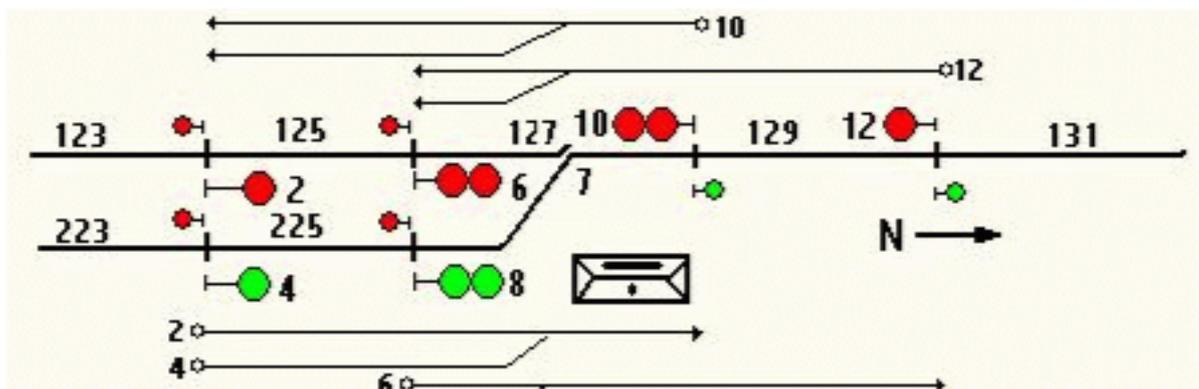
(b) Sinal "livre".

Figura 3.1: Sinais ferroviários.

3.1.2.1 Interlocking

O sistema de blocos fixos demonstrou a sua fiabilidade nos casos em que, em cada linha, todos os comboios viajavam na mesma direcção. Para controlar deslocações bidireccionais foi implementado um sistema, cuja essência ainda se mantém, de prevenção de colisões frontais: o *Interlocking*.

Este sistema serve para evitar que, em locais onde exista intercessão de linhas (junções), dois sinais em conflito levem a uma colisão frontal (Figura 3.2).

Figura 3.2: *Interlocking*.

Apesar de ser uma noção bastante antiga, continua a constituir o princípio básico de funcionamento dos actuais sistemas de controlo e sinalização, existindo várias empresas a disponibilizar sistemas computacionais específicos para o efeito (Figura 3.3).



Figura 3.3: Sistema integrado de *interlocking* (ALSTOM).

3.1.3 Desenvolvimento

A forma de controlo do tráfego ferroviário manteve-se relativamente estável durante várias décadas existindo apenas a necessidade de efectuar alguns ajustes, originados pelo aumento das velocidades alcançadas por cada comboio.

3.1.3.1 Sinais Multi-Aspectos

Paralelamente à noção de bloco foi idealizada uma sinalização de *dois aspectos* (verde, vermelho) indicando se o bloco estava ou não ocupado.

Devido à crescente velocidade, os comboios começaram a não ter espaço para travar ao receber informação da ocupação de um bloco. Por esta razão surgiram sinalizações de três e quatro aspectos, fornecendo informação do número de blocos livres que cada comboio tem à sua frente.

A figura 3.4 ilustra o funcionamento de um sistema de sinalização de quatro aspectos. O sinal verde indica que os próximos três blocos estão livres, dois sinais amarelos indicam dois blocos livres e um único sinal amarelo indica que apenas o próprio bloco está livre.

Na sinalização de três aspectos suprime-se o sinal "duplo amarelo".

3.1.3.2 *Automatic Train Protection* - ATP

Concebido especialmente para as redes metropolitanas, onde a concentração de composições é maior, foi pensada uma forma de prevenção de erros humanos, ou seja, a desobediência aos sinais apresentados (SPAD - Secção F). Com intervalos entre composições

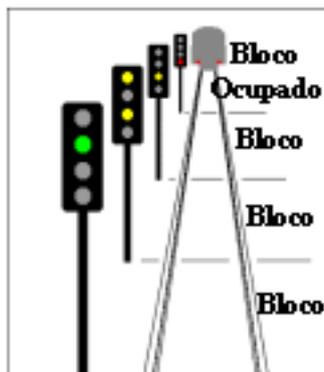


Figura 3.4: Sinalização de quatro aspectos.

diminutos existiam casos em que o maquinista de uma composição tomava para si um sinal que estava a ser mostrado à composição que seguia imediatamente à sua frente.

Foram idealizadas duas soluções:

Surgiu o sistema *Automatic Warning System - AWS*, que consiste num mecanismo implementado entre os carris que mecânica, electromagnética ou digitalmente, acciona o travão de emergência da composição que desrespeitou o sinal.

Foi criado um bloco de segurança adicional, designado por *overlap* na separação de composições adjacentes. Ao aumentar o espaçamento entre comboios, a segurança aumenta na razão inversa do volume máximo de tráfego para a rede, dando origem ao sistema descrito a seguir.

3.1.3.3 *Distance-To-Go*

O sistema de controlo baseado na noção de bloco tem o problema de operar numa sequência de valores limite discretos. Ainda mais grave é o facto de potencialmente existirem dois blocos vazios entre composições, o que diminui a capacidade da rede.

No método *Distance-To-Go*, através de um sistema computacional incorporado no comboio com informação detalhada de toda a rede e recebendo informação acerca da sua localização, é calculada a força de travagem necessária para parar antes do próximo obstáculo (provavelmente no final de um bloco).

É necessário actualizar constantemente a posição (através da colocação de sensores de rotação nas rodas e implementação de dispositivos de localização nas linhas) e refazer os cálculos da força de travagem a aplicar. A informação destinada ao comboio pode ser transmitida sob a forma de sinais enviados através dos carris e captados por uma antena localizada na parte inferior do comboio.

Este sistema tem como desvantagem a deterioração do sinal a partir dos 300 metros, o que obriga à instalação de equipamento adicional para o manter estável.

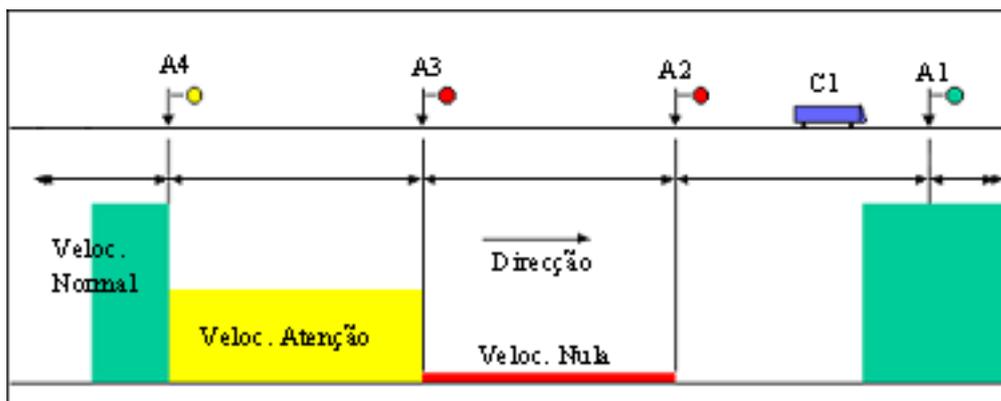


Figura 3.5: Velocidade permitida pelo método tradicional de blocos fixos.

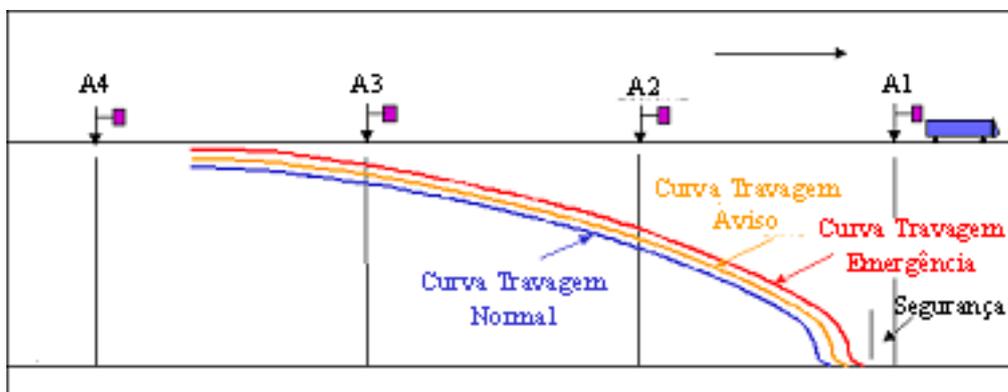


Figura 3.6: Velocidade permitida pelo método *Distance-To-Go*.

As figuras 3.5 e 3.6 exemplificam o acréscimo de velocidade permitida a um comboio.

Imagine-se uma composição localizada atrás do bloco A_4 . No caso do método tradicional a cor do sinal condiciona a velocidade permitida para o comboio durante todo o bloco. Desta forma, como A_2 está reservado para zona de segurança, aparece à sua entrada o sinal vermelho (apesar de estar desocupado), não podendo por isso ser ocupado.

Neste método cada comboio possui informação precisa sobre a distância a percorrer até ao final de cada bloco, pelo que lhe bastará calcular a força de travagem necessária para se imobilizar antes do limite de segurança estabelecido (*overlap* - Secção F).

São normalmente enviados três valores limite para o comboio se imobilizar. O valor de referência (normal), outro ligeiramente mais permissivo (aviso) e um terceiro (emergência) que não poderá em caso algum ser ultrapassado. Nesta eventualidade é automaticamente accionada a travagem de emergência para a composição, constituindo regra dos transportes ferroviários que uma travagem de emergência não poderá, em caso algum, ser anulada.

Evita-se o desperdício de um bloco, garantindo a segurança e aumentando a capacidade da rede.

3.1.4 *On-Cab Signalling*

A possibilidade de transmissão de informação entre o comboio e o sistema de controlo permitiu o envio da sinalização directamente para a cabine da locomotiva.

A informação, transmitida via rádio ou através da própria linha, é captada por receptores colocados na cabine de controlo do comboio e apresentada num painel específico para o efeito.

Este sistema apresenta como vantagens o facto de eliminar a possibilidade de interpretações incorrectas da sinalização, em que um comboio interpreta como para si uma ordem enviada para outro, e exigir a instalação de menos equipamento físico.

3.1.5 Blocos Móveis

Por ser bastante intuitiva, a ideia de bloco móvel surgiu há bastante tempo e apenas dificuldades técnicas ao nível da fiabilidade das comunicações, impediram a sua implementação. A ideia essencial consiste em fixar os blocos tomando como referência os comboios e não a pista.

Sendo um bloco uma distância de segurança destinada a evitar o choque entre comboios, será mais lógico medir essa distância tendo por base o comboio do que a linha. O problema é que tomando o comboio como referência, devido à sua mobilidade é necessário recalculá-lo a cada instante a dimensão e localização de cada bloco. A ideia consiste em implementar uma distância de segurança à frente e atrás de cada comboio, que não pode ser ultrapassada por nenhuma outra composição.

Estes dois limites formam um véu protector para a composição e garantem a sua segurança nesse intervalo. O dinamismo da rede obriga a que continuamente sejam recalculados os valores fronteira para os blocos e transmitidos para o respectivo comboio.

3.1.5.1 Exemplo de Funcionamento

Tome-se o exemplo ilustrado na figura 3.7.

No instante t_1 um comboio recebe informação acerca do bloco a si associado. São fixadas d_1 e d_2 como distâncias delimitadoras da área exclusiva do comboio à frente e à retaguarda respectivamente. Esta é a noção de *Movement Authority Limit - MAL* (Secção F).

Ao receber esta informação o comboio vai calcular a força de travagem necessária para parar após percorrer d_1 .

No instante de tempo t_2 o comboio recebe informação actualizada sobre o bloco a si associado, anulando a curva de travagem entretanto calculada e transferindo a paragem prevista para d_1' (Figura 3.8).

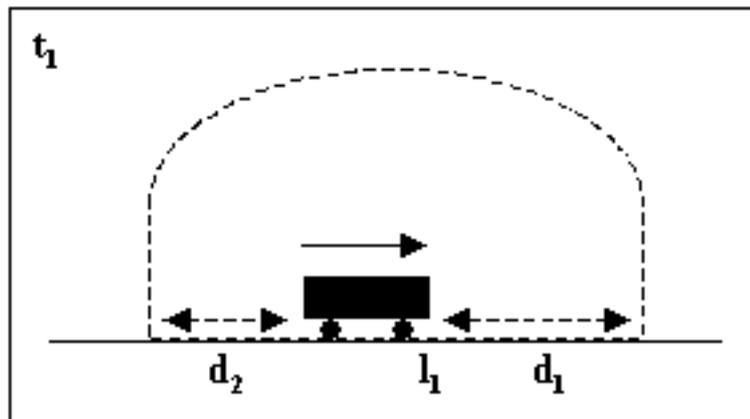


Figura 3.7: Definição da área exclusiva de um comboio (MAL).

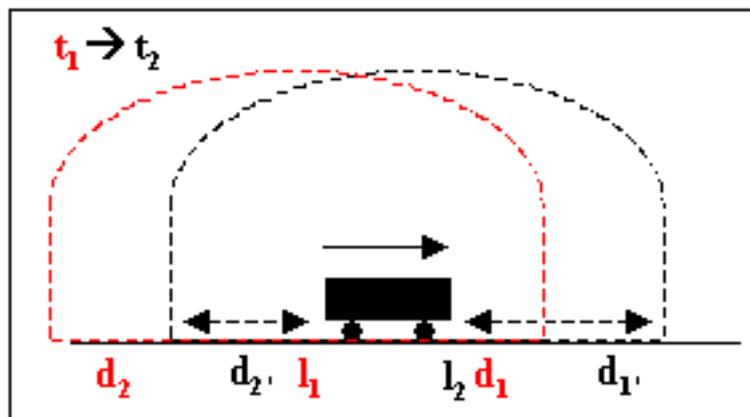


Figura 3.8: Atualização da área exclusiva de um comboio.

Para ordenar a paragem de um comboio bastará num dado momento fornecer o local de paragem como valor limite para o bloco e simplesmente deixar de actualizar este valor.

Constata-se que é um sistema baseado no método *Distance-To-Go* que o desenvolvimento das tecnologias de comunicação permitiu implementar.

Cada comboio transmite continuamente, através de rádio ou fibra óptica, a sua localização, velocidade e direcção a um computador. Aí, essa informação é processada e devolvidos os valores limite para o bloco associado ao comboio. Esta foi a génese dos sistemas de controlo baseados em comunicações (Secção 3.2).

3.1.6 Automatic Train Operation - ATO

Se o sistema ATP é o que evita as colisões entre comboios, este serve para administrar de forma automática as chegadas e partidas das estações.

O objectivo é informar o comboio que se aproxima de uma estação sobre o local, plataforma

de paragem e linha onde deve parar e, dentro destes, o exacto local onde esta deve ser feita.

Monitoriza a imobilização do comboio dentro da plataforma e garante que nenhum passageiro sai do comboio enquanto esta não tiver acontecido. Simultaneamente determina qual o lado do comboio em que devem ser abertas as portas, consoante a localização relativa à plataforma de paragem.

Após o tempo de paragem predeterminado, verifica se todas as portas estão devidamente fechadas e permite a saída do comboio da estação.

3.1.7 Automatic Train Supervision - ATS

Um sistema ATS encarrega-se, num sistema de blocos móveis, de supervisionar a localização de cada comboio na rede ferroviária, essencialmente através da comparação com a tabela prevista (*Timetable* - Apêndice F). Recebe informação do sistema de localização dos comboios e pode interferir na forma de controlo efectuada a partir do sistema ATP, hierarquizando a prioridade atribuída a cada comboio.

3.1.8 Automatic Train Control - ATC

O sistema ATC engloba os três anteriormente descritos, ATS, ATO e ATP. É um sistema que elimina a necessidade de intervenção humana. Normalmente é composto por várias unidades de ATP que comunicam directamente com um nó central de controlo. A informação é depois direccionada para os sistemas auxiliares ATS e ATO, respectivamente para monitorizar a localização dos comboios e as operações de entrada e saída em estações.

Actualmente existe um vasto conjunto de empresas a disponibilizar sistemas de controlo e sinalização ATC. Multinacionais como a *Siemens*, *Alstom* ou *Alcatel* possuem departamentos específicos de produção e investigação de sistemas específicos para o efeito (Figura 3.9).

A figura 3.10 (Cova 2002) mostra o esquema de funcionamento de um sistema completo de controlo automático. Os dispositivos de operação automática (ATO) estão localizados nas imediações das estações enquanto os de protecção (ATP) estão regularmente distribuídos pela rede. Cada um destes envia informação para um nó central (ATS), onde esta é processada. As ordens de controlo são devolvidas e reencaminhadas para a cabine de cada comboio.

Como forma de garantir segurança ao sistema implementam-se sistemas redundantes através da colocação de mais que um nó supervisão (ATS).



Figura 3.9: Sistema de controlo automático ATC (ALSTOM).

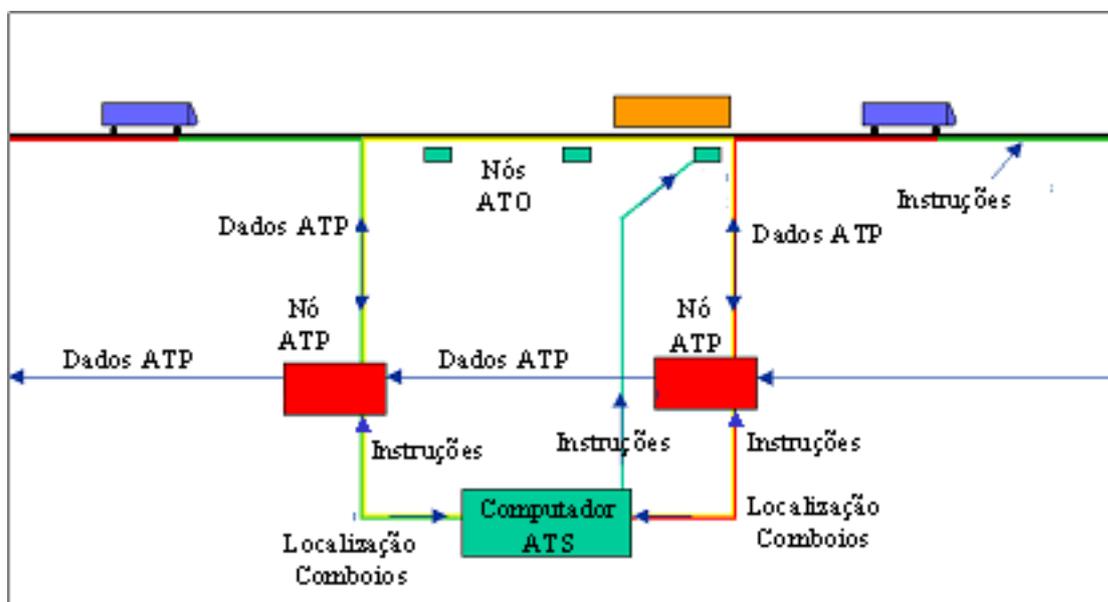


Figura 3.10: Esquema de um sistema de controlo automático (ATC).

3.2 *CBTC-Communications Based Train Control*

Os sistemas *CBTC* - "*Communications Based Train Control*" (Sullivan 2003) são sistemas automáticos de controlo de tráfego ferroviário baseados na localização dos comboios independentes dos tradicionais sensores de linha (*track-circuits*) e na implementação de canais de comunicação bi-direccionais e contínuos entre os comboios e o sistema de controlo (2 1996).

Comparados com os sistemas tradicionais, têm menores custos de instalação e de manutenção, maior capacidade, menores tempos de intervalo entre composições (*headways*), maior segurança e controlo das operações ferroviárias, justificando desta forma a sua crescente utilização em redes ferroviárias metropolitanas, sub-urbanas, regionais e nacionais.

Permitem a circulação de mais comboios no sistema, reduzem a necessidade de investimentos nas infra-estruturas, e tornam o sistema mais determinístico, aumentando a fiabilidade da informação disponibilizada aos utilizadores do sistema (Administration 2003).

3.2.1 Limitações dos Sistemas Tradicionais

Nos sistemas tradicionais de controlo ferroviário a localização dos comboios apenas pode ser determinada no momento da passagem pelos sensores localizados nas linhas. A implementação destes circuitos, além de reduzir a velocidade permitida para cada comboio e aumentar os custos de instalação, resulta num método ineficiente de localização.

Os sinais de controlo da rede, limitados pela existência de canais de comunicação unidireccionais, resumem-se à sinalização na pista ou à implementação de uma reduzida gama de comandos enviados directamente para a cabine de cada comboio (*On-Cab signalling*).

Cada comboio integrado num sistema *CBTC* define distâncias limite à retaguarda e vanguarda para a aproximação de quaisquer outras composições. Estas distâncias, calculadas dinamicamente, formam um véu protector que minimiza a probabilidade de existência de choques.

Este espaço exclusivo de cada composição é tecnicamente designado por "*Movement Authority Limit (MAL)*" e é calculado baseado na hipotética (mas fisicamente impossível) paragem instantânea do comboio que segue imediatamente à frente e na distância segura de travagem do próprio. Por sua vez esta distância é função da velocidade, inclinação, lotação e capacidade de travagem de cada composição.

3.2.2 Funções Básicas

Seguem-se as funcionalidades básicas associadas a um sistema de controlo *CBTC*:

Localização, velocidade e direcção de cada comboio. Um sistema *CBTC* deve ser capaz

de determinar de forma automática (sem qualquer intervenção humana) em cada instante a posição das extremidades de cada comboio com erros mínimos (inferiores a 10 pés) inferindo automaticamente a velocidade e direcção das composições.

Manutenção da Distância Segura entre Comboios. Deve ser continuamente verificada a violação do espaço exclusivo momentâneo de cada composição (MAL), e recalcular este espaço em tempo real com base nas condições verificadas.

Protecção contra Excesso de Velocidade. A velocidade permitida é determinada como a que permite a paragem do comboio antes de interceptar a MAL de qualquer outra composição. Factores como falhas temporárias do sistema ou condições climatéricas poderão condicionar a velocidade permitida para cada comboio.

Detecção de Paragem. O sistema deve ser capaz de determinar a permanência do comboio durante pelo menos dois segundos a velocidades superiores a duas milhas por hora. Desta forma garante-se que qualquer movimento, mesmo que mínimo, será detectado e controlado.

Bloqueio de Portas e de Partidas. Deve ser possível prevenir a abertura de qualquer porta de um comboio quando alguma das seguintes condições não for satisfeita:

- O comboio a velocidade zero.
- A porta está numa zona onde a abertura de portas é permitida, ou seja, sobre uma plataforma de paragem.

O bloqueio de partidas funciona no sentido contrário, isto é, deve ser automaticamente negada a partida de qualquer composição cujas portas não se encontrem todas completamente fechadas.

Interlocking O sistema deve disponibilizar mecanismos que previnam colisões entre comboios equivalentes ao mecanismos de reserva exclusiva de trajecto. Deve também ser garantido que o MAL de um comboio não abrange nenhuma agulha de selecção de trajecto excepto nos casos em que essa rota já esteja reservada para o próprio comboio.

Protecção de final de linha O MAL atribuído a um comboio nunca se deve prolongar para além do final de um troço por forma a permitir a travagem de forma segura.

Protecção contra carruagens soltas Um sistema *CBTC* deve ser capaz de detectar a desagregação de um comboio, accionando nestes casos a travagem de emergência ao próprio e a outros comboios.

Zonas de Trabalho Deve ser permitida a criação de zonas onde a circulação é temporariamente restrita ou mesmo interdita para determinado tipo de composições.

Colocação nas Plataformas Devem ser disponibilizadas funções de colocação nas plataformas (*ATO - Automatic Train Operation*), cabendo ao sistema a decisão acerca do melhor local para imobilizar o comboio dentro de uma plataforma.

Travagem de Emergência O sistema *CBTC* deve ser capaz de accionar os mecanismos de travagem de emergência. Esta travagem, designada tecnicamente por "*Penalty Stop*" e resulta na aplicação de toda a força de travagem ao comboio, não podendo esta ordem ser anulada sem que entretanto o comboio tenha atingido velocidade zero, ou seja, tenha parado.

Identificação dos Comboios O sistema deve atribuir uma identificação exclusiva para cada comboio, e que deve ser utilizada por todos os componentes para comunicar com ele.

Escalonamento de Paragens Cabe ao sistema *CBTC* a monitorização do tempo de paragem dos comboios em cada estação, podendo ser dadas ordens de paragem ou manutenção imprevista numa estação, ou ordem para ignorar a próxima paragem prevista.

Sistemas de Informação aos Passageiros É uma das componentes mais ambiciosas dos sistemas *CBTC*. Aproveitando o manancial de informação recebida e emanada do sistema, pode ser disponibilizada informação específica para cada composição. Esta pode ser função da hora do estado da rede ou de factores imprevistos, podendo ser enviada para os comboios, estações ou painéis de informação nas linhas.

Abaixo resumem-se as características e vantagens dos sistemas *CBTC* relativamente aos tradicionais.

- Todo o equipamento físico pode estar localizado nas estações reduzindo as operações nos túneis ao mínimo.
- Aumento da capacidade através da imposição de menores distâncias de espaçamento entre composições sem comprometer a segurança.
- Flexibilidade. Ao ser um sistema baseado em *software* torna-se mais fácil a alteração de limites ou restrições, de valores para o espaçamento entre composições.
- Redundância. O sistema, apesar de centralizado, é implementado de forma redundante pelo que cada composição recebe em cada momento duas ordens de controlo. Em caso de falha ou discrepância nestas deve ser sempre obedecida a ordem mais restrita.
- Controlo automático e contínuo dos comboios.
- Processo de localização dos comboios independentes dos sensores de linha.
- Comunicação contínua e bi-direccional entre os comboios e o sistema de controlo.
- Menores custos de instalação e de manutenção, derivado da necessidade de menos equipamento.
- Monitorização e controlo das operações em redes ferroviárias facilitada.
- Desenvolvimento de um standard pelo IEEE-"*Institute of Electrical and Electronic Engineers*". Um standard aceite pelos diferentes fabricantes facilita a manutenção ou integração de sistemas híbridos.

3.2.3 Implementação

Requisitos de Comunicação. Normalmente é implementado um sistema de comunicações baseado em fibra óptica por toda a extensão da linha. Em circunstâncias pontuais utilizam-se frequências de rádio de espectro superior a 2.4GHZ(Centonlanzi 2002).

Localização dos Comboios. A localização de cada comboio é feita através da combinação de sensores localizados nas pistas, tacómetros nas rodas dos comboios, acelerómetros e cálculos (aproximações) feitas pelos controladores. Normalmente os sensores são colocados em intervalos que podem variar entre os 300 e o 1000 metros de linha.

Detecção de Velocidade. A velocidade é calculada através das rotações das rodas dos comboios, existindo simultaneamente mecanismos fora do comboio que validam o funcionamento destes velocímetros.

3.2.3.1 Informação

Em 1999 existiam já diversas redes ferroviárias que implementavam ou perspectivavam a implementação de um sistema CBTC (*San Francisco's Bay Area Rapid Transit, MTA NYCT, Philadelphia's SEPTA, Long Island Rail Road,...*). Foi considerado importante manter, no mínimo, nestes novos sistemas o grau de interoperabilidade existente nos sistemas anteriores. O "*Transit Cooperative Research Program*", suportado pela oficial "*Federal Transit Administration*" dos Estados Unidos da América, criou um grupo de trabalho com o objectivo de promover um consenso entre os vários fabricantes e elaborar um standard relativo aos requisitos de funcionamento e, principalmente, aos fluxos de informação num sistema deste tipo.

Este standard deveria seguir os procedimentos da "*American National Standards Institute (ANSI)*" sob a égide do "*Institute of Electrical and Electronic Engineers (IEEE)*"

Apesar de alguma renitência inicial dos fabricantes em partilhar informação foi possível alcançar uma solução de compromisso e elaborar o futuro "*IEEE Standard 1474.1-1999*".

Este documento regulamenta o tipo e frequência da circulação da informação num sistema CBTC. É definida a periodicidade de comunicação entre o sistema de controlo e cada comboio, os itens de informação essenciais (posicionamento de cada um, velocidade, distância relativa pontos relevantes, etc...). Esta informação está especificada no apêndice D.

3.2.4 Utilização

Os sistemas CBTC estão em franca expansão, existindo um considerável número de empresas que os implementa e que investiga neste domínio. Neste particular pensa-se que a implementação completa do sistema no metropolitano de Nova Iorque, previsto para o final de 2004, leve a que bastantes mais cidades possam seguir o exemplo, aumentando a representação que este tipo de sistemas já possui presentemente.

Recentemente (Novembro de 2001), numa iniciativa apoiada pelo *Federal Transit Administration, E.U.A.* foi criada uma comissão com o objectivo de recomendar standards para a industria de transporte ferroviário de passageiros usando procedimentos consensuais do *American National Standards Institute(ANSI)* sob a égide do *Institute of Electrical and Electronic Engineers* (of *Electrical Engineers* 1999). Esta comissão depois de estudar as tecnologias actualmente em funcionamento e em investigação no domínio dos caminhos de ferro apontou os sistemas CBTC como o standard de maior interesse e para onde devem ser canalizados mais recursos (Program 2001).

Refere-se no relatório final que:

”O impacto desta tecnologia será imediato. A comparação entre custos e benefícios deverá encurtar o tempo de introdução e aceitação desta tecnologia que pode melhorar as operações de tráfego e simultaneamente reduzir os custos de investimento e operação” (Board 1999).

Perspectiva-se que este facto aumente a quantidade de recursos disponibilizados para a investigação nesta área e potencie a sua difusão.

Existem várias empresas com sistemas em funcionamento e com investigação reconhecida neste domínio. Algumas das mais relevantes são¹:

General Railway Signal (Alstom). É uma empresa fornecedora de produtos, serviços e sistemas para linhas urbanas e regionais. A divisão de transportes emprega mais que 25000.

Endereço WEB:<http://www.transport.alstom.com/>.

Siemens. Uma das empresas líderes no fornecimento de sistemas para a indústria ferroviária. Cobre vários domínios desde sistemas de controlo até fornecimento de energia de tracção.

Endereço WEB: <http://www.siemens.com/ts/>.

Alcatel. Opera nos Estados Unidos, Canadá e Europa (Portugal incluído) e produz um dos sistemas CBTC mais avançados: o Seltrac.

Endereço WEB: <http://www.alcatel.com/tas/>.

Bombardier. Desenvolve e comercializa sistemas de controlo e planeamento de tráfego ferroviário.

Endereço WEB: <http://transport.bombardier.com/>.

Westinghouse. Com mais de 100 anos de experiência no mercado dos transportes ferroviários fornece sistemas integrados de protecção, supervisão e operação automática.

Endereço WEB: <http://www.westinghouserail.co.uk/>.

Invensys Automation Systems. Desenha, produz e instala equipamento de sinalização e controlo ferroviário entre os quais sistemas de protecção e supervisão automática de

¹Informação recolhida nos endereços WEB de cada empresa

comboios.

Endereço WEB: <http://www.invensysrail.com>.

General Electric Transportation. Com mais de 100 anos de experiência acumulados no sector emprega actualmente mais de 2300 empregados distribuídos por 20 países. Implementa também sistemas de controlo automático de comboios.

Endereço WEB: <http://www.getransportation.com/>.

Segue-se uma lista de sistemas baseados em comunicações já implementados e em pleno funcionamento, existindo bastantes mais em fase de construção e de planeamento.

- *San Francisco Municipal Railway (Metro)*
- *Detroit Downtown People Mover System*
- *Vancouver Ski Train*
- *Toronto Scarborough Rapid Transit*
- *London Docklands Light Rail*
- *Paris Metro Meteor Line*
- *Ankara Rapid Transit System*
- *Kuala Lumpur Putra Light Rail*

Conclusão

O desenvolvimento das tecnologias de comunicação e de processamento de informação permitiu a implementação de novos métodos de controlo e sinalização de tráfego ferroviário. Tal como atrás referido, espera-se que o impacto destas novas metodologias possa aumentar a segurança, rapidez e capacidade das redes de transporte ferroviário, razão pela qual estão a ser implementados em diversas regiões do mundo, especialmente as tecnologicamente mais desenvolvidas.

Capítulo 4

Características Gerais do Sistema Proposto

Este capítulo é dedicado à apresentação geral do sistema, à descrição das entidades nele presentes, sua organização e interacção com vista à aquisição do objectivo proposto. Descrevem-se as suas principais características e a sub-divisão em *Controlo* e *Aprendizagem*. Apresentam-se as competências de cada sub-sistema e a interacção existente entre eles.

4.1 Apresentação - Sistema MARCS

O sistema **MARCS** - **Multi Agent Railway Control System** é um sistema multi-agente para controlo de tráfego ferroviário baseado em comunicações.

Tem por objectivo básico o controlo e ordenação automática do tráfego numa rede ferroviária com capacidades de comunicação entre todos os intervenientes.

Simultaneamente pretende explorar as vantagens resultantes da noção de *Agente* com vista à apresentação de uma alternativa fiável para a resolução do problema.

Na sua arquitectura distinguem-se dois sub-sistemas: *Controlo* e *Aprendizagem*.

O sistema de controlo é responsável pela supervisão e regulação do tráfego, tendo como função garantir a segurança e fluidez na rede.

O sistema de aprendizagem deverá, com base em informação proveniente do sistema de controlo, inferir regras que optimizem os processos de controlo, com o objectivo de minimizar o tempo de espera e ordens de paragem enviadas a cada composição.

4.2 Requisitos

O sistema MARCS está condicionado aos seguintes pré-requisitos:

- Existência de uma rede ferroviária com suporte à comunicação bidireccional em toda a sua extensão.
- Existência em todos os comboios de mecanismos de sinalização directa para a cabina e estruturas físicas computacionais apropriadas à instalação do agente que os representa.
- Estações equipadas com sistemas computacionais apropriados para a instalação do respectivo agente representante.
- Rede informática com capacidade de comunicação bi-direccional em toda a sua extensão para a instalação dos agentes supervisores de tráfego.

4.3 Arquitectura

No sistema MARCS interagem quatro tipos de agentes:

- Supervisor
- Estação
- Comboio
- Aprender

A figura 4.1 esquematiza a arquitectura proposta e a interacção estabelecida entre os seus componentes, estando simultaneamente representados os dois sub-sistemas de *Controlo* e *Aprendizagem*.

Tal como descrito na figura, as suas características principais são as seguintes:

- Vários agentes supervisores cooperam com vista à ordenação e controlo do tráfego.
- Cada agente *Supervisor* envia aos agentes representantes dos comboios informação acerca da distância livre para percorrer.
- Os agentes *Comboio* contêm informação sobre a sequência de estações onde devem ser efectuadas paragens, devendo disponibilizá-la aos supervisores.
- Aos agentes *Comboio* cabe também o envio do valor de utilidade aos agentes supervisores. Este valor funciona como heurística na atribuição de prioridade.

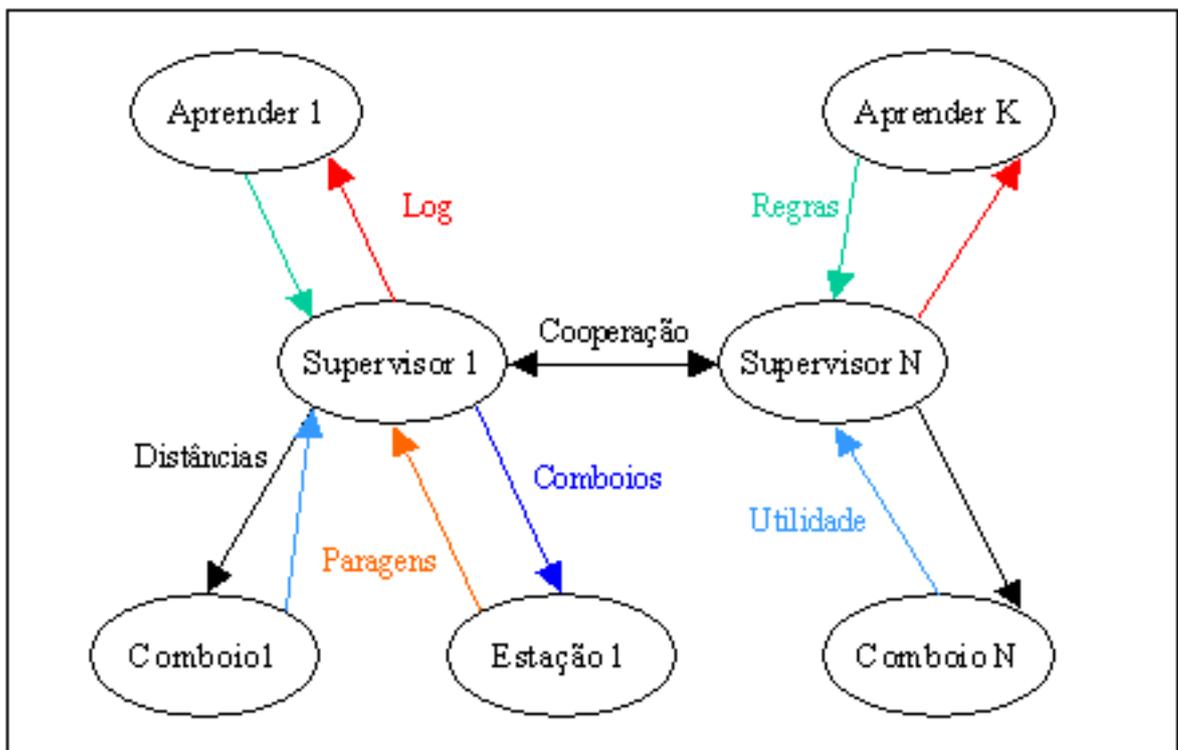


Figura 4.1: Arquitectura geral do sistema MARCS.

- Os supervisores notificam os agentes *Estação* da aproximação de comboios à estação por si representada. Estes devem indicar a plataforma e local exacto onde deve ser efectuada a paragem.
- Cada *Supervisor* transmite aos agentes *Aprender* o conteúdo do seu registo de actividade.
- Depois da análise do respectivo registo de actividade o agente *Aprender* infere regras que podem melhorar a fluidez do tráfego ferroviário. Estas regras são transmitidas aos agentes supervisores.

Foi adicionalmente implementado um agente (*Simulador*) incluído na aplicação de simulação com conhecimento de todos os restantes e com a função de transmitir a informação relativa à simulação do tráfego. No processo de aprendizagem acumula características de facilitador, sendo exclusivamente estas as suas funções na implementação de um sistema real.

4.4 Sistema de Controlo

Os sistemas de controlo CBTC (Secção 3.2) actualmente existentes, apesar de possuírem arquitecturas distribuídas, têm controlo centralizado ou semi-centralizado.

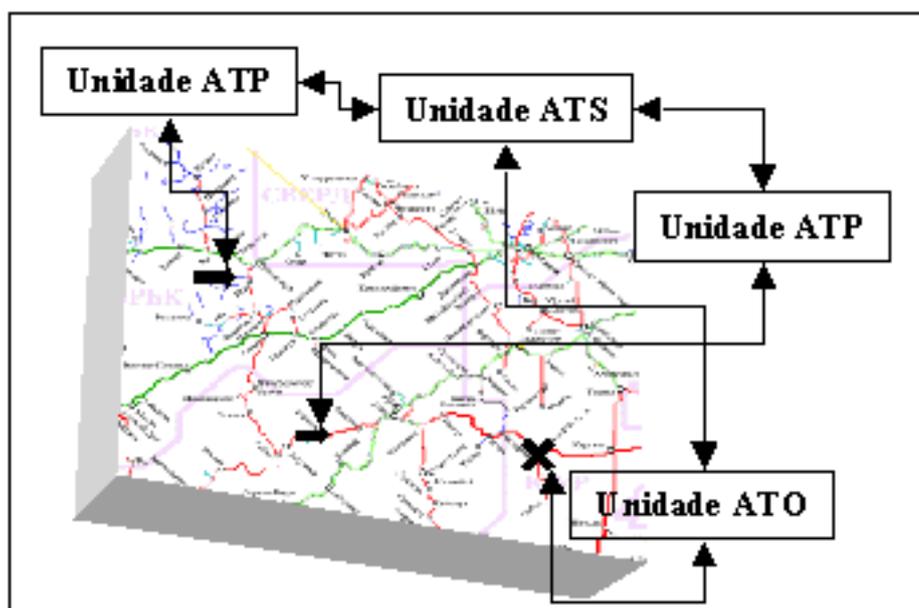


Figura 4.2: Esquema de um sistema com um nó de supervisão.

O sistema que supervisiona a posição de cada comboio na rede e efectua a comparação com a respectiva *timetable* (*Automatic Train Supervision*) é geralmente um nó central (Unidade ATS - Figura 4.2), para onde deve fluir toda a informação e de onde sai a prioridade atribuída a cada composição. Apenas por questões de segurança (redundância) se instala mais que um nó supervisor.

Apesar da eficiência alegada pelos fabricantes será lógico pensar que o aumento do volume de tráfego e da extensão das redes abrangidas traga dificuldades acrescidas de comunicação e exija recursos computacionais com maior capacidade por parte do sistema de controlo.

A abordagem por nós proposta prevê que a supervisão do tráfego seja distribuída por várias entidades (agentes), cada um deles com responsabilidade sobre uma parcela da rede. Cada comboio e estação é também representado pelo respectivo agente.

A ideia consiste em **tornar independentes o volume de informação a processar e a dimensão da rede**. Esta apenas terá correspondência directa no número de supervisores participantes no sistema de controlo.

A figura 4.3 esquematiza a abordagem implementada. Existem agentes responsáveis (Supervisor) por cada parcela da rede ferroviária e agentes que representam os comboios (*Comboio*) e estações (*Estação*) existentes. Cada um possui objectivos próprios que deverá satisfazer, no cumprimento do seu papel no sistema.

Cada *Supervisor* é responsável pela gestão do tráfego na respectiva área, tendo como funções:

1. Garantir a segurança dos trajectos sob sua supervisão.

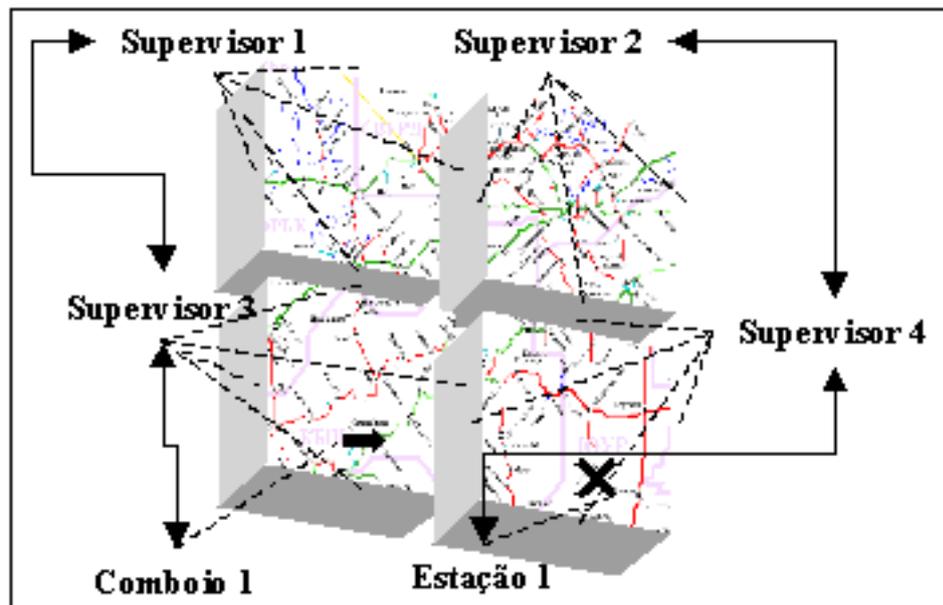


Figura 4.3: Esquema geral do sistema de controlo MARCS.

2. Determinar a rota ideal para cada comboio supervisionado.
3. Accionar os pontos de selecção de rota (*Switch* - Apêndice F) localizados na sua área, de acordo com os dois pontos anteriores.
4. Cooperar (trocar informação e propor planos) com os agentes vizinhos.

Os agentes *Estação* têm como função, subordinados aos supervisores:

1. Administrar as partidas e chegadas dos comboios às plataformas de paragem. Cada *Estação* é responsável pela definição da plataforma onde o comboio deve parar, respectiva posição e tempo previsto de paragem.
2. Providenciar informação para o estacionamento dos comboios que terminam viagens.

A responsabilidade dos agentes *Comboio* consiste em:

1. Seleccionar a ordem de controlo a enviar para o comboio por si representado. Nos troços supervisionados por mais que um agente podem ser enviadas ordens distintas. Neste caso o *Comboio* deve seleccionar a ordem a obedecer pela composição.
2. Calcular a força de travagem a aplicar para que a composição se imobilize antes de atingir o limite definido pelo agente supervisor.
3. Manter um índice de utilidade que servirá como heurística nos processos de atribuição de prioridade.
4. Manter registo da sequência de locais de paragem planeados.

4.5 Sistema de Aprendizagem

Trabalhos de investigação anteriores têm demonstrado que os modelos de controlo de tráfego baseados na comparação entre um arquivo histórico de informação e as condições momentâneas de tráfego tendem a produzir melhores resultados (Hobeika Kim 1994).

Com base nesta conclusão foi implementado, paralelamente ao sistema de controlo, um outro tipo de agente (*Aprender*) que tem como função a análise do registo de actividade dos agentes supervisores e, utilizando técnicas de extracção de conhecimento, inferir regras que optimizem o processo de controlo do tráfego.

A ideia consiste em, depois de recebida informação acerca da actividade (sequência de acções, mensagens recebidas e enviadas) do agente supervisor, analisar essa informação, extrair padrões e inferir possíveis regras. Veja-se um exemplo¹:

```
13323121 localizacaoComboio 10001 Latitude 12.3 Longitude 11 Destino d1
13323121 localizacaoComboio 10002 Latitude 26.3 Longitude 18 Destino d2
13323122 localizacaoComboio 10001 Latitude 10.3 Longitude 11 Destino d1
13323122 localizacaoComboio 10002 Latitude 18.5 Longitude 14 Destino d2
13323123 localizacaoComboio 10001 Latitude 9.3 Longitude 11 Destino d1
13323123 localizacaoComboio 10002 Latitude 12.3 Longitude 11 Destino d2
13323124 Perigo! Conflito entre os comboios 10001 e 10002 no Local L1
```

Com base no alerta lançado pelo supervisor, logo que detectou que dois agentes viajavam em posições demasiado próximas, é possível inferir uma regra que preveja um conflito da próxima vez que ocorrer uma situação de tráfego similar.

```
"
SE
    ComboioX Localizacao (12,11) Destino d1
E
    ComboioY Localizacao (26,18) Destino d2
ENTÃO
    Conflito no local L1 dentro de 3 segundos
"
```

Esta regra é posteriormente transmitida aos agentes supervisores responsáveis pelos troços em questão, que a deverão levar em conta na próxima situação de tráfego similar.

