

# **Plataforma de Desenvolvimento de Cadeiras de Rodas Inteligentes**

**Rodrigo Antonio Marques Braga**



Universidade do Porto

Faculdade de Engenharia

**FEUP**

**Faculdade de Engenharia da Universidade do Porto**

Departamento de Engenharia Informática

Rua Roberto Frias, s/n, 4200-465 Porto, Portugal

**Setembro de 2010**



**Faculdade de Engenharia da Universidade do Porto**



**Universidade do Porto**

**Faculdade de Engenharia**

**FEUP**

**Plataforma de Desenvolvimento de Cadeiras  
de Rodas Inteligentes**

**Rodrigo Antonio Marques Braga**

Dissertação submetida para a obtenção do grau de Doutor em  
Engenharia Informática pela Faculdade de Engenharia da  
Universidade do Porto

Programa de Doutoramento em Engenharia Informática

Trabalho realizado sob orientação do Professor Doutor

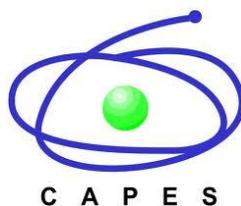
**Luís Paulo Reis**

e co-orientação do Professor Doutor

**António Paulo Gomes Mendes Moreira**

Porto, Setembro de 2010

Apoio



CAPES

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

**Programa Doutorado Pleno no Exterior**

Dedico esta tese aos meus queridos e dedicados familiares e, especialmente, aos meus pais  
Marlene C. Marques e Marcos Thomé da Silva Ferreira.

Ao meu pai e avós, Wanderley, Bruno e Célia.

E às minhas mães de coração, Yara e Tânia

E aos amigos que me acompanharam ao longo deste caminho.

“O fácil vem sempre depois do difícil”

J.F. Haffner

"No fim tudo dá certo, e se não deu certo é porque ainda não chegou ao fim."

Fernando Tavares Sabino, (1923 - 2004), escritor e jornalista brasileiro.



# Agradecimentos

Primeiro a Deus por tudo;

A todos aqueles que me acompanharam nesta jornada;

Especialmente aos meus pais Marlene C. Marques e Marcos Thomé da Silva Ferreira, pois sem eles, não teria chegado até aqui;

A todos meus familiares pelo apoio, em especial minhas irmãs;

Aos colegas de laboratório, em especial, Marcelo Petry e Rosaldo Rossetti;

Ao Prof. Doutor Eugénio Oliveira, pelo apoio e acolhimento em seu grupo de investigação;

Ao LIACC e PRODEI;

A todos os meus amigos, em especial, Marco Aurélio Moreira Saran, Lúcio Sanchez Passos, Sílvio Costa Sampaio, Brígida Mónica Faria e Leonardo Bremermann, os quais, para além da amizade, nesta fase final me deram muito apoio;

Novamente ao Sílvio Costa Sampaio pelo seu empenho incansável em ajudar-me na revisão deste documento;

A CAPES pelo apoio financeiro com a bolsa de doutoramento no exterior;

E em especial, aos meus orientadores,

Prof. Doutor Luís Paulo Reis e Prof. Doutor António Paulo Moreira, pelos conhecimentos repassados, apoio, preocupação na realização deste e, além de tudo, a amizade.



# Índice

ÍNDICE.....	IX
LISTA DE FIGURAS.....	XIII
LISTA DE TABELAS .....	XIX
LISTA DE ACRÓNIMOS E SIGLAS .....	XXI
RESUMO .....	XXV
ABSTRACT.....	XXVII
RÉSUMÉ .....	XXIX
<b>1 INTRODUÇÃO .....</b>	<b>31</b>
1.1 MOTIVAÇÃO .....	33
1.2 OBJECTIVOS.....	37
1.3 CONTRIBUIÇÕES CIENTÍFICAS DA TESE.....	38
1.4 ESTRUTURA DA TESE .....	39
<b>2 DEFINIÇÕES E FUNDAMENTOS TEÓRICOS .....</b>	<b>41</b>
2.1 INTRODUÇÃO .....	41
2.2 DEFINIÇÕES, PREMISSAS E RESTRIÇÕES .....	45
2.3 ROBÓTICA INTELIGENTE.....	47
2.3.1 <i>Agentes Reactivos e Arquitectura de Subsunção</i> .....	47
2.3.2 <i>Modelo do Mundo</i> .....	48
2.3.3 <i>Grafos</i> .....	49
2.3.4 <i>Algoritmo de Pesquisa/Procura</i> .....	52
2.3.5 <i>Algoritmo AStar (A*)</i> .....	57
2.3.6 <i>Planeamento Automático</i> .....	58
2.3.7 <i>Navegação e Controlo</i> .....	64
2.4 INTERACÇÃO E INTERFACES MULTI-MODAIS .....	70
2.4.1 <i>Arquitecturas de Sistemas de Interacção Homem-máquina</i> .....	72
2.4.2 <i>Projectos de Interface Multimodal</i> .....	73
2.5 SISTEMAS MULTI-AGENTE .....	74
2.5.1 <i>Plataforma de Comunicação</i> .....	82
2.5.2 <i>Metodologias de Modelação de SMA</i> .....	90
2.6 SUMÁRIO .....	97
<b>3 ESTADO DA ARTE .....</b>	<b>99</b>