Este método só será viável se não interferir com a segurança do sistema. É necessária garantia que:

- Qualquer regra inferida permanece subordinada aos critérios de segurança previamente definidos.
- O sistema de controlo funcione sem nenhuma regra inferida.

¹Formato simplificado, detalhado na secção 7.3

4.5.1 Algoritmo de Aprendizagem

No processo de aprendizagem foi utilizado o algoritmo APRIORI (Agrawal Strikant 1994). Tem como objectivo a identificação de conjuntos frequentes e descoberta de regras de associação entre itens, sendo normalmente utilizado em problemas do tipo "market-basket".

Neste género de problemas, mediante uma série de transacções (conjunto de itens), pretende-se identificar os elementos que aparecem frequentemente nas mesmas transacções. Posteriormente, é possível identificar regras do tipo:

"Normalmente, quando uma transacção inclui os itens A e B também inclui o item C "

Efectuando um paralelismo entre os itens de uma transacção com o posicionamento dos comboios e os conflitos de tráfego, pretende-se identificar regras como a seguinte:

"Quando existe um comboio localizado no ponto P_1 com destino à estação E_1 e outro no ponto P_2 com destino a E_2 vai ocorrer um conflito de tráfego no local L_1 ".

Com base nestas regras pretende-se que os agentes supervisores de tráfego possam pedir aos comboios para alterarem temporariamente a sua velocidade na tentativa de evitar o conflito.

Conclusão

Através da sub-divisão do sistema MARCS em dois ("Controlo" e "Aprendizagem"), potenciou-se a sua modularidade, facilitando simultaneamente o processo de desenvolvimento. No que respeita ao sistema de controlo, descreveu-se a interacção entre os agentes intervinientes e as principais competências de cada um. Relativamente ao sistema de aprendizagem, apresentaram-se as motivações que o originaram, o algoritmo utilizado no processo (APRIORI) e o conhecimento a inferir deste processo.

Capítulo 5

Linguagens e Metodologias Utilizadas

Este capítulo é dedicado à apresentação das metodologias seguidas na elaboração do sistema MARCS e das linguagens escolhidas para tal. Apresentam-se razões justificativas para as opções tomadas e efectuam-se comparações com possíveis alternativas.

5.1 Modelação

Num projecto com alguma dimensão torna-se imprescindível a prévia análise de requisitos e potenciais problemas, por forma a antecipar a sua resolução.

A primeira fase do trabalho consistiu na análise dos sistemas de controlo existentes e das suas características mais relevantes com vista à geração de um modelo inicial. Para tal foi utilizada a *Unified Modelling Language* (UML).

5.1.1 UML - *Unified Modelling Language*

A *UML* é uma linguagem de modelação de sistemas genérica, usada na especialização, visualização, construção e documentação de sistemas de *software*. Resulta da tentativa de unificação das várias linguagens utilizadas na modelação de sistemas uniformizando (na medida do possível) a prática de implementação de *software* (Rumbaugh, Jacobson Booch 1999).

A *UML* possibilita o registo de informação acerca da estrutura estática e comportamento dinâmico de um sistema, por exemplo através de diagramas de classe ou de sequência respectivamente.

O sistema é modelado como uma colecção de objectos que interagem na implementação do comportamento desejado.

A estrutura estática define os tipos de objectos necessários e respectivas relações. O com-

portamento dinâmico define o ciclo de vida de cada objecto e a interacção necessária para satisfação do seu papel no sistema.

É possível a gestão hierárquica dos modelos através da introdução da noção de *package*, facilitando desta forma a compreensão e implementação de sistemas de maior dimensão.

5.1.1.1 Modelo

Pode-se definir um modelo como a representação de algo numa forma mais perceptível para determinado fim, podendo-se falar em modelos como a representação de figuras geométricas 3-D em papel ou o desenho de edifícios num computador. O modelo de um sistema de *software* pode ser feito através da linguagem UML (Rumbaugh et al. 1999).

Os modelos podem ser usados com várias finalidades:

- Pensar o desenho de um sistema. Um arquitecto usa modelos desenhados no papel ou em aplicações específicas para o efeito. Um modelo de um sistema computacional ajuda os informáticos na idealização da arquitectura de um sistema antes de iniciar a fase de implementação.
- Organizar, pesquisar e filtrar informação em sistemas de grandes dimensões. Tal como a construção de um edifício envolve informação sobre vários aspectos (estrutura, rede eléctrica, ventilação, decoração), também na modelação de um sistema computacional se organiza a informação segundo várias perspectivas (estrutura estática, máquinas de estados, interacções).
- Explorar soluções múltiplas. Existem situações onde várias opções se apresentam como soluções a um problema, mas não é exequível a implementação de cada uma delas para testar a sua real eficiência. Nestes casos a elaboração de modelos pode ajudar a clarificar as vantagens e inconvenientes de cada abordagem idealizada.

5.1.1.2 Características da UML

Standardização A *UML* foi unanimemente adoptada pelo *OMG - Object Management Group* em Novembro de 1997, tendo este assumido a responsabilidade pelo futuro desenvolvimento do standard. A emergência da *UML* parece acarretar benefícios ao domínio da computação ao consolidar experiências de vários autores de diferentes áreas.

Estrutura Estática A definição de qualquer modelo deve ser antecedida da definição do universo a abranger, dos conceitos-chave, propriedades e relações recíprocas. A definição deste conjunto de estruturas origina o Modelo Estático (*Static View*). Os conceitos são modelados por classes, cada uma descrevendo um comportamento e implementando formas de relacionamento com as restantes. A abordagem orientada

a objectos permite a utilização de conceitos como a generalização, definindo comportamento genéricos que abrangem vários tipos de objectos e depois se detalha o comportamento específico de cada um.

Comportamento dinâmico Existem basicamente duas formas de modelar o comportamento: detalhando o ciclo de vida de objecto e a sua interacção com o resto do mundo, ou modelando as formas de comunicação de conjuntos de objectos. A análise do comportamento de um objecto isolado dá origem a uma máquina de estados, enquanto a modelação de um conjunto de objectos e respectiva interacção dá origem a diagramas de colaboração. Este tipo de diagramas especifica o fluxo de mensagens entre os vários objectos pertencentes a um sistema.

Implementação física Os modelos *UML* podem também representar a componente física de um sistema, introduzindo a noção de *Nodo*. Um nodo é um recurso computacional existente num determinado espaço e que executa um conjunto de procedimentos, através dos componentes e objectos em si contidos. Os diagramas de componentes descrevem a configuração física de um sistema e a relação entre nodos, podendo incluir possíveis migrações ou comutações de nodos.

Organização do modelo Os computadores lidam bem melhor que os humanos com modelos de elevada dimensão e com relações bastante complexas. Neste tipo de sistemas surge a necessidade de introdução da noção de *Package*. Uma *package* é uma estrutura lógica que agrupa um conjunto de objectos e é responsável pela implementação de um determinado comportamento.

Mecanismos de extensibilidade Independentemente do detalhe e exactidão da análise e implementação de um modelo, por regra serão sempre necessárias extensões ao modelo, quer seja pela alteração ou adição de requisitos. A *UML* fornece a noção de *Estereotipo*. Consiste num elemento com a mesma estrutura que outro existente no sistema mas com possíveis restrições adicionais. Uma restrição é um conjunto de caracteres que pode ser anexado a um modelo por forma a guardar informação arbitrária. A introdução de restrições pode, em certos casos, evitar a necessidade de proceder a alterações mais profundas nos modelos ou no sistema a eles associado.

5.1.1.3 Diagramas UML

A *UML* fornece diferentes perspectivas de análise consoante o tipo de sistema a modelar. Sistemas de controlo baseados na alteração dinâmica de estados terão formas diferentes de análise de sistemas essencialmente organizacionais, mais centralizados na definição da estrutura.

A tabela 5.1.1.3 mostra as diferentes perspectivas pelas quais se podem modelar sistemas.

Perspectiva	Diagramas	Conceitos Principais
Estática	Diagrama de classe	Classe, Associação, Generalização, Dependência, Realização, Interface
Casos	Diagrama de casos	Caso, Actor, Extensão, Inclusão
Implementação	Diagrama de componentes	Componente, Interface Dependência, Realização
Estados	Diagrama de estados	Estado, Evento, Transição, Acção
Actividade	Diagrama de actividade	Estado, Actividade, Junção, Separação
Interacção	Diagrama de Sequência	Interacção, Mensagem, Activação

Tabela 5.1: Diagramas e perspectivas UML.

5.1.1.4 Diagramas Utilizados

Pelas características do sistema, a modelação dinâmica é bastante mais complexa que a estrutural justificando a preferência por diagramas de sequência na modelação dos processos e da interacção entre agentes integrantes do sistema.

Um diagrama de sequência mostra a interacção entre as partes numa linha temporal. Neste caso, a execução de cada agente é representada por uma barra vertical e as mensagens enviadas e recebidas através de setas a outros agentes.

A figura 5.1 exemplifica um diagrama de sequência para a autenticação de um utilizador numa caixa electrónica. Mostra as entidades intervenientes no processo e a forma como interagem na execução da tarefa planeada.

Na autenticação e apresentação do menu ao utilizador intervêm três entidades: o quiosque, a central e o serviço de cartões.

A ordem pela qual eles interagem na sequência de mensagens é a seguinte:

1. O quiosque notifica a central do número de cartão inserido.
2. A central indica ao quiosque para apresentar o menu onde o código é inserido.
3. O quiosque envia informação com o código introduzido pelo utilizador.
4. Após receber o número do cartão e o código, a central envia esta informação ao serviço de segurança onde é efectuada a validação.
5. É confirmada a validade da informação recebida.

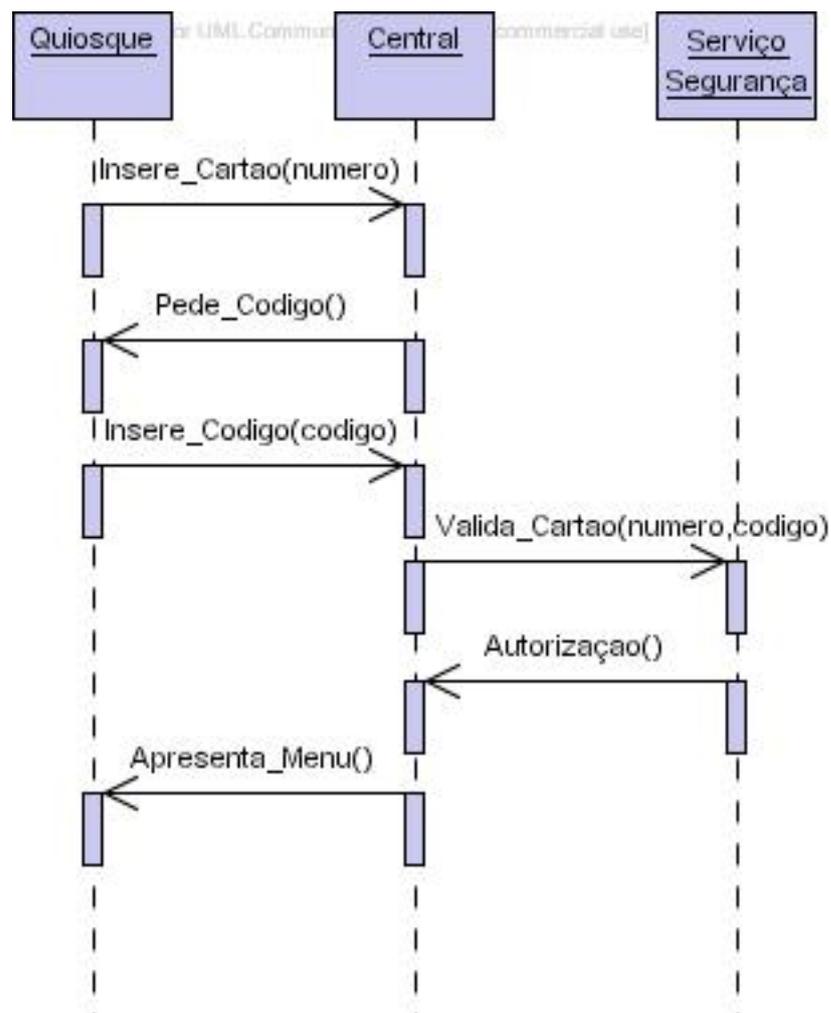


Figura 5.1: Exemplo de um diagrama de sequência.

6. A central comunica ao quiosque que pode apresentar ao utilizador o menu de operações.

5.2 Representação da Informação

Para registar a informação necessária a cada interveniente do sistema foi utilizada a linguagem XML. A sua crescente relevância e utilização em sistemas informáticos, portabilidade e escalabilidade foram factores determinantes na escolha.

5.2.1 XML - *Extensible Markup Language*

A *Extensible Markup Language* - XML é um standard do *World Wide Web Consortium* (W3C) para a estruturação de informação. Pode-se definir como uma sintaxe genérica para uma estruturação simples e facilmente perceptível pelo ser humano através da definição de um formato standard para documentos digitais (Harold Means 2001).

Este formato é flexível o suficiente para poder ser utilizado em domínios bastante diversos como a implementação de sítios WEB, transferência de informação, serialização de objectos ou invocação remota de procedimentos.

5.2.1.1 Características

É uma meta-linguagem para estruturação de documentos de texto. A informação incluída em documentos XML consiste em cadeias de caracteres (*strings*) inseridas em elementos que definem a estrutura do documento (*tags* ou marcas). Uma marca é uma sequência de texto delimitada pelos sinais "<" e ">" respectivamente à esquerda e direita.

A unidade mínima de informação designa-se por **elemento**. A especificação da linguagem XML define a forma como os elementos devem ser delimitados por marcas, a forma de uma marca e os nomes que podem ser atribuídos a elementos.

O aspecto mais importante reside no facto de, tal como referido anteriormente, ser uma meta-linguagem, o que significa que não contém um conjunto fixo de marcas e elementos supostamente adequados a qualquer domínio. O utilizador tem a possibilidade de definir a linguagem através da introdução dos elementos que achar conveniente.

A única restrição imposta pela XML reside na estrutura do documento, isto é, o número de marcas possíveis e a ordem pela qual elas devem aparecer. Este facto, que poderá parecer redutor, permite a implementação e disponibilização de avaliadores genéricos, capazes de ler qualquer documento XML e concluir acerca da sua boa-formação, ou seja, do respeito pelas regras anteriormente enunciadas. Esta funcionalidade é incorporada pelas versões mais recentes dos *browsers* mais comuns (*Microsoft Internet Explorer* e *Netscape Navigator*).

5.2.1.2 DTD - *Document Type Definition*

A estrutura que um documento XML deve seguir, o conjunto de marcas possíveis e a ordem pela qual podem aparecer, é definida num *DTD*. Veja-se um exemplo:

```
<?xml version="1.0"?>
<!DOCTYPE exemplo SYSTEM "exemplo.dtd" [
<!ELEMENT pessoa (nome, profissao*)>
<!ATTLIST pessoa bi ID #REQUIRED>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT profissao (#PCDATA)>
```

] >

Está acima definida a estrutura que um documento XML destinado a guardar informação sobre uma pessoa deve respeitar. Explicita que:

- O elemento "pessoa" é composto por um elemento obrigatório ("nome"), zero ou mais elementos "profissao" e um atributo.
- O atributo "bi", é identificador da pessoa e obrigatório no documento.
- Os elementos "nome" e "profissao" consistem em sequências de caracteres.

Podem ser criados documentos XML que respeitem o DTD acima especificado, por exemplo:

```
<?xml version="1.0"?>
<PESSOA bi=12345678>
<NOME>José Alberto</NOME>
<PROFISSAO>Arquitecto</PROFISSAO>
</PESSOA>
```

Este documento, guardado como ficheiro de texto, pode ser alterado através de qualquer editor, não existindo preocupações relativamente à capitulação das letras, ao número de espaços ou quebras de linha entre marcas.

A crescente utilização da XML para os mais diversos domínios conduziu ao aparecimento de outras linguagens de suporte à sua utilização:

XSL É uma aplicação da XML que permite que documentos XML sejam visualizados em *browsers*.

XLL Relaciona os documentos XML num ambiente de rede, através da marca HTML "<A>".

XHTML Uma reformulação do HTML como aplicação XML bem-formada.

XML Query Language Linguagem de pesquisa de elementos num documento XML.

XML Signatures Meio standard de assinar digitalmente documentos XML, embebendo assinaturas e autenticando documentos assinados.

5.2.1.3 Vantagens

Depois de focadas as características mais relevantes da linguagem XML importa justificar a sua utilização neste domínio.

A XML foi utilizada como forma de registo da informação relativa aos mapas ferroviários e às definições de configuração dos agentes (Apêndice E), essencialmente pelas seguintes razões:

Formato facilmente perceptível Este foi o aspecto decisivo. A transformação da XML num standard da estruturação digital de informação leva a que qualquer pessoa minimamente conhecedora da linguagem possa perceber o conteúdo de um ficheiro XML. Deixa de ser necessário o conhecimento minucioso do formato dos ficheiros que guardam informação, bastando a consulta do respectivo DTD para proceder à alteração ou inserção de informação.

Facilidade de alteração da estrutura O formato XML permite que alterações na estrutura de um documento possam ser facilmente feitas no respectivo DTD e actualizadas posteriormente nos documentos que o devam respeitar.

Capacidade de visualização por *browsers* Durante o processo de implementação e depuramento, foi várias vezes necessário consultar a informação relativa aos mapas. Qualquer versão moderna de um *browser* verifica a integridade do ficheiro e a sua boa-formação segundo os requisitos XML, tal como mostrado na figura 5.2.

Facilidade de alteração do documento Ao guardar a informação em formato de texto, torna-se bastante mais fácil a inserção, alteração ou remoção de componentes do ficheiro. Para tal basta usar um qualquer editor de texto.

5.3 Implementação

Considerando a experiência anterior em linguagens de programação e mediante a análise dos requisitos, exigências e especificidades do sistema, duas opções se apresentaram: Java ou C (C++).

A linguagem Java tem sido cada vez mais utilizada, tanto no âmbito académico como empresarial, e apresenta características que a distinguem das demais: É compatível com diferentes sistemas operativos, tem API's extensas e bem documentadas, gere a memória de forma automática e permite a programação orientada a objectos.

A linguagem C (C++) além de permitir a programação orientada a objectos, constitui a base para grande parte das aplicações e sistemas operativos existentes. Constitui um standard para a programação estruturada e permite operações de baixo nível (operações com *bits* e ponteiros para estruturas), o que pode resultar em aplicações mais eficientes.

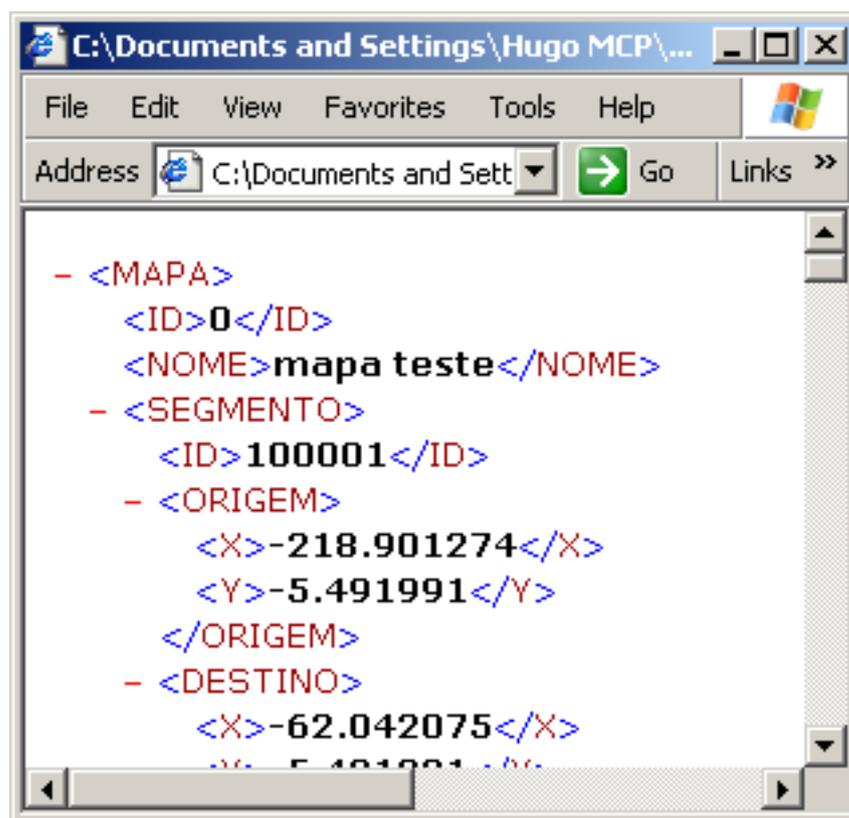


Figura 5.2: Visualização de um ficheiro XML num *browser* WEB.

5.3.1 Java versus C

Em 1995, o aparecimento da linguagem Java causou grande impacto na comunidade científica e empresarial, tendo sido apresentada como uma linguagem puramente orientada a objectos, compatível para todas as plataformas, fácil de integrar na internet e de rápida aprendizagem.

Esperava-se que representasse um avanço significativo, especulando-se mesmo na sua provável evolução para sistema operativo e o desafio que representaria para o próprio sistema *Windows*. O passar do tempo ajudou a clarificar as vantagens e desvantagens da linguagem Java relativamente a outras tendo em conta domínios de aplicações específicos.

A linguagem Java tem bastantes vantagens. Ao ser puramente orientada a objectos, propicia a reutilização do código. Tem uma sintaxe muito parecida ao C (C++), o que desde logo facilita a migração entre linguagens. Tem um sistema automático de gestão de memória, e a sua API incorpora estruturas de dados e algoritmos para a rápida produção de interfaces gráficos. A portabilidade é também uma vantagem relevante.

Quando um programa Java é *compilado*, não é traduzido para código nativo da máquina onde foi feita a compilação, sendo antes produzido um *bytecode* que poderá ser interpretado por qualquer máquina virtual de java (específica para cada sistema). Esta característica, que constitui a essência da sua portabilidade, é simultaneamente uma das suas maiores desvantagens.

Apesar do Java produzir código neutro, portátil entre arquitecturas, o método que utiliza para tal é ineficiente. Tal como foi referido anteriormente, após o código fonte ser *compilado* para *bytecode* vai ser interpretado por uma máquina virtual de Java específica para cada sistema, que o transforma em instruções nativas da máquina. Ao contrário do código nativo, sequência de instruções específicas para processador, o interpretador Java efectua primeiro a tradução para a correspondente instrução nativa e só depois efectua a execução.

É esta a razão por que é mais correcta a utilização do termo *interpretação* relativamente à *compilação* no caso de programas Java. Obviamente esta tradução consome tempo e recursos, induzindo à menor eficiência da linguagem Java ("Java, being a interpreted system, is currently an order of magnitude slower than C" (von der Linden 2002)).

Podem-se acrescentar outras razões que justificam a menor eficiência da linguagem Java relativamente ao C. O Java é optimizado em tempo de execução, o C em tempo de compilação. O acesso a vectores é verificado em tempo de execução, o que garante a inexistência de acessos indevidos à memória, mas tem custos em termos de performance. A livraria standard de strings é de 16-bits Unicode, em vez de 8 bits ASCII, o que quer dizer que todas as strings ocupam o dobro do espaço relativamente ao C. A gestão de memória é feita automaticamente, e ainda não existem CPU's optimizadas para Java.

Todas estas características facilitam a tarefa do programador mas têm obviamente custos na eficiência da aplicação resultante. As aplicações de simulação gráfica são normalmente de grande exigência computacional, ao que acresce o facto desta ter características de tempo

real, pelo que a eficiência foi o factor primordial na escolha da linguagem.

Os resultados dos testes de eficiência (*benchmarks*) da linguagem Java mostram resultados entre 0,3 a 3 vezes mais lento que o respectivo programa em C (Y.F.Hu, R.J.Allan K.C.F.Maguire 2000).

Refira-se, no entanto, a considerável melhoria que a linguagem Java apresentou em termos de desempenho, existindo já situações e casos (ainda) bastante específicos onde uma aplicação Java é mais eficiente que a respectiva em C (J.P.Lewis 2001).

O aparecimento dos *JIT-Just-In-Time Compilers* aumentou significativamente a eficiência da linguagem (à custa da portabilidade), na medida em que traduzem o *bytecode* (igual para todas as plataformas e interpretado pela máquina virtual de java específica de cada uma) para instruções nativas de um dado sistema em particular.

O objectivo de um *JIT* é converter o *bytecode* para instruções nativas, sendo requisito que esta conversão seja efectuada o mais rapidamente possível consumindo pequenas quantidades de memória. A satisfação destes requisitos leva a que ainda não sejam feitas as optimizações efectuadas pelos compiladores C e C++. Apesar do considerável aumento de eficiência, ainda fica aquém do C na maioria dos casos (Coha 2000).

A tabela 5.3.1 compara tempos de execução obtidos por aplicações C++ e Java na implementação de tarefas de leitura/escrita em consola, em ficheiro, inserção em listas, e iteração em ciclos. A forma como estes valores foram obtidos e detalhes acerca do código e testes implementados pode ser consultada em (Connell 2002).

Tarefa	Java (s)	C++ (s)
I/O Consola	33.58	3.6
I/O Ficheiro	3.68	1.04
Listas	2.71	0.19
Speed	1.18	0.18

Tabela 5.2: Comparação entre tempos de execução.
(Connell 2002)

Outra das razões que me levou a optar pela implementação em linguagem C foi a biblioteca *OpenGL*. A versão original foi implementada em linguagem C, sendo previsível que seja mais estável que a respectiva versão Java. Ao utilizar a *OpenGL* achei razoável usar também a linguagem que há mais tempo a incorpora. Esta biblioteca utiliza essencialmente operações com matrizes para implementar as transformações sobre os componentes gráficos.

A tabela 5.3.1 efectua a comparação entre tempos de execução de programas Java e C++ nas operações com matrizes (adição e multiplicação de matrizes de inteiros e reais de dimensão variável).

Tarefa	C++ (s)	Java (s)
Adição(Reais)	0.3	2.3
Adição (Inteiros)	0.292	2.143
Multiplicação (Reais)	0.35	2.57
Multiplicação (Inteiros)	0.343	2.534

Tabela 5.3: Comparação entre tempos de execução (Matrizes).
(Galyon 2002)

5.3.2 Linguagem C (C++)

5.3.2.1 *Standard Template Library (STL)*

Uma das limitações históricas da linguagem C++ consistiu na ausência de estruturas genéricas contentoras de objectos, levando a que cada programador que necessitasse de estruturas deste tipo tivesse que as implementar de raiz.

Recentemente ¹ um grupo criado sob a égide do ANSI propôs a noção de *template* que veio resolver este problema. Agregada à noção de classe, um *template* é uma implementação genérica de funções e classes que poderão ser utilizadas em diferentes aplicações.

A STL disponibiliza uma biblioteca com implementações genéricas das estruturas mais comuns - listas ligadas, vectores, filas, pilhas, conjuntos, grafos - e com algoritmos que vão operar sobre elas (Wise 1995).

Por ser comumente utilizada e testada, o programador, além de poder dedicar mais tempo a resolver os problemas específicos da aplicação, tem garantias de eficiência e robustez nas estruturas disponibilizadas pelas bibliotecas.

5.3.2.2 *Vectores Dinâmicos*

As estruturas a que será necessário aceder mais frequente e rapidamente estão guardadas em vectores dinâmicos.

Por serem de mais baixo nível, apesar de mais trabalhoso para o programador, tornam o manuseamento da informação mais eficiente.

Paralelamente à estrutura (vector) que guarda a informação, é necessário criar uma variável que guarde o seu tamanho em cada instante. Cada vez que se inserem ou eliminam elementos do vector tem que se fazer a correspondente actualização da variável.

Veja-se um exemplo relativo à declaração de um vector dinâmico:

```
TIPO *variavel=NULL;
int totalElementos=0;
```

¹Primeira proposta em 1990

A inserção de um novo elemento é feita da seguinte forma:

```
variavel=(TIPO*)realloc(variavel,++totalElementos*sizeof(TIPO));
```

e depois preencher a posição entretanto criada:

```
variavel[totalElementos-1].campo=...;
```

Quando se pretende eliminar o elemento da posição *i* é necessário efectuar a troca entre o último elemento e o da posição a eliminar:

```
variavel[i].campo=variavel[totalElementos-1].campo;  
...
```

e depois actualizar a dimensão da estrutura de dados:

```
variavel=(TIPO*)realloc(variavel,--totalElementos*sizeof(TIPO));
```

5.3.3 *Open GL*

Na implementação da componente gráfica foi utilizada a biblioteca de funções *OpenGL*².

Pode-se definir a *OpenGL* como "um *interface* de *software* para o *hardware* gráfico de um computador" (Jr. Sweet 2000). É uma livraria para modelação gráfica 2D e 3D extremamente rápida e portátil. Foi desenvolvida pela *SGI-Silicon Graphics, Inc.*, uma empresa líder no mercado da computação gráfica e animação.

Não deve ser confundida com uma linguagem de programação como o C ou o Java, não sendo correcto falar-se em programas *OpenGL*. O termo correcto será o de "programas que utilizam a *OpenGL* como uma das suas API's". Da mesma forma como se pode utilizar a API do *Windows* para aceder a um ficheiro, analogamente se utiliza a *OpenGL* para a modelação de objectos gráficos 2D e 3D.

É usada numa vasta gama de aplicações, desde programas de engenharia CAD, aplicações de arquitectura até animação gráfica para a industria cinematográfica.

Apesar de relativamente recente³, a sua crescente utilização levou a que já se possa falar em standard. O seu antecessor foi o *IRIS GL* da *Silicon Graphics*, a API de programação tridimensional que a empresa incorporava nas estações gráficas IRIS. Eram computadores

²<http://www.opengl.org>

³A versão 1.0 surgiu em Julho de 1992

especializados no desenho de sofisticadas componentes gráficas, cujo *hardware* estava optimizado para a transformação de matrizes. Ao tentar alargar o *IRIS GL* a outro tipo de sistemas surgiram problemas de compatibilização, residindo aqui a génese do *OpenGL*.

A *OpenGL* resultou dos esforços levados a cabo pela *SGI* para melhorar a portabilidade do *IRIS GL*. A nova *API* mantém as características *GL* da antecessora mas permanece compatível com diferentes plataformas e sistemas operativos.

A *API OpenGL* é hoje comumente aceite como um standard. Incluí mais de 300 comandos e funções usados basicamente para desenhar um conjunto de primitivas (pontos, linhas e polígonos) e depois operar transformações sobre elas.

Não inclui funções de gestão de janelas, interacção com o utilizador ou leitura e escrita de ficheiros uma vez que cada sistema operativo tem formas distintas de implementar tais funcionalidades.

5.3.3.1 Implementações genéricas

Existem implementações da *OpenGL* específicas para determinado tipo de *hardware*. Neste caso existem algumas transformações que são efectuadas pelo próprio *hardware* optimizando o desempenho.

Uma implementação genérica usa apenas *software* para a implementação de todas as funções *OpenGL*. Esta foi a versão utilizada, não condicionando a funcionalidade do sistema ao *hardware* utilizado.

5.3.3.2 Duplo Buffer

Uma das características mais importantes numa biblioteca como a *OpenGL* é o suporte a duplo *buffer*. Esta característica permite que enquanto uma cena (um conjunto de primitivas) está a ser mostrado no écran, outra esteja a ser desenhada num *buffer* paralelo. No final basta efectuar a troca. A qualidade do refrescamento da imagem melhora significativamente no modo de duplo *buffer*.

Para activar o modo de duplo *buffer* na implementação de um programa deve-se introduzir a seguinte linha de código:

```
glutInitDisplayMode(GLUT_DOUBLE);
```

Depois basta definir qual a função que desenha o *buffer* activo e o inactivo.

5.3.3.3 GLUT

Tal como foi atrás referido a *OpenGL* não contém funções para gestão de janelas, ou interacção com o utilizador, deixando estas tarefas para as *API's* de cada linguagem e plataforma

em particular.

Por esta razão surgiram bibliotecas auxiliares para facilitar a implementação de programas que usam a *OpenGL* sem que o programador se perca em tarefas demasiadamente minuciosas relacionadas com o sistema operativo ou plataforma específica.

A primeira biblioteca de apoio foi a *AUX*, considerada pouco estável e de sintaxe pouco intuitiva.

Posteriormente surgiu a GLUT (*OpenGL Utility Toolkit*), escrita por Mark Kilgard⁴. A GLUT é uma biblioteca de suporte à criação de programas que utilizam a biblioteca *OpenGL* independente da plataforma. Facilita a aprendizagem e implementação ao permitir a execução de programas em plataformas diferentes. Disponibiliza uma API portátil que permite que a mesma aplicação funcione em OS/2, *Windows 95*, *Windows NT*, *Mac* e *Unix*.

As suas principais funcionalidades consistem em:

- Possibilidade de refrescamento simultâneo de múltiplas janelas. Esta característica permite que de uma forma aparentemente simultânea se mostrem gráficos em várias janelas ou zonas de uma aplicação.
- Chamadas a funções a partir de eventos. Define formas de invocar funções em resposta a eventos.
- Rotinas de temporização e modo de desenho em *background*.
- Suporte para fontes *bitmap* ou *stroke*.

Está disponível gratuitamente para integração em aplicações C, Fortran e ADA.

5.3.3.4 GLUI

A GLUI (Rademacher 1999) é outra biblioteca auxiliar no desenvolvimento de programas que utilizam a *OpenGL* e que permite a criação de componentes de controlo para a aplicação de uma forma rápida e independente da plataforma e sistema operativo. Esta característica é bastante importante, uma vez que evita a necessidade de implementar versões diferentes para o *interface* com o utilizador consoante o sistema operativo.

As suas principais funcionalidades são:

- Forma simples de criar janelas de interacção com o utilizador.
- Suporte a múltiplas janelas de *interface*.
- Integração de botões.

⁴<http://www.opengl.org/developers/documentation/glut/>

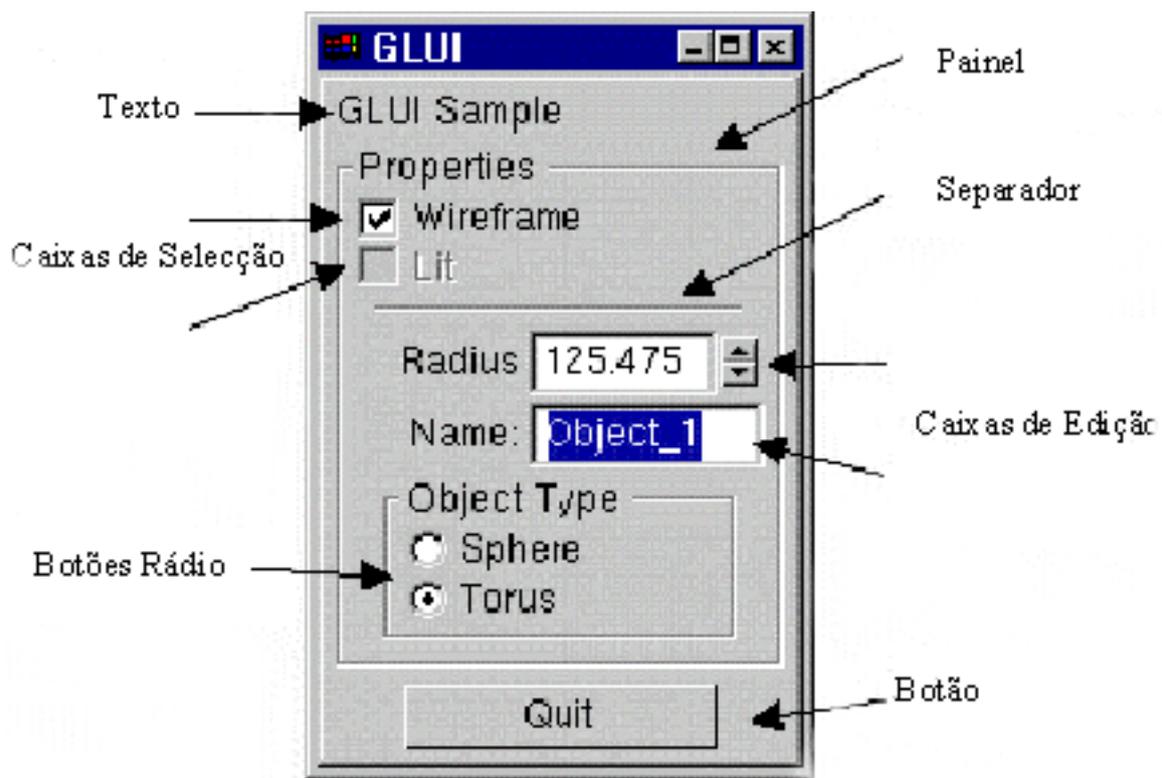


Figura 5.3: Exemplo de uma janela de interação GLUT.

- Caixas de selecção para variáveis booleanas.
- Botões rádio para opções mutuamente exclusivas.
- Caixas de texto editáveis.
- Texto estático.
- Painéis para agrupar elementos de controlo
- Organização e dimensão dos elementos de controlo automática.

A criação de eventos relacionados com os elementos de controlo é feita através da definição de funções específicas para cada controlo.

A figura 5.3 mostra a aparência de uma janela criada através da biblioteca GLUT.

5.4 Comunicação

Um sistema multi-agente baseia o seu funcionamento na interacção entre os seus componentes. É necessário possuir uma estrutura que permita a comunicação entre os agentes,

como forma de transmissão da informação necessária. Esta secção diz respeito à plataforma de comunicação utilizada para tal.

5.4.1 Plataforma de Comunicação

A crescente relevância dos agentes e dos sistemas neles baseados levou ao aparecimento de vários programas auxiliares para a sua concepção, implementação e teste. Existe hoje em dia, um vasto número de aplicações destinadas a facilitar a implementação de agentes. A sua complexidade varia desde as que apenas proporcionam um suporte para a comunicação, até às plataformas integradas que fornecem mecanismos de análise de sistema, implementação, teste e depuramento de sistemas baseados em agentes.

5.4.1.1 Plataformas Disponíveis

Segue-se uma descrição de algumas das plataformas de suporte à criação de agentes mais conhecidas ⁵:

AgentTalk Disponibiliza protocolos para coordenação de agentes. Linguagem LISP.

Endereço WEB: <http://www.kecl.ntt.co.jp/csl/msrg/topics/at/>.

AgentBuilder Ambiente integrado para o desenvolvimento de sistemas com agentes. Linguagem Java.

Endereço WEB: <http://www.agentbuilder.com/>.

Aglets Destinada à criação de agentes móveis. Linguagem Java.

Endereço WEB: <http://www.trl.ibm.com/aglets>.

JAM Disponibiliza arquitecturas para agentes. Linguagem Java.

Endereço WEB: <http://www.marcush.net/>.

Agent TCL Suporte à criação de agentes móveis. Linguagem TCL.

Endereço WEB: <http://agent.cs.dartmouth.edu/general/agenttcl.html>.

Bee-gent Ambiente de desenvolvimento de software com agentes. Linguagem Java.

Endereço WEB: <http://www2.toshiba.co.jp/beegent/index.htm>.

JADE Ambiente para desenvolvimento de sistemas multi agente. Linguagem Java.

Endereço WEB: <http://sharon.cselt.it/projects/jade/>.

JAFMAS Ambiente de desenvolvimento para sistemas multi-agente. Linguagem Java.

Endereço WEB: <http://www.ececs.uc.edu/~abaker/JAFMAS>

Zeus Ambiente para desenvolvimento de agentes. Linguagem Java.

Endereço WEB: <http://193.113.209.147/projects/agents.htm>.

⁵Fonte: www.agentbuilder.com

JATLite Plataforma para criação e comunicação entre agentes. Linguagem Java.

Endereço WEB: <http://java.stanford.edu>

MadKit Plataforma para o desenvolvimento de sistemas multi-agente. Linguagem Java.

Endereço WEB: <http://www.madkit.org/>.

Apesar de cada uma apresentar características próprias, todas partilham o objectivo de proporcionar ao utilizador um nível de abstracção que lhe permita preocupar-se apenas com as características e detalhes do sistema específico.

Cada empresa ou universidade que implementou uma plataforma de suporte à criação de agentes utiliza formas diferentes de justificar a sua necessidade. Por exemplo a *Reticular Systems* diz que fornece uma "ferramenta integrada para a construção de agentes inteligentes" (J.Pachl 2002), enquanto os autores do JADE definem a sua aplicação como "um *framework* de *software* para facilitar o desenvolvimento de agentes em sistemas multi-agente" (F.Bellirfemine, A.Poggi, G.Rimassa P.Turci 2000).

Outra característica relevante é o facto de incorporarem características de programação visual, com *interfaces* agradáveis para o utilizador, que permitem a utilização por parte de programadores menos experientes.

De entre a vasta gama de plataformas actualmente disponíveis, nem todas são indicadas para qualquer tipo de sistema. Existem especialmente indicadas para a implementação de agentes de comércio electrónico, outras para os agentes inteligentes, e algumas para os sistemas multi-agente. Nesta última categoria, as mais utilizadas são as seguintes (Serenko Detlor 2002):

Madkit Linguagem JAVA

Zeus Linguagem JAVA

JADE Linguagem JAVA

JatLite Linguagem JAVA

FIPA-OS Linguagem JAVA

Após a análise das características destas plataformas, da troca de experiências com colegas que anteriormente tinham utilizado algumas e pela própria experiência na utilização do *JatLite* e *JatLiteBean* decidi-me pela não utilização de nenhuma plataforma.

Quais as razões que sustentaram esta opção?

A eventual escolha de uma plataforma está umbilicalmente ligada à escolha da linguagem de implementação do sistema. Ao olhar para a lista acima exposta nota-se imediatamente que todas as plataformas especialmente concebidas para a implementação de sistemas multi-agente estão implementadas em JAVA, o que desde logo pode trazer inconvenientes em termos de eficiência do sistema (Secção 5.3.1).

O sistema a implementar tinha como crítico o factor tempo. A acrescentar ao sistema multi-agente existia a necessidade de implementação de um simulador. As aplicações de simulação consomem normalmente elevados recursos computacionais.

É imprescindível que a resposta do sistema a uma dada situação, além de conveniente, seja dada em tempo útil. (De nada valerá dar ordem de paragem a dois comboios após a sua colisão.)

Outro factor residiu na proporção entre as funcionalidades disponibilizadas pelas plataformas e as efectivamente necessárias para o sistema em questão.

Comparados com sistemas de negociação, de pesquisa ou extracção de informação, os sistemas de controlo são bastante mais rígidos nas tarefas a desempenhar, não havendo simultaneamente necessidade de funcionalidades como multi-difusão, reencaminhamento de mensagens ou reenvio de mensagens. Cada agente tem que saber precisamente para quem, o quê e quando comunica.

Tendo em consideração que a linguagem Java é menos eficiente que o C (Secção 5.3.1), decidida a utilização da biblioteca *OpenGL* (mais estável na linguagem C) e atendendo às funcionalidades pretendidas, decidi-me pela implementação de um sistema de comunicações através de *sockets* implementado em linguagem C.

Resumindo, quais as razões fundamentais para esta escolha?

Eficiência Este foi o factor primordial. A implementação em Java e incorporação de funcionalidades desnecessárias para o sistema em questão, tornariam a troca de mensagens mais lenta.

OpenGL A escolha das bibliotecas *OpenGL* para a componente gráfica da simulação e a sua maior estabilidade para a linguagem C condicionou a escolha da linguagem. As plataformas de apoio à criação de sistemas multi-agente mais utilizadas estão implementadas em Java.

Familiarização com a plataforma Ao serem concebidas para a implementação de sistemas genéricos estas plataformas requerem algum tempo de familiarização para poderem ser utilizadas de forma conveniente. Nos casos em que a utilização é mais fácil (*JatLite*), isso é conseguido através de programação de mais alto nível, o que deteriora bastante a eficiência da plataforma.

Funcionalidades desnecessárias O sistema idealizado não utilizaria grande parte das funcionalidades (facilidades) proporcionadas pelas plataformas (reencaminhamento de mensagens, multi-difusão, reenvio, esqueletos de agentes, ...).

5.4.2 KQML - *Knowledge and Query Manipulation Language*

Subjacente à introdução de tecnologias para agentes, linguagens de programação, protocolos de comunicação e teorias para agentes está a capacidade dos elementos de um sistema

trocarem informação e conhecimento entre si. Esta interacção deve ser possível independentemente das características físicas, sistemas operativos, arquitecturas ou linguagens de programação.

A linguagem de comunicação constitui uma componente crucial na implementação dos sistemas multi-agente, permitindo a partilha de significados e sentido na informação trocada entre os intervenientes.

A transferência de informação entre os intervenientes no sistema MARCS é feita através de mensagens de texto formatadas na linguagem *KQML*⁶, funcionando esta como suporte à comunicação.

Pode-se definir como "uma linguagem e protocolo para troca de informação e conhecimento vocacionada para o nível pragmático da comunicação" (Oliveira 2002), sendo utilizada para partilha de conhecimento, informação ou como forma de coordenação de actividades em sistemas multi-agente.

Decompõe-se em três níveis:

Envelope Define os agentes emissor e destinatário da mensagem. Caso existam intermediários na comunicação, define o último agente receptor da mensagem.

Performativa Identifica a classe da mensagem, podendo servir como critério de prioridade no tratamento de mensagens ou indicar a necessidade de executar alguma acção em resultado da recepção da mensagem.

Conteúdo Pode possuir uma linguagem própria e é neste nível que é especificada a informação a transmitir.

Segue-se o exemplo de uma mensagem KQML:

```
(reply :sender Agente1
      :receiver Agente2
      :in-reply-to Assunto1
      :content precoPeca(101,12.99))
```

A mensagem anteriormente descrita tem performativa "reply", indicando que se trata da resposta a uma mensagem anteriormente recebida pelo agente. É identificado o agente emissor (Agente1), receptor (Agente2) e o assunto a que a mensagem se refere (Assunto1). Finalmente surge o conteúdo da mensagem, onde é verdadeiramente incluída a informação a transmitir. Neste caso o conteúdo está expresso na forma de predicados *Prolog*, podendo a informação estar representada noutra linguagem (*XML*, *HTML*, ...) ou qualquer outro formato perceptível por ambos os agentes. Outros campos podem ser incluídos na mensagem, como sejam o da ontologia e o da linguagem do conteúdo.

⁶<http://www.cs.umbc.edu/kqml/>

5.4.2.1 Performativas Utilizadas

Apesar do KQML reservar uma série de palavras-chave identificadoras de predicados e campos da mensagem, permanece uma linguagem extensível, isto é, o utilizador pode aumentar o vocabulário disponível, ou decidir quais as performativas a usar, bem como os campos integrantes de cada tipo de mensagem.

No sistema MARCS são transmitidas mensagens com as seguintes performativas:

alert Mensagens a indicar a ocorrência de uma situação excepcional, tal como inexistência de agentes supervisores de tráfego. São compostas pelos seguintes atributos:

- sender → Identificação do agente emissor da mensagem.
- receiver → Identificação do agente destinatário da mensagem.
- subject → Assunto sobre o qual se reporta a mensagem de alerta.

ask Esta performativa è utilizada sempre que um agente necessita de qualquer informação. Cada mensagem com esta performativa contém os seguintes atributos:

- sender → Identificação do agente emissor da mensagem.
- receiver → Identificação do agente destinatário da mensagem.
- reply-width → Identifica o assunto sobre o qual se efectua a questão. Este deverá ser incluído na correspondente mensagem de resposta.

control As ordens de controlo provenientes dos agentes *Comboio* utilizam esta performativa. Simulam as ordens enviadas para os mecanismos físicos de controlo, contendo os seguintes atributos:

- sender → Identificação do agente emissor da mensagem.
- receiver → Identificação do agente destinatário da mensagem.
- identificação objecto → Identificação do objecto sobre o qual se reporta a ordem de controlo.

reply Correspondem a respostas sobre questões anteriormente formuladas através de mensagens com performativa "ask". Contêm os seguintes atributos:

- sender → Identificação do agente emissor da mensagem.
- receiver → Identificação do agente destinatário da mensagem.
- in-reply-width → Assunto sobre o qual se efectua a resposta. Corresponde ao atributo "reply-width" da mensagem com performativa "ask" que motivou a resposta.

tell Esta performativa indica mensagens de informação de rotina, tal como o posicionamento de cada comboio. São compostas pelos seguintes atributos:

- sender → Identificação do agente emissor da mensagem.
- receiver → Identificação do agente destinatário da mensagem.
- content → Conteúdo da mensagem.

register Antes de poderem trocar informação, os agentes têm que se registar mutuamente. A intenção de registo é expressa através de uma mensagem com esta performativa e os atributos:

- sender → Identificação do agente emissor da mensagem.
- receiver → Identificação do agente destinatário da mensagem.
- endereço → Endereço onde o agente emissor escuta mensagens através de *sockets*.

Esta é uma descrição superficial das performativas utilizadas e dos atributos que compõem cada uma delas. A descrição pormenorizada pode ser consultada no apêndice C.

Conclusão

Neste capítulo fez-se uma apresentação geral das linguagens utilizadas no desenvolvimento do sistema (UML, XML, KQML, C++), tendo, através da comparação com possíveis alternativas, sido apresentados os principais factores que determinaram tais escolhas. No caso específico da plataforma de suporte à comunicação entre agentes, apresentaram-se razões justificativas para a sua construção de raiz em detrimento da utilização de uma existente para o efeito.

Capítulo 6

Implementação do Sistema

Neste capítulo descrevem-se os aspectos mais relevantes relativos à implementação e consequente comportamento do sistema. Detalha-se o processo de controlo e ordenamento de tráfego e a informação circulante no sistema.

6.1 Arquitectura Geral

Na implementação do sistema consideraram-se os seguintes agentes:

- Supervisor
- Comboio
- Estação
- Aprender

Todos partilham a mesma arquitectura, representada na sua forma mais abstracta na figura 6.1. Tal como ilustrado, pode-se identificar em todos os agentes do sistema MARCS:

Módulo de Comunicação Responsável pela interacção com os restantes intervenientes do sistema.

Módulo de Conhecimento Conjunto de estruturas de dados que permitem a manutenção da informação essencial para o desempenho da função prevista para o agente.

Módulo de Decisão Responsável pelo tratamento das mensagens recebidas e pela sistemática análise do estado do ambiente, possuindo capacidade para tomar a iniciativa de interacção com outros.

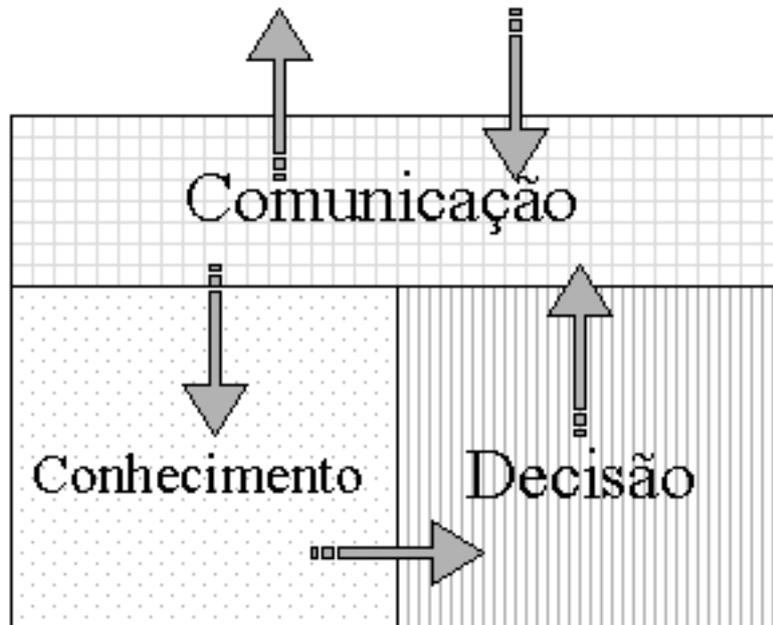


Figura 6.1: Arquitectura abstracta dos agentes.

6.2 Representação do Conhecimento

Esta secção è relativa às estruturas de dados implementadas para o armazenamento da informação a processar por cada agente interveniente no sistema.

6.2.1 Mensagens

Uma vez que não existe limite sobre o número de mensagens simultaneamente recebidas por cada agente, foi necessário implementar uma fila de espera para todas as mensagens à espera de tratamento. Para tal foi criada uma estrutura *Mensagem*, composta pelos seguintes atributos:

socketOrigem Identificador do *socket* através do qual foi enviada a mensagem. Este valor identifica automaticamente o agente remetente.

conteudo Estrutura dinâmica destinada ao conteúdo da mensagem. Uma vez que as mensagens podem ter tamanhos distintos, não existindo limite para a sua dimensão, foi implementado através de um vector dinâmico de caracteres.

6.2.2 Grafos

Para possibilitar o controlo do tráfego, os agentes supervisores necessitam de manter informação sobre a estrutura da rede e respectiva área de supervisão.

A forma clássica de representação de redes de transporte é feita através de grafos dirigidos (Dijkstra 1959). Arestas representam troços e vértices representam locais onde existe intersecção entre troços.

6.2.2.1 Grafo Dual

Existem redes que acrescentam restrições de mudança de direcção na chegada a um vértice. A direcção permitida para um elemento prestes a atingir um vértice varia de acordo com a aresta em que se encontra.

No sistema MARCS apenas é permitido aos comboios executarem mudanças de direcção até 45 graus. Ao aproximar-se de um ponto de selecção de trajecto, o comboio apenas pode seguir pelos trajectos que fizerem um ângulo menor ou igual a 45 graus relativamente ao actual.

Na figura 6.2 está representada uma rede ferroviária com um ponto de selecção de trajecto (V_1), três troços (T_1 , T_2 e T_3) e três pontos terminais de linha (F_1 , F_2 e F_3).

Um comboio que circule no troço T_1 em direcção a V_1 não poderá atingir T_2 . Esta restrição já não se aplica se o comboio estiver localizado em T_3 .

Após a construção do grafo, fazendo corresponder troços a arestas e pontos de selecção ou terminais a vértices, obtém-se uma estrutura que com quatro vértices e três arestas que não contém informação sobre as restrições de direcção descritas.

Uma solução consiste em expandir cada vértice com restrições em vários virtuais por forma a que, para cada um, não existam restrições. Na figura 6.3 mostra-se o grafo resultante após a divisão do vértice com restrições de direcção (V_1) em dois fictícios (V_{1A} e V_{1B}). Neste grafo está explícito que um comboio proveniente de V_4 (correspondente ao ponto terminal T_3), pode seguir em duas direcções diferentes. Vindo de qualquer outro ponto terminal (V_2 ou V_3) apenas se pode seguir numa direcção.

A obrigatoriedade de criação de vários vértices e arestas fictícios torna esta solução pouco eficiente, especialmente para redes de grande dimensão ou com elevado número de restrições.

A alternativa proposta consiste na criação de um grafo **dual**, com informação intrínseca sobre as restrições de direcção (Añez Pérez 1994). O grafo dual é construído com base em duas propriedades:

1. Os vértices duais representam arestas do grafo original.

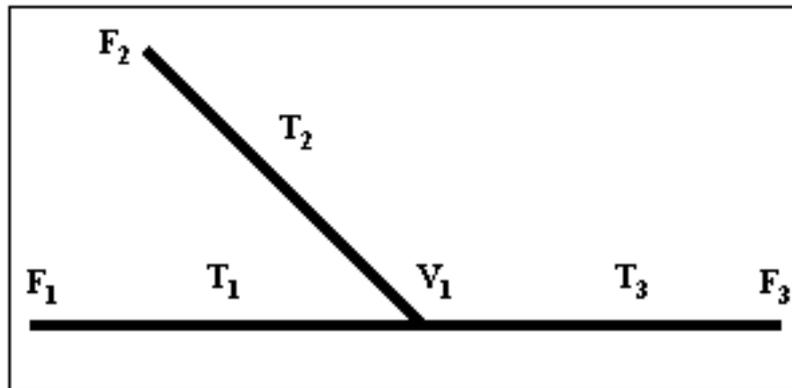


Figura 6.2: Vértice com restrições de direcção.

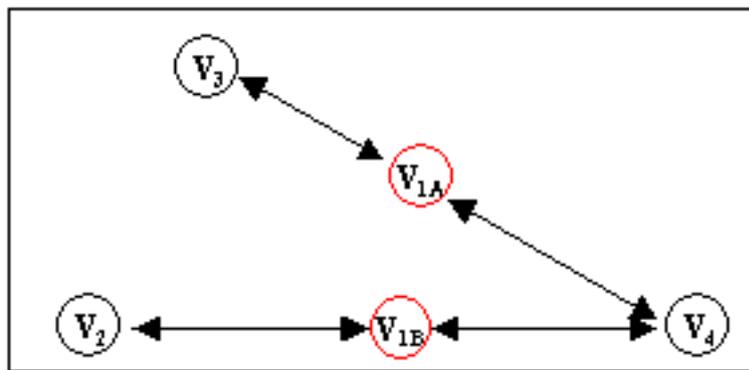


Figura 6.3: Criação de vértices e arestas fictícias.

2. As arestas do grafo dual representam possibilidade de circulação entre arestas adjacentes do grafo original.

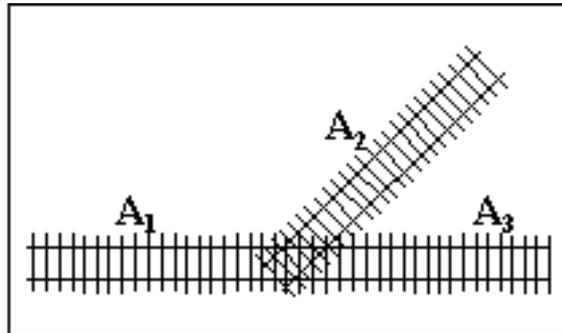
Desta forma o algoritmo de construção do grafo dual, executado por cada um dos agentes supervisores do tráfego, é bastante simples de construir e contém a informação necessária para gerir as restrições de direcção (Algoritmo 6.1).

Exemplo

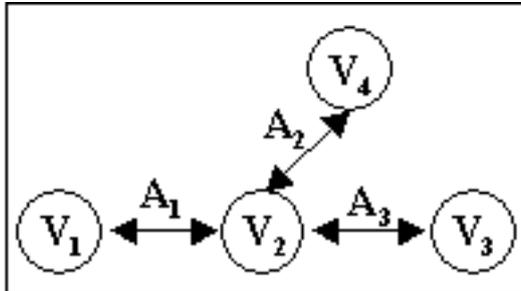
A figura 6.4a mostra uma rede ferroviária com um agulha de selecção, três arestas e três pontos terminais. Na figura 6.4b mostra-se o grafo obtido a partir da rede anterior, não contendo informação sobre as restrições de mudança de direcção. Finalmente a figura 6.4c é respeitante ao grafo dual onde as mudanças de direcção superiores a 45 graus não são permitidas.

6.2.2.2 Estrutura de Dados

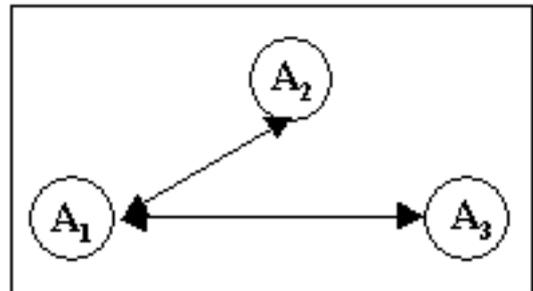
A figura 6.5 esquematiza a estrutura implementada para representar um grafo. É composta por dois tipos de sub-estruturas: *Aresta* e *Vertice*.



(a) Rede ferroviária (Exemplo).



(b) Grafo original obtido a partir de 6.4a.



(c) Grafo dual obtido a partir de 6.4b.

Figura 6.4: Construção do grafo dual.

Algoritmo 6.1 Construção do grafo dual

```

Adicionar_Vertice_Dual(VERTICE_ORIGEM)
for  $x = 0; x < TotalArestas; x ++$  do
     $a \leftarrow Devolve\_Aresta(x)$ 
    Adicionar_Vertice_Dual(a)
end for
for  $x = 0; x < TotalVerticesDual; x ++$  do
     $v \leftarrow Devolve\_Aresta(x)$ 
    for  $y = x + 1; y < TotalVerticesDual; y ++$  do
         $u \leftarrow Devolve\_Aresta(x)$ 
        if Adjacentes(u,v) E Direcoes_Compativeis(u,v) then
            Adicionar_Aresta_Dual(u,v)
        end if
    end for
end for

```

Os elementos de *Vertice* (V_0, V_1, \dots, V_n) estão guardados num vector dinâmico com referências para os elementos de *Aresta* (A_0, A_1, \dots, A_m) adjacentes. Estes estão também guardados num vector dinâmico específico para o efeito.

As razões que motivaram a escolha desta estrutura prendem-se essencialmente com a eficiência. Uma vez que cada vértice contém referências para as arestas adjacentes e estas para os respectivos vértices, minimiza-se o número de vezes que é necessário percorrer o vector na pesquisa de elementos.

6.2.2.3 Cálculo do Trajecto Ideal

Mediante a próxima paragem definida para cada comboio, é necessário calcular o melhor trajecto para o atingir. O algoritmo 6.2 permite calcular o trajecto de menor custo entre quaisquer dois pontos de um grafo. (Morris 2002).

Tendo definido um grafo como o ilustrado na figura 6.6:

Conjunto de Vértices Pontos terminais do grafo, ou locais onde é possível mudar de direcção.

Conjunto de Arestas Ligação entre dois vértices.

Cada aresta ($A_i : V_j \leftrightarrow V_k$) tem um custo associado, que pode ser descrito como o "custo da deslocação entre os vértices V_j e V_k ". O significado deste custo determina o resultado da execução do algoritmo, podendo ser um dos seguintes:

Distância euclidiana Caso o custo de cada aresta represente a distância entre dois vértices, o algoritmo permite descobrir o caminho mais curto entre quaisquer dois pontos do grafo.

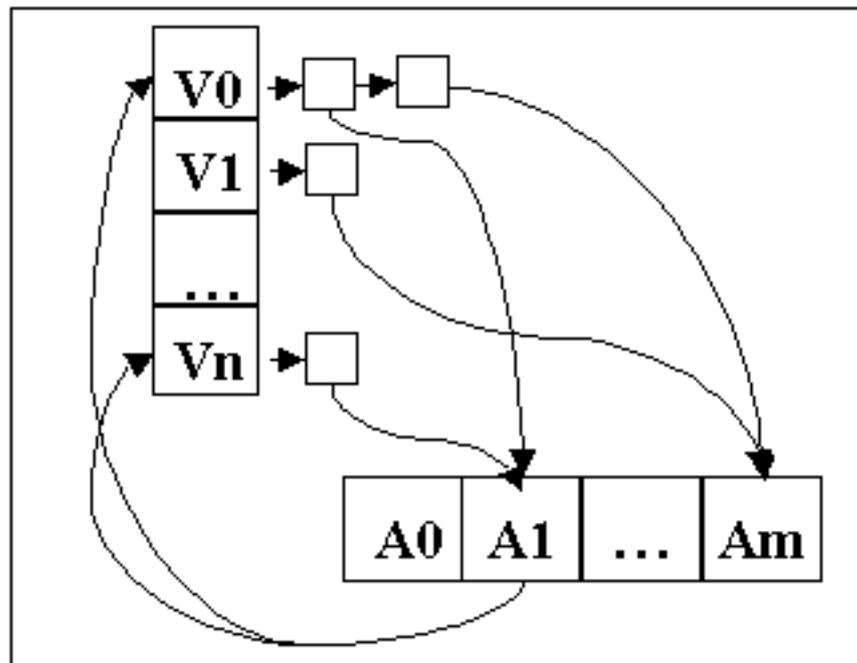


Figura 6.5: Estrutura para representar um grafo.

Tempo A execução do algoritmo permite determinar o trajecto mais rápido entre dois pontos.

Mudanças de Direcção É possível o cálculo do trajecto que implique menos mudanças de direcção.

Consumo Neste caso pode ser calculado o trajecto mais económico entre dois pontos.

Etc

Podem-se associar vários destes parâmetros por forma a que o trajecto encontrado seja óptimo considerando a combinação de vários factores (Winter Grünbacher 2003).

No sistema MARCS o custo associado a cada aresta é calculado com base em dois parâmetros: *Distancia* e *Momento*.

Distancia representa a distância euclidiana entre dois vértices adjacentes, sendo consequentemente um valor estático atribuído na inicialização do sistema.

Momento é um valor dinâmico calculado em função do estado do tráfego e da participação em planos de supervisão. Possibilita que na definição de trajectos para um comboio se evitem os troços com maior volume de tráfego, ou onde o razões de planeamento desaconselham a circulação. É inicializado com o valor 1.

Assim, para qualquer aresta A_i pertencente ao grafo, tem-se:

$$Custo(A_i) = Distancia(A_i) \times Momento(A_i)$$

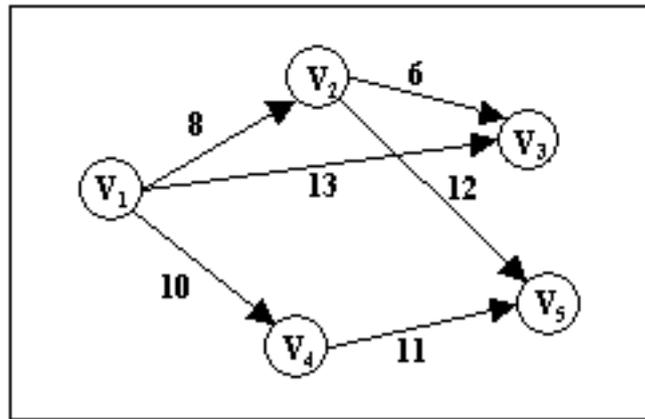


Figura 6.6: Grafo (Exemplo).

Algoritmo 6.2 Algoritmo de Dijkstra.

```

V ← VerticesGrafo(G)
S ← NULL
origem ← LocalizacaoComboio(i)
while V-S!=NULL do
  Ordena(V-S,origem)
  u ← RetiraInicioLista(V - S)
  AdicionaLista(S,u)
  RelaxaVertices(V-S,u)
end while

```

6.2.3 Comboios Supervisionados

Cada agente supervisor necessita de manter informação acerca dos comboios sob sua supervisão. Para cada um destes é criada uma estrutura de dados com os seguintes atributos:

ID Identificação do comboio.

idAgente Identificação do agente que representa o comboio.

velocidade Última velocidade conhecida para o comboio.

arestaF Identificador da aresta onde se encontra a parte da frente do comboio.

arestaT Identificador da aresta onde se encontra a parte traseira do comboio.

distânciaF Distância entre o próximo vértice e a parte dianteira do comboio.

distânciaT Distância entre a parte traseira do comboio e o último vértice por onde esta passou.

verticeOrigemF Identificador do último vértice por onde passou a parte dianteira do comboio.

verticeDestinoF Identificador do próximo vértice por onde a parte dianteira do comboio previsivelmente irá passar.

verticeOrigemT Identificador do último vértice por onde passou a parte traseira do comboio.

verticeDestinoT Identificador do próximo vértice por onde a parte traseira do comboio previsivelmente irá passar.

trajecto Lista com a sequência de arestas por onde está previsto que o comboio circule.

proximaParagem Identificação da próxima estação onde o comboio deve parar.

6.2.4 Plataformas de Paragem

Os agentes *Estação* necessitam de guardar informação sobre as plataformas de paragem, a sua localização relativa na rede e respectiva dimensão. Para este efeito foi criada uma estrutura de dados cujos atributos mais relevantes são os seguintes:

agente1 Identificador do agente responsável pela supervisão do troço onde se localiza a paragem.

agente2 Identificador do segundo agente supervisor, caso a paragem esteja localizada numa zona de transferência de supervisão.

aresta Identificador da aresta onde se localiza a paragem.

vertice1 Identificador de um vértice adjacente à aresta onde se localiza a paragem.

vertice2 Identificador do segundo vértice adjacente à aresta onde se localiza a plataforma de paragem.

distância1 Distância entre uma extremidade da plataforma de paragem e *vertice1*.

distância2 Distância entre uma extremidade da plataforma de paragem e *vertice2*.

6.3 Atribuição de Troços

Nesta parte descreve-se o critério de atribuição da competência de supervisão de troços e pontos de selecção pelos agentes participantes no controlo do tráfego. Este critério induz à noção de **vizinhança** entre supervisores.

6.3.1 Critério de Atribuição

Cada *Supervisor* possui valores limite para a latitude e longitude da sua zona de supervisão, sendo a responsabilidade sobre o tráfego atribuída da seguinte forma:

1. O agente é responsável pela supervisão de todas as agulhas de selecção localizadas dentro destes limites.
2. Adicionalmente, o agente é responsável pela supervisão de todos os trajectos adjacentes às anteriores.

Na figura 6.7 pode ver-se um exemplo relativo à distribuição da competência de supervisão dos trajectos e agulhas de selecção.



Figura 6.7: Distribuição da competência de supervisão.

As zonas assinaladas a verde e vermelho (troços de linha e agulhas de selecção) são exclusivamente supervisionadas por um agente, enquanto o troço a amarelo, por ser adjacente a agulhas controladas por agentes diferentes, é controlado pelos dois. Nesta zona, em caso de ordens contraditórias, cabe ao agente *Comboio* a decisão sobre qual obedecer, normalmente a mais restritiva.

6.3.2 Noção de Vizinhança

A existência de vários agentes supervisores, cada um responsável por uma parcela da rede, leva à necessidade de comunicação entre eles, nomeadamente no processo de transição de um comboio entre supervisores, na informação sobre o tráfego em zonas vizinhas ou na elaboração conjunta de planos.

No sistema MARCS dois supervisores consideram-se *vizinhos* se **partilharem a supervisão de algum troço de linha**.

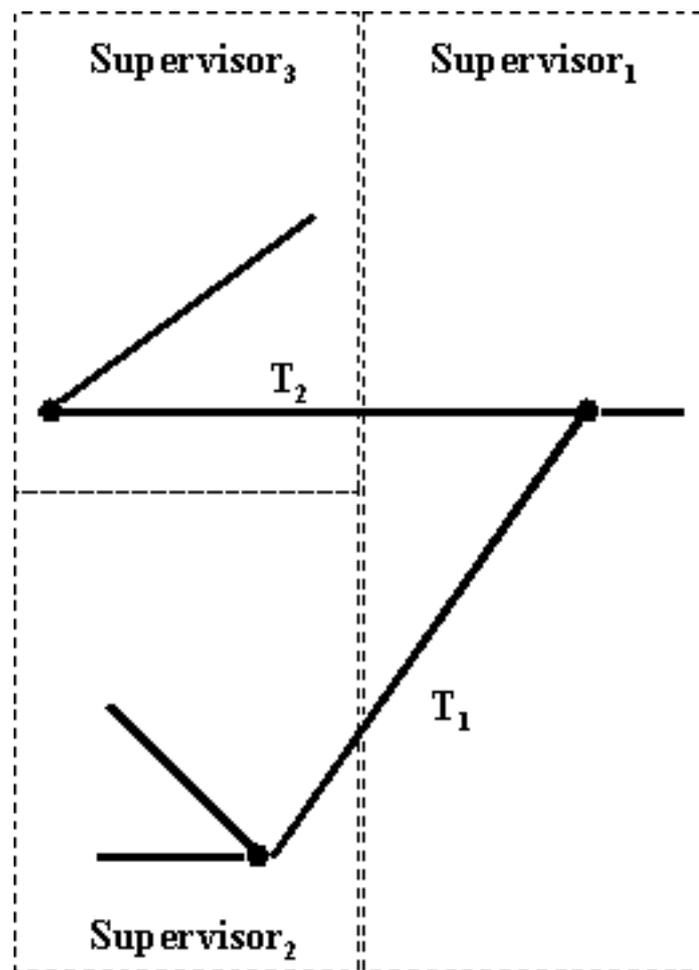


Figura 6.8: Noção de vizinhança (Exemplo).

A figura 6.8 exemplifica a atribuição de troços por três agentes supervisores. O agente *Supervisor₁* partilha a supervisão do troço T_1 com o agente *Supervisor₂* e a de T_2 com o agente *Supervisor₃*. Nesta situação os vizinhos de cada um são os seguintes:

$Supervisor_1 \rightarrow Supervisor_2$ e $Supervisor_3$.

$Supervisor_2 \rightarrow Supervisor_1$.

$Supervisor_3 \rightarrow Supervisor_1$.

Apesar de *Supervisor₂* e *Supervisor₃* supervisionarem zonas adjacentes, não são considerados vizinhos, uma vez que não partilham a supervisão de nenhum troço.

6.4 Informação Proveniente do Simulador

Esta secção é relativa à informação enviada pela aplicação de simulação a cada um dos agentes. Esta informação será a que, num sistema real, não será trocada entre agentes, possuindo cada um capacidades de percepção específicas para tal.

Na aplicação de simulação define-se a topologia da rede, o número de estações, comboios e o trajecto para cada um. É necessário passar esta informação aos respectivos agentes, bem como o estado inicial de cada objecto no processo de simulação.

A estrutura e periodicidade com que a informação é enviada foi definida tendo por base o standard 1494.1-1999 proposto pelo *Institute of Electrical and Electronic Engineers* (IEEE - Apêndice D).

6.4.1 Inicialização

Após a criação ou leitura da rede ferroviária, é necessário proceder à inicialização do sistema. Neste processo, a aplicação de simulação envia aos agentes a seguinte informação:

Agentes Supervisores

- Topologia da rede ferroviária.
- Total de zonas de supervisão.
- Limites das zonas de supervisão.
- Identificação das estações localizadas nas respectivas zonas de supervisão.
- Identificação e localização de todas as plataformas de paragem.
- Identificação dos comboios inicialmente localizados em zonas sob supervisão do agente.
- Endereços dos agentes representantes dos comboios e estações mencionados anteriormente.
- Endereços dos agentes supervisores responsáveis por zonas vizinhas.

Agentes Comboios

- Características do comboio representado.
- Sequência de locais de paragem previstos.
- Endereço do agente supervisor inicial.

Agentes Estações

- Total de plataformas de paragem.
- Localização relativa de cada plataforma de paragem.
- Endereço dos agentes supervisores responsáveis por troços onde existam plataformas de paragem.

6.4.2 Simulação

Agentes Supervisores

Durante o desenrolar da simulação, a aplicação de simulação envia periodicamente, em intervalos de um segundo, informação sobre a localização relativa e velocidade de cada comboio.

Por forma a simular um sistema de localização real, GPS por exemplo, esta informação é enviada a todos os agentes, independentemente de supervisionarem ou não os respectivos comboios. Cabe a cada *Supervisor*, com base na informação recebida, decidir se esta é ou não relevante para si. Os atributos enviados são os seguintes:

- Identificação do comboio.
- Próxima paragem prevista.
- Comprimento.
- Vértice origem da parte dianteira.
- Vértice destino da parte dianteira.
- Aresta de localização da parte dianteira.
- Distância entre a parte dianteira e o próximo vértice destino.
- Vértice origem da parte traseira.
- Vértice destino da parte traseira.
- Aresta de localização da parte traseira.
- Distância entre a parte traseira e o último vértice origem.
- Velocidade instantânea do comboio.

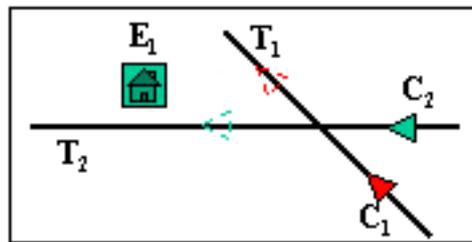


Figura 6.9: Notação das situações de tráfego (Exemplo).

6.5 Sistema de Controlo

Uma rede ferroviária pode ser vista como um conjunto de recursos (troços) que diferentes entidades (comboios) devem partilhar. Cabe ao sistema de controlo garantir a exclusividade na alocação de recursos (segurança) e gerir a prioridade de cada elemento nessa alocação (eficiência).

No caso específico do tráfego ferroviário, imperativos de segurança exigem que, além do espaço físico ocupado por cada comboio, exista uma margem de segurança dentro da qual não poderá estar qualquer outro.

O sistema de controlo deve ter como objectivo a minimização do tempo de espera de cada comboio e a maximização da fluidez do tráfego.

6.5.1 Conflito

No sistema implementado emerge a noção de **conflito**.

Um **conflito** de interesses surge no momento em que **diferentes entidades** desejam aceder ao **mesmo recurso** no **mesmo instante**. No caso das redes ferroviárias, quando vários comboios têm a pretensão de circular pelo mesmo troço ou agulha de selecção ao mesmo tempo.

Notação

A notação de todas as figuras relativas a situações de tráfego é similar à da figura 6.9. Aí temos dois comboios (C_1 e C_2) que viajam em direcção aos troços T_1 e T_2 respectivamente. O triângulo preenchido indica a posição e sentido actual, enquanto o tracejado de cor respectiva indica a direcção pretendida para a composição. As estações (E_1) serão sempre assinaladas com a respectiva figura.

6.5.2 Resolução de Conflitos

A decisão de atribuição de um recurso é feita com base nos seguintes critérios:

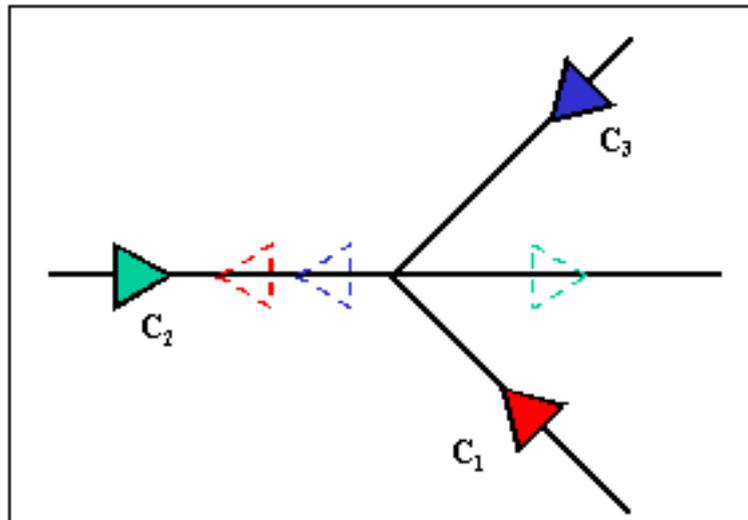


Figura 6.10: Situação de conflito entre 3 comboios.

1. Análise do benefício para a fluidez do tráfego (Benefício global). O agente responsável pela atribuição do recurso (*Supervisor*) deve começar por analisar as vantagens para a circulação resultantes da atribuição do recurso a cada pretendente. Na figura 6.10 ilustra-se uma situação em que, independentemente dos valores de utilidade (Secção 6.5.5) de cada elemento, será conveniente a atribuição de prioridade ao comboio C_2 , uma vez que desbloqueia o trajecto de C_1 e C_3 .
2. Análise de trajectos. Sempre que existam vários comboios aos quais pode ser atribuída a prioridade sem prejuízo para a fluidez do tráfego é analisado o trajecto planeado para cada um deles. A prioridade será atribuída à composição que tenha uma maior distância para percorrer, evitando que esta tenha adicionais tempos de espera em função das paragem dos comboios que seguem à sua frente.
3. Análise do benefício individual. Caso não resultem vantagens dos pontos anteriores, a prioridade é atribuída com base no benefício resultante para cada comboio em particular. É feito de duas formas distintas:
 - (a) Comparação dos valores de utilidade. O valor de utilidade associado a cada comboio indica o tempo que, proporcionalmente, já esteve à espera em anteriores conflitos. É atribuída a prioridade ao que apresentar maior valor de utilidade, constituindo este critério uma tentativa de harmonizar a distribuição dos recursos por todos os elementos, e de evitar discrepâncias entre elementos de igual prioridade.
 - (b) Distância a percorrer para utilizar o recurso. É altamente improvável o recurso a este critério, uma vez que pressupõe a existência de um conflito em que da selecção do vencedor não resultam benefícios óbvios para a rede e todos os candidatos têm trajecto semelhante e valor de utilidade igual. Está enunciado apenas como forma de garantir decisões determinísticas para quaisquer situações

de conflito. Neste caso a alocação do recurso será feita ao comboio que estiver à menor distância do troço ou agulha em questão.

O sistema MARCS está preparado para, autonomamente, resolver conflitos de interesses entre vários comboios com rotas mutuamente bloqueadas.

Pode, devido ao número de comboios envolvidos, colocar-se uma situação de conflito que o sistema não tenha capacidade para resolver.

É importante que o sistema não bloqueie e não funcione como obstáculo à resolução de conflitos. Ao detectar uma situação de conflito que não for capaz de resolver, o sistema automaticamente activa o modo manual de controlo, devendo o estado de cada uma das composições e agulhas de selecção ser definido manualmente.

Este modo corresponde à desactivação dos agentes supervisores de tráfego, continuando os agentes comboio a garantir um espaçamento mínimo entre composições.

Após o restabelecimento das premissas normais de funcionamento pode voltar a activar-se o modo automático de controlo e continuar a execução normal do sistema.

6.5.3 *Distance-To-Go*

Tal como detalhado na secção 3.1.3.3, este método baseia-se na distância que cada comboio tem à sua frente até encontrar o próximo obstáculo. A velocidade permitida para cada comboio é função desta distância e da sua capacidade de travagem. Esta varia de acordo com a capacidade do sistema de travagem, mas também com as condições atmosféricas, o peso ou comprimento do comboio.

No sistema MARCS os agentes supervisores são responsáveis por, regularmente, enviar a cada agente *Comboio* informação sobre a distância livre à frente do comboio por si representado.

Ao receber esta informação, e de acordo com a velocidade recomendada para a respectiva composição, o agente *Comboio* deverá seleccionar a ordem de controlo a enviar ao seu representado. Esta ordem será uma das que se indica na secção 6.5.4.

6.5.4 *Ordens de Controlo*

Num sistema de blocos móveis deixa de fazer sentido o uso dos sinais de controlo colocados em locais fixos da rede. Ao invés, são enviadas ordens directamente para a cabine de cada comboio que determinam a respectiva velocidade.

6.5.4.1 Sinalização

Com o intuito de regular a velocidade do comboio por si representado, o agente *Comboio* pode enviar-lhe uma das seguintes ordens de controlo, representadas pelo respectivo sinal:



Incremento de Velocidade. Ordem para o comboio incrementar a velocidade.



Manutenção de Velocidade. Ordem para o comboio manter a velocidade actual.



Decremento Mínimo de Velocidade. Ordem para reduzir a velocidade através da aplicação da força de travagem mínima.



Decremento Máximo de Velocidade. Ordem para reduzir a velocidade aplicando a força de travagem máxima.

6.5.5 Função de Utilidade

Cada agente *Comboio* tem associado um valor de utilidade destinado à obtenção de prioridade nas situações de conflito. É um valor numérico calculado periodicamente pelo agente e cujo princípio é o seguinte:

- Se um comboio viaja a velocidade inferior à velocidade média desejada aumenta o seu valor de utilidade.
- Se viajar a uma velocidade aproximada à desejada diminui o seu valor de utilidade.

Em conflitos onde os pretendentes estão em igualdade de circunstâncias, é atribuída prioridade ao comboio que tiver um valor de utilidade maior.

6.5.5.1 Definição

O conjunto V que representa a velocidade instantânea dos comboios pode ser definido como:

$$V = \{v \in \mathfrak{R} : v \geq 0\}$$

Cada comboio tem associado um índice de prioridade. Este valor serve para definir a proporção de conflitos de que sairão vencedores.

Sejam V_M e V_m respectivamente os valores máximo e mínimo de prioridade atribuível. O conjunto dos valores de prioridade pode ser definido da seguinte forma:

$$P = \{p \in \mathbb{N} : p \geq V_m \wedge p \leq V_M\}$$

Na inicialização do sistema cada agente *Comboio* cria 2 contadores $Trav$ e Tot destinados respectivamente ao total de vezes que o comboio circula a velocidade inferior à desejada e o total de análises efectuadas. Tem-se:

$$Trav, Tot \in \mathbb{N} : Trav \geq 0 \wedge Tot \geq Trav$$

Na inicialização do agente tem-se:

$$Trav_0 = Tot_0 = 0$$

Seja p a prioridade, v_d a velocidade desejada para o comboio e α a tolerância a essa velocidade. A actualização dos contadores é feita da seguinte forma:

$$Trav_{i+1} = \begin{cases} Trav_i & |v - v_d| \leq \alpha \\ Trav_i + 1 + Max(0, p - (V_M + V_m)/2) & |v - v_d| > \alpha \end{cases}$$

$$Tot_{i+1} = Trav_i + 1 + Max(0, ((V_M + V_m)/2) - p)$$

Finalmente o valor de utilidade é definido pela seguinte função:

$$f : (V \times P) \longrightarrow [0, 1]$$

$$(v, p) \longrightarrow f_i(v, p) = \frac{Trav_i}{Tot_i}$$

6.5.5.2 Razões para a escolha da função de utilidade

O cálculo do valor de utilidade segundo o processo descrito na secção 6.5.5.1 garante que:

1. Comboios que esperam por um recurso aumentam a sua utilidade por forma a aumentarem a sua probabilidade de sucesso em futuros conflitos.
2. Comboios que não esperam por recursos mantêm utilidade baixa o que os coloca em situação de desvantagem em futuros conflitos.
3. Comboios de maior prioridade aumentam a sua utilidade mais facilmente.
4. Comboios de menor prioridade mais facilmente diminuem o seu valor de utilidade.

6.5.6 Situações Especiais

O planeamento antecipado do horário e trajecto de cada composição induz a que, em situações normais, não existam conflitos entre interesses de diferentes comboios.

Para redes de maior dimensão existem métodos e sistemas específicos de planeamento cuja função é evitar a existência de situações onde um comboio tenha que esperar ou alterar a sua rota em função doutro.

Este planeamento será eficiente para a maioria dos casos onde todos os actores da rede ferroviária funcionam da forma prevista.

A ocorrência de um imprevisto pode criar situações de tráfego completamente novas, podendo estas ter por motivo:

- Retenção inesperada de um comboio numa estação.
- Impedimento temporário de um ou vários troços de linha.
- Avaria num comboio.
- etc...

Qualquer uma destas situações pode alterar completamente a situação planeada e originar o aparecimento de conflitos entre os trajectos de vários comboios.

O sistema MARCS, pela suas características de multi-agente, deve saber lidar com estas situações, isto é, detectar e responder autonomamente a situações não esperadas à partida.

6.5.6.1 Resolução

Nesta secção descrevem-se, através de exemplos, os processos de resolução de conflitos de tráfego. Mediante situações específicas e através do método descrito na secção 6.5.6 mostra-se qual seria o comportamento esperado do sistema MARCS.

Na figura 6.11a C_1 e C_2 desejam seguir pelo mesmo troço e passar através do mesmo ponto de selecção. Como aparentemente não existem vantagens relevantes para a circulação da atribuição de prioridade a qualquer um dos elementos (Bem Global), esta será atribuída com base no benefício individual de cada um dos elementos.

São comparados os valores de utilidade de cada composição e atribuída prioridade à que apresentar maior valor (Figura 6.11b). Esta composição será a que, potencialmente, já esteve mais tempo à espera em conflitos anteriores.

A situação descrita na figura 6.12a, apesar de semelhante à anterior, obriga à análise dos trajectos de cada composição. Uma vez que a paragem de C_2 é anterior à de C_1 este seria

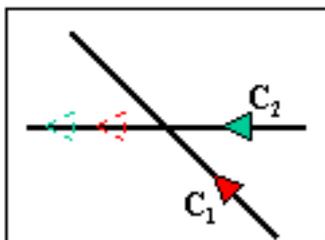
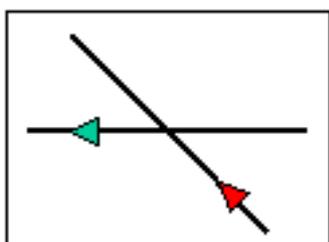
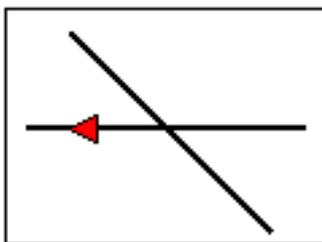
(a) C_1 e C_2 em conflito.(b) Atribuída prioridade a C_1 (c) Passagem de C_2 .

Figura 6.11: Controlo - Igualdade de circunstâncias.

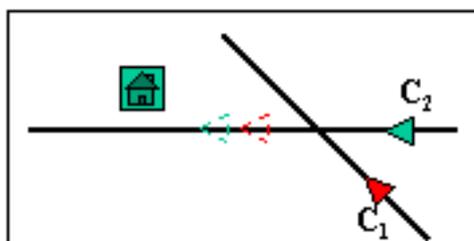
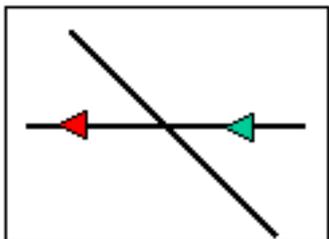
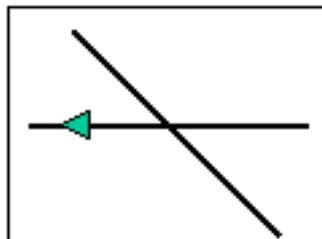
(a) C_1 e C_2 em conflito.(b) Atribuída prioridade a C_1 .(c) Passagem de C_2 .

Figura 6.12: Controlo - Análise de trajectos.

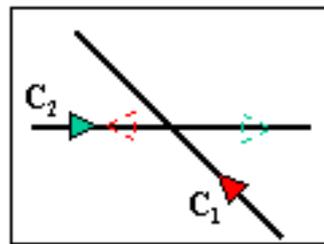
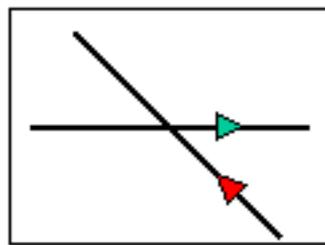
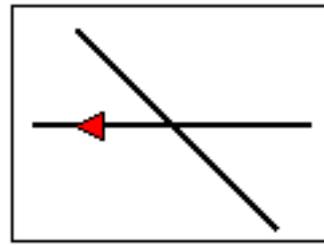
(a) C_1 e C_2 em conflito.(b) Atribuída prioridade a C_2 (c) Passagem de C_1 .

Figura 6.13: Controlo - Vantagem global.

obrigado a esperar em função da paragem da outra composição. Nesta situação é atribuída prioridade ao comboio cuja paragem não bloqueie outros.

A figura 6.13a ilustra uma situação em que os dois candidatos não estão em igualdade de circunstâncias, uma vez que o trajecto de C_1 está bloqueado por C_2 . Nestes casos, independentemente do valor de utilidade de cada comboio, é do interesse para a fluidez do tráfego a atribuição da prioridade a C_2 (Figura 6.13b).

A figura 6.14a diz respeito a conflitos de interesses cuja resolução se detalha na secção seguinte. Neste caso C_1 e C_2 bloqueiam-se mutuamente e será necessário que um deles altere temporariamente o trajecto planeado (Figura 6.14b).

Depois da passagem de C_1 , tendo C_2 atingido um troço não planeado, é necessário proceder à redefinição dos objectivos, isto é, do trajecto planeado para atingir o objectivo. Avaliam-se os benefícios em atingir o anterior estado planeado alternativamente à definição de novos objectivos.

6.5.7 Alteração Temporária de Objectivos

Em situações excepcionais de tráfego como as descritas na secção 6.5.6 é necessário que os comboios alterem temporariamente os seus objectivos para possibilitar a fluidez do tráfego.

Esta alteração constitui um exemplo onde o benefício global da rede se sobrepõe ao de cada

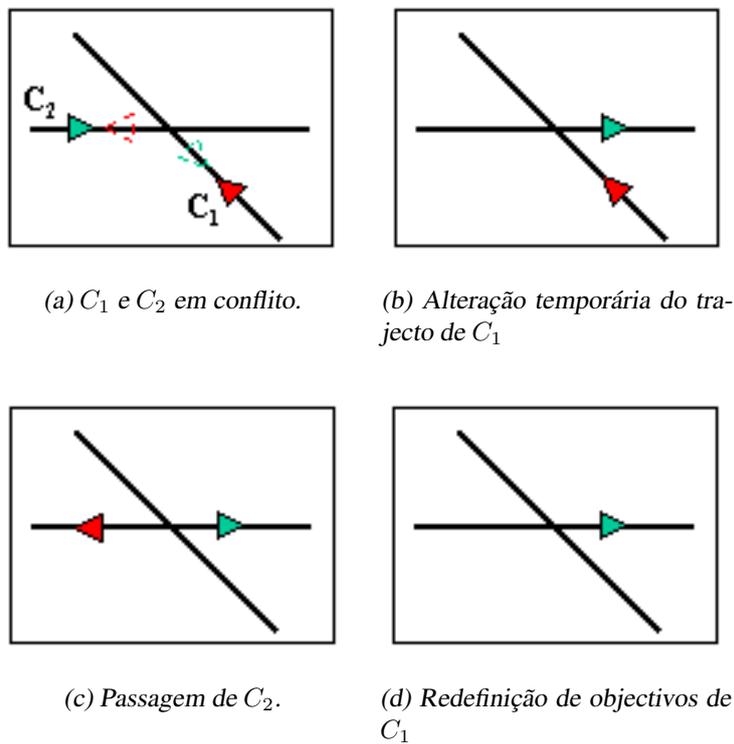


Figura 6.14: Controlo - Igualdade de circunstâncias (Trajectos bloqueados).

elemento em particular. O comboio que altera o trajecto inicialmente planeado, apesar de prejudicado, fá-lo em função de interesses prioritários.