3.1	PRINCIPAIS PROJECTOS DE CADEIRAS DE RODAS INTELIGENTES.....	99
3.1.1	<i>Cadeira de Rodas Autónoma da Universidade do Estado do Arizona</i> .....	100
3.1.2	<i>CALL Centre</i> .....	101
3.1.3	<i>OMNI</i> .....	103
3.1.4	<i>NavChair</i> .....	104
3.1.5	<i>Tin Man</i> .....	104
3.1.6	<i>Wellman</i> .....	105
3.1.7	<i>FRIEND:</i> .....	106
3.1.8	<i>Smartchair</i> .....	107
3.1.9	<i>VAHM</i> .....	107
3.1.10	<i>Robô SENA</i> .....	109
3.1.11	<i>RoboChair</i> .....	110
3.1.12	<i>ACCoMo</i> .....	111
3.1.13	<i>RobChair:</i> .....	112
3.2	SÍNTESE DOS PROJECTOS DE CRI .....	113
3.3	CARACTERÍSTICAS GERAIS DOS PROJECTOS .....	116
3.3.1	<i>Funções Presentes nos Projectos de CRI</i> .....	117
3.3.2	<i>Comportamento Baseado em Habilidades</i> .....	119
3.3.3	<i>Navegação</i> .....	120
3.3.4	<i>Interface Homem-máquina</i> .....	120
3.3.5	<i>Sistema Multi-agente e Cadeira de Rodas Inteligente</i> .....	121
3.4	SUMÁRIO .....	122
<b>4</b>	<b>CONCEITO E IMPLEMENTAÇÃO DA PLATAFORMA INTELLWHEELS ...</b>	<b>123</b>
4.1	CONCEITO E PROJECTO.....	123
4.2	HARDWARE DA PLATAFORMA .....	126
4.3	SIMULADOR DE CADEIRA DE RODAS.....	129
4.4	PLANEAMENTO INTELIGENTE.....	138
4.5	NAVEGAÇÃO E CONTROLO .....	145
4.5.1	<i>Controlo de Acções Básicas</i> .....	149
4.6	INTERFACE MULTIMODAL .....	155
4.7	SISTEMA MULTI-AGENTE .....	158
4.7.1	<i>Plataforma de Comunicação</i> .....	160
4.7.2	<i>Modelização do SMA</i> .....	164
4.8	SUMÁRIO .....	183
<b>5</b>	<b>EXPERIMENTAÇÃO E RESULTADOS .....</b>	<b>185</b>
5.1	INTRODUÇÃO E METODOLOGIA DE TESTE .....	185
5.2	PROTÓTIPOS.....	187

5.2.1	<i>Protótipo IntellWheelsChair-I</i> .....	188
5.2.2	<i>Protótipo IntellWheelsChair-II</i> .....	190
5.2.3	<i>Protótipo IntellWheelsChair-III</i> .....	193
5.2.4	<i>Dispositivos de Interação com o Utilizador</i> .....	196
5.3	CENÁRIOS DE TESTE.....	199
5.3.1	<i>Teste da MMI em Ambiente Real</i> .....	200
5.3.2	<i>Navegação Autónoma</i> .....	200
5.3.3	<i>Realidade Mista</i> .....	201
5.3.4	<i>Plataforma de Comunicação</i> .....	201
5.3.5	<i>Controlo Compartilhado em Firmware</i> .....	203
5.3.6	<i>Interação da CRI com Macro Agentes</i> .....	204
5.4	RESULTADOS DOS TESTES .....	207
5.4.1	<i>Teste da MMI em Ambiente Real</i> .....	207
5.4.2	<i>Navegação Autónoma</i> .....	208
5.4.3	<i>Realidade Mista</i> .....	210
5.4.4	<i>Plataforma de Comunicação</i> .....	211
5.4.5	<i>Controlo Compartilhado em Firmware</i> .....	215
5.4.6	<i>Interação da CRI com Macro Agentes</i> .....	217
5.5	SUMÁRIO .....	219
<b>6</b>	<b>CONCLUSÕES</b> .....	<b>223</b>
6.1	SÍNTESE DO TRABALHO .....	223
6.2	RESULTADOS .....	225
6.3	TRABALHOS FUTUROS.....	228
	<b>BIBLIOGRAFIA</b> .....	<b>233</b>



# Lista de Figuras

FIGURA 1: CRESCIMENTO MUNDIAL DA POPULAÇÃO IDOSA .....	33
FIGURA 2: ÍNDICES DE ENVELHECIMENTO EM PORTUGAL .....	34
FIGURA 3: ÍNDICES DE ENVELHECIMENTO NO BRASIL.....	34
FIGURA 4: POPULAÇÃO SEM DEFICIÊNCIA E COM DEFICIÊNCIA, PORTUGAL 2001 .....	35
FIGURA 5: DIAGRAMA QUE REPRESENTA UM GRAFO SIMPLES $G$ COMPOSTO POR 5 VÉRTICES E 7 ARESTAS.....	49
FIGURA 6: DIAGRAMA QUE REPRESENTA O GRAFO DIRIGIDO $D$ .....	50
FIGURA 7: GRAFO $G_1$ QUE REPRESENTA A RELAÇÃO DAS PESSOAS DA FAMÍLIA CASTRO .....	50
FIGURA 8: GRAFO $G_2$ NÃO ORIENTADO COM LAÇOS .....	51
FIGURA 9: LINHA DO TEMPO NO ESTUDO DE PLANEADORES AUTOMÁTICOS .....	59
FIGURA 10: REPRESENTAÇÃO DE UM ROBÔ NÃO-HOLONÓMICO .....	68
FIGURA 11: REPRESENTAÇÃO DOS VECTORES VELOCIDADES DAS RODAS DA CADEIRA.....	69
FIGURA 12: REPRESENTAÇÃO DOS VECTORES VELOCIDADE NO CENTRO DE MOVIMENTO $C_M$ .....	70
FIGURA 13: ESTRUTURA DE UM SISTEMA MULTI-AGENTE.....	75
FIGURA 14: TOPOLOGIA DE AGENTES PROPOSTA POR Nwana. ....	77
FIGURA 15: DIAGRAMA ESQUEMÁTICO DE UM AGENTE REACTIVO .....	78
FIGURA 16: DIAGRAMA ESQUEMÁTICO DE UM AGENTE REACTIVO COM ESTADO INTERNO .....	78
FIGURA 17: DIAGRAMA ESQUEMÁTICO DE UM AGENTE GERIDO POR OBJECTIVOS.....	79
FIGURA 18: DIAGRAMA ESQUEMÁTICO DE UM AGENTE BASEADO EM UTILIDADES .....	79
FIGURA 19: DIAGRAMA ESQUEMÁTICO DE UM AGENTE ADAPTATIVO.....	80
FIGURA 20: FORMAS DE COMUNICAÇÃO ENTRE AGENTES .....	82
FIGURA 21: ARQUITECTURA JADE.....	89
FIGURA 22: FASES DA METODOLOGIA MASE.....	91
FIGURA 23: MODELO GAIA v.2 (ADAPTADO DE(ZAMBONELLI ET AL., 2003)).....	95
FIGURA 24: RESULTADOS DA PESQUISA PELO TÓPICO "CADEIRA DE RODAS INTELIGENTE" .....	100
FIGURA 25: PROTÓTIPO DESENVOLVIDO POR MADARASZ .....	101
FIGURA 26: CRI – CALL CENTRE DA UNIVERSIDADE DE EDIMBURGO .....	102
FIGURA 27: CADEIRA DE RODAS OMNIDIRECCIONAL DO PROJECTO OMNI.....	103
FIGURA 28: PROTÓTIPO DA CRI - NAVCHAIR .....	104
FIGURA 29: SISTEMA TIN MAN.....	105

FIGURA 30: CADEIRA DE RODAS EQUIPADA COM DUAS PERNAS .....	105
FIGURA 31: PROJECTO FRIEND .....	106
FIGURA 32: PROTÓTIPO SMARTCHAIR .....	107
FIGURA 33: PROTÓTIPOS DO PROJECTO VAHM .....	108
FIGURA 34: CADEIRA DE RODAS SENA.....	109
FIGURA 35: ROBOCHAIR: CADEIRA DE RODAS COM COMANDO BASEADO EM GESTOS COM A CABEÇA .....	110
FIGURA 36: PROTÓTIPO ACCOMO .....	111
FIGURA 37: ROBÔ MÓVEL ROBCHAIR.....	112
FIGURA 38: CRI QUE SÃO CONTROLADAS SEM USAR AS MÃOS .....	117
FIGURA 39: EXEMPLO DE PROJECTOS DE CRI QUE UTILIZAM SENSORES PROXIMIDADE .....	118
FIGURA 40: MÓDULOS BÁSICOS DA PLATAFORMA INTELLWHEELS .....	124
FIGURA 41: ARQUITECTURA DE SOFTWARE DA PLATAFORMA INTELLWHEELS.....	125
FIGURA 42: MICRO E MACRO AGENTES.....	125
FIGURA 43: MODOS DE OPERAÇÃO .....	126
FIGURA 44: ARQUITECTURA DE HARDWARE DO PROJECTO INTELLWHEELS.....	127
FIGURA 45: ARQUITECTURA DO CIBER-RATO (ADAPTADO DE (LAU ET AL., 2002)).....	130
FIGURA 46: ARQUITECTURA DO SIMULADOR INTELLWHEELS .....	131
FIGURA 47. CONEXÃO COM O SIMULADOR INTELLWHEELS .....	132
FIGURA 48: REPRESENTAÇÃO CONTÍNUA DO MUNDO REAL AO MUNDO VIRTUAL .....	133
FIGURA 49: MODIFICAÇÃO DO CORPO DO ROBÔ DE CIRCULAR PARA RECTANGULAR .....	133
FIGURA 50: AGENTE ROBÓTICO SIMULADO (CRI).....	134
FIGURA 51: CURVAS DE EVOLUÇÃO DA POTÊNCIA COM O PARÂMETRO CURVE = (0,1, 0,5 E 0,8).....	135
FIGURA 52: VERIFICAÇÃO DA INTERSECÇÃO DOS SEGMENTOS DE RECTA.....	136
FIGURA 53: MODIFICAÇÕES NOS SENSORES DE PROXIMIDADE .....	136
FIGURA 54: JANELA 2D DO VISUALIZADOR .....	137
FIGURA 55: VISUALIZADOR 3D, A) MODO CÂMARA LIVRE E B) MODO PRIMEIRA PESSOA .....	138
FIGURA 56: PLANEADOR COM BASE EM STRIPS.....	139
FIGURA 57: PLANEADOR COM BASE EM GRAPHPLAN.....	140
FIGURA 58: JANELAS DE CONFIGURAÇÕES DO PLANEADOR .....	141
FIGURA 59: DESCRIÇÃO DO DOMÍNIO DA CRI EM PDDL.....	142
FIGURA 60: DESCRIÇÃO DO PROBLEMA DA CRI EM PDDL.....	143
FIGURA 61: PLANEADOR DE TRAJECTÓRIAS.....	143