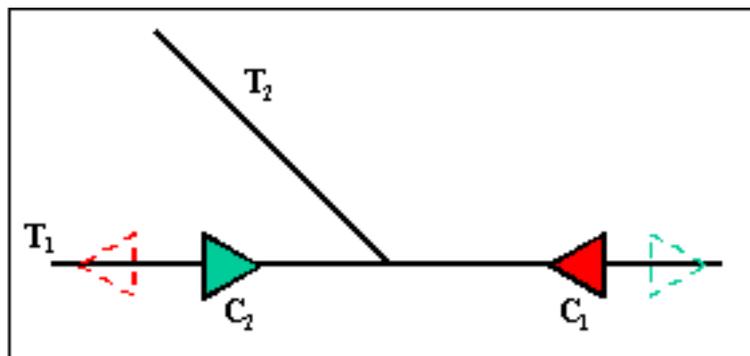


Figura 6.15: Alteração temporária de objectivos.

A figura 6.15 ilustra uma situação de alteração de objectivos. O comboio C_1 , apesar de não ter planeado seguir pelo troço T_2 , terá que o fazer para possibilitar a resolução do conflito existente.

Quando atingir T_2 deve ser reavaliado o trajecto previsto para C_1 . Pode ser mais vantajoso inverter o sentido e retomar o trajecto anteriormente planeado, ou uma vez que chegou a T_2 , optar por outro trajecto.

6.5.8 Entrada e Saída nas Estações

Nesta secção descreve-se o processo de cooperação estabelecido entre os agentes na entrada e saída dos comboios nas estações. A figura 6.16 mostra um diagrama de sequência relativo ao processo.

6.5.8.1 Entrada nas Estações

Cada agente *Comboio* possui uma tabela (*timetable* - Secção F) onde estão indicados as sucessivas estações onde devem ser feitas paragens, constituindo seu objectivo seguir pelo trajecto mais curto até ao próximo local de paragem.

Na aproximação à estação é necessário definir a plataforma onde o comboio deverá parar.

Segue-se uma descrição de cada um dos passos:

Calculo do trajecto Cada agente *Comboio* tem informação sobre a sequência de paragens que a composição por si representada deve executar. O agente supervisor responsável pela área onde o comboio está localizado pergunta-lhe qual a sua próxima paragem, e com base nessa informação vai calcular o trajecto mais curto para atingir essa estação (independentemente da plataforma de paragem).

Especificação da plataforma Durante a aproximação de um comboio à estação, o agente *Supervisor* notifica o *Estação* e pergunta-lhe sobre a plataforma e local de paragem indicados para a composição. Caso a estação não esteja localizada na respectiva área de supervisão, é necessário difundir o pedido aos agentes vizinhos que se encarregarão de o transmitir à estação.

Aproximação da plataforma Após ter sido especificada a plataforma de paragem, e encontrando-se o comboio no troço onde esta deve ser executada, o agente *Supervisor* fornece informação acerca do exacto local de paragem ao agente *Comboio* e espera que este calcule a correspondente força de travagem.

Fiscalização da paragem Após a definição do local de paragem dentro da plataforma o agente *Supervisor* mantém um papel fiscalizador sobre o movimento do comboio dentro da plataforma. Caso receba informação que a composição ultrapassou o limite previamente definido vai automaticamente ordenar-lhe a paragem.

Paragem Quando o comboio se imobiliza dentro da plataforma de paragem, o agente *Supervisor* efectua a correspondente notificação aos agentes *Estação* e *Comboio* correspondentes. Esta informação será útil para os sistemas paralelos de informação aos passageiros.

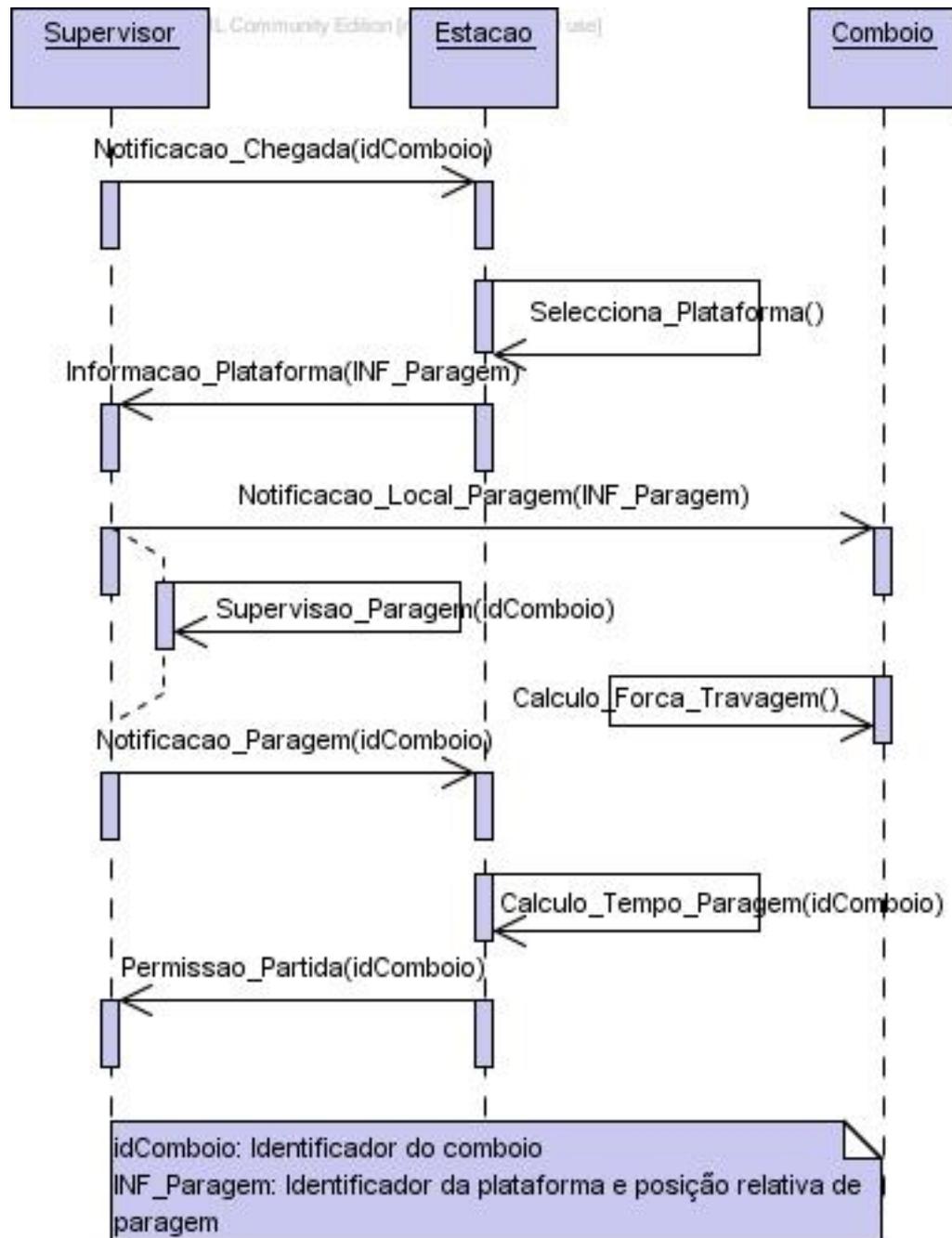


Figura 6.16: Entrada e saída de um comboio numa estação (Diagrama de sequência).

6.5.8.2 Permanência nas Estações

Após a notificação da paragem do comboio na plataforma, cabe ao agente *Estação* a definição do tempo mínimo de paragem, tendo por base o número de passageiros na respectiva estação.

Sendo, tal como referido no capítulo 3.2, requisito básico de um sistema CBTC a mo -ni -to -ri -za -ção da paragem dos comboios nas estações, neste caso será o agente *Estação* o responsável por tal processo. Quando este julgar apropriada a partida do comboio vai transmitir essa informação ao agente *Supervisor* da sua área.

Neste caso simulou-se o processo introduzindo uma aleatoriedade no cálculo do tempo de paragem de cada comboio. A probabilidade para cada valor segue uma distribuição normal tendo por base o tempo-padrão de paragem de um comboio.

6.5.8.3 Saída das Estações

Permissão de partida Após o cumprimento do tempo de paragem determinado pelo agente *Estação* este notifica o agente *Supervisor* de permissão de partida.

Calculo do trajecto O agente *Supervisor* pede ao agente *Comboio* informação sobre o próximo local de paragem. Caso a paragem acabada de cumprir não seja a última é calculado o trajecto ideal, comunicado ao agente *Comboio* e reiniciado o processo de aproximação a uma estação.

6.5.9 Transferência de Supervisor

A responsabilização de cada agente por uma parcela do mapa ferroviário, implica que cada comboio seja supervisionado por vários agentes durante a totalidade do seu percurso.

Nesta parte explica-se o processo de transferência de um comboio entre dois agentes supervisores, estando representado na figura 6.17 o diagrama de sequência do processo.

Cada agente *Supervisor* é responsável por informar os seus vizinhos (Secção 6.3.2) acerca dos comboios actualmente sob sua supervisão.

Esta necessidade é justificada por dois motivos:

Primeiramente por uma questão de segurança. Numa eventual falha do agente, e após o leilão (Secção 7.2.1) pela respectiva área de supervisão, é necessário que o agente vencedor saiba quais os comboios que adicionalmente terá de supervisionar.

Por outro lado constitui uma forma de avisar os agentes vizinhos da provável eminência de supervisão de um novo comboio. Caso o comboio não termine a sua viagem dentro da área de supervisão terá sempre que seguir para a área de um agente vizinho. Desta forma,

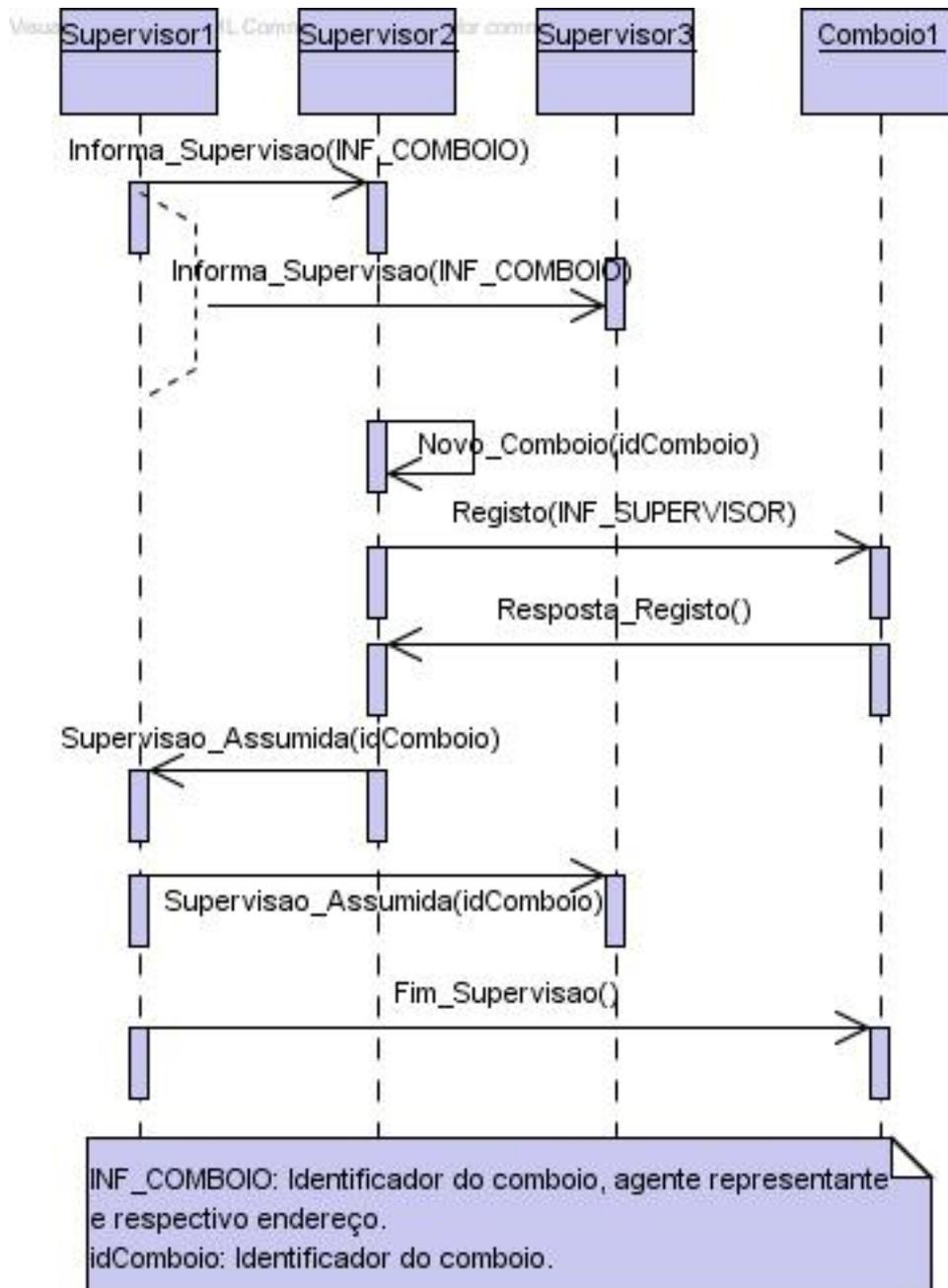


Figura 6.17: Transferência entre agentes supervisores (Diagrama de sequência).

ao enviar previamente informação acerca do nome e endereço do agente que representa o comboio, potencia-se a transferência com maior rapidez.

Ao receber informação sobre a provável entrada de um novo comboio na sua zona de supervisão, cada agente vizinho vai guardar essa informação numa estrutura específica para o efeito.

Quando um *Supervisor* detecta a entrada do comboio na sua zona de supervisão, efectua a ligação ao respectivo agente e, caso tenha sido bem sucedido, notifica o *Supervisor* anterior.

Cabe aos dois agentes supervisores notificar os vizinhos da alteração entretanto efectuada.

O agente que está prestes a deixar a supervisão do comboio deve notificar os seus vizinhos (excepto o que recebeu a supervisão) indicando que o comboio já está supervisionado por outro agente. Estes vão eliminar a informação que tinham guardada acerca do comboio, uma vez que em princípio já não será necessária.

O agente receptor da supervisão reinicia o processo, isto é, notifica os seus vizinhos (entre os quais o que terminou anteriormente a supervisão) da supervisão que efectua sobre o comboio.

Veja-se um exemplo relativo à transferência de supervisão de um comboio:

O comboio C_1 está num troço sob supervisão exclusiva de S_1 que tem como agentes vizinhos S_2 e S_3 (Figura 6.18).

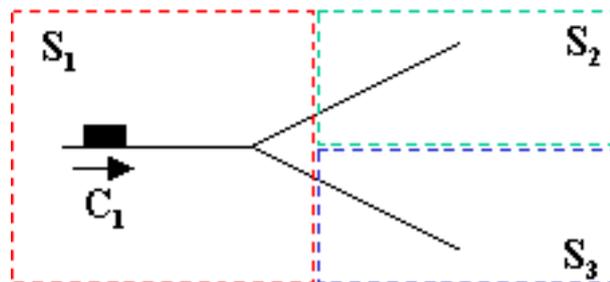


Figura 6.18: Comboio sob supervisão exclusiva.

O agente supervisor S_1 notifica S_2 e S_3 acerca do comboio C_1 . Estes ficam com a informação que C_1 será um comboio que potencialmente poderá entrar na respectiva área de supervisão e guardam o nome e endereço do agente que o representa.

Imagine-se que o trajecto previsto para C_1 passa através da zona supervisionada por S_2 . Após detectar a entrada na respectiva zona consulta a informação previamente guardada de C_1 e efectua a ligação ao agente representante (Figura 6.19).

É agora responsável por informar os seus vizinhos da alteração efectuada. Vai-lhes enviar uma mensagem a indicar que potencialmente irão supervisionar o comboio C_1 . Para S_1 envia adicionalmente informação a indicar que assumiu a supervisão do comboio.

Finalmente S_1 termina a ligação a C_1 indicando-lhe o final da competência de supervisão.

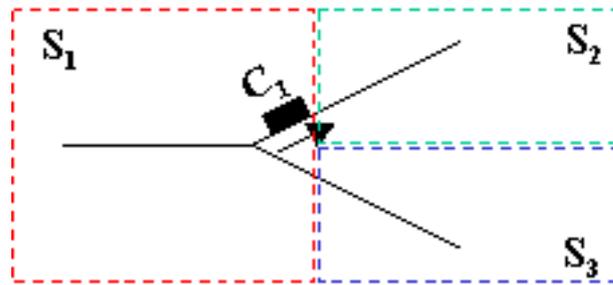


Figura 6.19: Entrada na zona de transferência.

Informa os agentes vizinhos que não receberam a supervisão do comboio da alteração entretanto efectuada.

6.5.9.1 Falha na Transferência

Caso o processo de transferência não seja bem sucedido é necessário garantir a segurança das operações ferroviárias, nomeadamente através do despoletar do modo de emergência. Durante o processo de transferência existem três situações que o podem causar:

1. Transferência não efectuada em tempo útil. O processo de transferência é sempre efectuada num troço comum a dois supervisores. Caso o comboio esteja prestes a chegar ao término do troço (Localizado a uma distância inferior a ξ , (> 0)) e o receptor da supervisão ainda não tiver efectuada a notificação.
2. Quando o agente receptor da supervisão não consegue efectuar uma ligação ao agente representante do comboio.
3. Quando em algum momento o agente representante do comboio não tiver uma ligação a um agente supervisor.

Conclusão

Apresentaram-se os principais aspectos referentes à implementação do sistema, designadamente os critérios de atribuição de prioridade e alteração temporária de objectivos por parte dos comboios circulantes na rede. Descreveu-se o método de distribuição da competência de supervisão pelos respectivos agentes e a noção de vizinhança daqui resultante. Apresentou-se a noção de "Conflito" e descreveram-se os critérios utilizados para a sua resolução, (benefício global, análise de trajectos, valor de utilidade e distância a percorrer), bem como a ordem pela qual são considerados.

Capítulo 7

Características Avançadas do Sistema

Neste capítulo descrevem-se as características mais específicas do sistema. Aquelas que, derivadas das noções de agente e sistema multi-agente, produzem comportamentos que de alguma forma diferenciam o sistema proposto dos restantes. São detalhados os processos de cooperação entre agentes, de aprendizagem e tolerância a falhas.

7.1 Processos de Cooperação entre Agentes

Tal como referido anteriormente, uma das características mais relevantes de um sistema multi-agente reside na interacção estabelecida entre os seus componentes. Podem ser estabelecidas relações de cooperação, competição ou antagonismo, com o objectivo de facilitar a resolução de um problema.

No sistema MARCS os agentes supervisores estabelecem processos cooperativos que podem ser divididos em duas categorias:

Troca de Informação Consiste no envio aos agentes vizinhos de informação sobre o tráfego na respectiva zona. Esta terá utilidade no controlo do tráfego, permitindo uma visão mais global no processo de atribuição de prioridades.

Troca de Planos É expressa através da transmissão de objectivos entre agentes vizinhos. Estes objectivos, da forma "manter livre o melhor caminho entre o troço T_i e a estação E_j ", permitem a elaboração de planos conjuntos e acções concertadas por forma a propiciar a fluidez do tráfego.

7.1.1 Troca de Planos

Situações específicas, como a existência de comboios com índice de prioridade máximo, podem levar o *Supervisor* a elaborar um objectivo da forma:

”Manter o melhor caminho entre o troço T_i e a estação E_j livre”

Como provavelmente ele não é responsável pela supervisão de todo o trajecto, o objectivo é transmitido sob a forma de plano aos agentes vizinhos, com o objectivo que estes também o adoptem e se possa falar em ”**acção concertada**”.

Ao receber informação sobre o plano de um vizinho cada *Supervisor* tem autonomia para:

Ignorar o plano Caso a estação destino e o caminho para a atingir não estejam sob sua supervisão, o agente pode simplesmente ignorar a informação recebida.

Rejeitar o plano Quando existem comboios nos troços que deveriam estar desimpedidos ou quando os trajectos desses comboios os irão interceptar, o agente pode rejeitar a colaboração no plano previsto e efectuar a correspondente notificação do agente emissor.

Aceitar o plano Caso não resultem dificuldades acrescidas para o tráfego na própria área o agente vai reajustar os processos de controlo por forma a satisfazer o plano recebido. Neste caso o agente deve notificar o emissor do plano da concordância na participação.

7.1.2 Troca de Informação

A troca de informação entre agentes supervisores consiste no envio de informação sobre o tráfego circulante na área de cada supervisor. Esta pode manifestar-se de três formas distintas:

1. Informação sobre a assumpção ou término de supervisão.
2. Informação acerca da localização de comboios em zonas específicas.
3. Informação sobre congestionamentos de tráfego.

7.1.2.1 Assumpção ou Término de Supervisão

Sempre que um comboio entra numa nova zona, o respectivo supervisor notifica todos os agentes vizinhos da ocorrência. Este facto tem justificação sob duas perspectivas: segurança e eficiência.

Se cada supervisor tiver informação sobre os comboios e respectivos agentes representantes que os seus vizinhos supervisionam, não existem supervisores imprescindíveis no processo de controlo de tráfego.

Tal como detalhado na secção 7.2.1, se um qualquer agente supervisor deixar de funcionar, os seus vizinhos podem rapidamente decidir quem tem mais disponibilidade para assumir

uma nova zona e iniciar a supervisão do tráfego aí circulante, uma vez que têm informação sobre todos os agentes com os quais necessitam de interagir.

As razões de eficiência prendem-se com o processo de transferência de supervisão de um comboio. Uma vez que antes de entrar numa zona o agente supervisor já contém informação sobre o agente que o representa e as respectivas características, torna-se mais rápido o processo de registo e início da supervisão.

7.1.2.2 Tráfego em Zonas Específicas

Este é um aspecto relevante para a funcionalidade do sistema. Em situações como a ilustrada na figura 7.1, é necessário que os dois supervisores troquem informação sobre o tráfego circulante na área do outro.

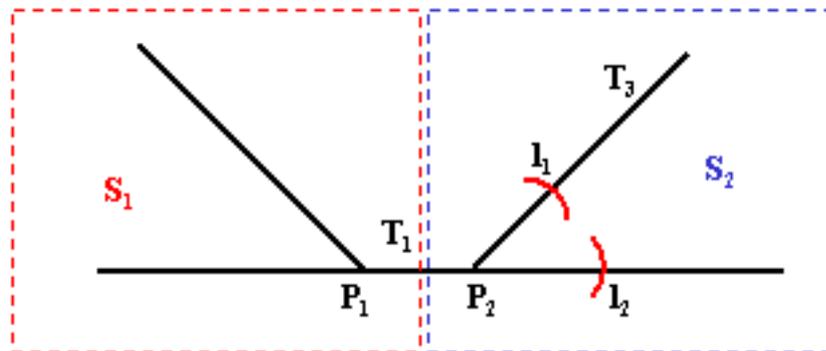


Figura 7.1: Troca de informação sobre o tráfego.

Sempre que um ponto de selecção está localizado perto do limite de uma área, é necessário que o agente que o supervisiona peça a colaboração do vizinho para efectuar o seu controlo.

Veja-se a situação ilustrada na figura 7.1. O agente S_1 partilha a supervisão de T_1 . No entanto este troço não tem comprimento suficiente para permitir o controlo seguro do ponto P_1 .

Neste caso S_1 vai pedir a S_2 para ser informado sempre que existirem comboios a distância α de P_2 (l_1 e l_2), por forma a poder antecipar as acções necessárias em relação ao ponto P_1 . Assim sendo, um comboio que circule no troço T_3 apesar de estar sob supervisão exclusiva de S_2 pode determinar a forma de resolução de um conflito em P_1 .

7.1.2.3 Congestionamentos

Cada *Supervisor* deve avisar os seus vizinhos da ocorrência e resolução de conflitos na sua área. Esta informação será tida em conta na resolução dos próprios conflitos e no encaminhamento das composições dentro da respectiva área.

Resolução dos Próprios Conflitos

Sempre que é necessário resolver uma situação de conflito entre comboios, o *Supervisor* leva em conta o volume de tráfego nas zonas vizinhas no processo de atribuição de prioridade.

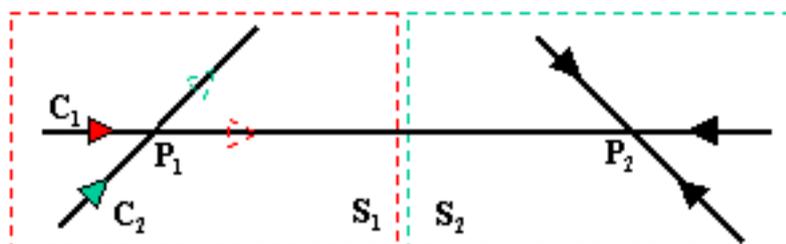


Figura 7.2: Informação de conflito.

Tome-se o exemplo da figura 7.2. Mesmo que, pela comparação dos valores de utilidade, a prioridade deva ser atribuída a C_1 , tal não acontecerá, uma vez que apenas iria atrasar a resolução do conflito em P_2 .

Ao analisar o conflito de P_1 , e estando os comboios aparentemente em igualdade de circunstâncias, o *Supervisor* S_1 deve analisar também a situação numa área, que não a sua, no processo de decisão de um conflito.

Encaminhamento de Composições

Ao receber informação da existência de congestionamentos em parcelas da rede, o *Supervisor* deve reanalisar o trajecto definido para cada uma das composições da sua área. O objectivo é avaliar se não existirá um trajecto alternativo que possa momentaneamente ser vantajoso para a composição.

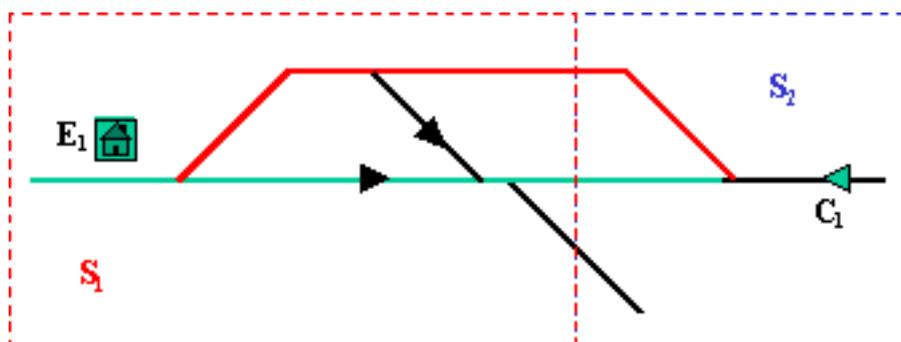


Figura 7.3: Informação de conflito - Reencaminhamento.

A figura 7.3 ilustra um caso em que será mais vantajoso para o comboio C_1 seguir por outro trajecto em direcção à estação E_1 . O trajecto inicialmente planeado (Verde), apesar de mais curto, representa mais tempo de espera para a composição em relação ao alternativo (Vermelho). Cabe ao agente que supervisiona o comboio (S_2) redefinir o seu trajecto, em função da informação recebida de S_1 .

7.1.3 Repercussões

A realização de processos cooperativos apenas será benéfica se dessa interacção resultarem alterações no comportamento individual de cada agente. É necessário que cada um altere os seus processos de controlo de tráfego no sentido de satisfazer um plano pedido, ou em resultado de informação transmitida por um vizinho.

No sistema MARCS esta alteração de comportamento manifesta-se na alteração do valor da variável "Momento" (Secção 6.2.2.3) associada a cada troço ferroviário.

Esta variável é inicializada com valor "1" para todas as arestas do grafo que representa a rede, funcionando como factor de ponderação ao custo de circular pelo respectivo troço.

Desta forma, o aumento do seu valor influencia directamente o custo de percorrer o troço, evitando a sua utilização e propiciando a definição de trajectos alternativos.

7.1.3.1 Exemplo

Veja-se um exemplo (Figura 7.4) relativo à troca de planos entre agentes supervisores e consequentes repercussões no processo de controlo de tráfego.

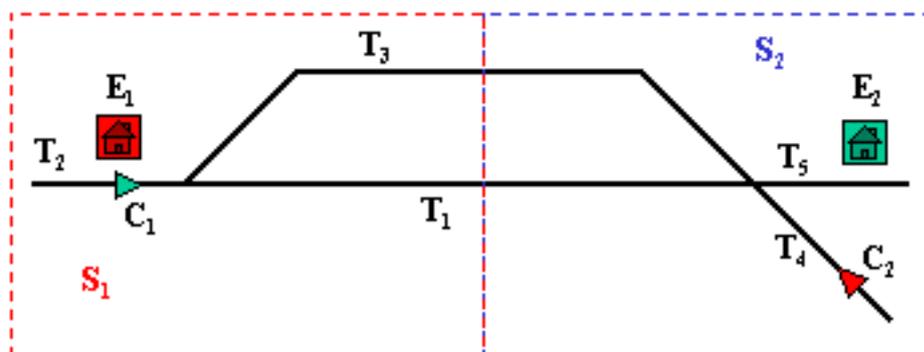


Figura 7.4: Exemplo de cooperação.

Os agentes supervisores S_1 e S_2 , por partilharem a supervisão dos troços T_1 e T_3 , são considerados vizinhos existindo um canal de comunicação entre ambos.

Apesar de possuírem visão parcial do tráfego da rede, ambos mantêm uma estrutura (grafo) com a sua topologia global. Esta estrutura é utilizada no encaminhamento das composições dentro das respectivas áreas, tendo no estado inicial os valores apresentados na figura 7.5.

Sendo S_1 responsável pela definição do trajecto para o comboio C_1 (prioridade máxima) irá calcular o trajecto ideal para atingir a próxima paragem prevista (E_2). Este consiste na sequência de troços $T_2 \rightarrow T_1 \rightarrow T_5$.

Com o objectivo de minimizar o tempo de espera para C_1 , S_1 propõe a S_2 um plano da forma:

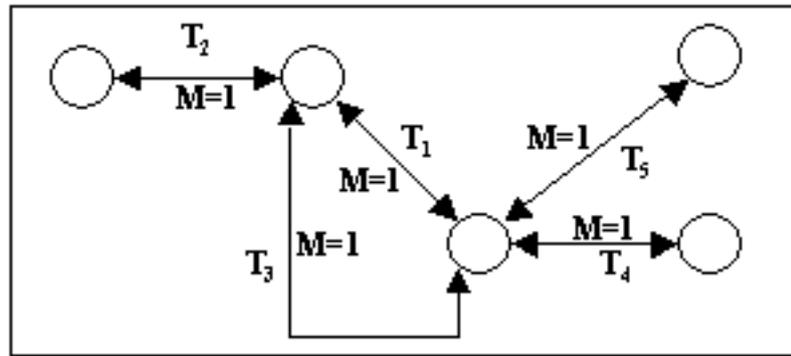


Figura 7.5: Estado inicial do grafo.

”Manter o melhor trajecto entre o troço T_2 e a estação E_2 livre”

Caso este plano não fosse comunicado ao supervisor vizinho e surgisse um comboio no troço T_4 com destino a E_1 seria calculado um trajecto que entraria em conflito com o do comboio C_1 .

Quando S_2 recebe informação sobre o plano do vizinho vai aumentar o valor de ”Momento” para os troços em questão (T_2 , T_1 e T_5) passando o grafo a ter o aspecto da figura 7.6.

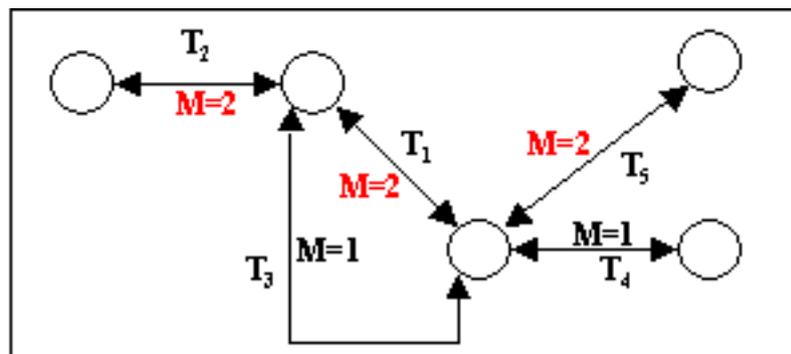


Figura 7.6: Estado final do grafo.

A partir deste momento, o custo de percorrer o troço T_1 aumentará e, na existência de alternativas, não será escolhido no trajecto ideal para atingir E_1 , encaminhando-o através do troço T_2 .

O ajuste do valor de ”Momento” significou para o supervisor S_2 a concordância e participação com o plano recebido de S_1 .

7.2 Tolerância a Falhas

Num sistema de controlo é bastante importante a capacidade de tolerar falhas ou situações excepcionais. Além do tratamento eficiente das situações para as quais foi planeado, é

relevante a sua capacidade para lidar com situações imprevistas. É importante que o sistema seja capaz de se adaptar de forma eficiente à ocorrência de situações excepcionais, minimizando a degradação do seu desempenho e garantindo a segurança até que a normalidade seja reposta.

Um sistema multi-agente, associado a noções de descentralização, distribuição ou autonomia pode apresentar vantagens relativamente aos sistemas tradicionais. Cada agente deve possuir conhecimento para, de forma autónoma, se adaptar a falhas ou situações excepcionais.

7.2.1 Falha num Agente Supervisor

No sistema MARCS cada agente supervisor tem um papel fulcral na regulação do tráfego numa determinada parcela da rede ferroviária. É importante garantir que falhas num elemento possam ser colmatadas pelos restantes sem que o desempenho do sistema se deteriore.

Cada agente supervisor está ligado a cada um dos seus vizinhos, isto é, àqueles com os quais partilha a supervisão de troços de linha. Estas ligações servem para troca de informação relativa ao tráfego e à sua transferência entre zonas.

Quando, por qualquer razão, ocorre uma falha num dos agentes supervisores, a ligação é quebrada sendo este evento perceptível aos agentes vizinhos.

Torna-se competência dos agentes vizinhos a participação num processo de leilão pela posse da área entretanto sem supervisor. A figura 7.7 mostra o diagrama de sequência relativo à falha num agente supervisor e respectivas acções associadas.

7.2.1.1 Leilão

A ideia subjacente ao processo de leilão consiste em encontrar o agente com menor volume de trabalho. Este será o que, potencialmente estará em melhores condições para assegurar a supervisão de uma nova zona.

É igualmente importante que o leilão se conclua no mais curto espaço de tempo possível, uma vez que a área a leilão está momentaneamente sem supervisor. Esta é a razão pela qual o processo consiste na "abertura de envelope" e selecção do vencedor (leilão fechado).

Os agentes trocam informação acerca da dimensão das zonas por si supervisionadas e do volume de tráfego associado enviando aos restantes um valor numérico indicatriz da ocupação de cada um.

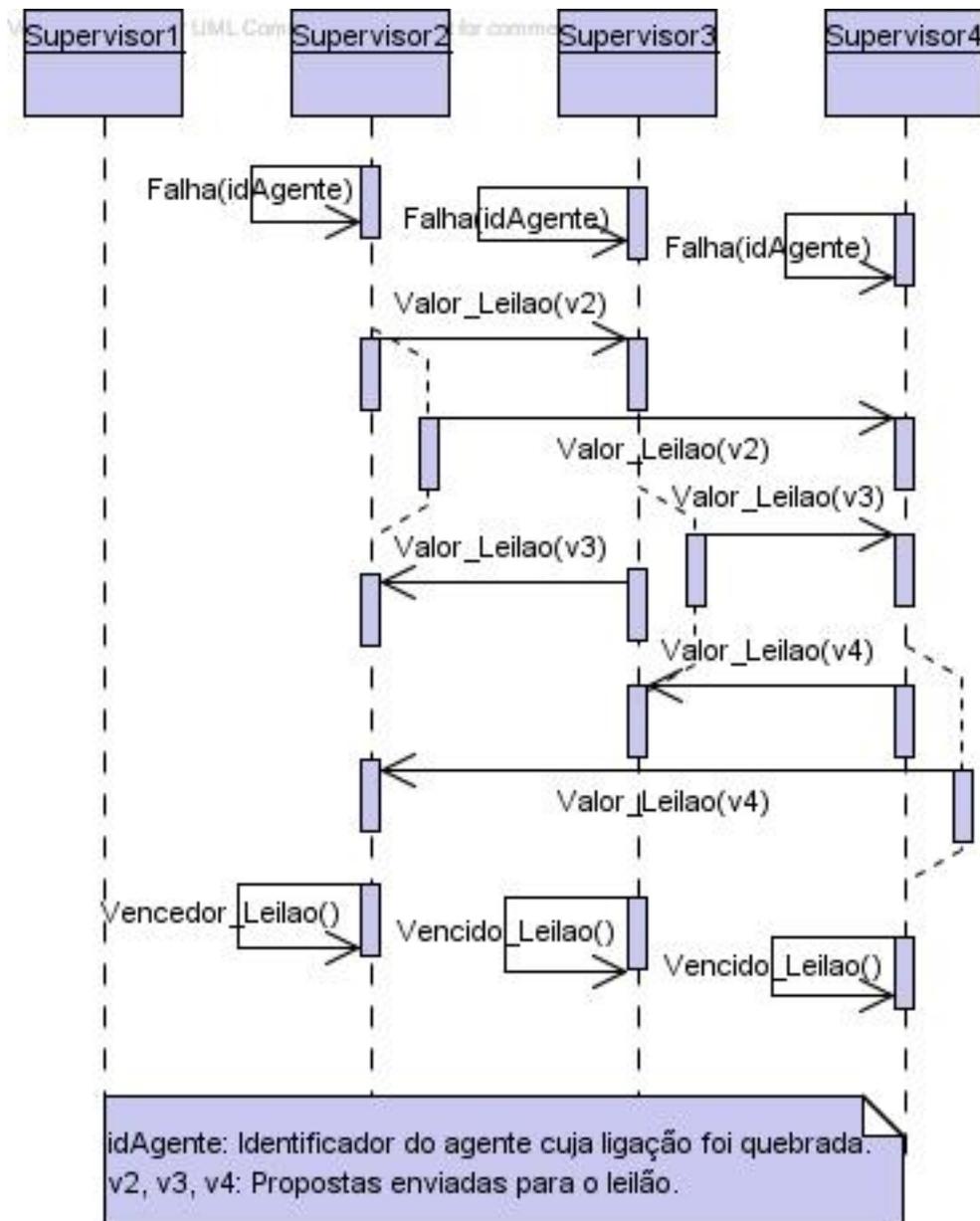


Figura 7.7: Falha num agente supervisor (Diagrama de sequência).

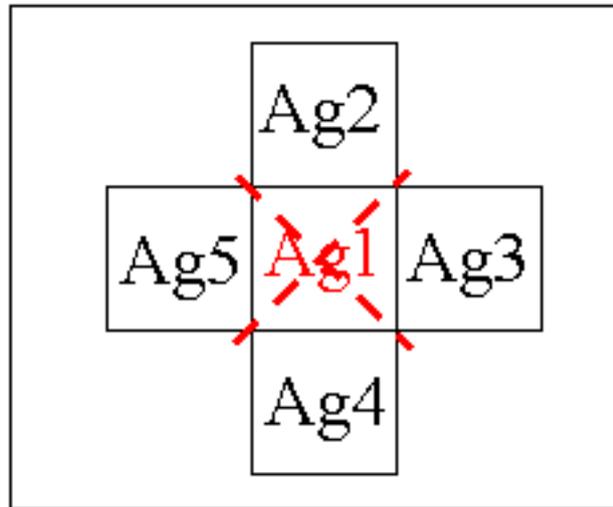


Figura 7.8: Falha num agente supervisor.

7.2.1.2 Valor da Zona

Sejam C , T e A respectivamente o total de comboios, total de troços e de agulhas de selecção sob a supervisão do agente. Sejam p_1 , p_2 e $p_3 \in \mathbb{R}$.

O valor final Ind_{Zona} , representando o volume de trabalho de cada supervisor, é dado pela expressão:

$$Ind_{Zona} = p_1 \times C + p_2 \times T + p_3 \times A$$

7.2.1.3 Exemplo

A figura 7.8 ilustra a existência de uma falha no agente supervisor $Ag1$. A ligação estabelecida a cada um dos seus vizinhos é quebrada, sendo este evento perceptível por cada um.

Após detectar a falha, os supervisores vão iniciar um processo de leilão pela recepção da zona anteriormente supervisionada pelo agente $Ag1$.

Cada supervisor contém informação acerca dos seus vizinhos (endereço e porto de escuta de ligações) e, relevante para este caso, informação acerca da vizinhança de cada um dos seus vizinhos. Neste exemplo, $Ag2$, $Ag3$, $Ag4$ e $Ag5$ sabem que $Ag1$ é seu vizinho e sabem que este tem como vizinhos $Ag2$, $Ag3$, $Ag4$ e $Ag5$. Desta forma, ao detectar uma falha na comunicação com o agente $Ag1$ cada um sabe quais os parceiros de negociação no leilão pela supervisão da área entretanto livre.

Cada agente efectua uma ligação aos vizinhos de $Ag1$, sendo estas as seguintes:

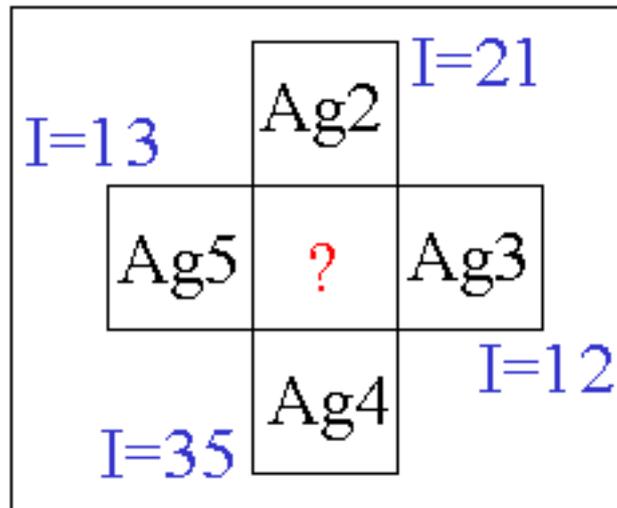


Figura 7.9: Índices associados a cada agente vizinho.

Agente Ag2 → Agentes Ag3, Ag4 e Ag5.

Agente Ag3 → Agentes Ag2, Ag4 e Ag5.

Agente Ag4 → Agentes Ag2, Ag3 e Ag5.

Agente Ag5 → Agentes Ag2, Ag3 e Ag4.

Note-se que alguns destes agentes poderão já estar ligados entre si, no caso partilharem a supervisão de troços de linha, isto é, serem vizinhos.

Suponha-se que, pelo processo anteriormente descrito, cada agente supervisor calcula os índices indicados na figura 7.9.

A fase seguinte consiste na troca desta informação entre todos, por forma a permitir a avaliação da melhor proposta. Após a recepção de todos os valores, o leilão é dado por encerrado e selecciona-se o vencedor.

Cada agente pode imediatamente assumir-se como vencedor ou vencido no processo de leilão, devendo em qualquer dos casos efectuar as actualizações correspondentes.

Caso não seja autor da proposta mais baixa, basta actualizar a informação acerca da sua vizinhança, enquanto o vencedor terá adicionalmente que reconstruir a estrutura que representa a sua área de supervisão e notificar o agente facilitador da alteração entretanto efectuada.

No exemplo apresentado Ag3 é o que menor volume de trabalho tem, passando a assumir a supervisão da área anteriormente assegurada por Ag1.

A figura 7.10 mostra o novo estado de cada agente. Ag3, vencedor do leilão, assume agora uma área de supervisão maior, e verá o seu índice de ocupação incrementado o que o deixará

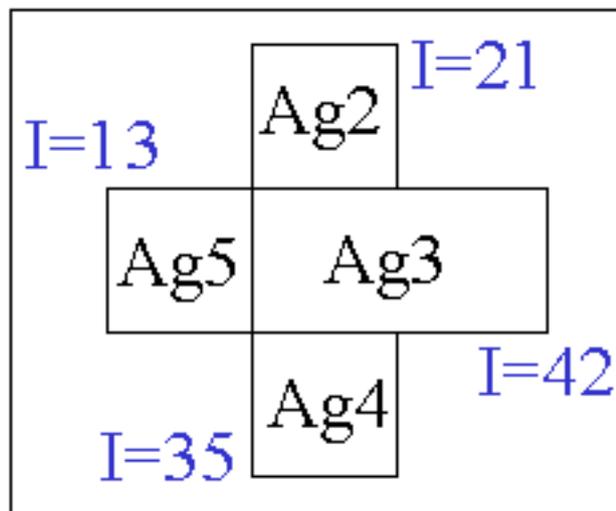


Figura 7.10: Nova situação de supervisão.

em posição de desvantagem para os próximos leilões. Esta é uma situação desejável no sentido de evitar demasiadas discrepâncias entre a dimensão das áreas de supervisão de cada agente.

Como forma de desempate, na eventualidade de dois ou mais agentes apresentarem propostas (valores) iguais, o vencedor do leilão será aquele cujo identificador for menor, após ordenação alfabética. Como não são permitidos agentes com nome igual, este critério garante uma decisão determinística na selecção do vencedor de qualquer leilão.

7.2.2 Avarias em Troços de Linha

As redes ferroviárias, particularmente as de maior extensão, estão potencialmente expostas à ocorrência de situações imprevistas (Avarias mecânicas, quedas de objectos sobre os carris, situações climatéricas adversas, manifestações de cariz político, etc...).

É necessário que cada agente supervisor saiba ajustar o seu modo de actuar em função do estado momentâneo da rede e dos recursos disponíveis. O cálculo do trajecto para cada comboio deve ser feito tendo em conta as secções de linha disponíveis, e poderem ser encontradas alternativas caso estas se alterem.

A figura 7.11 mostra o diagrama UML relativo à ocorrência de uma falha num troço da rede ferroviária. Cada agente supervisor é notificado pelo agente facilitador da ocorrência de uma falha, tornando-se responsável pela reavaliação do trajecto definido para cada comboio sob sua supervisão.

Se existir necessidade de proceder a alterações estas são comunicadas ao respectivo agente *Comboio* (Comboio1), caso contrário, este nem se aperceberá da ocorrência da anomalia (Comboio2).