FIGURA 62: JANELAS DE CONFIGURAÇÃO DO PLANEADOR DE TRAJECTÓRIAS.....	144
FIGURA 63: PLANEADOR DE TRAJECTÓRIAS COM MAPA MÉTRICO .....	145
FIGURA 64: ARQUITECTURA DE CONTROLO MULTI-NÍVEL DA PIW.....	146
FIGURA 65: APLICAÇÃO DE RECONHECIMENTO DE MARCOS ARTIFICIAIS .....	148
FIGURA 66: SISTEMA UBISENSE: ANTENA, TAG FINA E TAG COMPACTA .....	149
FIGURA 67: JANELA DE CONFIGURAÇÃO DOS PARÂMETROS DE CONTROLO .....	149
FIGURA 68: REPRESENTAÇÃO ESQUEMÁTICA DO CONTROLADOR PARA SEGUIMENTO DE RECTA.....	151
FIGURA 69: REPRESENTAÇÃO ESQUEMÁTICA DO CONTROLADOR PARA IR PARA O PONTO (X,Y) .....	152
FIGURA 70: REPRESENTAÇÃO ESQUEMÁTICA DO CONTROLO GIRAR PARA ( $\theta$ ).....	153
FIGURA 71: ESQUEMA DA UTILIZAÇÃO DOS SENSORES LATERAIS PARA O CONTROLO DE SEGUIR PAREDES .....	154
FIGURA 72: ARQUITECTURA DE SUBSUNÇÃO UTILIZADA NO CONTROLO DE DESVIO DE OBSTÁCULO.	154
FIGURA 73: ARQUITECTURA DO MÓDULO MMI.....	156
FIGURA 74: LISTAS DE ACÇÕES E DISPOSITIVOS DE ENTRADA DISPONÍVEIS .....	156
FIGURA 75: LISTA QUE ARMAZENA AS ASSOCIAÇÕES ENTRE AS SEQUÊNCIAS DE ENTRADAS E SAÍDAS .....	157
FIGURA 76: EXEMPLOS DE UMA SEQUÊNCIA DE ENTRADA .....	157
FIGURA 77: ARQUITECTURA MULTI-CAMADAS DA PLATAFORMA DE COMUNICAÇÃO.....	161
FIGURA 78: BIBLIOTECAS QUE COMPÕEM A PLATAFORMA DE COMUNICAÇÃO DA PIW.....	163
FIGURA 79: CAMADAS DE COMUNICAÇÃO E SUAS FUNÇÕES.....	164
FIGURA 80: ESTEREÓTIPOS MAIS UTILIZADOS.....	165
FIGURA 81: ESTÁGIOS DA METODOLOGIA GAIA .....	165
FIGURA 82: A) PACOTE DE REQUISITOS; B) PACOTE DE ANÁLISE .....	166
FIGURA 83: A) PACOTE PROJECTO DA ARQUITECTURA; B) PACOTE PROJECTO DETALHADO .....	166
FIGURA 84: PROCESSO GAIA .....	167
FIGURA 85: FASES DA ANÁLISE .....	167
FIGURA 86: ESTÁGIO PROJECTO DA ARQUITECTURA .....	168
FIGURA 87: ESTÁGIO PROJECTO DETALHADO.....	168
FIGURA 88: ARQUITECTURA INFORMAL DO SISTEMA .....	170
FIGURA 89: ORGANIZAÇÃO DO SMA E SUAS SUB-ORGANIZAÇÕES .....	171
FIGURA 90: DIAGRAMA DE RECURSOS DO AMBIENTE.....	171
FIGURA 91: PAPEL SENSOR .....	173
FIGURA 92: PAPEL REACTIVO.....	173