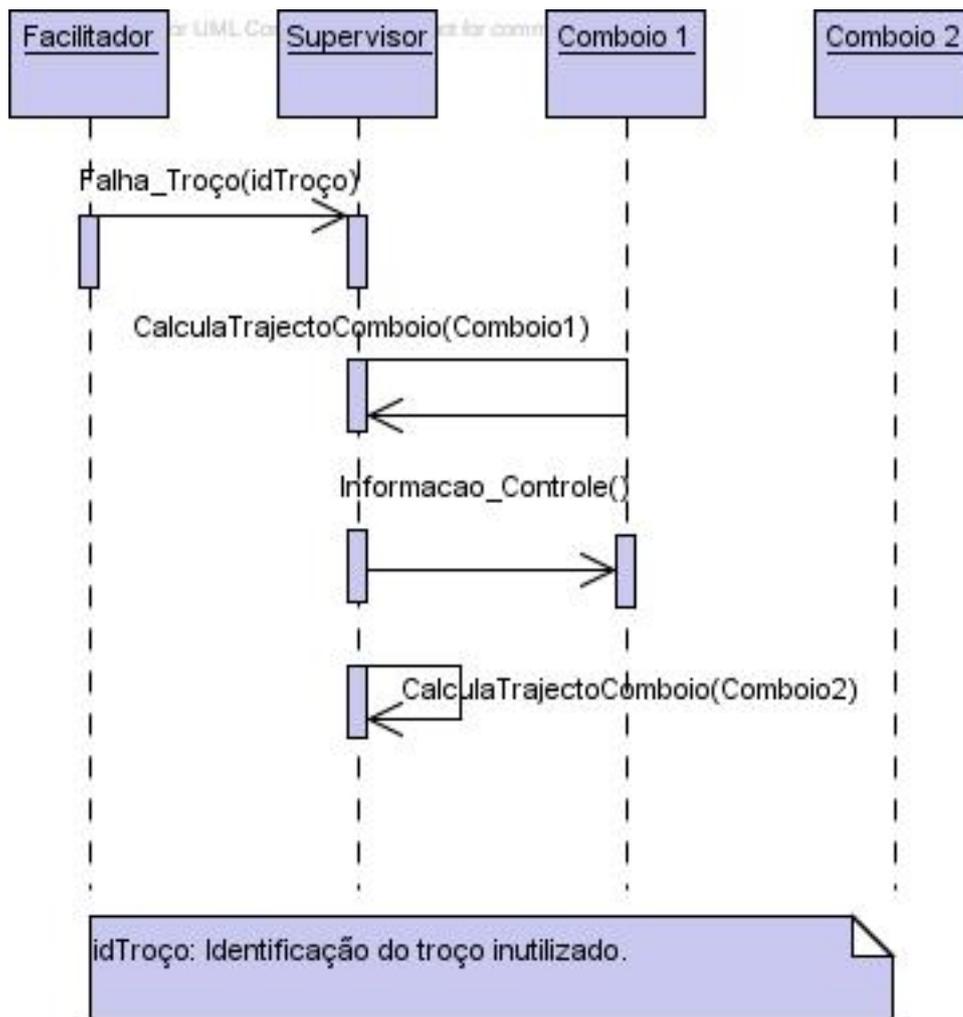


Figura 7.11: Falha num troço de linha (Diagrama de sequência).

7.3 Aprendizagem

Existe um tipo de agentes (Aprender) cuja tarefa consiste em analisar a actividade dos agentes supervisores e inferir possíveis regras de optimização do tráfego circulante na respectiva zona. Estas são depois transmitidas aos agentes supervisores para que estes possam adaptar os processos de controlo do tráfego.

7.3.1 Analogia

Esta secção tem por objectivo, através de uma analogia, mostrar o essencial do sistema de aprendizagem, a ideia primária que o originou e as repercussões esperadas para o sistema de controlo.

Imagine-se um espaço com 2 robôs (R_1 e R_2) com o objectivo de se deslocarem respectiva-

mente para D_A e D_B (Figura 7.12a), dispondo apenas de um trajecto possível tal.

Caso se desloquem à mesma velocidade, vão atingir os estados relativos às figuras 7.12b e 7.12c. Como a passagem de que ambos necessitam apenas permite a circulação de um robô de cada vez, cria-se um conflito de interesses cuja resolução passa obrigatoriamente pelo recuo de um elemento por forma a permitir a passagem do outro.

O robô que cedeu a prioridade só depois poderá atravessar a passagem para atingir o seu objectivo (Figura 7.12d).

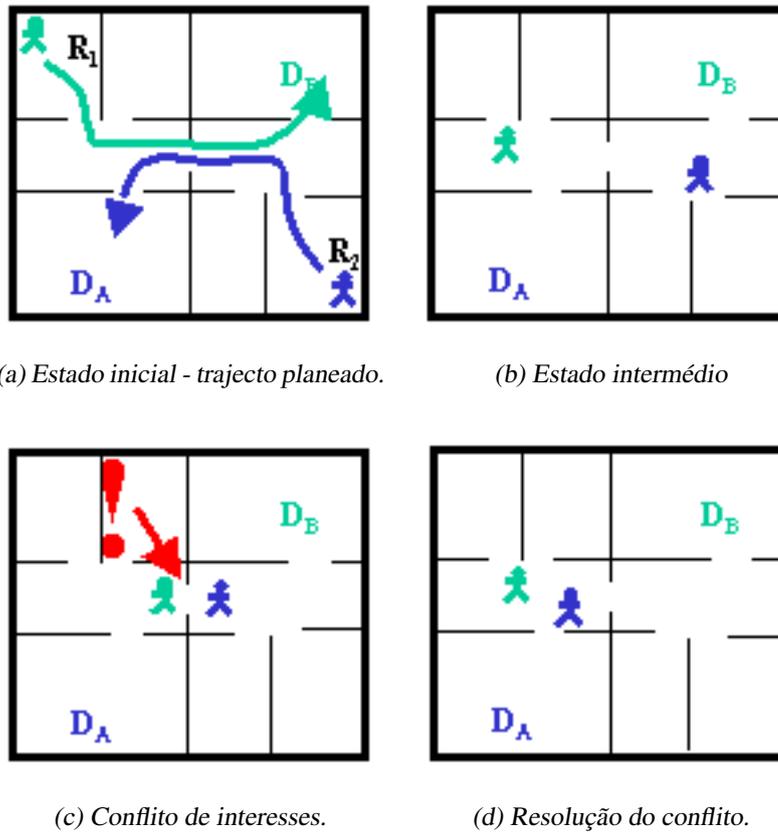
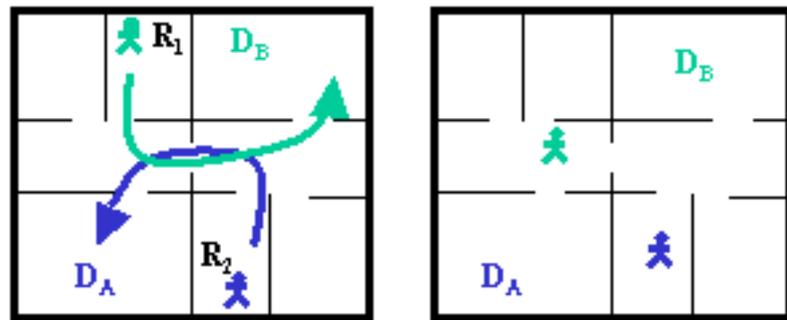


Figura 7.12: Conflito de interesses.

Esta forma de resolver o conflito de interesses, apesar de funcional, não é óptima uma vez que obriga à paragem de ambos os robôs e à inversão de sentido de um deles com consequentes custos ao nível de tempo.

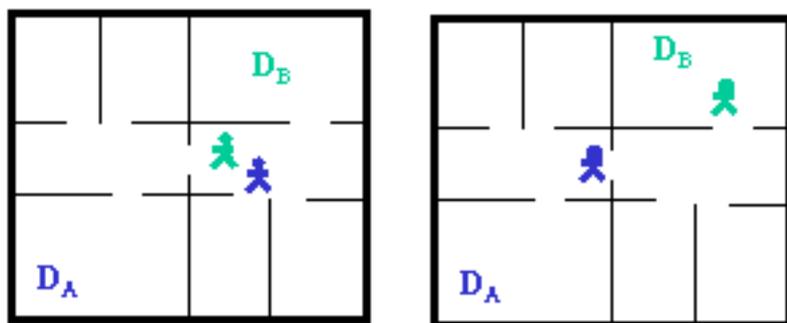
Imagine-se que, algum tempo depois, surge a situação ilustrada na figura 7.13a. Apesar de análoga à anterior, se ambos os elementos se deslocarem a velocidade semelhante vão originar um conflito similar ao ocorrido na figura 7.12c.

Uma solução passaria por mandar acelerar o passo a um dos elementos (P_1), abrandar o do outro (P_2) por forma a permitir a passagem de ambos sem que, no mesmo momento, o outro também a necessite (Figuras 7.13b , 7.13c e 7.13d).



(a) Estado inicial - trajecto planeado.

(b) Antecipação do conflito



(c) Passagem do primeiro elemento.

(d) Passagem do segundo elemento

Figura 7.13: Antecipação do conflito de interesses.

É exactamente esta a ideia subjacente ao sistema de aprendizagem. Com base no historial de actividade, pretende-se prever a ocorrência de conflitos de interesses, permitindo que o sistema de controlo possa tomar as acções adequadas para os evitar. Desta forma potencia-se a fluidez do tráfego, minimizando-se a necessidade de paragens e inversão em resultado de conflitos.

7.3.2 Recepção da Informação

Imediatamente após a criação, o agente *Aprender* regista-se no *Simulador (Facilitador)* e pergunta-lhe pelos agentes supervisores registados. Este devolve-lhe uma lista com a identificação e respectivo endereço de cada supervisor actualmente registado na rede.

O agente *Aprender* regista-se no *Supervisor* e pede a transmissão do respectivo registo de actividade. A figura 7.14 mostra o diagrama de sequência relativo à interacção entre os agentes neste processo.

7.3.2.1 Registo de Actividade

O registo de actividade de um agente consiste num ficheiro de texto onde estão descritas todas as mensagens recebidas e enviadas pelo agente. No caso dos agentes supervisores, são adicionalmente registadas as acções relativas à supervisão do tráfego.

Esta informação é guardada num ficheiro com o nome do agente e extensão ".log", com o seguinte formato:

```

AGENTE <NOME_AGENTE>    DATA    <DATA>
LINHA_REGISTO 1
LINHA_REGISTO 2
...
LINHA_REGISTO N

```

Cada linha de registo tem o seguinte formato:

```
<TEMPO> <TIPO> <DESCRIÇÃO>
```

TEMPO Instante em que a linha foi registada.

TIPO Tipo da linha. Pode ser relativa a uma mensagem recebida (RECEBE), uma mensagem enviada (ENVIA), uma acção (PROCESSA) ou uma excepção (EXCEPÇÃO).

DESCRIÇÃO Corresponde ao conteúdo da mensagem enviada ou recebida ou à descrição da acção correspondente.

7.3.2.2 Transmissão da Informação

Após receber o pedido relativo à transmissão do registo de actividade, o agente *Supervisor* envia o conteúdo do ficheiro com o seu nome e extensão ".log" através do *socket* que liga os dois agentes.

O agente *Aprender* activa o modo de transmissão de ficheiros e toda a informação recebida proveniente do supervisor é colocada num ficheiro com o mesmo nome.

Desta forma efectua-se a transmissão de ficheiros entre agentes (Secção B.2.4).

O caracter £ indica o final da transmissão do ficheiro, pelo que o agente *Aprender* pode reactivar o modo de transmissão de mensagens.

7.3.3 Pré-Processamento

Imediatamente após a transmissão do ficheiro de registo de actividade pode-se iniciar o processo de aquisição de conhecimento.

A fase inicial consiste no pré-processamento da informação. Esta é feita em quatro etapas (Brazdil 2002):

Seleção de casos Seleção das linhas relevantes para o processo de aprendizagem.

Eliminação de atributos Seleção dos atributos relevantes para a aplicação do algoritmo de aprendizagem.

Discretização de valores Discretização dos valores correspondentes à posição absoluta (latitude, longitude) dos comboios.

Redução do número de casos Eliminação de instâncias para conjuntos de dados de dimensão elevada.

7.3.3.1 Seleção de Casos

Através da seleção de casos pretende-se eliminar informação irrelevante para o processo de aprendizagem. Sabendo o formato do ficheiro de registo de actividade, consiste em aplicar um filtro que apenas permite a passagem de dois tipos de linhas:

1. Localização de Comboios. Informação acerca da localização de cada comboio supervisionado pelo agente. Contém informação sobre a latitude, longitude, destino, posição relativa na rede e velocidade do comboio.
2. Conflito de Interesses. Esta linha corresponde à ocorrência de um conflito de interesses entre pelo menos dois comboios.

7.3.3.2 Eliminação de Atributos

Nesta fase procede-se à redução do número de atributos das instâncias remanescentes da etapa anterior. Consoante o tipo de linha (Localização ou Conflito) seleccionam-se os atributos relevantes.

1. Localização de Comboios. Neste tipo de linhas considerou-se relevante a identificação do comboio, a sua localização absoluta (latitude e longitude), destino e tempo de registo. Assim sendo, cada instância deste tipo contém cinco atributos.
2. Conflito de Interesses. Neste caso seleccionam-se os atributos relativos ao tempo do conflito, identificação do local e comboios intervenientes.

7.3.3.3 Discretização de Valores

A discretização de valores foi aplicada relativamente ao posicionamento absoluto de cada comboio. Este valor é expresso através de um par de números reais, respectivamente latitude e longitude.

Com o objectivo de facilitar a generalização das regras geradas pelo sistema, agrupam-se as localizações absolutas dos comboios. Cada grupo possui um valor (centro) de latitude e longitude e estabelece-se uma vizinhança de raio ϵ , (> 0), considerando-se que todos os valores pertencentes a esta vizinhança estão localizados no mesmo local.

Seja (α, β) um par de valores representativos da localização absoluta de um comboio (latitude, longitude), tal que:

$$(\alpha, \beta) = \{\alpha, \beta \in \mathfrak{R}\}$$

Defina-se d como a distância euclideana entre dois pares de valores:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Para determinar o grupo onde inserir uma nova instância é necessário compará-la com os anteriormente definidos e englobá-la num já existente ou criar um novo, tal como descrito no algoritmo 7.1.

Algoritmo 7.1 Discretização de valores

```

(x, y) ← (latitude, longitude)
for i = 1 to totalInstancias do
  (L1, L2) ← (Latitudei, Longitudei)
  if d((x, y), (L1, L2)) < RAI0_VIZINHANCA then
    AtribuirEtiqueta(i)
    RecalcularCentro(i)
  RETORNAR
  end if
end for
CriarNovaEtiqueta(totalInstancias + 1)

```

7.3.3.4 Redução do Número de Casos

Uma vez que o agente *Supervisor* recebe informação acerca da localização dos comboios em intervalos de um segundo, simulações demoradas originam um número bastante elevado de instâncias relativas à localização de comboios. Nestes casos aplica-se um filtro final às instâncias de localização por forma a aumentar o espaço temporal mínimo entre instâncias de localização sucessivas.

Consoante o total de valores disponíveis, pode-se estabelecer um espaço mínimo de x segundos, com $x \in \mathbb{N}$, por forma a, diminuindo o total de instâncias, potenciar a rapidez de execução do algoritmo de aprendizagem.

7.3.4 Extracção de Conhecimento

Após a fase de pré-processamento da informação, os dados estão numa estrutura de dados tabular, contendo apenas as instâncias e atributos considerados relevantes.

O algoritmo de aprendizagem escolhido - APRIORI (Agrawal Strikant 1994) - permite a descoberta de regras de associação em problemas do tipo *market-basket*.

Tendo um conjunto de instâncias agrupadas em transacções, o objectivo consiste em descobrir potenciais padrões nessas transacções, isto é, instâncias que normalmente aparecem juntas em cada transacção. O resultado final é expresso em regras de associação com o formato "SE {premissas} ENTÃO conclusão".

7.3.4.1 Construção de Transacções

Uma transacção consiste num conjunto de instâncias relacionadas por algum critério. Para o problema em questão, as instâncias relativas ao posicionamento dos comboios envolvidos num conflito são agrupadas na mesma transacção.

Por exemplo, o seguinte conjunto de instâncias:

ID	Tempo	Tipo	Descrição	Destino
1	t_1	Localização	Comboio 1 (x_3, y_3)	Destino 1
2	t_2	Localização	Comboio 2 (x_2, y_2)	Destino 1
3	t_2	Localização	Comboio 1 (x_1, y_1)	Destino 1
4	t_3	Conflito L_1	Comboio 1 2	
5	t_8	Localização	Comboio 3 (x_2, y_2)	Destino 1
6	t_8	Localização	Comboio 4 (x_1, y_1)	Destino 1
7	t_9	Conflito L_1	Comboio 3 4	

Daria origem às transacções:

Transacção	Items
T_1	1, 2, 3, 4
T_2	5, 6, 7

7.3.4.2 Suporte e Confiança

Seja r uma regra de associação tal que:

$$r = \{i_1, i_2, \dots, i_n\} \Rightarrow c$$

Seja I o conjunto de todos os items da regra r :

$$I = \{i_1, i_2, \dots, i_n, c\}$$

Seja t o número total de transacções e t_p o número de transacções onde ocorrem todas as premissas da regra r . O suporte s da regra r , $s \in [0..1]$, pode ser definido como:

$$s = \frac{t_p}{t}$$

Cada regra r tem confiança $conf$, $conf \in [0..1]$ tal que:

$$conf = \frac{Suporte(I)}{Suporte(I - c)}$$

7.3.4.3 Algoritmo APRIORI

Tendo o conjunto de transacções definido, parte-se para a fase de aplicação do algoritmo com o objectivo de identificar conjuntos frequentes.

Tome-se o exemplo referido em 7.3.4.1. O objectivo é descobrir que, normalmente, **dois comboios localizados em (x1,y1) e (x2,y2) com destino "Destino₁" vão entrar em conflito em L_1 .**

Da próxima vez que dois comboios com determinado destino estiverem sensivelmente nessa posição, será necessário evitar que venham a entrar em conflito. Para isso pode pedir-se que, temporariamente, aumentem ou diminuam a sua velocidade média desejada.

Variante TID do Algoritmo

Existe uma correspondência directa entre o número de vezes que é necessário percorrer a base de dados e o tamanho dos conjuntos frequentes passíveis de identificação. Este facto pode atrasar a execução do algoritmo em conjuntos de dados de elevada dimensão.

Por esta razão surgiu uma variante, designada APRIORI-TID, que permite a identificação de regras com apenas uma passagem pela base de dados. Apesar de ter como inconveniente o facto de gastar mais memória, permite a construção de regras de uma forma mais rápida.

No caso dos agentes *Aprender* o factor tempo não foi considerado demasiado importante, justificando-se a implementação da versão original descrita no algoritmo 7.2 (Jorge 2002).

Algoritmo 7.2 Algoritmo APRIORI.

```

 $F_1 \leftarrow \{ConjuntosFreq.Tamanho1\}$ 
for ( $k = 2; F_{k-1} \neq \emptyset; k++$ ) do
   $C_k \leftarrow GeraCandidatos(F_{k-1})$ 
  for all Transacções  $t$  em  $D$  do
     $C_t \leftarrow SubConjunto(C_k, t)$ 
    for all Candidatos  $c$  em  $C_t$  do
       $c.contagem \leftarrow c.contagem + 1$ 
    end for
  end for
   $F_k \leftarrow \{c \text{ em } C_k | c.contagem \geq SUPORTE\_MINIMO\}$ 
end for
RETORNA  $\bigcup F_k$ 

```

7.3.4.4 Geração de Regras

Tal como descrito anteriormente, o algoritmo identifica e retorna os conjuntos frequentes nas transacções, isto é, os elementos que, com confiança e suporte mínimos (Secção 7.3.4.2), aparecem juntos nas transacções existentes.

A fase seguinte consiste em identificar as potenciais regras de associação, da seguinte forma:

Seja I um conjunto frequente de tamanho n .

$$I = \{i_1, i_2, \dots, i_n\}$$

Pode-se construir o seguinte conjunto de n regras de associação:

$$R = \{\forall i \in I : (I - i) \Rightarrow i\}$$

No problema específico, são consideradas relevantes as regras de associação cujo consequente é uma instância do tipo "Conflito de Interesses", isto é:

$$r \in R : \{i_1, i_2, \dots, i_n\} \Rightarrow c, Tipo(c) = "Conflito"$$

7.3.5 Envio de Regras

A ultima etapa do processo de aprendizagem consiste no envio de regras aos supervisores, para que estes possam otimizar os seus processos de controlo.

A forma de cada regra transmitida é a seguinte:

```

SE
    ComboioA em  $(x_1 \pm \delta, y_1 \pm \delta)$ , DestinoA
E
    ComboioB em  $(x_2 \pm \delta, y_2 \pm \delta)$ , DestinoB
E
    ...
ENTÃO
    Conflito Locali Tempot ComboioA, ComboioB

```

O algoritmo 7.3 é respeitante ao processo de envio de regras aos agentes supervisores.

Algoritmo 7.3 Envio de regras

```

for  $i = 0$  to  $totalRegras$  do
     $R \leftarrow Regra_i$ 
     $T \leftarrow TotalPremissas(R)$ 
    for  $j = 0$  to  $totalSupervisores$  do
         $S \leftarrow Supervisor_j$ 
         $ENVIA \leftarrow FALSE$ 
        for  $k = 0$  to  $T$  do
            if  $PertenceZona(LocalPremissa, S)$  then
                 $ENVIA \leftarrow TRUE$ 
            end if
        end for
        if  $ENVIA == TRUE$  then
             $EnviaRegraSupervisor(Regra_i, Supervisor_j)$ 
        end if
    end for
end for

```

Conclusão

Espera-se que da incorporação das características atrás descritas resultem benefícios para a fluidez e segurança do tráfego circulante numa rede ferroviária, seja pela antecipação de conflitos ou pela redefinição dinâmica de trajectos. Descreveram-se detalhadamente os processos de aprendizagem, de tolerância a falhas e os mecanismos de cooperação entre agentes supervisores, nomeadamente a troca de informação e de planos.

No próximo capítulo ilustram-se cenários onde as características aqui descritas contribuem para a optimização do controlo de tráfego.

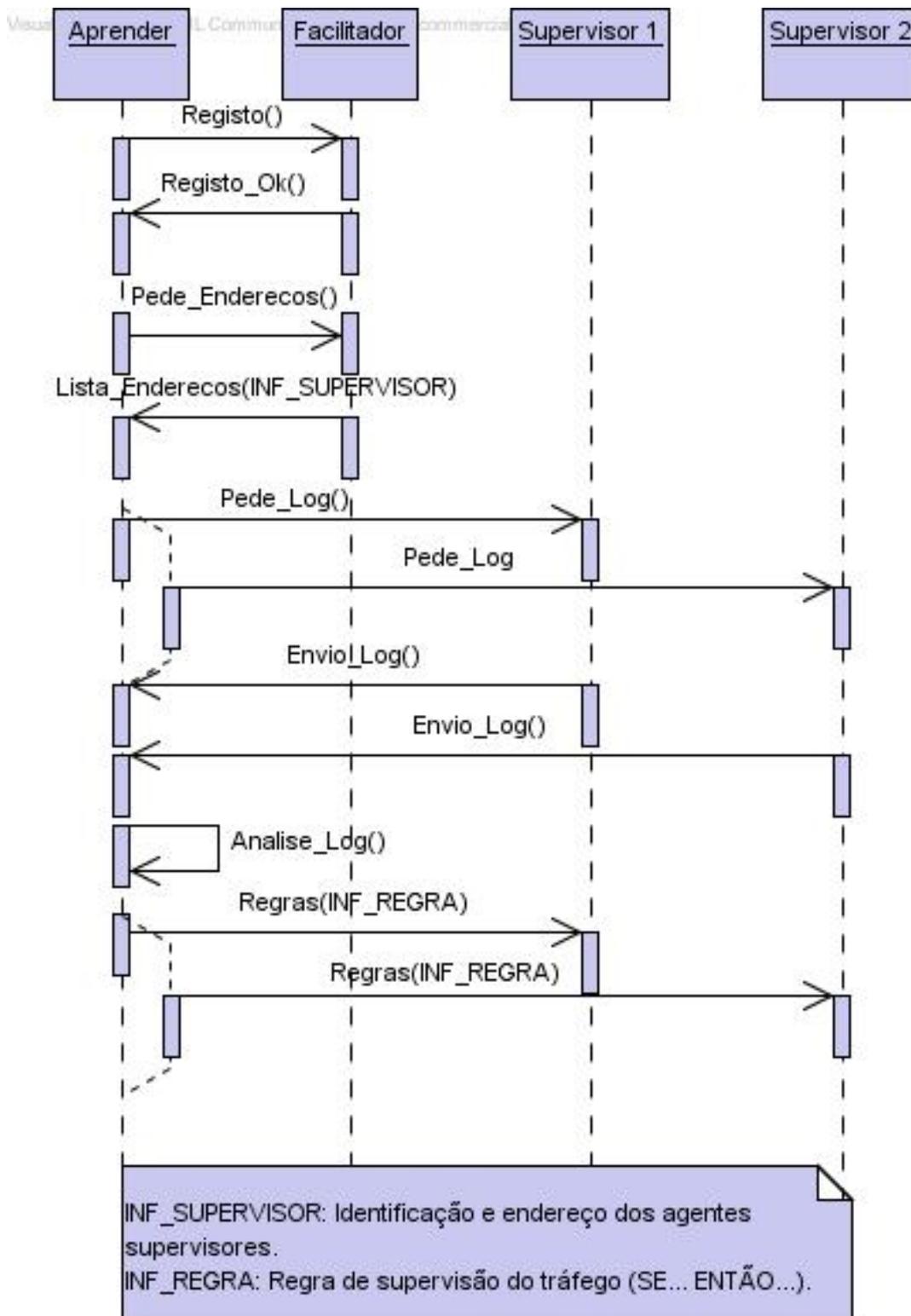


Figura 7.14: Diagrama de sequência do processo de aprendizagem.

Capítulo 8

Ilustração de Cenários, Avaliação e Conclusão

Neste capítulo apresentam-se cenários que ilustram o funcionamento do sistema MARCS, especialmente nos seus comportamentos mais relevantes.

Descrevem-se medidas de avaliação do sistema e retiram-se conclusões sobre o trabalho efectuado e o que futuramente se lhe poderá acrescentar.

Apresentam-se domínios a que, pela similaridade dos requisitos, se poderia aplicar um sistema do âmbito do descrito neste documento.

8.1 Avaliação

A avaliação de um sistema de controlo pode ser feita sob diferentes perspectivas, porventura subjectivas e susceptíveis de distintas conclusões. Quais as características mais relevantes para a avaliação?

Primariamente, o factor segurança é imprescindível num sistema de controlo de tráfego ferroviário. A impossibilidade de ocorrência de choques entre as composições circulantes ou descarrilamentos para além do término de linha deve ser o requisito básico para qualquer sistema do género.

A capacidade e volume máximo de tráfego permitido serão outros aspectos sobre os quais se poderá avaliar o sistema, se bem não tenham sido consideradas prioritários na implementação deste sistema.

Outra forma de avaliação pode consistir na medição dos tempos médios de paragem forçada para cada composição, bem como no total de ordens de paragem recebidas por cada composição durante o seu percurso.

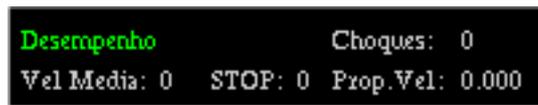


Figura 8.1: Painel de avaliação do desempenho do sistema.

8.1.1 Medidas de Avaliação Implementadas

A aplicação de simulação apresenta uma área onde são mostrados quatro valores que podem funcionar como forma de avaliação do sistema. Tal como ilustrado na figura 8.1, contém três valores com o seguinte significado:

Total de Choques Número total de comboios que chocaram ou descarrilaram (ultrapassando pontos terminais da rede) durante o processo de simulação.

Velocidade Média Velocidade média dos comboios presentes no processo de simulação.

STOP Somatório do número de vezes que cada comboio foi mandado parar em consequência de conflitos de tráfego.

Proporção de Tempo à Velocidade Desejada (Prop.Vel.). Complemento para um do valor médio de utilidade de cada um dos comboios. Uma vez que a utilidade de cada agente comboio é um valor compreendido no intervalo $[0,1]$ directamente proporcional ao tempo o comboio viajou a velocidade inferior à desejada, este valor representa a proporção de tempo que os comboios da rede ferroviária viajaram a velocidade aproximada à desejada.

Além destas medidas, também o tempo que os comboios demoram até realizar todas as paragens previstas e utilizado como medida de avaliação ao desempenho do sistema.

8.1.2 Situações Excepcionais de Tráfego

Tendo por base o mencionado nos objectivos previstos para este trabalho, um aspecto relevante na avaliação do sistema será a capacidade de lidar com situações não planeadas antecipadamente, situações a que, devido à ocorrência de factos imprevistos, é necessário saber responder.

A figura 8.2 ilustra uma situação excepcional de tráfego a que um supervisor consegue autonomamente dar resposta. O mapa aí ilustrado tem as seguintes características:

- 4 estações (E_1 , E_2 , E_3 e E_4).
- 4 comboios C_1 , C_2 , C_3 e C_4 com paragens previstas respectivamente em E_1 , E_2 , E_3 e E_4 .

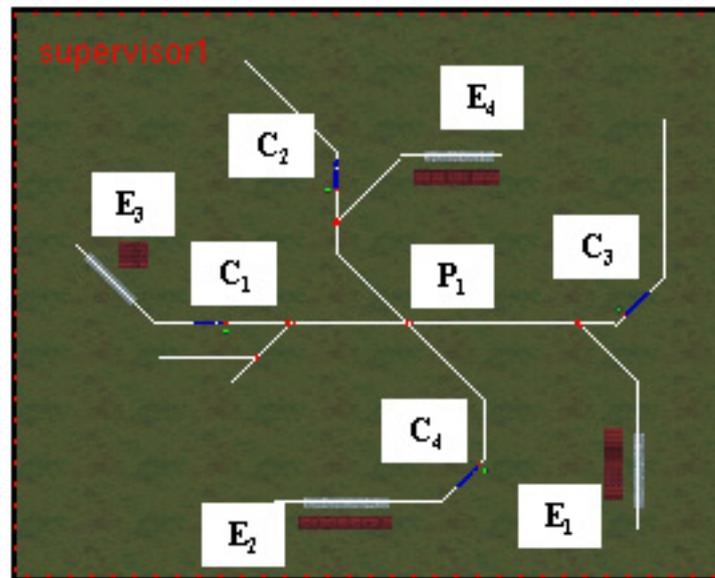


Figura 8.2: Situação excepcional de tráfego.

- 5 pontos de selecção de trajecto.
- 8 pontos terminais de linha.
- 2.2 quilómetros de extensão.
- 4 agentes *Comboio* representantes de C_1 , C_2 , C_3 e C_4 .
- 4 agentes *Estação* representantes de E_1 , E_2 , E_3 e E_4 .
- 1 agente *Supervisor*.

É óbvio que os sistemas de escalonamento de tráfego têm por função antecipar situações como a descrita na figura, prever e evitar que vários comboios necessitem simultaneamente dos mesmos recursos; mas o que fazer em caso de ocorrência de uma situação excepcional? Desactivar o sistema, comutar para o modo manual e apenas reiniciar o seu funcionamento quando as premissas de funcionamento estiverem restabelecidas?

Comportamentos usualmente associados a agentes, como autonomia e adaptabilidade a situações anómalas adquirem mais significado em situações como a descrita.

Apesar desta ser uma situação extrema e altamente improvável, em resultado de alteração de trajectos, atrasos em chegadas ou partidas de estações ou qualquer outra situação inesperada, C_1 , C_2 , C_3 e C_4 desejam seguir respectivamente para as estações E_1 , E_2 , E_3 e E_4 . Além de todos desejarem passar pelo mesmo ponto de selecção de trajecto, é necessário que o trajecto destino esteja livre, o que não se verifica.

A resolução por parte do sistema MARCS de uma situação deste género implicou o desvio temporário de comboios, o cálculo de novos trajectos para alcance do objectivo, criando-se uma sequência de manobras por forma a possibilitar a resolução.

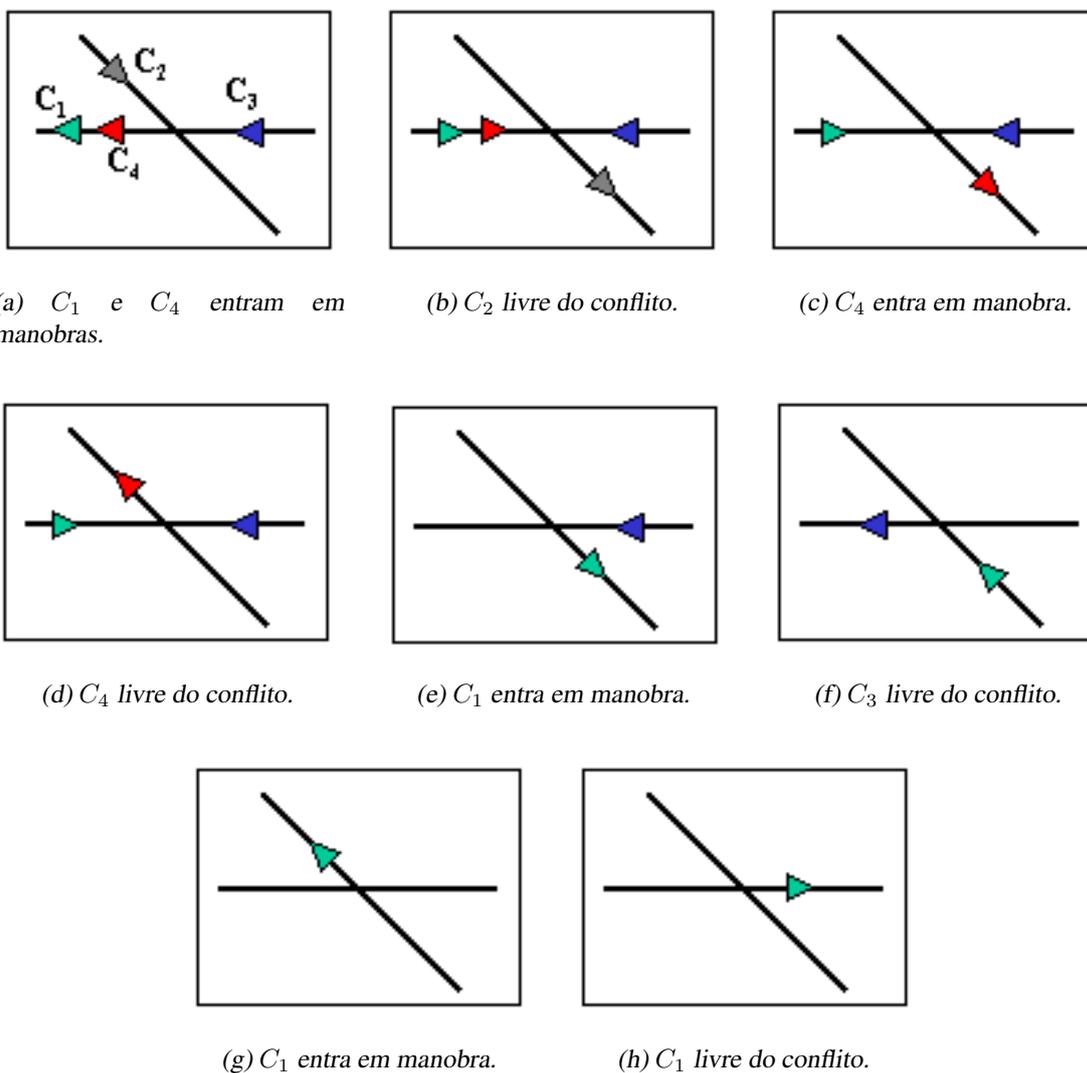


Figura 8.3: Resolução do conflito de interesses.

A figura 8.3 mostra o conjunto de acções que permitiram resolver o problema, apresentando os seguintes valores para as medidas de avaliação:

	Resultados
Choques	0
STOP	12
Prop.Vel.	0.008
Velocidade Média (Km/h)	25
Tempo de Simulação	2:55

A sequência de acções efectuada pelo agente supervisor responsável pelo ponto de selecção P_1 e troços adjacentes foi a seguinte:

Detecção do conflito. Quando algum comboio se coloca a distancia de P_1 inferior à limite, o $Supervisor_1$ verifica quais os comboios que desejam passar pelo ponto de selecção P_1 e detecta um conflito entre os interesses de quatro composições: C_1 , C_2 , C_3 e C_4 , tendo todas elas o troço destino bloqueado por outra (C_1 bloqueia C_3 , C_2 bloqueia C_4 , C_3 bloqueia C_1 e C_4 bloqueia C_2).

C_1 e C_4 **entram em manobras.** Uma vez que não existem comboios cujo trajecto destino está desimpedido, para resolver o conflito é necessária a desobstrução de pelo menos um troço, alterando temporariamente o trajecto dos comboios que nele estiverem localizados.

Nesta situação todos os troços adjacentes a P_1 contêm e bloqueiam igual número de composições (uma), pelo que é indiferente o resultado da escolha sobre qual desimpedir.

Tendo aleatoriamente determinado que o troço onde está localizado C_4 deve ficar desimpedido, é necessário desviar temporariamente esta composição. Devido às restrições de direcção, apenas lhe è possível seguir em frente ou virar à esquerda. Caso seguisse em frente continuaria a bloquear a única composição que deseja seguir em direcção a E_2 , pelo que foi determinada a mudança de direcção à esquerda.

Neste momento é importante garantir que existe espaço suficiente para que C_4 efectue a mudança de direcção à esquerda, sendo pedido a C_1 que recue, por forma a garantir tal espaço.

C_2 **livre do conflito.** Existe agora uma nova situação de conflito, com quatro pretendentes, mas em que um dos elementos tem o trajecto destino desbloqueado (C_2). Uma vez que a atribuição de prioridade a esta composição pode contribuir para a fluidez do tráfego ("bem global", secção 6.5.2), é determinada a sua passagem.

C_4 **entra em manobra.** A situação de conflito é agora entre três comboios com os trajectos bloqueados, mas em que um deles (C_4) apenas deseja atingir um troço a partir do qual possa inverter o sentido e seguir em direcção a E_4 . Neste caso é também conveniente para a fluidez na rede que seja atribuída prioridade a esse elemento.

C_4 **livre do conflito.** Nesta nova situação um dos elementos (C_4) tem o seu trajecto destino livre sendo-lhe por isso atribuída a prioridade.

C_1 **entra em manobra.** O conflito de interesses do ponto P_1 está agora limitado a duas composições. Ambas têm o seu trajecto bloqueado, sendo o processo de decisão sobre o troço a desbloquear semelhante ao do estado inicial do conflito. Ao seleccionar C_1 para temporariamente alterar o trajecto, desbloqueia-se o troço destino de C_3 .

C_3 **livre do conflito.** Uma vez que C_3 é o único comboio sem troço destino bloqueado, é-lhe atribuída prioridade.

C_1 **entra em manobra.** Apenas resta C_1 que, devido às restrições de direcção, tem que atingir primeiro um troço a partir do qual possa inverter sentido e seguir em direcção a E_1 .

C_1 **livre do conflito.** Finaliza-se a sequência de atribuição de prioridades, uma vez que apenas existe um elemento a necessitar da passagem através de P_1 .

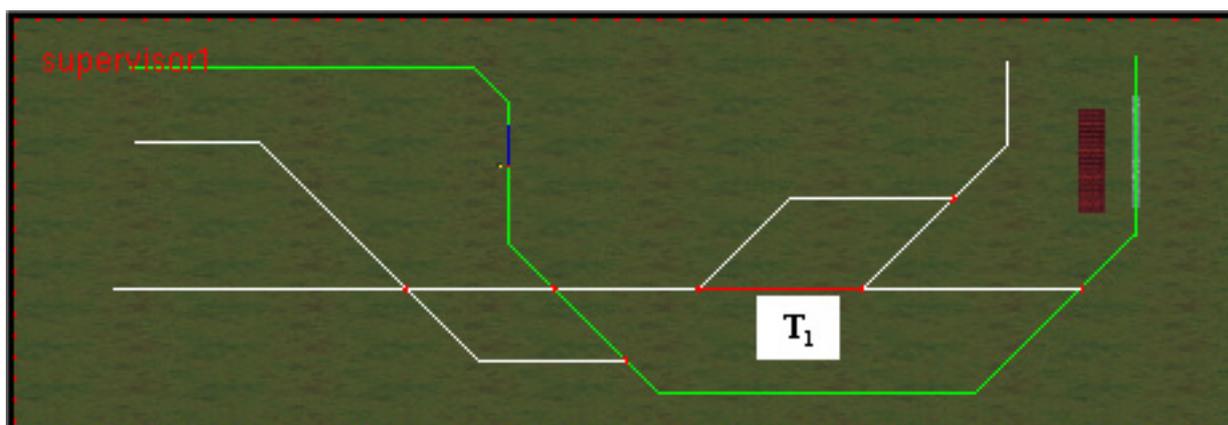
8.1.3 Avarias em Troços

Outro aspecto relevante resulta na capacidade de autónoma e dinamicamente serem calculados trajectos alternativos para cada composição em função do estado momentâneo da rede.

Uma rede ferroviária está sujeita a vários factores externos que podem influenciar a sua funcionalidade, tais como condições climáticas adversas, falhas físicas nos troços de linha, de comboios ou outros componentes. É importante que, ao receber informação sobre a ocorrência de uma situação inesperada, o sistema encontre possíveis trajectos alternativos para cada composição, minimizando os inconvenientes causados pela situação excepcional.



(a) Trajecto inicialmente previsto (a verde).



(b) Alteração autónoma do trajecto.

Figura 8.4: Avaria num troço de linha (Avaliação).

As figuras 8.4a e 8.4b ilustram o comportamento do sistema MARCS perante uma avaria num troço de linha.

O mapa aí ilustrado tem as seguintes características:

- 1 estação (E_1).
- 1 comboio (C_1).
- 7 pontos de selecção de trajecto.
- 5 pontos terminais de linha.
- 4.3 quilómetros de extensão.
- 1 agente *Comboio* representante de C_1 .
- 1 agente *Estação* representante de E_1 .
- 1 agente *Supervisor*.

Na figura 8.4a mostra-se a verde o trajecto inicialmente previsto para o comboio C_1 atingir a sua próxima paragem prevista (E_1).

Após simular a avaria num troço de linha (T_1 , segmento assinalado a vermelho), o *Supervisor* eliminou-o dos troços (recursos) disponíveis, e verificou se era necessário calcular um trajecto alternativo para alguma composição sob sua responsabilidade.

Uma vez que C_1 tinha passagem prevista pelo troço em questão, e existindo um trajecto alternativo, ele foi transmitido ao agente representante do comboio, passando a ser o destacado a verde na figura 8.4b.

O tratamento automático e autónomo de situações deste género pode constituir um avanço significativo na funcionalidade dos sistemas de controlo de tráfego.

8.1.4 Falhas em Agentes Supervisores

Nesta parte ilustra-se um cenário onde ocorre uma falha súbita num dos agentes supervisores de tráfego (Figura 8.5). O exemplo aqui representado tem as seguintes características:

- 8 estações ($E_1, E_2, E_3, E_4, E_5, E_6, E_7$ e E_8).
- 8 comboios ($C_1, C_2, C_3, C_4, C_5, C_6, C_7$ e C_8).
- 72 pontos de selecção de trajecto.
- 73.4 quilómetros de extensão.

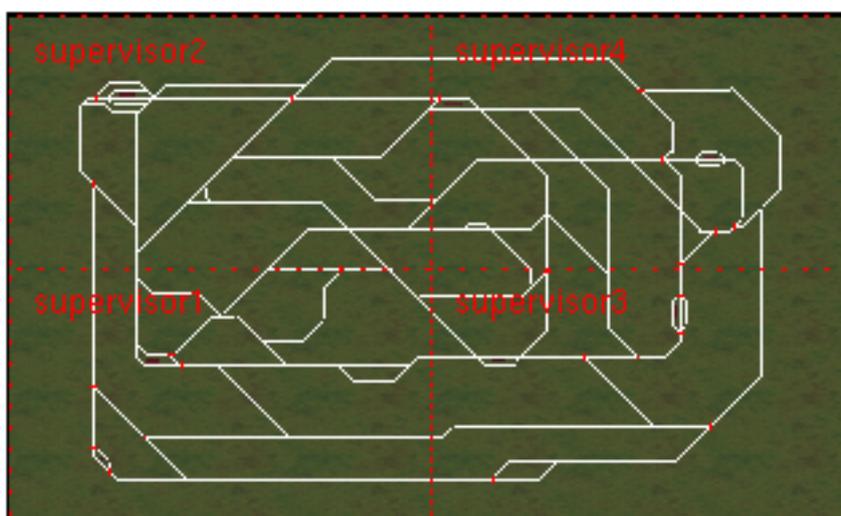


Figura 8.5: Distribuição inicial da supervisão.

- 8 agentes *Comboio* representantes de $C_1, C_2, C_3, C_4, C_5, C_6, C_7$ e C_8 .
- 4 agentes *Estação* representantes de $E_1, E_2, E_3, E_4, E_5, E_6, E_7$ e E_8 .
- 4 agentes *Supervisor* (S_1, S_2, S_3 e S_4).

Neste caso simulou-se uma falha abrupta no agente supervisor S_1 , terminando o funcionamento da correspondente aplicação. Tal como descrito na secção 7.2.1, os agentes com os quais partilha a supervisão de troços de linha (S_2 e S_3) participam num processo de leilão do tipo "envelope fechado".

Trocam entre si um valor indicador do respectivo volume de trabalho (Secção 7.2.1.2) e o que apresentou menor valor passou a supervisionar a zona anteriormente de S_1 .

Neste caso, com os seguintes valores de ponderação no cálculo do volume de ocupação de cada agente:

- Número de troços $\rightarrow 0.3$
- Número de comboios momentaneamente na zona $\rightarrow 0.2$
- Número de agulhas de selecção de trajecto supervisionadas $\rightarrow 0.5$

Os agentes supervisores apresentaram os seguintes valores:

- Supervisor $S_2 \rightarrow 25.9$
- Supervisor $S_3 \rightarrow 19.1$

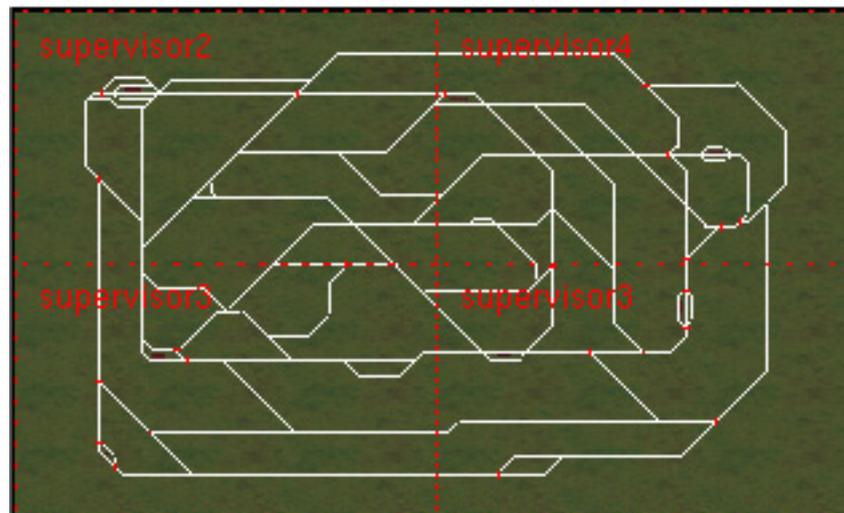


Figura 8.6: Distribuição final da supervisão após falha de S_1 .

Ao receber o valor de S_2 , o agente S_3 imediatamente se declarou como vencedor do processo de leilão (Secção 7.2.1.1) e assumiu uma nova zona de supervisão, tal como mostrado na figura 8.6.

Esta situação ilustra simultaneamente a utilidade de cada *Supervisor*, além de conhecer os seus vizinhos, conhecer os vizinhos dos seus vizinhos.

Repare-se que, analisando o mapa ilustrado na figura 8.5, baseado no método de distribuição de competência de supervisão detalhado na secção 6.3, os agentes supervisores S_2 e S_3 não eram vizinhos entre si, pelo que nem sequer existia alguma ligação estabelecida entre ambos.

Ambos eram vizinhos de S_1 e sabiam quais os vizinhos que este tinha. Desta forma, após detectarem a falha no agente puderam autonomamente estabelecer uma ligação com os parceiros no processo de leilão.

8.1.5 Cooperação entre Supervisores - Troca de Informação

Nesta secção apresenta-se um cenário onde a troca de informação entre agentes supervisores sobre congestionamentos (conflitos) de tráfego potencia a eficiência do sistema de controlo.

O mapa ferroviário apresentado na figura 8.7 tem as seguintes características:

- 2 estações (E_1 e E_2).
- 3 comboios C_1 , C_2 e C_3 , respectivamente com destino E_1 , E_2 e E_2 .
- 14 pontos de selecção de trajecto.
- 4 pontos terminais de linha.

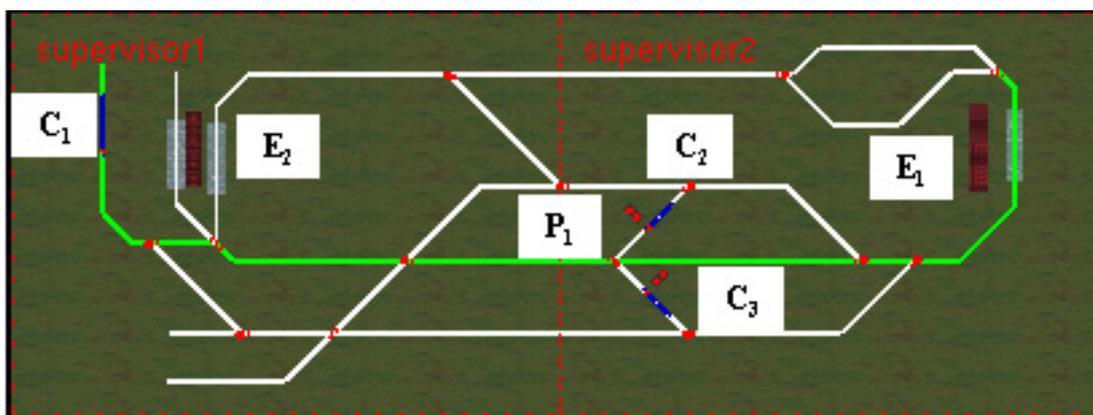
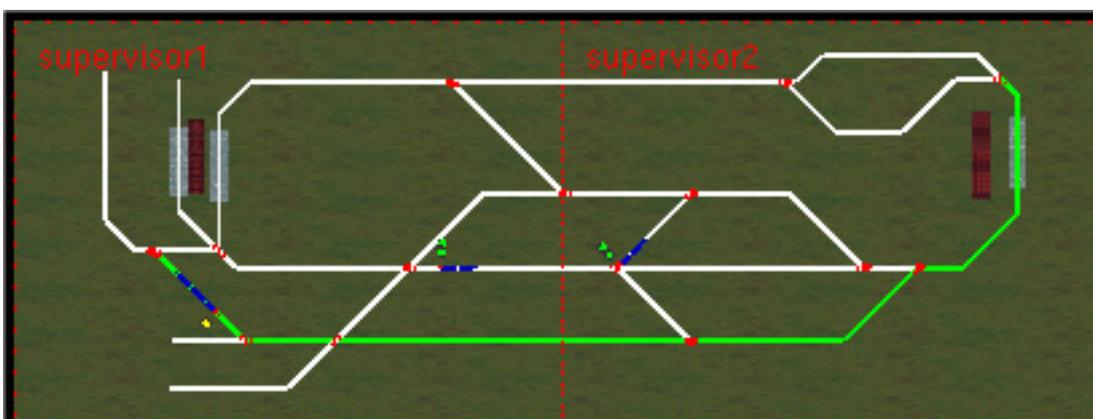
(a) S_2 informa S_1 de um conflito em P_1 .(b) Alteração do trajecto de C_1 em função do conflito em P_1 .

Figura 8.7: Cooperação entre supervisores (Troca de informação - Avaliação).

- 2.8 quilómetros de extensão.
- 3 agentes *Comboio* representantes de C_1 , C_2 e C_3 .
- 2 agentes *Estação* representante de E_1 e E_2 .
- 2 agentes *Supervisor* (S_1 e S_2).

O trajecto previsto para os comboios C_2 e C_3 resulta num congestionamento de tráfego no ponto de selecção de trajecto P_1 .

Ao detectar o conflito, o agente supervisor S_2 informa os agentes vizinhos (S_1) sobre o tipo e local de conflito existente. Com esta informação S_2 espera que os agentes vizinhos (e ele próprio) possam, sempre que possível, arranjar razoáveis trajectos alternativos, de forma a não sobrecarregar ainda mais o tráfego na zona congestionada.

Quando recebe esta informação, S_1 procede à reavaliação dos trajectos dos comboios sob sua supervisão, evitando sempre que possível a passagem pelo ponto de selecção P_1 . Neste caso o trajecto inicialmente previsto para C_1 (a verde na figura 8.7a) é alterado e comunicado ao respectivo agente representante (Figura 8.7b).

Tal como ilustra a figura 8.7b, esta alteração contribuiu claramente para a fluidez do tráfego, uma vez que além de evitar a saturação de tráfego no ponto P_1 , evita ainda um conflito que aconteceria entre os comboios C_1 e C_3 (vencedor do conflito do ponto P_1).

Na execução do processo de simulação nas duas variantes implementadas (com ou sem processos cooperativos) obtiveram-se os seguintes resultados:

	Sem cooperação	Com cooperação
Choques	0	0
STOP	5	1
Prop.Vel.	0.000	0.012
Velocidade Média (Km/h)	35	39
Tempo de Simulação	1:22	1:15

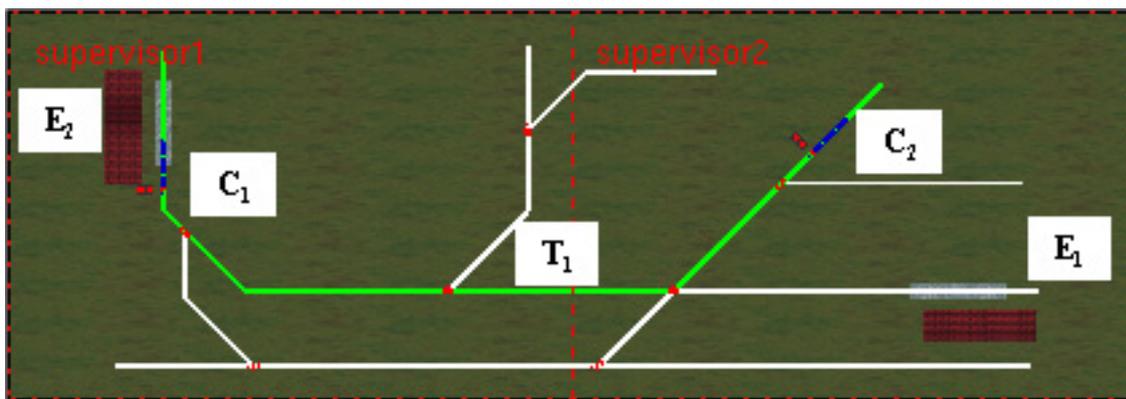
Tal como facilmente se conclui, a cooperação estabelecida entre supervisores através da troca de informação sobre congestionamentos de tráfego permitiu a obtenção de uma visão mais global do estado da rede e contribuir para a eficiência do sistema.

8.1.6 Cooperação entre Supervisores - Troca de Planos

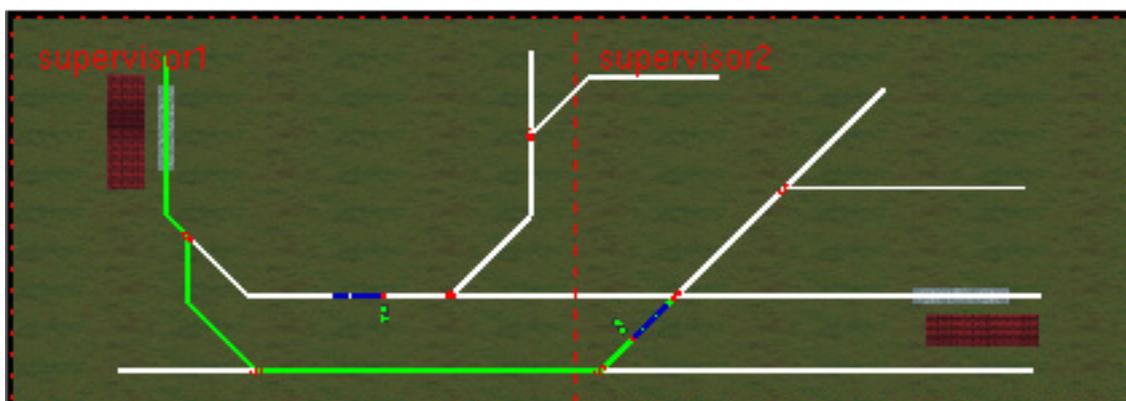
Nesta secção exemplifica-se uma situação de tráfego em que a construção de um plano comum entre agentes supervisores contribui para minimizar o tempo de espera de uma composição de alta prioridade e melhorar a fluidez do tráfego.

O mapa apresentado na figura 8.8 tem as seguintes características:

- 2 estações (E_1 e E_2).
- 2 comboios C_1 e C_2 com destino respectivamente a E_1 e E_2 .
- 7 pontos de selecção de trajecto.
- 8 pontos terminais de linha.
- 2.2 quilómetros de extensão.
- 2 agentes *Comboio* representantes de C_1 e C_2 .
- 2 agentes *Estação* representantes de E_1 e E_2 .
- 2 agentes *Supervisor* (S_1 e S_2).



(a) $Supervisor_1$ propõe plano para minimizar tempo de espera de C_1 .



(b) Alteração do trajeto de C_2 em função do plano comum.

Figura 8.8: Cooperação entre supervisores (Troca de planos - Avaliação).

O comboio C_1 é de prioridade máxima, sendo cada *Supervisor* responsável pela minimização do respectivo tempo de espera. Na definição do trajeto previsto para C_1 , S_1 propõe aos agentes vizinhos (S_2) um plano da forma:

”Tentar manter desimpedido o melhor caminho para a estação E_1 a partir do troço T_1 ”

Ao receber esta informação o *Supervisor* S_2 adopta este objectivo e procede à reavaliação dos trajectos de cada comboio sob sua supervisão.

Neste exemplo, o trajeto inicialmente previsto para o comboio C_2 (a verde na figura 8.8a) intersecta algum dos troços que devem permanecer desimpedidos.

Tal como ilustrado na figura 8.8b, uma vez que existia um trajeto **alternativo razoável** para o comboio C_2 , foi feita a respectiva substituição.

Esta acção teve como consequência directa a minimização do tempo que C_1 poderia perder

em resultados de conflitos de tráfego. Simultaneamente evitaram-se conflitos de interesses entre composições, maximizando a fluidez do tráfego.

Ao efectuar o processo de simulação permitindo ou inibindo a cooperação entre supervisores, obtiveram-se os seguintes resultados:

	Sem cooperação	Com cooperação
Choques	0	0
STOP	2	0
Prop.Vel.	0.000	0.019
Velocidade Média (Km/h)	41	53
Velocidade Média Comboio C_1 (Km/h)	47	53
Tempo de Simulação	1:50	1:13

8.1.7 Aprendizagem

Esta secção é dedicada à apresentação de um caso específico onde a adaptabilidade do sistema contribui para a fluidez do tráfego na rede.

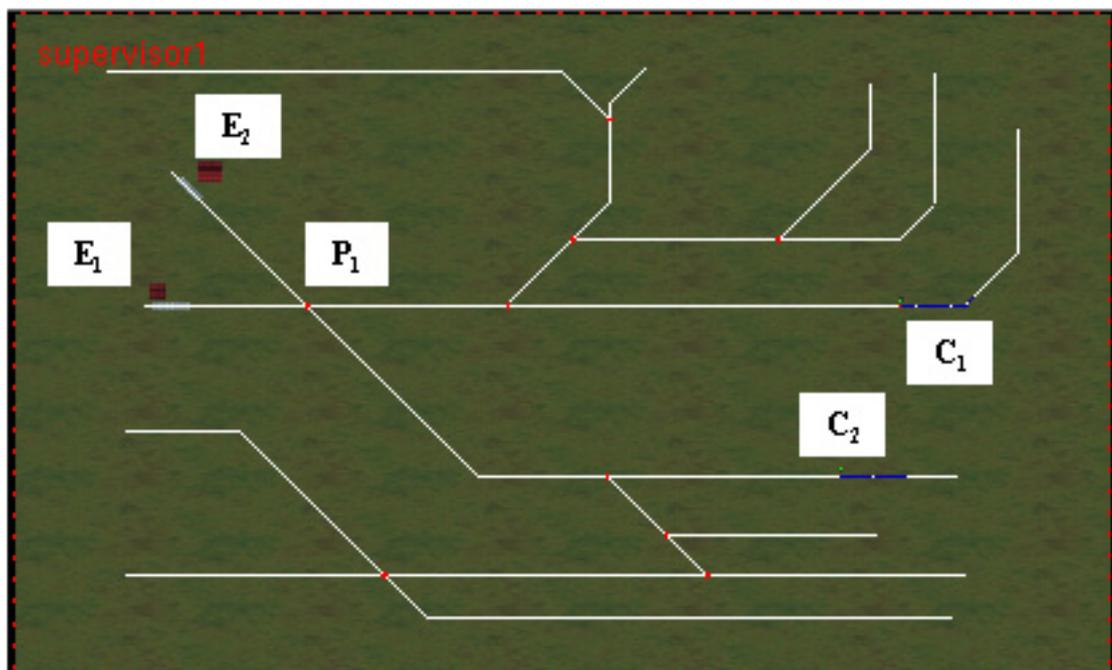


Figura 8.9: Sistema de aprendizagem (Avaliação).

A figura 8.9 ilustra um mapa ferroviário com as seguintes características:

- 2 estações (E_1 e E_2).
- 9 pontos de selecção de trajecto.

- 13 pontos terminais de linha.
- 8.3 quilómetros de extensão.
- 2 agentes *Comboio* representantes de C_1 e C_2 .
- 2 agentes *Estação* representantes de E_1 e E_2 .
- 1 agente *Supervisor*.

Nesta situação os trajectos dos dois comboios (C_1 e C_2) com direcções respectivamente a E_1 e E_2 vão-se interceptar e entrar em conflito no ponto de selecção de trajecto (P_1).

Uma vez que chegam sensivelmente ao mesmo tempo a este local será necessário que um deles tenha que parar para permitir a passagem do outro.

Nesta situação após as duas composições atingirem os respectivos destinos, o painel de avaliação do sistema regista os seguintes valores:

Resultados da primeira Simulação	
Choques	0
STOP	1
Prop.Vel.	0.366
Velocidade Média (Km/h)	61
Tempo de Simulação	1:54

O valor de *STOP* indica que foi necessário mandar parar uma composição em resultado de conflitos de tráfego.

Caso não existam agentes *Aprender* registados no sistema, esta é uma situação que se vai repetir sempre que dois comboios estiverem sensivelmente no mesmo local e com destinos semelhantes aos considerados neste exemplo.

Ao incorporar um agente *Aprender* no processo de simulação produzem-se resultados semelhantes aos anteriores na primeira execução da simulação, visto ser a que fornece um historial para análise.

Após parar a simulação e efectuar a sua reinicialização, verifica-se, através da análise do registo de actividade dos agentes, que *Aprender* transmitiu uma regra ao *Supervisor* prevendo uma situação de conflito sempre que dois comboios estiverem em situação semelhante à descrita. A mensagem enviada foi a seguinte:

```
(tell:sender Aprender1
  :receiver Supervisor1
  :content (REGRA
    Confianca 1.0
```

```

Tempo -67
TotalPremissas 2
  Premissa Local 861.0 -4.9 Destino estacao2
  Premissa Local 768.0 -260.0 Destino estacao1
Consequente
  Conflito Vertice 0)
)

```

Ao analisar o conteúdo da mensagem pode concluir-se que *Aprender1* gerou uma regra com confiança máxima (uma vez que o histórico resume-se a uma única situação de tráfego), que diz:

”Sempre que existirem comboios localizados sensivelmente a latitude e longitude de (861.0,-4.9) e (768.0,-260.0), respectivamente com destino a ”estacao2” e ”estacao1”, vão originar um conflito de tráfego no vértice (ponto de selecção de trajecto) 0, dentro de 67 segundos.”

No segundo processo de simulação, o *Supervisor* vai comparar o estado dos dois comboios com as premissas da regra entretanto inferida. Ao detectar uma situação de tráfego similar pede ao agente que representa C_1 para acelerar ligeiramente o andamento e ao de C_2 para o abrandar.

Estas acções são suficientes para evitar o conflito no ponto P_1 passando o painel de avaliação a mostrar os seguintes valores:

	Resultados da segunda Simulação
Choques	0
STOP	0
Prop.Vel.	0.282
Velocidade Média (Km/h)	66
Tempo de Simulação	1:48

Neste caso o facto mais relevante é a inexistência de ordens de paragem a qualquer dos comboios. Nesta situação foi possível antecipar o conflito de tráfego e evitar que alguma das composições tivesse que parar em resultado da cedência de prioridade a outra.

Após o processo de aprendizagem, foram melhorados os valores de avaliação para os parâmetros de ”Velocidade Média” e ”Tempo de Simulação”.

No caso da medida de eficiência do sistema, o decréscimo no seu valor é explicado pelo facto de representar a proporção de tempo que cada comboio viaja a velocidade aproximada à média prevista para a rede. Ao pedir a um dos agentes comboios que diminuísse temporariamente a sua velocidade está-se a baixar o seu índice de utilidade e consequentemente a medida de eficiência do sistema.

8.2 Trabalho Futuro

Após a implementação do sistema (Aplicação de simulação, plataforma de comunicação e agentes), existem algumas vertentes susceptíveis de complemento ou optimização e objecto de trabalho futuro.

O requisito principal do sistema, que consistia na implementação de um sistema multi-agente para controlo de tráfego ferroviário, permite pelas suas características de modularidade, a evolução do sistema para níveis de maior sofisticação e aproximação à realidade.

Selecionei três variantes que julgo serem passíveis de aprofundamento e objecto de trabalho futuro: Controlo, Agentes e Aprendizagem.

8.2.1 Controlo

A nível dos processos de controlo do tráfego será possível, alterando o comportamento dos agentes "Supervisor", incorporar ou modelar de forma mais exacta os métodos de controlo dos sistemas actualmente em funcionamento. A modularidade do sistema permite a fácil incorporação de novos processos de controlo.

8.2.2 Agentes

A nível dos agentes, uma vez que parecem resultar vantagens da cooperação entre entidades, será possível alargar o número e tipo de processos cooperativos.

Os processos de decisão podem também tornar-se mais sofisticados e aproximados à realidade.

Por exemplo no caso dos agentes "Estação". O processo de definição da plataforma de paragem para cada comboio é extremamente simples, sendo dada ordem de paragem no ponto central da primeira plataforma disponível.

Seria possível levar em conta o comprimento, destino ou tempo de paragem prevista para cada composição na selecção da plataforma de paragem e do preciso local onde esta deverá ser efectuada. Desta forma poderiam, tal como na realidade, existir vários comboios estacionados na mesma plataforma.

8.2.3 Aprendizagem

Por ser de carácter mais académico, esta será provavelmente a área à qual mais trabalho poderá ser acrescentado. Através de implementação de novos comportamentos no agente "Aprender", podem-se testar novos algoritmos de extracção de conhecimento ou incorporar vários métodos de aprendizagem simultaneamente.

Penso que será possível tornar o sistema de aprendizagem mais eficiente ao tentar extrair conhecimento das acções tomadas por cada *Supervisor* na resolução de um conflito.

No trabalho apresentado apenas é possível extrair informação do género: "Quando dois comboios estão em determinada posição vai existir um conflito algum tempo depois".

A análise das acções de cada *Supervisor* poderia levar a conhecimento do tipo: "Quando dois comboios (*A* e *B*) estão em determinada posição vai existir um conflito algum tempo depois **E** não se deve mandar abrandar o comboio *A* porque originará outro conflito".

8.3 Generalização do Sistema

Numa visão geral, o sistema MARCS pode ser entendido como de **alocação distribuída de recursos**. O seu objectivo básico consiste em afectar os recursos (troços e agulhas de selecção) aos elementos que temporariamente deles necessitam (comboios).

A afectação é efectuada por várias entidades (*Supervisor*) com conhecimento detalhado de uma parcela do ambiente e que implementam processos cooperativos por forma a facilitar a aquisição do objectivo.

Efectuando este exercício de abstracção podem-se encontrar pontos em comum entre o domínio do sistema MARCS e outros que se descrevem de seguida.

Empresas de Transportes Internacionais

As grandes empresas de transportes internacionais abrangem presentemente a quase totalidade da superfície terrestre, efectuando serviços de distribuição de mercadorias através das vias área, marítima e terrestre.

É necessário articular os meios de transporte disponíveis tendo em conta diversos objectivos: entrega das mercadorias dentro dos prazos ou maximizar a capacidade de carga de cada veículo por exemplo.

Em relação ao sistema MARCS, pode-se efectuar um paralelismo entre os recursos disponíveis (troço de caminho de ferro ↔ meios de transporte) e pretendentes (comboios ↔ mercadorias) por forma a encontrar alguns pontos de contacto.

Sendo igualmente sistemas distribuídos, pode-se imaginar a implementação de agentes "Zona" responsáveis por áreas geográficas locais, com conhecimento detalhado da situação na respectiva área (meios de transporte disponíveis e pedidos de transporte efectuados) e com capacidade para efectuar a alocação dos respectivos recursos (Figura 8.10).

A capacidade de colaboração e articulação entre os agentes "Zona" seria neste caso ainda mais relevante.

A transmissão de planos do sistema MARCS: "Tenta manter o caminho livre (alocação de recursos) entre o troço T_1 e a estação E_j " encontra paralelismo em:

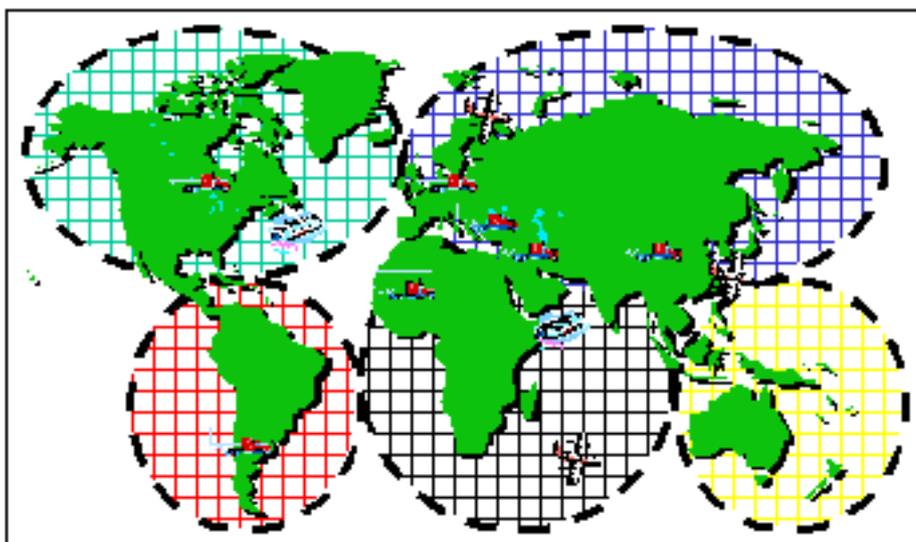


Figura 8.10: Divisão hipotética em agentes *Zona*.

”Eu asseguro o transporte marítimo até ao porto ”X”, e tu asseguras (alocação de recursos) a ligação terrestre até ao local de destino”.

A transmissão de informação sobre a situação específica de cada um seria, tal como no caso do sistema MARCS, de utilidade notória. A impossibilidade de alocação de recursos, ou a sobrecarga na alocação seria factor suficiente para que se evitasse determinado meio de transporte ou zona de circulação.

A implementação de agentes com autonomia suficiente para administrar os pedidos efectuados na respectiva zona iria diminuir as necessidades de comunicação, aumentar a eficiência do sistema, distribuir a informação e o controlo por várias entidades, melhorando a sua robustez.

Instituto Nacional de Emergência Médica

O Instituto Nacional de Emergência Médica - INEM - é o organismo do Ministério da Saúde ao qual cabe assegurar o funcionamento, no território de Portugal Continental, de um sistema integrado de emergência médica, de forma a garantir aos sinistrados ou vítimas de doença súbita a pronta e correcta prestação de cuidados de saúde.

A prestação de socorros no local da ocorrência, o transporte assistido das vítimas para o hospital adequado e a articulação entre os vários estabelecimentos hospitalares, são as principais tarefas do INEM. O INEM, através do número nacional de socorro (112), dispõe de vários meios para responder com eficácia, a qualquer hora, a situações de emergência médica (INEM 2003)

Cada chamada para o 112 é dirigida para o centro de controlo nacional que se encarrega, consoante a zona origem e os meios de socorro necessários, de notificar o respectivo centro regional para que disponibilize os meios necessários.

De forma análoga à divisão da competência de supervisão de uma rede ferroviária por vários agentes, poderia ser dividido o território nacional por vários agentes (*Zona*), cada um com conhecimento actualizado dos recursos efectivamente disponíveis (ambulâncias, bombeiros, médicos, para-médicos, . . .) e com autonomia para efectuar a respectiva alocação (Figura 8.11).

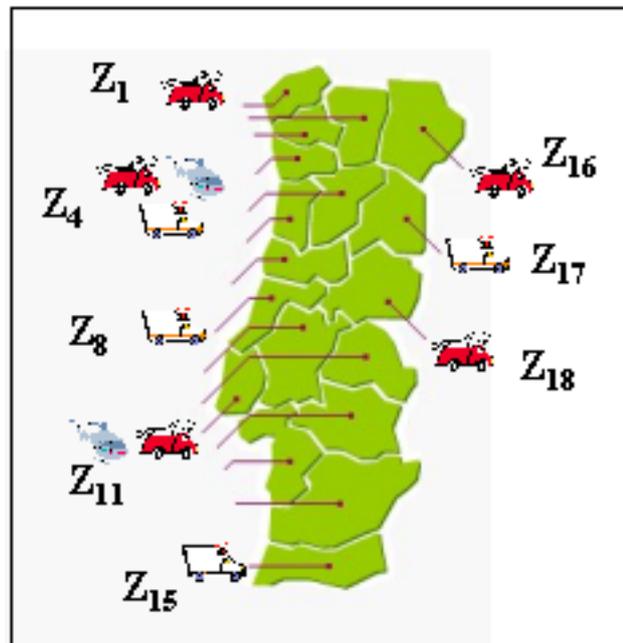


Figura 8.11: Divisão hipotética em agentes *Zona* - INEM.

Nesta figura, o território nacional está dividido por dezoito agentes *Zona* (Z_1, Z_2, \dots, Z_{18}). Cada um destes mantém ligação aos seus vizinhos e está encarregue de responder às ocorrências na respectiva área. Deve igualmente manter registo actualizado do estado dos recursos efectivamente disponíveis (veículos avariados, pessoal doente, de férias, . . .). Esta tarefa será potencialmente mais fácil quanto menor for a área controlada pelo agente.

É possível estabelecer um paralelismo entre as entidades presentes no sistema MARCS e as de um sistema deste género:

- Agentes *Supervisor* → Agentes *Zona*. Responsáveis por efectuar a alocação de recursos numa parcela do ambiente.
- Comboios → Sinistros. Entidades que necessitam os recursos alocados pelos elementos anteriores.
- Troços e agulhas de selecção → Ambulancias, bombeiros, médicos, . . . Recursos passíveis de atribuição.

Tal como no caso dos transportes ferroviários será necessário que os vários agentes *Zona* partilhem informação e planos de actuação por forma a poderem estabelecer acções conjuntas que lhes permitam responder aos sinistros de forma o mais eficiente possível.

A necessidade de recursos (existência de um sinistro) seria apenas comunicada ao respectivo agente *Zona* que, caso não tivesse capacidade efectuar a alocação, a transmitiria aos seus vizinhos.

Poderiam resultar como vantagens desta abordagem:

Robustez do sistema Ao descentralizar o controlo do sistema minimiza-se a sua dependência em relação a uma única entidade.

Facilidade na manutenção de informação Será mais fácil para cada agente manter informação actualizada dos recursos efectivamente disponíveis na sua zona, que a uma entidade manter informação sobre o estado global do sistema.

Flexibilidade na alocação de recursos A autonomia associada a cada agente permitiria que, em caso de necessidade de transferência de recursos entre zonas, eles fossem alocados ao agente e não ao sinistro. Esta situação pode permitir ao agente receptor dos recursos a reavaliação da situação dentro da sua zona e ajustar as acções correspondentes (Em paralelismo com o reencaminhamento dos comboios à entrada da área de cada *Supervisor*)

8.4 Conclusões

O trabalho descrito neste documento permitiu que me apercebesse da importância que, especialmente nas regiões tecnologicamente mais desenvolvidas, se está a dar à sofisticação dos sistemas de controlo de tráfego ferroviário.

O previsível aumento populacional em zonas urbanas e sub-urbanas originará requisitos de capacidade, segurança e rapidez cada vez maiores.

Estes requisitos tornam obsoletos os sistemas de controlo de tráfego ferroviário tradicionais - baseados na noção de bloco fixo - uma vez que, para manter a segurança, exigem distâncias entre comboios incompatíveis com as necessidades actuais.

Além disso, o controlo de tráfego ferroviário apresenta requisitos e especificidades não susceptíveis de identificação a partir de uma análise superficial.

A maior diferença relativamente a outros sistemas de controlo está exactamente na necessidade de encaminhamento (ordenação) de cada veículo.

Um sistema dedicado ao tráfego rodoviário, por exemplo, tem por função a **regulação** do volume de tráfego permitido para cada parcela da rede. O sistema não interfere directamente com o posicionamento de cada veículo, nem é necessário ordenar activamente a direcção e velocidade de cada elemento.

No caso dos transportes ferroviários, além do requisito da afectação de recursos, existe o de encaminhamento de cada composição, requisitando um sistema de **controlo e ordenação** efectivo.

O sistema implementado, MARCS - *Multi Agent Railway Control System*, contém características que podem constituir-se, mais do que alternativas, como complementos viáveis aos sistemas de controlo de tráfego ferroviário baseados em comunicações.

Este tipo de sistemas tem provado segurança e eficiência no desempenho da sua função. A qualidade e melhoria de desempenho tem sido manifestada de forma unânime em vários artigos e conferências sobre o tema.

Esta abordagem de características académicas apresenta comportamentos, essencialmente ao nível de tolerância a falhas e tratamento de situações excepcionais, que a distinguem dos demais e podem constituir-se como complemento aos sistemas actualmente existentes.

O facto de ser um sistema baseado em agentes propicia a implementação de comportamentos de adaptabilidade, autonomia e relacionamento. Penso que será perfeitamente viável a adopção por parte dos sistemas actuais de controlo de algumas das características apresentadas pelo MARCS.

Situações como o atraso ou avaria simultânea de várias composições ou troços de linha, apesar de altamente improváveis, exigem nos sistemas tradicionais a desactivação do sistema até ao restabelecimento das condições normais de funcionamento. É em situações como esta que a noção de agente autónomo pode responder com vantagem relativamente a modelos tradicionais.

Além destas características vantajosas, os agentes podem simultaneamente modelar os comportamentos actualmente implementados nos componentes dos sistemas de controlo em funcionamento.

Relativamente ao papel dos agentes no sistema, penso que as conclusões mais relevantes se prendem com os potenciais benefícios resultantes da cooperação entre supervisores de tráfego.

Tal como apresentado na secção de avaliação do sistema, uma vez que cada agente tem necessariamente uma visão parcial do ambiente, a troca de informação e de objectivos entre elementos pode proporcionar uma maior abrangência da perspectiva de cada um e potenciar a eficiência do sistema de controlo.

Nos casos apresentados, todas as medidas de avaliação permitem constatar os benefícios resultantes da implementação de processos cooperativos.

Considero que o comportamento dos agentes, apesar de passível de sofisticação futura, está desde logo condicionado pelo facto de se tratar de um sistema de controlo com características de tempo-real.

Este facto aumenta os imperativos de segurança, eficiência e rigidez, restringindo a autonomia e tempo de decisão disponível para cada agente.

Num sistema de negociação, por exemplo, será bastante mais intuitiva a implementação de comportamentos característicos dos agentes, como o não-determinismo, liberdade de actuação ou capacidade de iniciativa. Nestes casos emergem mais facilmente as propriedades

associadas a agentes, tornando mais enriquecedora a interação entre os intervenientes do sistema, em termos da "programação orientada para agentes".

Não obstante, penso que as premissas e requisitos apresentados para este trabalho foram globalmente atingidos e apresentado um sistema multi-agente que responde ao enunciado como tema proposto de tese.

Conclusão

Os cenários apresentados ilustram situações onde as características do sistema contribuem para a fluidez e segurança do tráfego. Apesar de a isso serem propícias, pode-se inferir que, em casos específicos, o sistema MARCS pode otimizar os processos de controlo, especialmente após a ocorrência de situações excepcionais.

Apêndice A

Execução e Utilização do Sistema

Neste apêndice mostra-se a forma de execução do sistema MARCS e a sua correcta utilização, nomeadamente através da interacção com a aplicação de simulação. Apresentam-se os comandos e funcionalidades disponibilizadas e exemplifica-se o processo de construção de um mapa ferroviário.

A.1 Execução do Sistema

Nesta parte mostra-se como se devem executar as aplicações que compõem o sistema: simulador e agentes.

Os exemplos apresentados são relativos à execução no sistema operativo *Microsoft Windows*, independentemente da versão utilizada.

As aplicações que fazem parte do sistema são as seguintes:

- "Simulador"
- "AgenteSupervisor"
- "AgenteComboio"
- "AgenteEstacao"
- "AgenteAprender"

Um aspecto comum a todas as aplicações é o facto de possibilitarem o registo da sua actividade de formas distintas. Podem enviar essa informação para o écran, para ficheiro, ambos ou nenhum. O utilizador especifica o tipo de registo pretendido através de um parâmetro passado no comando de execução da aplicação.

<Aplicação> [TIPO_LOG]

Os códigos associados às várias possibilidades de registo de actividade ([TIPO_LOG]) são os seguintes:

- 0 agente não efectua registo da sua actividade.
- 1 A informação é enviada para o écran.
- 2 A informação é enviada para o ficheiro <NOME_AGENTE>.log.
- 3 Informação enviada simultaneamente para o écran e para ficheiro.

A.1.1 Simulador

A aplicação ”*Simulador*” representa o mundo real, uma vez que é utópico o teste em ambiente real dum sistema com estas características. A execução desta aplicação è imprescindível, pois é ela que fornece informação aos agentes acerca do mapa ferroviário e da posição de cada comboio. Para tal é necessária a seguinte estrutura de ficheiros e directorias:

Simulador.exe Ficheiro executável que inicia a aplicação de simulação.

configSimulador.net Ficheiro que contém as definições necessárias para que as restantes aplicações possam comunicar com o simulador. Deve conter, em formato *XML*, informação acerca do porto onde o simulador aceita pedidos de ligação (Formato detalhado no apêndice E).

config.xml Este ficheiro contém todos as definições parametrizáveis pelo utilizador, desde o aspecto da aplicação, medidas dos componentes, tempos e distâncias (Formato detalhado no apêndice E).

Img Directoria que contém os recursos visuais da aplicação. Dentro desta directoria deverão estar os seguintes ficheiros:

→ acelerar.tga	→ agentes.tga	→ apagar.tga	→ apagar_all.tga
→ aprender.tga	→ baixo.tga	→ cima.tga	→ comboio.tga
→ direita.tga	→ divide.tga	→ divideB.tga	→ edit1.tga
→ edit.tga	→ esquerda.tga	→ final.tga	→ finalF.tga
→ fixa.tga	→ fixaNao.tga	→ help.tga	→ init.tga
→ novo.tga	→ open.tga	→ paralelos.tga	→ pause1.tga
→ relva.tga	→ sair.tga	→ save.tga	→ segmento.tga
→ Simmais.tga	→ Simmenos.tga	→ simula1.tga	→ start1.tga
→ stop1.tga	→ telhas.tga	→ texturas.tga	→ trajecto.tga
→ trajectoNao.tga	→ travar.tga	→ zoomIn.tga	→ zoomInit.tga
→ zoomOut.tga			

default.xml Se o utilizador tiver um mapa ferroviário previamente criado e desejar que a aplicação efectue a sua leitura no início da execução deve dar ao ficheiro o nome "default.xml". Este ficheiro deve estar em formato XML cujo DTD se mostra no apêndice E.

glut32.dll Biblioteca auxiliar de componentes gráficos.

opengl32.dll Biblioteca auxiliar para criação e visualização de objectos gráficos. As versões posteriores ao *Windows 98* já trazem este ficheiro não sendo necessário para a execução do simulador.

Numa sessão de *MS-DOS*, o comando para iniciar a aplicação de simulação é o seguinte:

Simulador [TIPO_LOG=0]

Após a execução deste comando aparece no écran uma aplicação com o aspecto indicado na figura A.1

Depois de ter sido criado o simulador, deve-se passar para a fase de criação dos agentes.

A.1.2 Agente Supervisor

Ao contrário do simulador, podem existir simultaneamente várias aplicações intervenientes no sistema. O comando para iniciar a sua execução é o seguinte:

AgenteSupervisor < NOME_AGENTE > [TIPO_LOG] [COOPERAÇÃO=1]

O parâmetro < *NOME_AGENTE* > serve como identificador do agente no sistema.

No caso dos agentes supervisores é possível especificar-se, no comando de execução da aplicação, se os agentes podem ou não implementar mecanismos de cooperação para supervisionar o tráfego. Os valores admissíveis para este parâmetro são os seguintes:

0 Neste caso os mecanismos de cooperação entre supervisores estão inibidos.

1 Este é o valor por defeito e indica a liberdade de cooperação entre agentes supervisores.

Os ficheiros necessários para a sua execução são:

AgenteSupervisor.exe Ficheiro executável que inicia a aplicação.

Config< NOME_AGENTE >.net Ficheiro que contém a informação necessária para o agente comunicar com os restantes, nomeadamente o endereço e porto de escuta do simulador e o porto onde o próprio agente recebe mensagens. Este ficheiro está em formato XML cujo DTD se encontra no apêndice E.



Figura A.1: Aplicação de simulação.

Exemplos:

”AgenteSupervisor A 2 1” → Inicia a execução de um agente supervisor de nome ”A”, que regista a sua actividade em ficheiro e implementa processos de cooperação.

”AgenteSupervisor B 0 0” → Inicia a execução de um agente supervisor de nome ”B”, que não efectua o registo da sua actividade e não coopera com outros agentes.

A.1.3 Agente Comboio

Podem também ser executadas várias aplicações deste tipo, em princípio tantas quantos os comboios existentes na simulação. O comando para iniciar a sua execução é o seguinte:

AgenteComboio < *NOME_AGENTE* > [TIPO_LOG=0]

O parâmetro < *NOME_AGENTE* > serve como identificador do agente no sistema e os ficheiros necessários são:

AgenteComboio.exe Ficheiro executável que dá início ao funcionamento da aplicação

Config< *NOME_AGENTE* >.net Ficheiro que contém a informação necessária para o agente comunicar com os restantes, nomeadamente o endereço e porto de escuta do simulador e o porto onde o próprio agente recebe mensagens. Este ficheiro está em formato XML com o respectivo DTD no apêndice E.

A.1.4 Agente Estacao

Analogamente ao caso dos agentes que representam comboios, devem ser criadas tantas quantas as estações existentes na simulação. O comando para iniciar a sua execução será:

AgenteEstacao < *NOME_AGENTE* > [TIPO_LOG=0]

Os ficheiros necessários são:

AgenteEstacao.exe Ficheiro executável que dá início à aplicação.

Config< *NOME_AGENTE* >.net Ficheiro que contém a informação necessária para o agente comunicar com os restantes, nomeadamente o endereço e porto de escuta do facilitador e o porto onde o próprio recebe mensagens. Este ficheiro está em formato XML cujo DTD se detalha no apêndice E.

A.1.5 Agente Aprender

A forma de iniciar a execução é similar à dos restantes agentes. O comando para tal é:

AgenteAprender < *NOME_AGENTE* > [TIPO_LOG=0]

Tal como nos anteriores casos, < *NOME_AGENTE* > funciona como identificador do agente perante o sistema e são necessários os seguintes ficheiros:

AgenteAprender.exe Ficheiro executável que dá início à aplicação.

Config< *NOME_AGENTE* >.net Ficheiro que contém a informação necessária para o agente comunicar com os restantes, nomeadamente o endereço e porto de escuta do simulador e o porto onde o próprio recebe mensagens. Este ficheiro está em formato XML cujo DTD se detalha no apêndice E.

A.2 Manual de Utilizador - Aplicação Simulador

Neste apêndice pretende-se apresentar a melhor forma de interagir com o sistema implementado. Indicam-se as funcionalidades e comandos permitidos para a aplicação de simulação, consoante o modo e estado em que esta se encontra.

Constituiu preocupação tornar a interacção com o utilizador o mais simples possível, nomeadamente através da criação de uma janela de ajuda ou da implementação de teclas de atalho para todas as opções disponíveis.

A aplicação que simula o mapa ferroviário e o respectivo tráfego pode funcionar em dois modos distintos:

Modo de Simulação É o modo onde se desenrola a simulação, isto é, onde se observa o resultado da interacção entre os vários agentes componentes do sistema. Neste modo não é permitida a alteração da estrutura do mapa ferroviário.

Modo de Edição Serve para criar os mapas ferroviários e cada um dos seus componentes. É possível ao utilizador especificar a topologia, o número de troços, o número de comboios e de estações, bem como o respectivo número de plataformas de paragem associadas. Foi prevista a leitura e gravação de mapas em ficheiro.

Quando o simulador está em modo de edição não aceita ligações de agentes, quebrando as que existirem no momento da activação.

Ambos os modos têm aparência semelhante, variando apenas as opções permitidas e correspondentes funcionalidades. A figura A.2 mostra as quatro zonas em que o *interface* com o utilizador se divide:

Zona 1 - Zona Principal de Visualização Esta é a zona onde se visualiza o desenrolar da simulação e se constroem os mapa ferroviários. Nesta zona, o utilizador pode navegar por todo o mapa ajustando a aproximação desejada.

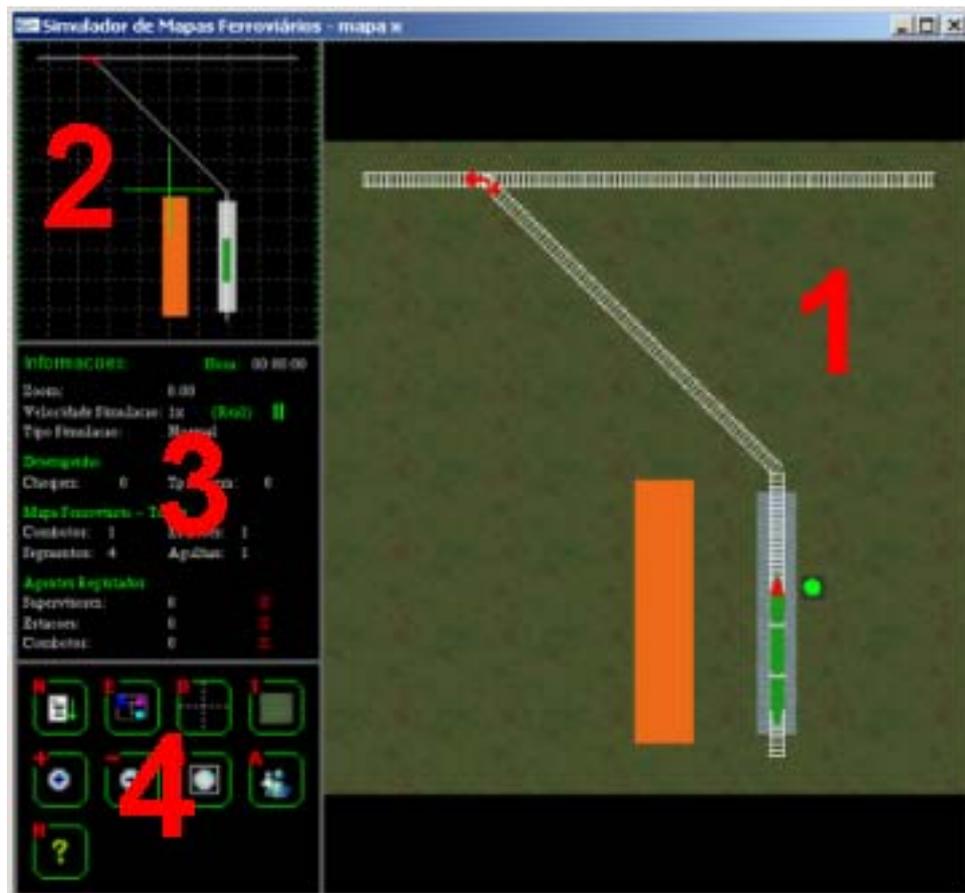


Figura A.2: Definição das quatro zonas do *interface*.