FIGURA 93: PAPEL PLANEADOR .....	174
FIGURA 94: PAPEL INTERFACE .....	175
FIGURA 95: PAPEL BROADCASTER .....	175
FIGURA 96: PAPEL GESTOR DE CONFLITO .....	176
FIGURA 97: PAPEL DESIGNADOR DE TAREFAS .....	177
FIGURA 98: PROTOCOLO TROCA PAPÉIS .....	177
FIGURA 99: PROTOCOLO BROADCAST .....	178
FIGURA 100: PROTOCOLO MONITOR DE AMBIENTE .....	178
FIGURA 101: PROTOCOLO RESOLUÇÃO DE CONFLITOS .....	178
FIGURA 102: PROTOCOLO SOLICITAÇÃO DE PLANO .....	179
FIGURA 103: PROTOCOLO ATENDER OBJECTIVO .....	179
FIGURA 104: DIAGRAMA DOS PAPÉIS E INTERACÇÃO .....	180
FIGURA 105: MODELO DE SERVIÇO .....	181
FIGURA 106: ARQUITECTURA COMPLETA DA PIW .....	182
FIGURA 107: CURVA DE ACELERAÇÃO DA CADEIRA DE RODAS REAL E VIRTUAL .....	186
FIGURA 108: MONTAGEM DOS SENSORES, A) POSICIONAMENTO DOS <i>ENCODERS</i> E B) INSTALAÇÃO DOS SENSORES DE ULTRA-SOM .....	188
FIGURA 109: POSICIONAMENTO DOS SENSORES DE ULTRA-SOM INSTALADOS NO PROTÓTIPO I .....	189
FIGURA 110: MONTAGEM DAS PLACAS ELECTRÓNICAS PARA A INTERFACE DOS DISPOSITIVOS E O COMPUTADOR .....	190
FIGURA 111: EVOLUÇÃO DO PROTÓTIPO INTELLWHEELSCHAIR-I .....	190
FIGURA 112: ARQUITECTURA PRELIMINAR DE HARDWARE .....	191
FIGURA 113: HARDWARE UTILIZADO NA CONSTRUÇÃO DO PROTÓTIPO 2 .....	191
FIGURA 114: PROTÓTIPO INTELLWHEELSCHAIR-II .....	192
FIGURA 115: PROTÓTIPO INTELLWHEELSCHAIR-III .....	193
FIGURA 116: PLACA ARDUÍNO .....	193
FIGURA 117: FORÇAS REPULSIVAS .....	194
FIGURA 118: FORÇAS RESULTANTES QUE AGEM SOBRE A CADEIRA .....	194
FIGURA 119: DISTÂNCIA SEGURA EM FUNÇÃO DA VELOCIDADE E DISTÂNCIA DA CRI .....	195
FIGURA 120: COMPONENTE DA DISTÂNCIA SEGURA .....	195
FIGURA 121: <i>DRIVER</i> PARA O JOYSTICK .....	196
FIGURA 122: INTERFACE DO MÓDULO DE CONTROLO POR EXPRESSÃO FACIAL .....	197
FIGURA 123: SENSOR PARA DETECTAR OS MOVIMENTOS DA CABEÇA .....	197

FIGURA 124: <i>DRIVER</i> DO SISTEMA DE DETECÇÃO DOS MOVIMENTOS DA CABEÇA.....	198
FIGURA 125: AGENTE INTERFACE .....	199
FIGURA 126: REPRESENTAÇÃO DO PERCURSO PARA O TESTE COM A MMI .....	200
FIGURA 127: MODELO DE INTERACÇÃO ESPERADA ENTRE OS AGENTES COMPRADOR/VENDEDOR DE LIVROS .....	202
FIGURA 128: REPRESENTAÇÃO DO AMBIENTE COM O CIRCUITO FECHADO ONDE OCORRERAM OS TESTES.....	204
FIGURA 129: DIAGRAMA DE INTERACÇÃO ENTRE OS AGENTES MÉDICO X CONTROLO .....	205
FIGURA 130: DIAGRAMA DE INTERACÇÃO ENTRE OS AGENTES CONTROLO X INTELIGÊNCIA .....	206
FIGURA 131: MAPA HIPOTÉTICO DO HOSPITAL COM O GRAFO DE TRAJECTÓRIAS.....	206
FIGURA 132: DIAGRAMA DOS AGENTES ENVOLVIDOS NESTE ENSAIO .....	207
FIGURA 133: CONTROLO DA CRI ATRAVÉS MMI EM UM CORREDOR.....	208
FIGURA 134: GERAÇÃO DE PLANO DE ALTO NÍVEL.....	209
FIGURA 135: ROTAS PLANEADAS E PERCORRIDAS EM MODO AUTÓNOMO.....	209
FIGURA 136: TESTE DE REALIDADE MISTA: CRI REAL (IMAGENS SUPERIORES) INTERAGINDO COM OBJECTOS VIRTUAIS (IMAGENS) .....	210
FIGURA 137: LABORATÓRIO DE REABILITAÇÃO EM REALIDADE MISTA .....	210
FIGURA 138: TEMPO NECESSÁRIO PARA QUE OS APLICATIVOS CONSIGAM ELEGER UM <i>CONTAINER</i> LOCAL .....	211
FIGURA 139: TEMPO NECESSÁRIO PARA REUNIR TODAS AS INFORMAÇÕES DOS APLICATIVOS E CRIAR UMA NOVA LAL .....	212
FIGURA 140: TEMPO NECESSÁRIO PARA INFORMAR TODOS OS APLICATIVOS DE UMA NOVA LAL E DISTRIBUÍ-LA.....	212
FIGURA 141: TEMPO NECESSÁRIO PARA ELEGER UM NOVO <i>CONTAINER</i> .....	213
FIGURA 142: RESULTADOS DA INTERACÇÃO COMPRADOR/VENDEDOR EM JADE OBTIDOS ATRAVÉS DO <i>SNIFFER</i> E <i>COMMAND LINE</i> . .....	213
FIGURA 143: MENSAGENS DAS INTERACÇÕES COMPRADOR/VENDEDOR NA PLATAFORMA DE COMUNICAÇÃO INTELLWHEELS.....	214
FIGURA 144: RESULTADOS OBTIDOS PARA A COMPRA DE LIVROS UTILIZANDO JADE .....	214
FIGURA 145: RESULTADOS OBTIDOS PARA A COMPRA DE LIVROS UTILIZANDO A PLATAFORMA INTELLWHEELS .....	215
FIGURA 146: NÚMERO DE COLISÕES POR VOLUNTÁRIO EM AMBIENTE SIMULADO .....	215
FIGURA 147: NÚMERO DE COLISÕES POR VOLUNTÁRIO EM AMBIENTE REAL .....	215
FIGURA 148: SENSAÇÃO DE SEGURANÇA COM E SEM A AJUDA DO CONTROLE COMPARTILHADO .....	216