Zona 2 - Visão Global Esta zona proporciona uma visão do mapa ferroviário na sua totalidade, por forma a orientar o utilizador relativamente à área a visualizar na zona 1. É também uma zona interactiva uma vez que o utilizador pode, ao carregar com o rato sobre esta zona, seleccionar a área visualizada na zona 1.

Zona 3 - Informação ao Utilizador Aqui é mostrada informação de interesse para o utilizador, como o número de agentes registados, o total de objectos do mapa ou a ocorrência de choques. Funciona também como ajuda às opções disponíveis para o utilizador. Este aspecto é especialmente útil no modo de edição para facilitar a criação, selecção ou eliminação de componentes.

Zona 4 - Comandos Permitidos Neste painel aparecem as opções permitidas ao utilizador, consoante o modo e o estado do simulador. Qualquer opção pode ser tomada através do clique com o rato no botão ou da tecla de atalho que cada um tem associada. Existem comandos específicos do modo de simulação, de edição ou comuns a ambos. Estes últimos são os seguintes:



Tecla + - Aumenta a aproximação ao mapa ferroviário (*Zoom in*).



Tecla – - Diminui a aproximação ao mapa ferroviário (*Zoom out*).



Tecla I - Coloca a aproximação por defeito, visualizando-se o mapa completo.



Tecla T - Oculta / mostra as texturas dos objectos.



Tecla ← - Mover visão do mapa para a esquerda.



Tecla → - Mover visão do mapa para a direita.



Tecla ↑ - Mover visão do mapa para cima.



Tecla ↓ - Mover visão do mapa para baixo.



Tecla P - Propriedades de um comboio. Define propriedades relativas ao comboio seleccionado e ao respectivo horário (sequência de locais de paragem).



Tecla Q - Terminar a aplicação.

A.2.1 Simulação

Com o modo de simulação activado, as quatro zonas anteriormente enunciadas têm as seguintes funções:

Zona Principal de Visualização Zona onde se desenrola a simulação e onde se podem seleccionar os objectos que nela interagem. É possível ao utilizador, através do rato, seleccionar qualquer objecto interveniente no processo de simulação (troços, comboios, estações ou plataformas de paragem), aparecendo as respectivas propriedades no painel de informação da aplicação. Esta funcionalidade é mais relevante para o caso dos comboios, permitindo saber qual a sua velocidade instantânea, a próxima paragem e utilidade associada.

Visão Global Mostra o aspecto geral do mapa ferroviário dando realce aos comboios que nela circulam. Em mapas de grandes dimensões é necessário aumentar a escala dos comboios para que permaneçam visíveis.

Informação ao Utilizador Apresenta ao utilizador informação sobre o processo de simulação, ou um objecto nela interveniente, caso o utilizador o tenha seleccionado. Se não tiver sido seleccionado nenhum objecto, o seu aspecto é o da figura A.3.

```

Informacoes:          Hora: 00:00:00
Zoom:                0.00
Velocidade Simulacao: 1x (Real) ||
Modo Simulacao:      Automatico

Desempenho          Choques: 0
Vel Media: 0 STOP: 0 Prop.Vel: 0.000

Mapa Ferroviario - Totais
Comboios: 8 Estacoes: 8
Comprimento: 73.4 Km. Agulhas: 72
Parag.Prev.: 45 Parag.Efect.: 0

Agentes Registrados X
Supervisores: 0 Comboios: 0
Estacoes: 0 Aprender: 0

```

Figura A.3: Informação disponibilizada sem objectos seleccionados.

Segue-se uma descrição de cada um dos items mostrados quando o utilizador não seleccionou nenhum objecto:

Hora Tempo decorrido de simulação. Cada vez que a simulação é inicializada este valor passa a zero.

Zoom Aproximação actual.

Velocidade Simulação Velocidade com que se desenrola a simulação.(Câmara-lenta, tempo-real ou acelerado).

Modo Simulação Indica se o sistema está a ser controlado no modo automático ou manual.

Choques Número de choques ou descarrilamentos ocorridos. Serve como medida de avaliação do sistema de controlo.

STOP Número de paragens forçadas em resultado de ordens de controlo.

Prop.Vel Proporção de tempo que os comboios da rede viajam a velocidade aproximada à desejada. Consiste no complemento para um do valor médio de utilidade dos agentes *Comboio* registados na rede.

Comboios Total de comboios presentes na simulação.

Estações Total de estações presentes na simulação.

Segmentos Total de segmentos que compõem o mapa ferroviário.

Agulhas Total de pontos de selecção de trajectória. Correspondentes à intersecção de troços de linha.

Parag. Previstas Total de paragens previstas para os comboios existentes na rede.

Parag. Efectuadas Total de paragens efectuadas pelos comboios existentes na rede.

Agentes Supervisores Total de agentes supervisor registados. Para efectuar uma simulação tem que existir pelo menos um.

Agentes Estações Total de agentes estação registados. Se este valor for inferior ao número de estações aparece um sinal de aviso.

Agentes Comboios Total de agentes comboio registados. Se o valor for inferior ao número total de comboios, aparece um sinal de aviso.

Comandos Disponíveis De acordo com o objecto seleccionado, esta zona mostra os comandos passíveis de execução.



Tecla I - Inicialização da simulação.



Tecla P - Avanço da Simulação (*Play*).



Tecla P - Paragem momentânea da simulação (*Pause*).



Tecla S - Fim da simulação (*Stop*)



Tecla A - Propriedades relativas aos agentes.



Tecla > - Incremento da velocidade de simulação.



Tecla < - Decremento da velocidade de simulação.



Tecla D - Activa / desactiva a mostragem das áreas associadas a cada agente supervisor.



Tecla X - Termina a simulação e mostra uma janela auxiliar com estatísticas a esta relativas.



Tecla X - Fecha a janela auxiliar de final de simulação.



Tecla F - Fixa o centro da zona principal de visualização ao comboio actualmente seleccionado. Permite seguir o percurso do comboio sem necessidade de actualizar a área visualizada.



Tecla F - Termina o modo de fixação do écran a um comboio.



Tecla R - Mostra na zona principal de visualização o trajecto definido para o comboio seleccionado.



Tecla R - Deixa de mostrar o trajecto para o comboio seleccionado.



Tecla Z - Propriedades de um segmento. Serve para gerar situações de excepção simulando a avaria de troços de linha.



Tecla V - Aumenta a velocidade de um comboio. Permite o controlo manual do comboio seleccionado.



Tecla B - Trava o comboio. Utilizado para o mesmo efeito do comando anterior.



Tecla M - Activação do modo automático de controlo dos comboios.



Tecla M - Activação do modo manual de controlo dos comboios.



Tecla E - Activação do modo de edição.

A.2.2 Edição

A aplicação é iniciada no modo de simulação, pelo que para activar o modo de edição o utilizador terá que carregar no botão acima especificado, ou digitar "E".

Este modo serve para o utilizador criar, alterar ou eliminar mapas ferroviários, bem como guardar ou recuperar mapas a partir de ficheiro. É possível a definição da topologia das linhas e a introdução, alteração ou eliminação de componentes, sejam eles comboios, estações ou plataformas de paragem.

A organização do espaço disponível é feita de forma similar ao modo de simulação. Neste caso cada uma das quatro zonas tem a seguinte função:

Zona Principal de Visualização Zona onde o utilizador cria novos objectos, ou selecciona os existentes.

Visão Global Mostra o mapa na sua globalidade.

Informação ao Utilizador Informa o utilizador do que deve ser feito tendo em conta o comando executado.

Comandos Disponíveis Tal como no modo de simulação, mostra os comandos disponíveis para o respectivo modo. Neste caso são:



Tecla N - Criação de novos componentes. Serve para definir o tipo de componente que se pretende criar.



Tecla X - Eliminação dos componentes seleccionados. Ao carregar com o rato sobre um componente efectua-se a sua selecção. Podem-se seleccionar vários componentes de diferentes tipos, aparecendo estes destacados na zona 1.



Tecla Z - Eliminação de todos os componentes. Apaga o mapa ferroviário por completo.



Tecla L - Leitura de Mapas. Efectua a leitura de mapas a partir de ficheiro.



Tecla G - Gravação de Mapas. Serve para efectuar a gravação do mapa actualmente em memória em ficheiro.

A.2.3 Criação de um Mapa Ferroviário - Exemplo

O aspecto inicial da aplicação será o da figura A.4, devendo o primeiro passo corresponder à activação do modo de edição (Tecla E).

Para criar um novo componente (Comboio, Linha, Estação ou Plataforma de Paragem) deve ser carregado o botão com a tecla N no canto superior esquerdo ou digitado o respectivo carácter. É apresentada ao utilizador uma janela (Figura A.5) onde deve ser especificado o tipo de componente desejado.



Figura A.4: Início do processo de construção.

A.2.4 Definição da Topologia

A fase inicial consiste na definição da topologia da rede ferroviária, através da construção de todos os troços que a compõem.

Para inserir um troço de linha, o utilizador basta clicar com o rato (Zona 1 - Figura A.2) no ponto de início do troço e arrastá-lo até ao fim desejado.

Caso o novo troço intercepte algum anteriormente definido a aplicação calcula o ponto de intersecção e cria uma agulha de selecção de rota. A figura A.6 mostra a aparência da zona principal de simulação após a definição de dois troços.

Este processo pode ser iterativamente repetido até que a topologia do mapa esteja completamente definida.

A.2.4.1 Criação de Estações e Plataformas de Paragem

Após a definição da topologia, pode-se passar para a definição das estações e respectivas plataformas de paragem.

Para criar uma nova estação basta seleccionar na janela da figura A.5 o respectivo tipo,

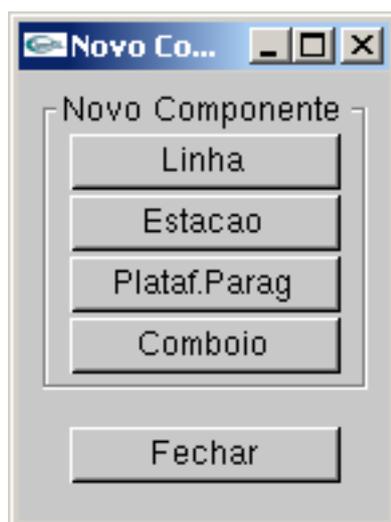


Figura A.5: Seleção de um novo componente.

atribuir-lhe um nome e definir a área que esta vai ocupar. Com o rato carrega-se num ponto extremo (Zona 1 - Figura A.2) e arrasta-se até à definição dos limites pretendidos.

A criação de plataformas de paragem está dependente da prévia existência de estações uma vez que cada plataforma é automaticamente associada à estação mais próxima.

Para criar uma plataforma de paragem basta carregar com o rato em dois locais sucessivos do mesmo segmento, tendo no final uma aparência semelhante à da figura A.6.

A.2.4.2 Criação dos Comboios

Tendo criados os componentes fixos da rede ferroviária (troços, estações e plataformas de paragem) pode-se proceder à criação dos comboios que nela vão transitar.

Seleccionando, como nos casos anteriores, o tipo de componente desejado, é mostrada uma janela onde se especificam as características do novo comboio (Número de carruagens, orientação inicial, prioridade, etc...). Depois de preenchida esta informação basta carregar com o rato sobre o troço de linha onde o comboio inicialmente estará localizado.

A.2.4.3 Definição de Trajectos

A etapa final de criação de um mapa ferroviário consiste na definição da sucessão de paragens previstas para cada comboio, que definirá o trajecto de cada um.

Para definir as paragens deve-se clicar com o rato sobre o comboio específico, e seguidamente carregar no botão da zona 4 (Figura A.2) com a letra *C* no canto superior esquerdo.

Abre-se uma janela de propriedades na qual se deve clicar no botão *Novo* dentro do painel

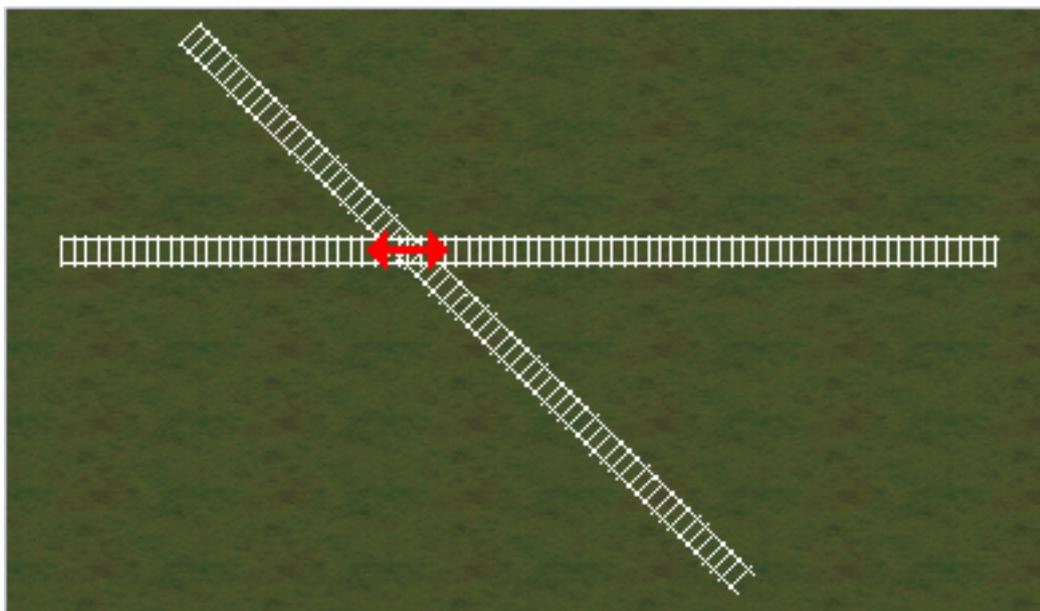


Figura A.6: Definição da topologia.

de *Horário*.

Finalmente o utilizador deve carregar com o rato sobre as estações onde, sequencialmente, o comboio deve parar (Zona 1 - Figura A.2). Quando tiver definido todas as paragens pretendidas basta clicar noutro qualquer componente que não uma estação.

Após o término deste processo para os comboios existentes na rede, está concluído o processo de criação de uma rede ferroviária, devendo a zona principal de visualização ter um aspecto semelhante ao apresentado na figura A.8.

A.2.5 Criação dos Agentes

Após a criação de todos os componentes do mapa ferroviário e activação do modo de simulação, podem ser criados os agentes que vão intervir no sistema.

Pelas suas características de autonomia, basta ao utilizador iniciar a sua execução da forma especificada no apêndice A.1.

O número de agentes *Comboio* e *Estação* deve ser igual ao respectivo número de objectos existentes na simulação. O número de agentes supervisor deve ser superior ou igual a um.

Com os agentes criados e tendo estes efectuado o respectivo registo no simulador (automático) pode-se iniciar a simulação.

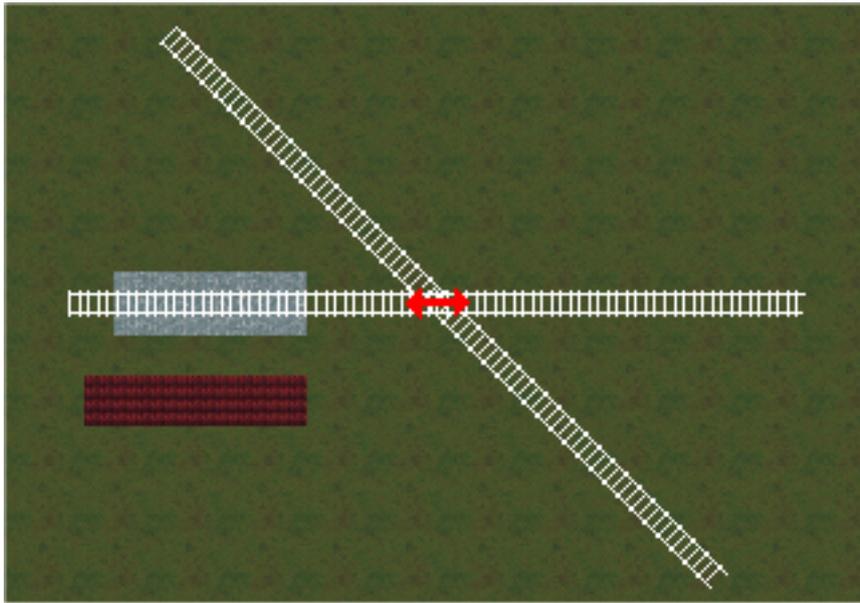


Figura A.7: Definição de componentes fixos.

A.2.6 Início da Simulação

Para iniciar a simulação basta carregar no botão com a tecla *N* que efectua a sua inicialização, e seguidamente no botão *Play*.

Caso algum dos pré-requisitos necessários para iniciar a simulação não seja satisfeito é mostrada uma janela indicadora do erro cometido. Este pode ser um dos seguintes:

- Número de supervisores insuficiente.
- Número de agentes *Estação* insuficiente.
- Número de agentes *Comboio* insuficiente.

Está disponível informação na zona 3, como o tempo e velocidade de simulação, o número de agentes registado ou o número de choques ocorridos. O utilizador pode alterar parar, acelerar, abrandar ou reiniciar a simulação a qualquer momento.

A.2.7 Modos de Controlo

O sistema MARCS está preparado para funcionar em dois modos distintos de controlo dos comboios: Modo automático e modo manual.

No modo automático, modo de funcionamento por defeito, o controlo da velocidade e trajecto de cada comboio é automaticamente assegurado pelo sistema, não deixando reservado nenhum papel activo ao utilizador.

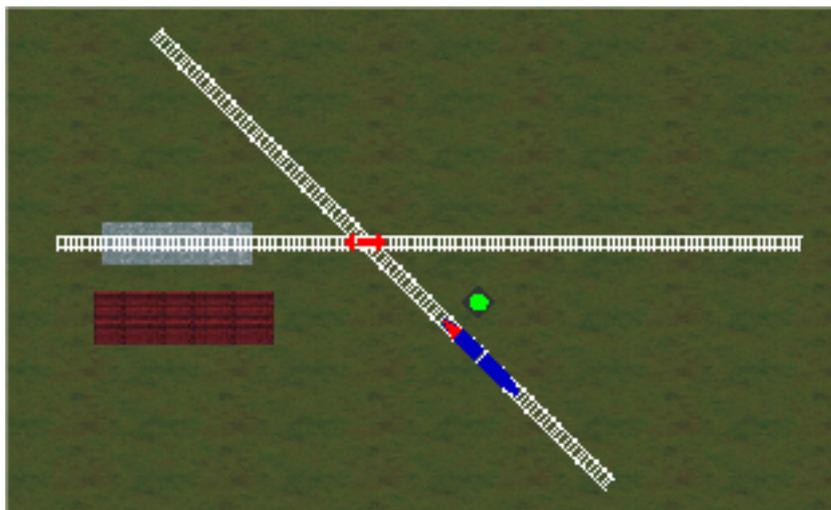


Figura A.8: Rede ferroviária definida.

Quando o utilizador desejar introduzir novas situações de tráfego, ou quando o sistema for incapaz de resolver algum conflito de interesses, pode ser activado o modo manual de controlo. Neste modo é permitido ao utilizador accionar as agulhas de selecção de trajecto e definir a velocidade para cada comboio, estando os agentes supervisores desactivados.

Para controlar individualmente uma composição basta seleccioná-la com o rato e depois accionar os comandos que permitem ajustar a velocidade pretendida (teclas "v" e "b").

Quando o utilizador desejar accionar uma agulha de selecção de trajecto, basta carregar com o rato sobre ela na zona principal de visualização. Automaticamente a agulha passa ao próximo estado coerente.

Apêndice B

Detalhes de Implementação

Neste apêndice apresentam-se aspectos complementares relativos à implementação do sistema MARCS. O ambiente de desenvolvimento utilizado, a estrutura dos agentes, o tipo de *sockets* implementado, as classes e estruturas criadas para cada aplicação.

B.1 Ambiente de Desenvolvimento

Como ambiente de desenvolvimento para o sistema MARCS (simulador e agentes) utilizei o "Microsoft Visual C++ 6.0". Demonstrou ser fiável e robusto, não tendo durante o processo de implementação e refinamento apresentado qualquer problema. O elevado volume de documentação e tutoriais disponíveis, o sistema de *debug*, a possibilidade de implementação de aplicações genéricas e a interacção agradável e intuitiva com o utilizador foram alguns dos factores que motivaram a escolha ¹ :

Tutoriais Juntamente com o ambiente de desenvolvimento pode ser instalado um elevado número de tutoriais que funcionam como suporte à realização de diversas tarefas (*MSDN- Microsoft Developer Network*). Paralelamente existe disponível na WEB um manancial de forums, tutoriais e exemplos onde grande parte dos problemas surgidos durante a implementação podem ser resolvidos.

Auto-compleição Outra das características que se revela bastante útil é o sistema que automaticamente apresenta ao programador informação sobre as acções permitidas para cada situação, quer sejam métodos disponíveis para um determinado tipo de objecto ou parâmetros com que uma função deve ser chamada. Desta forma é possível dirigir a concentração para o que é verdadeiramente importante, deixando de se preocupar com a memorização de parâmetros, métodos ou propriedades de componentes.

Visualização de métodos e classes dinâmica Possui uma área onde os métodos, classes ou variáveis implementadas são dinamicamente actualizados, característica relevante

¹<http://www.microsoft.com/catalog/display.asp?site=755&subid=22&pg=2>

tendo em conta que rapidamente a quantidade de informação a reter atinge proporções elevadas.

Edição durante o processo de *debug* (“*Edit and Continue*”) É possível fazer alterações ao código durante o *debug*, evitando que por cada falha detectada seja necessário parar o *debug* e reiniciá-lo posteriormente.

Suporte do ANSI C++ Suporta as mais recentes especificações ANSI/ISO, pelo que qualquer aplicação desenvolvida (desde que não inclua componentes ou classes específicas da *Microsoft*) será passível de compilação e execução noutra ambiente de desenvolvimento.

A figura B.1 mostra a janela de interação com o utilizador do ambiente de desenvolvimento utilizado. Conforme se descreve na tabela B.1, pode-se dividir em 4 áreas principais que apoiam o programador nas tarefas de codificação. O número de zonas visíveis e a aparência de cada uma é personalizável por forma a ajustar-se às preferências de cada um.

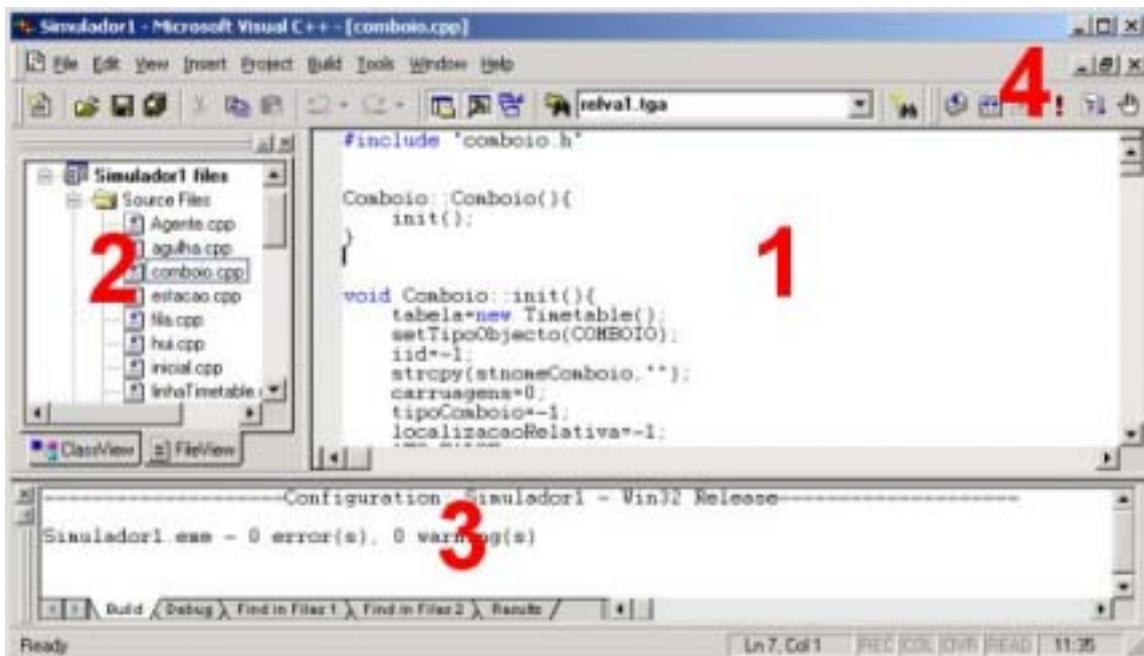


Figura B.1: Ambiente de desenvolvimento *Microsoft Visual C++*.

B.2 Detalhes de Implementação - Agentes

B.2.1 Transmissão da Informação

Na sua forma mais simples, a recepção de informação através de *sockets* bloqueia a aplicação.

Tal como numa instrução de leitura a partir do teclado (`scanf()`) a aplicação fica indefinidamente à espera até receber a informação pretendida, também no caso dos *sockets* a instrução de leitura obriga à espera de um determinado número de *bytes*.

Além disto é necessário sincronizar as duas aplicações, no sentido de saber qual a ordem pela qual cada uma vai ler e escrever no *socket*.

No sistema MARCS não existe este tipo de restrições. A ligação entre dois agentes é sempre feita através de dois *sockets*, cada um para leitura e escrita exclusiva por parte da respectiva aplicação.

Por outro lado foi implementado um sistema de "*non-blocking sockets*" que permite saber antecipadamente quando existe informação para ler no *socket* e evita as situações de bloqueio.

B.2.2 Tipo de Sockets

Sempre que possível foram utilizadas bibliotecas ANSI C, o que facilita a hipotética migração do sistema para outras plataformas ou sistemas operativos.

Na parte relativa aos *sockets* tal não foi possível, uma vez que as funções que os implementam são específicas de cada sistema operativo.

A versão implementada corresponde à do sistema operativo *Windows*, especificando-se de seguida os aspectos mais relevantes na criação e transmissão de informação através de *sockets*.

Inicialmente é necessário definir o tipo e família de *sockets* desejado.

Área	Função
1	Nesta zona efectua-se a implementação do código.
2	Listagem dos ficheiros pertencentes ao projecto, estruturas, funções, classes, métodos e variáveis implementadas.
3	Visualização dos erros ou avisos relativos aos processos de compilação e de construção do executável.
4	Barras personalizáveis para os comandos mais utilizados pelo implementador.

Tabela B.1: Ambiente de desenvolvimento (Zonas)

Existem duas famílias de sockets: *AF_UNIX* e *AF_INET*. A primeira serve para a comunicação entre processos de máquinas *UNIX*, enquanto a segunda é utilizada para comunicação entre aplicações através da rede (Frost 1999).

Quanto ao tipo, os dois mais comuns são o *SOCK_STREAM* e *SOCK_DGRAM*:

SOCK_STREAM Disponibiliza mecanismos de transmissão fiável de sequência de *bytes* sobre o protocolo TCP.

SOCK_DGRAM Permite a transmissão de datagramas. Um datagrama consiste numa estrutura de tamanho fixo (normalmente pequena) e indivisível. Utiliza o protocolo UDP, retirando-lhe alguma fiabilidade na transmissão.

B.2.2.1 Implementação

Para criar um novo *socket* é usada a função **socket()**. Esta tem o seguinte protótipo:

```
SOCKET socket (int af, int type, int protocol );
```

Parâmetros:

- *af* - Família do *socket*.
- *type* - Tipo do *socket*.
- *protocol* - Protocolo sobre o qual vão comunicar.

A função retorna um descritor (identificador inteiro) para o *socket* criado.

A fase seguinte consiste na atribuição do endereço e porto ao *socket* anteriormente criado. Para isso utiliza-se a função:

```
int bind (SOCKET s,const struct sockaddr FAR* name,int namelen );
```

Parâmetros:

- *s* - Identificador do *socket*.
- *name* - Endereço e porto atribuído.
- *namelen* - Tamanho da estrutura FAR em *bytes*.

Esta função retornará zero se tudo tiver corrido bem, ou outro valor em caso de ocorrência de algum erro.

Depois de ter o *socket* criado deve-se definir o número máximo de pedidos de ligação pendentes, utilizando para isso a função **listen()**:

```
int listen (SOCKET s,int backlog);
```

Parâmetros:

- *s* - Identificador do *socket*.
- *backlog* - Total de pedidos de ligação.

Após a configuração do *socket* pode-se passar para a fase de estabelecimento de ligações.

Na aplicação servidor aceitam-se pedidos de ligação através da função **accept()**:

```
SOCKET accept (SOCKET s,struct sockaddr FAR* addr,int FAR* addrlen );
```

Parâmetros:

- *s* - Identificador do *socket*.
- *addr* - Estrutura que regista o endereço da aplicação que está a estabelecer a ligação
- *addrlen* - Tamanho da estrutura que regista o endereço.

Os pedidos de ligação são feitos do lado da aplicação cliente através da função **connect()**.

```
int connect (SOCKET s, const struct sockaddr FAR* name,int namelen );
```

- *s* - Identificador do *socket*.
- *name* - Estrutura com o endereço onde se vai efectuar a ligação.
- *namelen* - Tamanho da estrutura que regista o endereço.

A partir do momento em que a ligação é efectuada pode-se transmitir informação através do *socket* que liga as duas aplicações. São usadas as funções **send()** e **recv()** respectivamente para enviar e receber informação.

send() usado para enviar informação:

```
int send (SOCKET s, const char FAR * buf, int len, int flags );
```

Parâmetros:

- *s* - Identificador do *socket*.
- *name* - Informação a transmitir.
- *namelen* - Número de *bytes* que se pretendem transmitir.
- *flags* - Parâmetro opcional de configuração da transmissão.

recv() usado para receber informação:

```
int recv (SOCKET s, char FAR* buf, int len, int flags );
```

Parâmetros:

- *s* - Identificador do *socket*.
- *name* - Estrutura que guarda a informação recebida.
- *namelen* - Número de *bytes* que se pretendem receber.
- *flags* - Parâmetro opcional de configuração da transmissão.

B.2.2.2 *Non-Blocking Sockets*

No sistema implementado era imperioso que as aplicações não esperassem indefinidamente na recepção de informação através de *sockets*.

Foi implementado um sistema que permite saber antecipadamente a quantidade de informação à espera de ser recebida e efectuar a respectiva leitura.

Neste caso, ao ser notificado da existência de informação num determinado *socket*, cada agente procede à sua leitura e continua a execução.

A função que permite saber antecipadamente se existe informação para ler, se é possível a escrita, ou se aconteceu alguma excepção num *socket* é a **select()**:

```
int select (int nfds,fd_set FAR * readfds,fd_set FAR * writefds, fd_set FAR * exceptfds,  
           const struct timeval FAR * timeout );
```

Parâmetros:

- *nfds* - Parâmetro ignorado, colocado apenas por razões de compatibilidade com versões anteriores.
- *readfs* - Conjunto de descritores identificando os *sockets* de leitura de informação.
- *writefds* - Conjunto de descritores identificando os *sockets* de escrita de informação.
- *exceptfds* - Conjunto de descritores identificando os *sockets* em que serão verificadas exceções.
- *timeout* - Tempo de espera.

Tal como indicado acima, esta função recebe um conjunto de descritores identificadores de *sockets* e devolve informação sobre os que estão aptos para escrita, os que contêm informação à espera de ser lida e aqueles onde ocorreram exceções. O seu esquema de funcionamento é o seguinte:

1. Colocam-se os *sockets* sobre os quais se pretende informação nos respectivos conjuntos de descritores. Para tal utiliza-se a função `FD_SET(conjunto, descritor)`
2. Efectua-se uma chamada à função `select()` tendo como parâmetros os conjuntos criados no passo anterior.
3. Esta função vai verificar os descritores passados como parâmetro e devolver apenas aqueles onde existe respectivamente informação para ler, passíveis de escrita ou com ocorrência de exceções.
4. Através da função `FD_ISSET(conjunto, descritor)` pode-se ver se um determinado descritor ainda está num conjunto. Se um descritor ainda estiver no conjunto *readfs* indica que nesse *socket* existe informação para receber.
5. Sabendo quais os *sockets* onde se pode receber e enviar informação pode-se agora chamar as respectivas funções de `send()` e `recv()` sem correr o risco de bloqueio da aplicação.

B.2.3 Estrutura dos Agentes

Os agentes implementados (*Simulador*, *Supervisor*, *Comboio*, *Estação* e *Aprender*) partilham a mesma estrutura e processos de comunicação, estando a sua representação na figura B.2.

A *thread* 1 está permanentemente à escuta de mensagens e de novas ligações. Ao detectar a recepção de uma mensagem completa (através da recepção do carácter "#") coloca-a na fila de tratamento de mensagens.

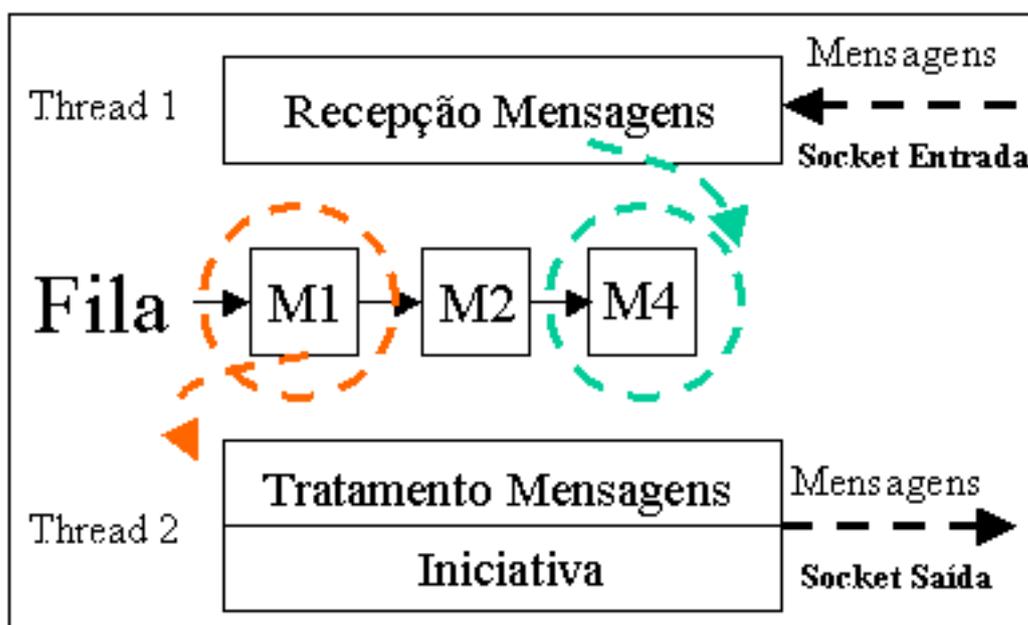


Figura B.2: Estrutura dos agentes.

A segunda *thread* é responsável pelo tratamento das mensagens recebidas, e o que confere aos agentes características de proactividade, pela análise sistemática do "estado do mundo" e iniciativa de comunicação com os restantes.

Os algoritmos B.1 e B.2 são relativos à execução das duas *threads* que compõem os agentes.

Algoritmo B.1 Recepção de mensagens

```

while true do
  NovasLigacoes ← verifica_Novas_Ligacoes(ListaLigacoes)
  if NovasLigacoes then
    adicionar_Lista(NovasLigacoes,ListaLigacoes)
  end if
  NovasMensagens ← verifica_Novas_Mensagens(ListasLigacoes)
  if NovasMensagens then
    adicionar_Fila(NovasMensagens,FilaMensagens)
  end if
end while

```

B.2.4 Modos de Recepção de Mensagens

Cada agente pode receber informação através de um *socket* em dois modos distintos: **Mensagem** e **Ficheiro**.

Algoritmo B.2 Tratamento de mensagens e iniciativa de actuação

```

while true do
  NovaMensagem ← retira_Fila_Mensagens(FilaMensagens)
  if NovaMensagem then
    trata_Mensagem(NovaMensagem)
  end if
  iniciativa()
end while

```

B.2.4.1 Modo Mensagem

Este é o modo de funcionamento por defeito. No modo *Mensagem* toda a informação recebida é enviada para um *buffer* até que seja lido o carácter "#". Este denota a separação entre mensagens pelo que, ao recebê-lo, o agente agrupa a informação existente no respectivo *buffer* e cria uma nova mensagem. Caso exista informação remanescente ela é deixada no *buffer* até à recepção do próximo separador. As figuras B.3 e B.4 ilustram o processo de transmissão de informação neste modo.

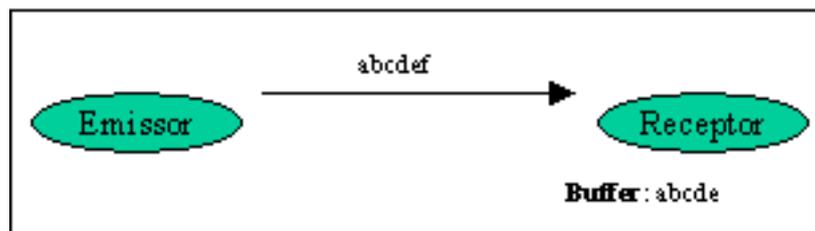


Figura B.3: Envio de informação (Modo Mensagem).



Figura B.4: Criação de uma nova mensagem (Modo Mensagem).

B.2.4.2 Modo Ficheiro

Este modo serve para efectuar a transmissão de ficheiros entre agentes, devendo no momento da activação o agente receptor já conter informação sobre o nome e remetente do ficheiro a transmitir.

Este vai abrir um descritor para ficheiro e aí colocar toda a informação recebida através do *socket*.

O caracter ” \backslash ” indica o final de transmissão do ficheiro, pelo que o agente receptor deve fechar o descritor e reactivar o modo de *Mensagem*.

Este processo é ilustrado pelas figuras B.5 e B.6.

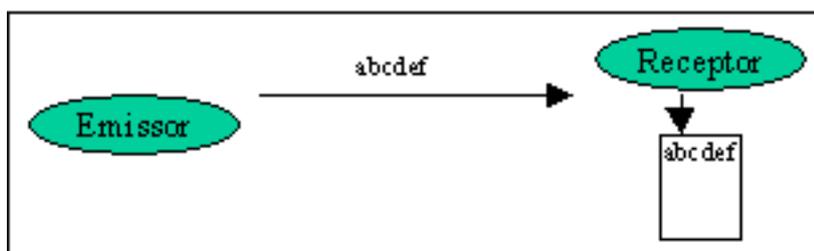


Figura B.5: Envio de informação (Modo *Ficheiro*).

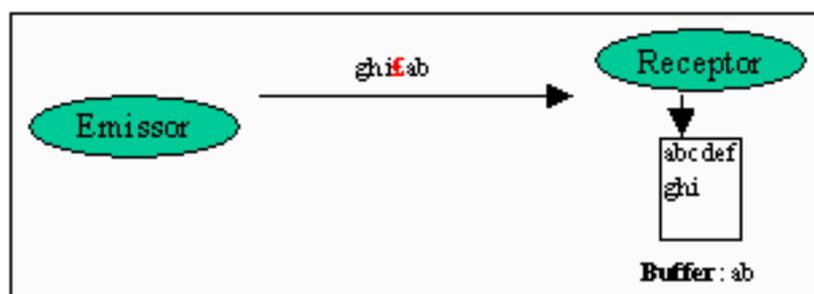


Figura B.6: Término do modo *Ficheiro*.

B.3 Classes Implementadas

Nesta secção descrevem-se as classes implementadas para cada uma das aplicações que compõem o sistema.

B.3.1 Aplicação de Simulação

Comboio Subclasse de "ObjectoVisual". Cada instância desta classe guarda informação relativa a um comboio.

Agulha Subclasse de "ObjectoVisual". Representa um ponto onde é possível optar por mais que uma direcção. Resulta do cruzamento de linhas.

Estacao Subclasse de "ObjectoVisual". Representa uma estação ferroviária.

ObjectoVisual Classe abstracta que agrupa as características dos objectos passíveis de representar graficamente na simulação.

PlataformaParagem Subclasse de "ObjectoVisual". Representa um local (normalmente contíguo a uma estação) destinado à paragem de comboios.

Segmento Subclasse de "ObjectoVisual". Representa um troço de linha ferroviária. Consiste numa linha recta a unir dois pontos.

Timetable Cada instância desta classe contém uma sequência de pontos de paragem (ordenados cronologicamente) para um determinado comboio.

LinhaTimetable Representa uma hora de chegada ou de partida para um comboio.

B.3.2 Agente Supervisor

Agente Classe abstracta que implementa o comportamento comum a todos os agentes que compõem o sistema.

AgenteSupervisor Sub-classe de "Agente". Implementa o comportamento específico dos agentes responsáveis pela supervisão do tráfego.

B.3.3 Agente Comboio

Agente Classe abstracta que implementa o comportamento comum a todos os agentes que compõem o sistema.

AgenteComboio Sub-classe de "Agente". Implementa o comportamento específico dos agentes representantes de cada comboio circulante na rede.

B.3.4 Agente Estação

Agente Classe abstracta que implementa o comportamento comum a todos os agentes que compõem o sistema.

AgenteEstacao Sub-classe de "Agente". Implementa o comportamento específico dos agentes representantes das estações existentes na rede.

B.3.5 Agente Aprender

Agente Classe abstracta que implementa o comportamento comum a todos os agentes que compõem o sistema.

AgenteAprender Sub-classe de "Agente". Implementa o comportamento específico dos agentes incumbidos de inferir regras de optimização e adaptação do fluxo de tráfego da rede.

B.4 Estrutura Genérica de Informação

A informação relativa à cooperação entre agentes supervisores é mantida numa estrutura genérica. Esta estrutura, esquematizada na figura B.7, permite no mesmo vector guardar elementos de qualquer tipo.

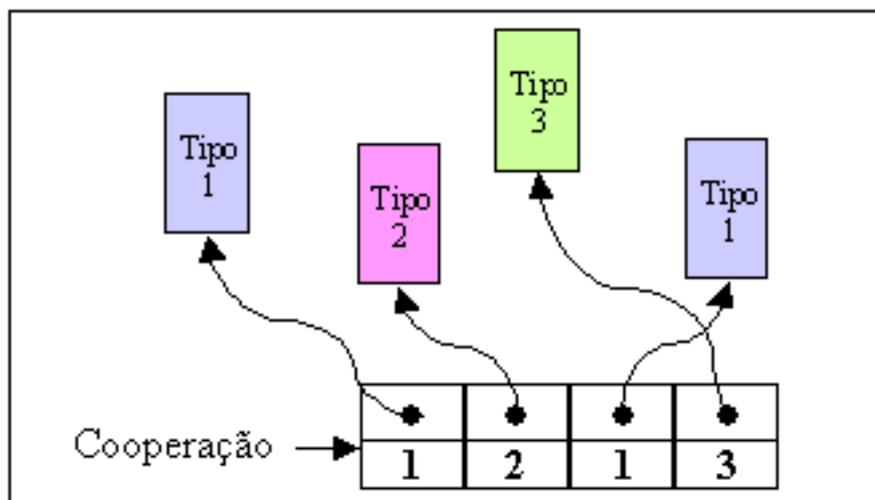


Figura B.7: Estrutura genérica de informação.

Este facto propicia a sofisticação dos processos de cooperação, podendo ser utilizada para guardar elementos de tipos que venham a ser criados em resultado de trabalho futuro.

Apêndice C

Descrição das Mensagens

Este apêndice é dedicado à descrição das mensagens trocadas entre os agentes intervenientes no sistema. Para cada uma delas descreve-se o(s) agente(s) emissor(es) e receptor(es), bem como o tipo e quantidade de parâmetros necessários para a sua correcta utilização.

As mensagens aparecem descritas na seguinte forma:

De: Campo 1	Para: Campo 2	Performativa: Campo 3
Mensagem: Campo 4		
Parâmetros: Campo 5		
Observações: Campo 6		

Cada campo tem o seguinte significado:

- **Campo 1** - Tipo(s) de agente(s) que envia(m) a mensagem.
- **Campo 2** - Tipo(s) de agente(s) receptores da mensagem.
- **Campo 3** - Performativa da mensagem. Pode ser uma das seguintes: *ask*, *control*, *register*, *reply* e *tell*.
- **Campo 4** - Protótipo da mensagem. Neste campo aparece a sequência de caracteres transmitida. Os itens parametrizáveis aparecem no seguinte formato: [**TIPO**]. O tipo pode ser um dos seguintes:
 - **D** : Números inteiros.
 - **F** : Números reais.
 - **S** : Cadeias de caracteres.
- **Campo 5** - Neste campo estão descritos o tipo e gama de valores permitidos para cada parâmetro da mensagem. A ordem de especificação de cada um tem correspondência directa com a ordem em que aparecem na mensagem.

- **Campo 6** - Campo de informação complementar relativa à mensagem.

Notação → Sempre que parâmetros aparecerem dentro de chavetas, indica que a mensagem pode conter sequências desses parâmetros. Por exemplo {[d]} indica um conjunto de parâmetros do tipo inteiro.

De: Supervisor	Para: Simulador	Performativa: alert
Mensagem: alert:sender [s]:receiver [s]:error [ComboioSemAgente — AgenteComboioDesconhecido] [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio		
Observações: Despoletar do estado de emergência		

De: Simulador	Para: Supervisor	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width TrajectoComboio:Comboio [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio		
Observações: Pergunta aos agentes supervisores qual o trajecto definido para o comboio. Estes passarão a enviar-lhe regularmente as alterações ao trajecto		

De: Supervisor	Para: Supervisor	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width Vizinhos		
Parâmetros: Agente emissor, Agente receptor		
Observações: Pesquisa de informação acerca dos vizinhos de um supervisor. Cada supervisor deve manter informação sobre os seus vizinhos e sobre os vizinhos de cada um destes		

De: Supervisor	Para: Comboio	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width proximaParagem		
Parâmetros: Agente emissor, Agente receptor		
Observações: Pedido de informação acerca da próxima paragem de um comboio		

De: Supervisor	Para: Supervisor	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply PlataformaParagemComboio:Estacao [s] Comboio [d] Comprimento [f]		
Parâmetros: Agente emissor, Agente receptor, ID da estação, ID do comboio, comprimento do comboio		
Observações: Supervisor pergunta ao vizinho pela plataforma de paragem para próxima estação. Este responderá se a estação estiver na sua área de supervisão		

De: Supervisor	Para: Supervisor	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width ParticipacaoPlano:Comboio [d] Aresta [d] Estacao [s]		
Parâmetros:		
Observações: Pergunta acerca da participação num plano conjunto		

De: Supervisor	Para: Estação	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width LocalParagem:Comboio [d] Plataforma [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Identificação da plataforma		
Observações: O supervisor pergunta à estação qual o local específico da plataforma onde o comboio deve parar		

De: Supervisor	Para: Estação	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width PlataformaParagem:Comboio [d] Comprimento [f] Parqueamento [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Comprimento do comboio		
Observações: Pergunta à estação sobre a plataforma de paragem indicada para o comboio		

De: Aprender	Para: Simulador	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width ListaSupervisores		
Parâmetros: Agente emissor, Agente receptor		
Observações: Pergunta acerca da lista de endereços dos agentes supervisores		

De: Aprender	Para: Supervisor	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width Log Porto [d] Inicio [d] Fim [d]		
Parâmetros: Agente emissor, Agente receptor, Porto de recepção de ligações, Tempo de início do log, Tempo final do log		
Observações: Pedido do registo de actividade entre início e fim		

De: Simulador, Supervisor	Para: Comboio	Performativa: ask
Mensagem: ask:sender [s]:receiver [s]:reply-width ValorUtilidade		
Parâmetros: Agente emissor, Agente receptor		
Observações: Pergunta sobre o valor de utilidade associado ao agente		

De: Supervisor	Para: Simulador	Performativa: control
Mensagem: control:sender [s]:receiver [s]:TipoObjecto Agulha LigacaoAgulha Agulha X [f] Y [f] Arestas [d] [d]		
Parâmetros: Agente emissor, Agente receptor, Latitude da agulha, Longitude da agulha, Aresta 1, Aresta 2		
Observações: Ordem de controle do supervisor sobre um ponto de selecção de trajecto		

De: Comboio	Para: Simulador	Performativa: control
Mensagem: control:sender [s]:receiver [s]:tipoObjecto [d] idObjecto [d] ordem [d]		
Parâmetros: Agente emissor, Agente receptor, Tipo do Objecto, Identificação do objecto, Código da ordem		
Observações: Ordem de controle sobre o comboio representado pelo agente		

De: Supervisor	Para: Comboio	Performativa: register
Mensagem: register:sender [s]:receiver [s]:reply-width SupervisaoRecebida porto [d]		
Parâmetros: Agente emissor, Agente receptor, Porto de recepção de mensagens		
Observações: Registo inicial do supervisor no comboio prestes a entrar na sua área de supervisão.		