FIGURA 149: QUÃO REALISTA OS VOLUNTÁRIOS ACHARAM O SIMULADOR .....	217
FIGURA 150: SENSAÇÃO DE AJUDA COM O CONTROLO COMPARTILHADO .....	217
FIGURA 151: A) CONSOLA DE COMUNICAÇÃO DO AGENTE CONTROLO. B) CONSOLA DE COMUNICAÇÃO DO AGENTE INTELIGÊNCIA E TABELA DE PLANOS. ....	218
FIGURA 152: LISTA DE ACÇÕES PENDENTES DO AGENTE CONTROLO.....	218
FIGURA 153: IMAGENS DO VISUALIZADOR EM MODO 3D QUE REPRESENTAM O MOVIMENTO DA CRI DO NÓ 19 PARA O NÓ 3. ....	219
FIGURA 154: TRAJECTO PERCORRIDO PELA CRI .....	219

# Lista de Tabelas

TABELA 1: PROPRIEDADES DOS GRAFOS .....	50
TABELA 2: TIPOS DE GRAFOS.....	51
TABELA 3: ALGORITMO A* .....	57
TABELA 4: DEFINIÇÃO DO DOMÍNIO DO MUNDO DOS BLOCOS .....	64
TABELA 5: DEFINIÇÃO DO PROBLEMA MUNDO DOS BLOCOS .....	64
TABELA 6: PROJECTOS DE INTERFACE MULTIMODAL .....	74
TABELA 7: TERMINOLOGIAS USADAS NOS SISTEMAS MULTI-AGENTE .....	76
TABELA 8: PRINCIPAIS CARACTERÍSTICAS ENCONTRADA NOS AGENTES.....	76
TABELA 9: PSEUDO-CÓDIGO AGENTE REACTIVO SIMPLES .....	78
TABELA 10: PSEUDO-CÓDIGO AGENTE MEMORIZANDO O MUNDO.....	78
TABELA 11: BENEFÍCIOS OBTIDOS PELO USO DE SMA EM PROBLEMAS DE IA.....	81
TABELA 12: CARACTERÍSTICAS ENCONTRADAS NAS PLATAFORMAS DE SMA .....	83
TABELA 13: LISTA DE PLATAFORMAS DE SISTEMAS MULTI-AGENTE.....	84
TABELA 14: FORMATO DUMA MENSAGEM FIPA-ACL .....	85
TABELA 15: PARÂMETROS CONTIDOS NAS MENAGENS FIPA ACL .....	86
TABELA 16: PERFORMATIVAS PROVIDENCIADAS PELA LINGUAGEM FIPA ACL .....	86
TABELA 17: PRINCIPAIS EXTENSÕES DA METODOLOGIA GAIA .....	94
TABELA 18: ALGUNS ESTEREÓTIPOS DE SPEM 2.0 .....	97
TABELA 19: PROJECTOS DE CADEIRAS DE RODAS INTELIGENTES.....	113
TABELA 20: CADEIRAS DE RODAS INTELIGENTES COMERCIAIS .....	116
TABELA 21: MODELO DE AGENTE .....	181
TABELA 22: CORRELAÇÃO ENTRE O METAMODELO E A PIW.....	182
TABELA 23: PARÂMETROS DE CONFIGURAÇÕES DA CADEIRA DE RODAS .....	186