De: Estação	Para: Comboio	Performativa: register
Mensagem: register:sender [s]:receiver [s]:reply-width OrientadorParqueamento Porto [d]		
Parâmetros: Agente emissor, Agente receptor, Porto de recepção de mensagens		
Observações: Apresentação da estação ao comboio que irá orientar nas manobras de estacionamento		

De: Supervisor, Comboio, Estação	Para: Supervisor, Comboio, Estação	Performativa: register
Mensagem: register:sender [s]:receiver [s]:reply-width status-Registry:type [d]:port [d]		
Parâmetros: Agente emissor, Agente receptor, Tipo do agente, Porto de recepção de mensagens		
Observações: Pedido de registo no simulador		

De: Simulador	Para: Supervisor, Comboio, Estação	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to status-registry:content [d]		
Parâmetros: Agente emissor, Agente receptor, Código da resposta		
Observações: Resposta a pedidos de registo		

De: Simulador	Para: Aprender	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to [d] Total [d] Agente [s] Endereco [s] Porto [d]		
Parâmetros: Agente emissor, Agente receptor, Total de agentes, Identificação do agente, Endereço do agente, Porto do agente		
Observações: Listagem com os endereços do tipo [d]		

De: Supervisor	Para: Comboio	Performativa: reply
Mensagem: tell:sender [s]:receiver [s]:content LocalParagem Actual [f] Velocidade [f] Paragem [f]		
Parâmetros: Agente emissor, Agente receptor, Localização relativa actual, Velocidade actual, Localização relativa do ponto de paragem		
Observações: Informação a indicar o local onde se deve processar a paragem na plataforma. O agente comboio deve calcular a força de travagem necessária para poder cumprir o local de paragem		

De: Supervisor	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to Vizinhos total [d] agente [s] endereco [s] porto [d]		
Parâmetros: Agente emissor, Agente receptor, Total de vizinhos, Identificação do agente vizinho, Endereço do agente vizinho, Porto de escuta de mensagens do agente vizinho		
Observações: Informação sobre os vizinhos do supervisor		

De: Supervisor	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to LocalParagemSupervisor:Comboio [d] Local [f]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio, Local de paragem		
Observações: Supervisor responde ao vizinho acerca do local onde deve ser feita a paragem do comboio		

De: Supervisor	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to PlataformaParagemComboio:Comboio [d] Plataforma [d] Aresta [d]		
Parâmetros: Agente emissor, Agente receptor, ID do Comboio, ID da plataforma de paragem, ID da aresta onde está localizada a plataforma de paragem		
Observações: Supervisor responde a anterior mensagem acerca da plataforma para um comboio cuja estação está na sua área de supervisão		

De: Supervisor	Para: Simulador	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to TrajectoComboio:Comboio [d] TotalTrajectos [d] [d] [d] ...		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Total de trajectos do comboio, Identificação do trajecto		
Observações: Informação com o trajecto do comboio		

De: Comboio	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to SupervisaoRecebida comboio [d] statusOk		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio		
Observações: Informação sobre a ligação ao novo agente supervisor		

De: Comboio	Para: Simulador, Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to ValorUtilidade:[f]		
Parâmetros: Agente emissor, Agente receptor, Valor de utilidade		
Observações: Resposta acerca do valor de utilidade associado ao agente		

De: Comboio	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to proximaParagem:content estacao [s] tempo [d] ultima [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação da estação, Tempo para a próxima paragem		
Observações: Informação sobre o local onde se deve processar a próxima paragem		

De: Estação	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to LocalParagem:Comboio [d] Local [f]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Localização relativa do ponto de paragem		
Observações: Informação acerca do local onde se deve processar a paragem de um comboio		

De: Simulador	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content partilhaTrajecto trajecto [d] supervisor [s]		
Parâmetros: Agente emissor, Agente receptor, Identificação do trajecto, Identificação do supervisor		
Observações: Informação sobre a partilha de um trajecto com outro supervisor		

De: Simulador	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content supervisor [s] endereco [s] porta [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do supervisor, Endereço do supervisor, porto de recepção de mensagens do supervisor		
Observações: Supervisor a que o comboio se deve ligar		

De: Simulador	Para: Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content característicasEstacao id [d] nome [s] totalPlataformas [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação da estação, Nome da estação, Total de plataformas de paragem		
Observações: Informação sobre as características da estação representada pelo agente		

De: Simulador	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content paragensComboio [d] total [d] Paragem [s] Tempo [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Total de paragens, Identificação da paragem, Tempo da paragem		
Observações: Informação sobre as paragens que o comboio representado pelo agente deve executar		

De: Simulador	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content característicasComboio [d] [d] [d] [d] [d] [f] Prioridade [d]		
Parâmetros: Agente emissor, Agente receptor		
Observações: Informação acerca das características do comboio representado pelo agente		

De: Simulador	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content EstadoAgulhas Total [d] Agulha [d] Estado [d] Estado [d]		
Parâmetros: Agente emissor, Agente receptor, Total de Agulhas, Identificação da agulha, Ligação segmento1, Ligação segmento2		
Observações: Informação acerca do estado (direcção) das agulhas localizadas na área de supervisão		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content SupervisaoAssumida comboio [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio		
Observações: O novo supervisor notifica o antigo que já efectuou a ligação ao comboio e assumiu a sua supervisão		

De: Supervisor	Para: Simulador, Comboio, Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content ParagemPlataforma Comboio [d] Plataforma [d] Origem [d] Destino [d] Local [f]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Identificação da plataforma, Vértice origem, Vértice Destino, Distância relativa do local de paragem		
Observações: Informação sobre a plataforma de paragem onde o comboio se imobilizou		

De: Supervisor	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content FimSupervisao		
Parâmetros: Agente emissor, Agente receptor		
Observações: Notificação do supervisor sobre o final da supervisão		

De: Supervisor	Para: Simulador	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content AssumpcaoZona AgenteAntigo [s] TotalZonas [d] X [f] [f] Y [f] [f]		
Parâmetros: Agente emissor, Agente receptor, Identificação do agente anterior, Total de zonas, Valores mínimos e máximos de latitude e longitude		
Observações: O supervisor notifica o simulador que, em resultado de um processo de negociação, assumiu a zona anteriormente supervisionada por outro agente		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content possivelTransicao comboio [d] agente [s] endereco [s] porto [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Identificação do agente representante, Endereço do agente, Porto do agente		
Observações: Informação acerca da possibilidade de um novo comboio entrar na zona supervisionada		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content Leilao [s] valorOcupacao [f]		
Parâmetros: Agente emissor, Agente Receptor, Agente que falhou, Índice de área		
Observações: Informação acerca da dimensão das zonas de supervisão de um agente		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content EliminaReservasComboio Comboio [d]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio		
Observações: Notificação para o agente eliminar todas as reservas feitas para o comboio		

De: Supervisor	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content VelocidadeMediaDesejada [f]		
Parâmetros: Agente emissor, Agente receptor, Velocidade desejada		
Observações: Notificação da velocidade desejada para o comboio		

De: Supervisor	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content VelocidadeMediaDesejada [f]		
Parâmetros: Agente emissor, Agente receptor, Velocidade		
Observações: Informação da velocidade média a que o comboio se deverá deslocar		

De: Supervisor	Para: Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content DistanciaObstaculo [f] Velocidade [f] Total-Manobras [d]		
Parâmetros: Agente emissor, Agente receptor, Distância livre, velocidade actual, Total de manobras em execução		
Observações: Informação da distancia do próximo obstáculo, para o comboio calcular a força de travagem necessária)		

De: Supervisor	Para: Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content EntradaParqueamento Comboio [d] Plataforma [d] Agente [s] Endereco [s] Porto [d]		
Parâmetros: Agente emissor, Agente receptor		
Observações: Informação de que o comboio deve iniciar as manobras de estacionamento		

De: Supervisor	Para: Aprender	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content ModoSocket [d]		
Parâmetros: Agente emissor, Agente receptor, Modo de transmissão de informação		
Observações: Selecciona o modo de escuta do socket entre os agentes		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content Congestionamento ID [d] Tipo [d] Total [d] Comboios [d] [d] Tempo [d]		
Parâmetros:		
Observações: Informação sobre a existência de congestionamento de tráfego		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content SupervisaoComboio ID [d] Agente [s] Endereco [s] Porto [d]		
Parâmetros: Agente emissor, Agente receptor, Endereço, Porto de escuta		
Observações: Supervisor informa os vizinhos que está a supervisionar um comboio		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content EliminaCandidaturasComboio Comboio [d]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio		
Observações: Notificação para o agente eliminar todas as candidaturas feitas para o comboio		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content GaranteEspacoManobra Comboio [d] Distancia [f] TotalTrajectos [d] [d] [d]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio, Distância a guardar, Total de elementos do trajecto, elementos do trajecto		
Observações: Pedido ao agente supervisor para garantir um determinado espaço pelo trajecto especificado		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content CandidaturaVertices Comboio [d] Distancia [f] TotalTrajectos [d] [d][d]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio, Distancia actual do vértice, Total de trajectos, Elementos do trajecto		
Observações: Informação sobre a necessidade de efectuar candidaturas para um determinado comboio		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content InformacaoVizinhosZonas TotalVizinhos [d] Vizinho [s] Endereco [s] Porto [d] TotalZonas [d] Zona X [f] [f] Y [f] [f]		
Parâmetros: Agente emissor, Agente receptor, Total de vizinhos, Nome vizinho, Endereço do vizinho, Porto do vizinho, Total de zonas, Inicio e fim latitude, Inicio e fim longitude		
Observações: Resposta sobre o número de zonas que supervisiona e os vizinhos que tem		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content PlataformaComboio Comboio [d] Plataforma [d] Aresta [d]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio, ID da plataforma, ID da aresta onde se encontra a plataforma		
Observações: Supervisor notifica os vizinhos acerca do local onde deve ser feita a paragem do comboio		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content TrajectoComboio ID [d] TotalTrajectos [d] [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Sequência de troços		
Observações: Informação do trajecto definido para um comboio		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content VelocidadeMediaDesejada Comboio [d] Velocidade [f] Tempo [d] Valor [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Velocidade desejada, Tempo de duração da velocidade desejada, Valor de duração actual para a velocidade desejada		
Observações: Notificação para os vizinhos saberem da velocidade desejada para o comboio		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content ProximaParagemComboio ID [d] Proximaparagem [s] Parqueamento [d]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio, Próxima estação onde o comboio deve parar		
Observações: Informação sobre a próxima estação onde o comboio deve parar		

De: Supervisor	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content ComboioAssumidoSupervisor ID [d] Agente [s]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Identificação do supervisor		
Observações: Informação de que o comboio já está a ser supervisionado por outro supervisor (os receptores vão tirá-lo dos prováveis)		

De: Estação	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content PermissaoPartida Comboio [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio		
Observações: Notificação de permissão dada ao comboio para abandonar a estação		

De: Aprender	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content FicheiroRecebido		
Parâmetros: Agente emissor, Agente receptor		
Observações: Indicação de que recebeu correctamente um ficheiro		

De: Supervisor, Comboio, Estação	Para: Simulador	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content OKInicioSimulacao		
Parâmetros: Agente emissor, Agente receptor		
Observações: Sinal de permissão para iniciar processo de simulação		

De: Estação	Para: Supervisor	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to PlataformaParagem:Comboio [d] Plataforma [d] Aresta [d]		
Parâmetros: Agente emissor, Agente receptor, Identificação do comboio, Identificação da plataforma, Identificação da aresta da plataforma		
Observações: Resposta acerca da plataforma de paragem em que um comboio deve parar		

De: Supervisor, Comboio, Estação	Para: Supervisor, Comboio, Estação	Performativa: reply
Mensagem: reply:sender [s]:receiver [s]:in-reply-to Status-Registry:content [d]		
Parâmetros: Agente emissor, Agente receptor, Código de resposta		
Observações: Resposta a um pedido de registo		

De: Simulador	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content ZonasSupervisao totalZonasSupervisao [d] Zona X [f] [f] Y [f] [f]		
Parâmetros: Agente emissor, Agente receptor, Total de zonas, Limite mínimo e máximo de latitude, Limite mínimo e máximo de longitude		
Observações: Informação sobre os limites das zonas supervisionadas pelo agente		

De: Simulador	Para: Supervisor, Comboio, Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content reset		
Parâmetros: Agente emissor, Agente receptor		
Observações: Pedido de reinicialização		

De: Simulador	Para: Supervisor, Comboio, Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content Desliga		
Parâmetros: Agente emissor, Agente receptor		
Observações: Pedido para terminar a ligação entre os agentes		

De: Simulador	Para: Supervisor, Comboio	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content localizacaoComboio Comboio [d] Destino [s] Cm [f] Pri [d] Fr [d] [d] [d] [d] [f] Tr [d] [d] [d] [d] [f] Veloc [f]		
Parâmetros: Agente emissor, Agente receptor, ID do comboio, Próximo destino, Comprimento, Prioridade,		
Observações: Informação periódica acerca da localização de cada comboio localizado na respectiva zona de supervisão (Frente + Traseira)		

De: Simulador	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content areaControle volume [d] of [d] [s]		
Parâmetros: Agente emissor, Agente receptor, Volume actual, Volume total, Informação do mapa		
Observações: Informação acerca da área supervisionada (linhas, pontos de selecção, estações)		

De: Simulador	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content comboio [s] [d] endereco [s] porta [d]		
Parâmetros: Agente emissor, Agente receptor, Agente representante do comboio, Identificação do comboio, Endereço do agente, Porto do agente		
Observações: Comboio a que o supervisor se deve ligar		

De: Simulador	Para: Supervisor	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content ModoControlo [d]		
Parâmetros: Agente emissor, Agente receptor, modo de controlo		
Observações: Activa um modo de controlo (Automático—Manual)		

De: Simulador	Para: Supervisor, Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content [s] [d]		
Parâmetros: Agente emissor, Agente receptor, Avaria ou Arranjo, Identificação do troço		
Observações: Avaria ou recomposição de uma parcela do trajecto		

De: Simulador	Para: Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content VelocidadeSimulacao Numerador [f] Denominador [f]		
Parâmetros: Agente emissor, Agente receptor, Numerador, Denominador		
Observações: Notificação da Velocidade de simulação		

De: Simulador	Para: Estação	Performativa: tell
Mensagem: tell:sender [s]:receiver [s]:content Supervisores ID [d] Trajecto [d] V [d] V [d] C [f] Total Supervisor [s] endereco [s] porto [d] Supervisor [s] endereco [s] porto [d]		
Parâmetros: Agente emissor, Agente receptor		
Observações: Supervisores responsáveis pelos trajectos onde estão inseridas as plataformas de paragem da estação		

C.1 Constantes

Nesta secção apresentam-se algumas das constantes definidas durante a implementação do sistema, e que são necessárias na transmissão de algumas mensagens anteriormente mencionadas.

Tipos de agentes

→ TIPO_AGENTE_SIMULADOR	0
→ TIPO_AGENTE_SUPERVISOR	1
→ TIPO_AGENTE_ESTACAO	2
→ TIPO_AGENTE_COMBOIO	3
→ TIPO_AGENTE_APRENDER	99

Registos de actividade

→ AGENTE_SEM_LOG	0
→ AGENTE_LOG_ECRAN	1
→ AGENTE_LOG_FICHEIRO	2
→ AGENTE_LOG_ECRAN_FICHEIRO	3

Tipos de simulação

→ TIPO_SIMULACAO_OK	0
→ TIPO_SIMULACAO_EMERGENCIA	1

Modos de transmissão de mensagens

- MODO_SOCKET_MENSAGEM 1
- MODO_SOCKET_FICHEIRO 2

Apêndice D

Standard 1494.1-1999 IEEE

Extracto do "IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements".

Informação On-Board.

- Taxa de aceleração
- Capacidade de travagem
- Velocidade máxima
- Identificação ATP

Informação da Pista para o Comboio(Periodicidade 1s)

- Identificação do controlador
- Identificador do comboio no sistema ATP
- Sector
- Segmento
- Offset
- Número de switches dentro do limite de autoridade.
- Posições dos switches dentro do limite de autoridade.
- Estado do procedimento de travagem de emergência
- Informação redundante para assegurar a integridade da transmissão.

Informação do Comboio para a pista(Periodicidade 1s)

- Identificação do controlador

- Identificador do comboio no sistema ATP
- Sector da parte dianteira do comboio
- Segmento da parte dianteira do comboio
- Offset da parte dianteira do comboio
- Sector da parte traseira do comboio
- Segmento da parte traseira do comboio
- Offset da parte traseira do comboio
- Informação redundante para assegurar a integridade da transmissão.

Informação recebida do controlador (ATP)

- Identificação do sector
- Limites da zona
- Zona seguinte a oeste
- Zona seguinte a este
- Identificação do segmento
- Segmento seguinte a oeste
- Segmento seguinte a este
- Identificação dos pontos de switch
- Limite de velocidade do segmento
- Plataformas de paragem
- Sinalização na pista

Informação transmitida em caso de necessidade (Controlador para Comboio)

- Actualização de limites de velocidade
- Definição de zonas de trabalho

Informação calculada on-board

- Posição da parte dianteira do comboio
- Posição da parte traseira do comboio
- Velocidade instantânea

- Direcção do comboio
- Modo de operação
- Limite da zona CBTC

Informação calculada pelo controlador

- MAL de cada composição
- Transferência entre zonas

Apêndice E

DTD

Neste apêndice apresentam-se os DTD's dos ficheiros de formato XML utilizados para guardar a informação relativa ao sistema.

- **Ficheiro de representação do mapa ferroviário. "mapa.dtd"**

```
<?xml version="1.0"?>
<!DOCTYPE mapa SYSTEM "mapa.dtd" [
<!ELEMENT mapa (id,nome,segmento+,estacao*,agulha*)>
<!ELEMENT segmento (id,origem,destino,caracteristica_segmento*)>
<!ELEMENT origem (x,y)>
<!ELEMENT destino (x,y)>
<!ELEMENT agulha (id,x,y)>
<!ELEMENT estacao (id,nome,localizacao,plataforma_paragem+,
                    caracteristica_estacao*)>
<!ELEMENT caracteristica_estacao (gera_pedidos|informacao_estacao)>
<!ELEMENT gera_pedidos (sim|nao)>
<!ELEMENT informacao_estacao (#PCDATA)>
<!ELEMENT plataforma_paragem (id,inicio,fim,largura,segmentoPlataforma)>
<!ELEMENT caracteristica_parqueamento (tipo_parqueamento|oficina)>
<!ELEMENT tipo_parqueamento (coberto|descoberto)>
<!ELEMENT oficina (sim|nao)>
<!ELEMENT localizacao (v1,v2,v3,v4)>
<!ELEMENT inicio (x,y)>
<!ELEMENT fim (x,y)>
<!ELEMENT v1 (x,y)>
<!ELEMENT v2 (x,y)>
<!ELEMENT v3 (x,y)>
<!ELEMENT v4 (x,y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT largura (#PCDATA)>
<!ELEMENT y (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT segmentoPlataforma (#PCDATA)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT caracteristica_segmento( tipo_segmento |
                                velocidade_maxima |
                                electrificado |
                                ATS |
                                ATP |
                                oncab |
                                sinalizacao_lateral)>
<!ELEMENT tipo_segmento (normal|tipo_tunel|tipo_ponte)>
<!ELEMENT electrificado (sim|nao)>
<!ELEMENT sinalizacao_lateral (sim|nao)>
```

```

<!ELEMENT comboio (id,nome,local,tipo_comboio,carruagens,
    ATS,oncab,prioridade,avaria,choque,localizacao,caracteristica_comboio*)>
<!ELEMENT tipo_comboio ( mercadorias |
    passageiros |
    misto)>
<!ELEMENT localizacao (absoluta_x, absoluta_y, orientacao_xy, relativa,
    vertice_origem, vertice_destino, aresta, sentido)>

<!ELEMENT absoluta_x (#PCDATA)>
<!ELEMENT absoluta_y (#PCDATA)>
<!ELEMENT orientacao_xy (#PCDATA)>
<!ELEMENT relativa (frente, traseira)>
<!ELEMENT aresta (frente, traseira)>
<!ELEMENT vertice_origem (frente, traseira)>
<!ELEMENT vertice_destino (frente, traseira)>
<!ELEMENT frente (#PCDATA)>
<!ELEMENT traseira (#PCDATA)>

<!ELEMENT prioridade (#PCDATA)>
<!ELEMENT caracteristica_comboio ( comprimento |
    velocidade_maxima
    ATS |
    ATO |
    oncab)>
<!ELEMENT velocidade_maxima (#PCDATA)>
<!ELEMENT ATS (sim|nao)>
<!ELEMENT ATP (sim|nao)>
<!ELEMENT oncab (sim|nao)>
<!ELEMENT carruagem (tipo_carruagem)
<!ELEMENT tipo_carruagem (mercadoria | passageiros)>
<!ELEMENT mercadoria (carga)>
<!ELEMENT carga (#PCDATA)>
<!ELEMENT passageiros (lotacao)>
<!ELEMENT lotacao (#PCDATA)>
<!ELEMENT caracteristica_comboio (ATS,ATO,On_Cab)>
<!ELEMENT ATO (sim|nao)>
<!ELEMENT ATS (sim|nao)>
<!ELEMENT On_Cab (sim|nao)>
<!ELEMENT timetable (prioridade,tempo_paragem*)>
<!ELEMENT prioridade (#PCDATA)>
<!ELEMENT tempo_paragem (id_local,data)>
<!ELEMENT id_local (#PCDATA)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT data (dia,mes,ano,hora,minuto,segundo)>
<!ELEMENT dia (#PCDATA)>
<!ELEMENT mes (#PCDATA)>
<!ELEMENT ano (#PCDATA)>
<!ELEMENT hora (#PCDATA)>
<!ELEMENT minuto (#PCDATA)>
<!ELEMENT segundo (#PCDATA)>
]>

```

- **Ficheiro de configuração do simulador. "ConfigSimulador.dtd"**

```

<?xml version="1.0"?>
<!DOCTYPE ConfigSimulador SYSTEM "ConfigSimulador.dtd" [
<!ELEMENT configuracoes (agente_simulador)>
<!ELEMENT agente_simulador (identificador,porto)>
<!ELEMENT identificador (#PCDATA)>
<!ELEMENT porto (#PCDATA)>
]>

```

- **Ficheiro de configuração dos agentes Supervisor. "ConfigSupervisor.dtd"**

```

<?xml version="1.0"?>
<!DOCTYPE ConfigSupervisor SYSTEM "ConfigSupervisor.dtd" [
<!ELEMENT configuracoes (agente_supervisor)>
<!ELEMENT agente_supervisor
(porto_entrada,id_simulador,endereco_simulador,porto_simulador)>
<!ELEMENT porto_entrada (#PCDATA)>
<!ELEMENT id_simulador (#PCDATA)>
<!ELEMENT endereco_simulador (#PCDATA)>
<!ELEMENT porto_simulador (#PCDATA)>
]>

```

- **Ficheiro de configuração dos agentes *Comboio*. "ConfigComboio.dtd"**

```

<?xml version="1.0"?>
<!DOCTYPE ConfigComboio SYSTEM "ConfigComboio.dtd" [
<!ELEMENT configuracoes (agente_comboio)>
<!ELEMENT agente_comboio
(porto_entrada,id_simulador,endereco_simulador,porto_simulador)>
<!ELEMENT porto_entrada (#PCDATA)>
<!ELEMENT id_simulador (#PCDATA)>
<!ELEMENT endereco_simulador (#PCDATA)>
<!ELEMENT porto_simulador (#PCDATA)>
]>

```

- **Ficheiro de configuração dos agentes *Estacao*. "ConfigEstacao.dtd"**

```

<?xml version="1.0"?>
<!DOCTYPE ConfigEstacao SYSTEM "ConfigEstacao.dtd" [
<!ELEMENT configuracoes (agente_estacao)>
<!ELEMENT agente_estacao
(porto_entrada,id_simulador,endereco_simulador,porto_simulador)>
<!ELEMENT porto_entrada (#PCDATA)>
<!ELEMENT id_simulador (#PCDATA)>
<!ELEMENT endereco_simulador (#PCDATA)>
<!ELEMENT porto_simulador (#PCDATA)>
]>

```

- **Ficheiro de configuração dos agentes *Aprender*. "ConfigAprender.dtd"**

```

<?xml version="1.0"?>
<!DOCTYPE ConfigAprender SYSTEM "ConfigAprender.dtd" [
<!ELEMENT configuracoes (agente_aprender)>
<!ELEMENT agente_aprender
(porto_entrada,id_simulador,endereco_simulador,porto_simulador)>
<!ELEMENT porto_entrada (#PCDATA)>
<!ELEMENT id_simulador (#PCDATA)>
<!ELEMENT endereco_simulador (#PCDATA)>
<!ELEMENT porto_simulador (#PCDATA)>
]>

```


Apêndice F

Terminologia

Neste apêndice apresentam-se alguma da terminologia associada ao domínio dos transportes ferroviários.

Absolute Block System Sistema de sinalização baseado no princípio da divisão de uma linha em blocos ou secções. Normalmente nenhuma composição tem autorização para entrar num bloco se este estiver ocupado por outro.

Arm Painel de um semáforo que contém um sinal. Nos US é também aplicado aos sinais luminosos. Cada um disponibiliza 1 unidade de informação: 1 cor ou 1 posição.

Aspect Aparência do sinal para o engenheiro/conductor que pode incluir luzes, cor de luzes, posição de luzes, placas com números posição de braços, etc. Numa linha podem coexistir vários aspectos.

Automatic Signaling Sistema onde os sinais visuais de controle de tráfego são accionados inteiramente pela passagem dos comboios ao accionarem sensores (track circuits). Usado onde não existem junções ou *turnouts*.

Automatic Train Control Sistema automático de controlo e sinalização ferroviária. É composto por sistemas de ATS, ATP e ATO.

Automatic Train Operation Sistema automático de entrada e saída de comboios em estações.

Automatic Train Protection Sistema automático de segurança de tráfego ferroviário.

Ballast Gravelha onde a pista (carris) assenta.

Bi-Directional Signalling Sistema usado para permitir a passagem de composições em ambas as direcções.

Blade Parte móvel de uma agulha que permite a mudança de direcção de um comboio.

Block Porção de pista protegida por sinalização que não poderá ser ocupada por mais que uma composição.

Blocking Back File de comboios parados devido a uma composição atrasada ou qualquer outra obstrução.

Board Gíria para painel de um semáforo.

Brakeman Pessoa responsável por accionar manualmente os pontos de selecção.

Cab Signalling Sistema onde as indicações são mostradas ao condutor directamente na sua cabina. Pode co-existir com os sinais visuais. É usado em conjunção com algum tipo de protecção automática para prevenir que 2 composições entrem no mesmo bloco ou excedam determinada velocidade.

Car Carruagem (US).

Carriage Carruagem (UK).

Centre Siding Sistema usado para inverter composições, colocado normalmente perto de estações.

Coach Carruagem de passageiros.

Color-Light Signal Um sinal onde a cor das lâmpadas tem significado.

Color-Position-Light Signal Um sinal onde tanto a cor das lâmpadas como a sua posição têm significado.

Conductor Supervisor do manuseamento do comboio.

Conductor Rail Um terceiro rail existente nas linhas electrificadas que transmite a energia à composição.

Crossing Parte de pista que permite que duas linhas se cruzem.

Crossover Porção de pista que liga 2 linhas paralelas através de 2 conjuntos de *Points*.

Crosstie Barras horizontais que suportam os rails. Começaram por ser de madeira, depois de ferro e mais recentemente os carris começaram a assentar directamente sobre betão.

Detection Capacidade de determinar que um bloco está ocupado por uma composição. A detecção é normalmente feita através de track circuit ou equivalente equipamento eléctrico. Usado também para verificar que um *Point* funcionou correctamente numa operação de *interlocking*.

Dispatching Acto de ordenar a uma composição para se dirigir a uma determinada estação.

Dragging Detector Dispositivo para avisar que uma composição ou parte dela saiu da linha. Normalmente consiste numa barra colocada entre os rails e é mais útil para as composições de mercadorias onde a dificuldade para detectar uma roda fora do trilho é maior.

Driver Pessoa primeiramente responsável pela operação da locomotiva.

Dwarf (Low) Signal Sinal colocado muito perto do chão, normalmente utilizado em linhas de muito baixa velocidade.

Emergency Evacuation Essencial para sistemas subterrâneos, consistem em sistemas com uma disposição distinta dos rails. Normalmente é requerido um passeio lateral perto dos rails para eventual evacuação dos passageiros.

EMU (*Electric Multiple Unit*) Termo genérico para uma composição sub-urbana ou metropolitana onde uma locomotiva separada não é necessária por que o sistema de tracção e controle está distribuído pelas várias composições do comboio.

Engineer (US) Equivalente a Driver.

Face Parte dos sinais luminosos consistindo num painel mais largo que a parte que emite luz, e que funciona como fundo da luz emitida, por forma a evitar interferências no sinal. São tipicamente placas de metal negro.

Facing Points Pontos onde uma pista diverge para duas. O oposto são *Trailing Points*.

Fireman Pessoa encarregue de abastecer a locomotiva, especialmente no caso das locomotivas a vapor.

Four Foot Nome de gíria que designa a distancia entre rails. Deriva da medida standard 4.5ft (1435mm).

Hight Signal Sinal colocado a grande altura especialmente indicado para as linhas principais.

Hight-Wide load Detector / High Car Detector Sensor óptico para detectar excessos de carga nos comboios.

Interchange Uma estação onde os passageiros podem mudar de composição para seleccionar uma rota diferente. Também designada por *Transfer*.

Interlocking Sistema usado para impedir que duas composições entrem em rota de colisão. No início eram sistemas mecânicos, depois electro-mecânicos sendo actualmente altamente computadorizados usando sistemas de votação 2 em 3.

Junction (US) Termo geral para o local onde duas pistas de 2 companhias diferentes se aproximam e incluem uma pista de interface entre elas.

Levelling Valve Dispositivo que mede as variações no peso das composições causadas pela entrada e saída de passageiros.

Leverman Pessoa que acciona as alavanca dos switches, sob a direcção da torre de controle.

Light Rail Vehicle (LRV) Termo genérico para *Streetcar*. Um veículo eléctrico que usa rails embebidos nas estradas ou linhas dedicadas de rails. Nos dias de hoje são preferidos em relação aos sistemas tradicionais devido aos menores custos de construção e manutenção.

Line Capacity Numero máximo de composições que podem operar sobre uma linha numa determinada direcção. Normalmente expressa por hora.

Lineside Signals Sinais visuais localizados ao longo da linha para serem observados pelo condutor.

Loop Linha que vai dar novamente a si mesma permitindo que as composições invertam o sentido.

Main Line Designam-se por Main Lines as principais rotas de uma companhia ou país.

Main Track Principal pista geralmente utilizada pelas composições mais rápidas e mais prioritárias de uma rota.

Metro Termo usado para designar uma linha ferroviária urbana, parcialmente ou totalmente subterrânea, com elevada quantidade de passageiros. É sinónimo de *Subway* (US).

Motor Car Veículo de passageiros que contém equipamento de tracção.

Operator, Tower Operator, Towerman Pessoa controlando uma porção de pista relativamente pequena, normalmente apenas um ponto de *interlocking*.

Overhead Termo que designa os cabos eléctricos que fornecem energia e que estão localizados acima das composições.

Overlap Distância segura de travagem fornecida para o caso do comboio não parar num determinado sinal.

Overrun Distância permitida para um comboio parar em caso de não ter parado no local devido. Esta distância depende da velocidade, peso e capacidade de travagem do comboio.

Point Refere-se nos US como sendo um switch, isto é, um ponto onde uma linha se divide em duas. Também conhecido como *Turnout*.

Possession Quando um troço de pista deve ser interdito para reparação, diz-se que está sob posseção.

Rail Barra de ferro onde as rodas das composições assentam.

Repeater Signal Sinal que é uma cópia exacta de outro sinal, usados para prevenir que um condutor não visualize correctamente um determinado sinal.

Route Locking Acção efectuada pelo sistema de controle na qual uma porção da linha está bloqueada (mecânica e electricamente) enquanto um comboio se aproxima e passa através dela.

Semaphore signal Um sinal que disponibiliza informação através da posição do seu braço. São tipicamente colocados à direita/esquerda das pistas consoante estas sejam de *Right/Left-Hand Operation*.

Shifted Load Detector Detector óptico de excesso de carga.

Shuttle Comboio expresso que efectua um determinado trajecto entre duas estações sem efectuar nenhuma paragem.

Siding Linha de baixa prioridade colocada paralelamente (ou não) ao lado da linha principal.

Signalman Termo usado para a pessoa que opera ou controla os sinais. Tradicionalmente alojados numa *Signal box*, mais recentemente em torres de controlo.

Signals Indicação visual transmitida ao maquinista para avisar da velocidade recomendada, velocidade máxima e da direcção do comboio.

Signal-Passed-At-Danger (SPAD) Transgressão originada por falhas técnicas ou humanas de um comboio ao sistema de sinalização.

Stabling Acto de estacionar um comboio fora de serviço.

Switch Nos US define-se como a estrutura que permite aos comboios mudarem de uma linha para outra. Equivalente aos *Points* (UK).

Timetable Sequência de locais de paragem que devem ser efectuados por uma composição no decurso de uma viagem.

To Cross (UK) Termo para designar o acto de dois comboios se cruzarem em direcções opostas.

To Pass (UK) Acto de um comboio ultrapassar outro que viaja na mesma direcção num local apropriado para o efeito.

Track Conjunto de estruturas onde os comboios se deslocam. A infra-estrutura.

Track Circuits Sensores colocados nas linhas de caminho de ferro e que detectam a passagem de comboios. Usados principalmente para accionar os sinais de controle. Fisicamente consistem num relay com um circuito de baixa voltagem, sendo cada um isolado dos restantes. À passagem da composição as rodas do comboio vão accionar o circuito efectuando a respectiva comutação de sinais.

Trailer Car Carruagem de passageiros.

Train Comboio.

Train Crew Tripulação de um comboio.

Turnout Termo para designar *Switches* ou *Points*, isto é, locais onde uma linha se divide em duas.

UIC (Union Internationale de Chemin de Fer) Organismo europeu que regula os standards para a industria dos caminhos de ferro.

Wagon Vagão.

Bibliografia

- 2, CBTC Working Group 1996, Communications based train control standards, *in* 'Summary Minutes of Meeting 3', p. 1.
- Administration, Federal Transit 2003, 'Communications based train control'.
<http://www.fta.dot.gov/research/equip/raileq/cbtc/cbtc.htm>
- Añez, J. B. Pérez 1994, 'Dual graph representation of transport networks'.
- Agrawal, R. A. Strikant 1994, *Fast algorithms for mining association rules*, VLDB.
- Antoniotti, M. A. Göllü 1997, Shift and smart-ahs: A language for hybrid system engineering, modeling and simulation., *in* 'USENIX Conference on Domain Specific Languages, Santa Barbara, CA'.
- Bej, Mark D. 1998, 'Introduction to railroad signalling'.
<http://broadway.pennsyrr.com/Rail/Signal/intro.html>
- Board, Transportation Research 1999, 'Committe research problem statements. group 2. section m railway systems'.
- Brazdil, Pavel 2002, *Apontamentos da Disciplina Extração de Conhecimento de Dados*, MIACC.
- Brooks, R. 1986, A robust layered control system for a mobile robot, *in* 'IEEE Journal of Robotics and Automation, V.RA-2'.
- Carlos 2003, 'Crash simulation system'.
<http://jlagunez.iquimica.unam.mx/~carlos/crash/>
- Centonlanzi, Patrick M. 2002, *Communications Based Train Control – An Overview*, Brooklyn Polytechnic University, E.U.A.
- Chavez, A. P. Maes 1996, An agent marketplace for buying and selling goods, *in* 'First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Systems'.
- Coha, Joseph 2000, 'Performance of java vs. c/c++ compiled code'.
<http://h21007.www2.hp.com/dspp/tech/>
- Connell, Mike 2002, 'Python vs. perl vs. java vs. c++ runtimes'.
<http://www.flat222.org/mac/bench/>
- Corporation, Magenta 2003, 'Why multi-agent software?'.
http://magenta.corp.com/ma_faq.htm

Cova, Mustafa 2002, 'Railway technical web pages'.

<http://www.trainweb.org/railwaytechnical>

Cuppari, Alessio Pier Luigi Guida n.d., 'Prototyping freight trains traffic management using multi-agent systems'.

citeseer.nj.nec.com/514832.html

Dautenhahn, Kerstin 1995, Getting to know each other. artificial social intelligence for autonomous robots, *in* 'Robotics and Autonomous Systems 16', pp. 333–356.

Dijkstra, E. W. 1959, A note on two problems in connexion with graphs, *in* 'Numerische Mathematik 1', pp. 269–271.

Durfee, E. V. Lesser 1995, Using partial global plans to coordinate problem solvers, *in* 'Proc. of IJCAI'.

F.Bellirfemine, A.Poggi, G.Rimmassa P.Turci 2000, An object-oriented framework to realize agent systems, *in* 'Proceedings of WOA 2000 Workshop', Parma, Italia, pp. 52–57.

Frost, Jim 1999, 'Windows sockets. a quick and dirty primer'.

Gabri, Tihomir, Emma Norling, Gil Tidhar, Liz Sonenberg Nicholas Howden 1994, Multi-agent design of a traffic-flow control system, *in* 'Technical Report 94'.

Galyon, Eric 2002, 'C++ vs java performance'.

<http://www.cs.colostate.edu/~cs154/PerfComp/>

Ganguly, Sri Subhasis 2002, 'The history of railway signalling'.

http://railindia.tripod.com/hist_rly_signalling

Gieseler, Charlie, Brett Graves Philip Maher n.d., 'A multi-agent intelligent traffic light system'.

www.cs.iastate.edu/~bgraves/CS573Project/573ProjectArchive

Harold, Elliotte Rusty W. Scott Means 2001, *XML in a Nutshell*, O'Reilly, E.U.A.

Hayes-Roth, B. 1995, An architecture for adaptative intelligent systems, *in* 'ArtificialIntelligence: Special Issue on Agents and Interactivity', pp. 329–365.

Hayes-Roth, B., M. Hewett, R. Waashington, R. Hewett A. Seiver 1995, Distributing intelligence within an individual, *in* 'L. Gasser, M. N. Huhns (Eds.) Distributed AI, Volume II', pp. 385–412.

Hobeika, A.G. C. F. Kim 1994, Traffic flow prediction systems based on upstream traffic, *in* 'Vehicle navigation and information systems IEEE conference'.

Honavar, Vasant 1999, 'Intelligent agents and multi agent systems'.

Huber, Alexander Bernd Ludwig 2002, 'A natural language multi-agent system for controlling model trains'.

http://www-wv.informatik.uni-erlangen.de/~bdludwig/pubs/huber_ludwig_c%amready

INEM 2003, 'Instituto nacional de emergência médica'.

http://www.inem.min-saude.pt/inem_pub/

- Jennings, Nicholas R. Michael J. Wooldridge 1998, Applications of intelligent agents, in 'Agent Technology: Foundations, Applications, and Markets', Springer-Verlag: Heidelberg, Germany, pp. 3–28.
citeseer.nj.nec.com/jennings98applications.html
- Jorge, Alípio 2002, *Apontamentos da Disciplina Extração de Conhecimento da Internet*, MIACC.
- J. Pachl 2002, *The Agent Builder User Guide*, Reticular Systems, San Diego. E.U.A.
- J. P. Lewis 2001, 'Performance of java versus c'.
<http://www.idiom.com/~zilla/Computer/javaCbenchmark.html>
- Jr., Richard S. Wright Michael Sweet 2000, *OpenGL Superbible, Second Edition*, Waite Group Press, E.U.A.
- K. Decker V. Lesser 1995, Designing a family of coordination algorithms, in 'Proc. of the 1st ICMAS'.
- Laboratory, Kajitany Ming 1992, 'Mubot'.
<http://www.crystaliz.com/logicware/mubot.html>
- Ljungberg, Magnus Andrew Lucas 1992, The OASIS air-traffic management system, in 'Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence', Seoul, Korea.
citeseer.nj.nec.com/ljungberg92oasis.html
- Ltd, Zuno 1997, 'Zuno'.
<http://www.dlib.com/>
- Maes, P. 1994, Agents that reduce work and information overload, in 'Communications of the ACM, 37(7)', pp. 31–40.
- Maes, Pattie 1995, Artificial life meets entertainment: Life like autonomous agents, in 'Communications of the ACM', pp. 108–114.
- Mamei, Marco Michael Mahan 2003, Engineering mobility in large multi-agent systems, in 'SELMAS 2002, LNCS 2603', pp. 110–122.
- Manikonda, Vikram, Renato Levy, Goutam Satapathy, David J. Lovell, Peter C. Chang Anna Teittinen 2003, Autonomous agents for traffic simulation and control, in 'Transportation Research Record 1774', pp. 1–10.
- Mataric, Maja J. 1994, Interaction and intelligent behavior, in 'AITR-1495', p. 177.
citeseer.nj.nec.com/article/mataric94interaction.html
- Mayfield, James, Yannis Labrou Tim Finin 1995, Desiderata for agent communication languages, in 'AAAI Spring Symposium on Information Gathering'.
citeseer.nj.nec.com/mayfield95desiderata.html
- Moncaster, Toby 2002, 'More room on the tracks. the future of railway signalling'.
<http://www.essex.ac.uk>
- Morris 2002, 'Data structures and algorithms'.
<http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/>

- Newell, A. H. Simon 1976, Computer science a empirical inquiry: symbols and search, in 'Comm. ACM 19(4)', pp. 113–126.
- Nwana, Hyacinth S. 1996, Software agents. an overview, in 'Knowledge Engineering Review, Vol. 11', pp. 1–40.
- of Electrical, Institute Electronic Engineers 1999, 'Ieee standard for communication based train control. performance requirements and functional requirements'.
- Oliveira, Eugénio 2002, *Apontamentos da Disciplina Sistemas Multi-Agente*, MIACC.
<http://www.fe.up.pt/~eol>
- Oliveira, Eugénio José M. Fernandes 1999, 'Tramas: Traffic control through behaviour based multi-agent system'.
<http://www.ieeta.pt/~jfernand/tramas/>
- Oliveira, Eugénio, Klaus Fisher Olga Stepankova 1999, Multi-agent systems: Which research for which applications, in 'Robotics and Autonomous Systems 28', pp. 1–16.
- Parunak, H. V. D. 1987, Manufacturing experience with the contract net, in 'M. N. Huhns (Ed.) Distributed AI'.
- Program, Transit Cooperative Research 2001, Concensus standards for the rail transit industry, pp. 9–12.
- Rademacher, Paul 1999, 'A glut based user interface library'.
http://www.cs.unc.edu/~rademach/glui/src/release/glui_manual_v101.pdf
- Rocha, Ana Paula 2003, *Formação de Empresas Virtuais*, Faculdade de Engenharia da Universidade do Porto, Portugal.
- Roozmond, Danko A. Jan L.H. Rogier 2000, Agent controlled traffic lights, in 'ESIT 2000'.
- Rumbaugh, James, Ivar Jacobson Grady Booch 1999, *The Unified Modeling Language Reference Manual*, Addison-Wesley, E.U.A.
- Russell, J. Stuart Peter Norvig 1995, *Artificial Intelligence A Modern Approach*, Prentice Hall, E.U.A.
- Sarmiento, Luís Eugénio Oliveira 2003, 'Agentes emocionais'.
- Serenko, Alexander Brian Detlor 2002, *Agent Toolkits: A general overview*, McMaster University, Hamilton, Ontario.
- Smith, R. G. 1980, The contract net prototol: High level comunciations and control in a distributed problem solver, in 'IEEE Transactions on Computers C29(12)', pp. 1104–1113.
- Sullivan, Tom 2003, 'Communications based train control'.
<http://www.tsd.org/communic.htm>
- Sycara, Katia P. 1998, Multiagent systems, in 'American Association for Artificial Intelligence Magazine', pp. 79–92.

- Technologies, IntelliOne 2001, 'Why, when and where to use software agents'.
<http://www.agentbuilder.com/Documentation/whyAgents.html>
- von der Linden, Peter 2002, *Just Java*, Prentice Hall.
- Wavish, P. M. Graham 1996, A situated action approach to implementing characters in computer games, in 'Int. Journal of Applied Artificial Intelligence, 10(1)', pp. 53–74.
citeseer.nj.nec.com/ljungberg92oasis.html
- Winter, S. A. Grünbacher 2003, 'Modeling costs of turns in route planning'.
<http://citeseer.nj.nec.com/491840.html>
- Wise, G. Bowden 1995, 'An overview of the standard template library'.
<http://www.cs.rpi.edu/~wiseb/xrds/ovp2-3b.html>
- Wooldridge, Michael Nicholas R. Jennings 1995, *Agent Theories, Architectures and Languages: a Survey, Intelligent Agents*, Springer-Verlag, Berlin.
- Y.F.Hu, R.J.Allan K.C.F.Maguire 2000, 'Comparing the performance of java with fortran and c for numerical computing'.
<http://www.dl.ac.uk/TCSC/UKHEC/JASPA/bench.html>