# Lista de Acrónimos e Siglas

ACC	Agent Communication Channel
ACCoMo	Autonomous, Cooperative, COLlaborative MOBILE
ACL	Agent Communication Language
ADL	Action Description Language
ADQ	Actividades Diárias do Quotidiano
AGV	Automatic Guided Vehicle
AID	Agent Identifier
AIP	Agent Interaction Protocol
AIPS	Artificial Intelligence Planning Systems
AMS	Agent Management System
AOS	Agent Oriented Software
AOSE	Agent Oriented Software Engineering
API	Application Program Interface
AUML	Agente UML
BT	British Telecommunications
CAN	Controller Area Network
CAS	Common Agent Structure
CCD	Charge-Coupled Device
CIIM-PLEX	Consortium for Intelligent Integrated Manufacturing Planning-Execution
CRI	Cadeira de Rodas Inteligente
DARPA	Defense Advanced Research Projects Agency
DF	Directory Facilitator
EOG	Eletroculografia
FCT	Fundação para a Ciência e a Tecnologia
FEUP	Faculdade de Engenharia da Universidade do Porto
FIPA	Foundation for Intelligent Physical Agents
FRIEND	Functional Robot arm with user-frIENdly interface for Disabled people
GPS	Global Positioning System
GRASP	General Robotics and Active Sensory Perception
GUI	Graphical User Interface

IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
IDA*	Interactive Deepening A*
IHM	Interface Homem-Máquina
IPC	International Planning Competition
IW	IntellWheels
JADE	Java Agent DEvelopment framework
JADEX	JADE eXtension
KIF	Knowledge Interchange Format
KIPR	KISS Institute for Practical Robotics
KISS	Keep It Simple, Stupid
KQML	Knowledge Query and Manipulation Language
KSE	Knowledge Sharing Effort
LAF	Lista de Agentes Facilitadores
LAG	Lista de Agentes Global
LAL	Lista de Agentes Local
LASC	Laboratório de Automação e Sistemas Co-operativo
MaSE	Multiagent Systems Engineering
MMI	MultiModal Interface
NTP	Network Time Protocol
OCR	Optical Character Recognition
OMG	Object Management Group
OMS	Organização Mundial da Saúde
PDDL	Planning Domain Definition Language
PDT	Prometheus Design Tool
PF	Potential Field Methods
PID	Proportional–Integral–Derivative
PIW	Plataforma IntellWheels
PTP	Precision Time Protocol
PWM	Pulse Width Modulation
RFID	Radio-Frequency IDentification
RIM	Representação Interna do Mundo
RM	Robô Móvel

RMA	Remote Management Agent
RMI	Remote Method Invocation
RN	Redes Neuronais
ROADMAP	Role Oriented Analysis and Design for Multi-Agent Programming
RTLS	Real Time Location Systems
SAPI	Speech Application Programming Interface
SGC	Sistema Global de Coordenadas
SLAM	Simultaneous Localization And Mapping
SMA	Sistemas Multi-Agente
SMA*	Simplified Memory-bounded A*
SPEM	Software Process Engineering Metamodel
SQL	Structure Query Language
STRIPS	Stanford Research Institute Problem Solver
TILAB	Telecom Italia Lab
TTS	Text To Speech
UDP	User Datagram Protocol
UML	Unified Modeling Language
UWB	Ultra-WideBand
VAHM	Véhicule Autonome pour personnes Handicapés Moteurs
WIMP	Window, Icon, Menu, Pointing device
XML	eXtended Modeling Language



# Resumo

Nos últimos anos, a quantidade de indivíduos idosos tem aumentado significativamente e pensa-se que o envelhecimento social é cada vez maior. Para muitos destes indivíduos, o envelhecimento traz consigo dificuldades na mobilidade, o que afecta sensivelmente a sua qualidade de vida e sua independência. Paralelamente, o aumento das actividades sociais voltadas para indivíduos com mobilidade reduzida, cada vez mais, revela a inadequação e a insuficiência funcional dos equipamentos actuais direccionados para este grupo de indivíduos, sobretudo, para aqueles com níveis mais elevados de necessidade. Outro problema que se revela neste contexto é a insuficiência de profissionais da saúde responsáveis por zelar e promover o bem-estar social a estes grupos de indivíduos.

Embora as dificuldades de mobilidade dos idosos e deficientes possam ser diminuídas com o uso de Cadeiras de Rodas tradicionais, manuais ou motorizadas, em alguns casos, a sua utilização de forma independente ainda é um desafio. O desenvolvimento de novas tecnologias, que proporcionem maior segurança e comodidade a estes indivíduos, é um problema científico (mobilizando cientistas de diversas áreas do conhecimento) e social (mobilizando políticos e empresários). Nos últimos anos, robôs móveis, autónomos, do tipo Cadeira de Rodas Inteligentes (CRIs) têm sido estudados e desenvolvidos para aplicações em hospitais ou em residências, com o objectivo de proporcionarem maior bem-estar e independência para indivíduos idosos e/ou indivíduos com mobilidade reduzida.

Este trabalho aborda o estudo, projecto e implementação de uma plataforma de desenvolvimento de CRI, que permita facilitar o desenvolvimento de sistemas sensoriais e de actuação, de novas metodologias de navegação e planeamento inteligente voltados para as CRI.

O trabalho propõe uma estrutura genérica e modular para o desenvolvimento de CRI baseada em Sistemas Multi-Agente (SMA). Relativamente ao SMA desenvolvido, destaca-se a aplicação dos conceitos de micro agente e macro agente. Assim, cada CRI representa um macro agente no SMA global, interagindo com outros agentes (outras CRI, Agente Porta, Agente Assistente, entre outros). No entanto, internamente, cada CRI é um SMA formado pelo agrupamento de diversos módulos distribuídos em agentes autónomos – micro agentes. A plataforma de comunicação de SMA para robôs móveis reais implementada baseia-se num conceito de cooperação entre os agentes para manter a organização do sistema de comunicação, reforçando a constante preocupação com aspectos de segurança crítica no sistema de comunicação.

Outro conceito proposto é a aplicação de interface multimodal, flexível, que permita uma interação mais natural entre o utilizador e a CRI através de sequências de diferentes *inputs* do utilizador para comandar a cadeira.

O conceito de Realidade Mista aplicado ao simulador IntellWheels desenvolvido permite testar a CRI em ambientes mais complexos e próximos daqueles que os seus utilizadores enfrentarão no mundo real, de uma forma mais segura e económica. Outra utilização possível para a Realidade Mista é na reabilitação de pacientes. Esta característica distingue o simulador IntellWheels dos demais existentes, uma vez que a maioria dos simuladores aplicados a esta área não explora este conceito.

No decurso do trabalho foram desenvolvidos três protótipos reais revelando os aperfeiçoamentos da plataforma de hardware, bem como a melhoria do software de controlo e restantes módulos. Testes e experiências realizadas com o protótipo real de CRI e apresentados nesta tese permitiram validar a plataforma IntellWheels, tanto em termos do funcionamento integrado de seus módulos, bem como da sua integração com outras CRI e outros macro agentes. A análise dos resultados permite mostrar a aplicabilidade e eficácia da abordagem proposta e, conseqüentemente, da plataforma desenvolvida.

# Abstract

The number of elder individuals has increased significantly during the last years and the social aging is also increasing. For most of these individuals the aging process brings also mobility problems affecting their life quality and independence. In parallel, the number of social activities for these individuals reveals the functional insufficiency of the devices used for helping them with emphasis on the more needed ones. Another problem is concerned with the reduced number of health professionals responsible for promoting the well being of these individuals.

Although the mobility difficulties of the elder and handicapped people may be diminished by the use of traditional wheelchairs, in some cases the use of such devices, in an independent manner is still a challenge. The development of new technologies that may bring a safety and commodity for these individuals is a scientific and social problem. In the last years, autonomous mobile robots, known as intelligent wheelchairs (IW) have been studied and developed for applications in hospitals or even at home, with the objective of promoting a better life quality for these individuals.

This thesis is concerned with the study, design and implementation of a development platform for intelligent wheelchairs that enables the easy development of new sensorial and actuator systems, navigation methodologies and intelligent planning algorithms for IW.

This work proposes a generic and modular framework for the development of IW based on multi-agent systems (MAS). The MAS developed is based on the concepts of micro and macro agent. Each IW represents a macro agent in the global MAS, interacting with the other agents (other IW, intelligent doors, among others). However, internally, each IW is a MAS composed by several modules distributed in autonomous agents - micro-agents. The communication platform for mobile robots developed is based on the concept of cooperation maintaining the global organization of the system and enforcing the concern with critical safety of the global system.

Another concept proposed is the use of a multimodal and flexible interface enabling a more natural interaction between the user and the IW through the use of input sequences composed by several inputs that trigger distinct actions of the device.

The mixed reality concept is also applied in the context of the IntellWheels simulator developed in order to enable to test the IW in more complex environments closer to the ones the users will face in the real world, in a more safe and economic manner. Another possible use of this concept is on patient rehabilitation. This characteristic distinguish the IntellWheels simulator from others since they do not explore this concept.

During this work, three prototypes were developed revealing the improvement of the hardware platform as well as the improvement in the control software and all the other modules. Tests and experiments using IntellWheels prototypes presented on this work enabled to validate the development platform. Both the global functioning, the behavior of each of its modules as well as the integration of the IW with other macro agents were tested and validated. The result analysis enables to show the applicability and effectiveness of the proposed approach and of the platform developed.

# Résumé

Ces dernières années, le nombre de personnes âgées a considérablement augmenté et l'on pense que le vieillissement de la société est de plus en plus. Pour beaucoup de ces personnes le vieillissement entraîne dès difficultés dans la mobilité, ce qui affecte considérablement leur qualité de vie et de l'indépendance. En parallèle, l'augmentation des activités sociales adaptées aux personnes à mobilité réduite, révèle de plus en plus de l'insuffisance et les troubles fonctionnels de l'équipement existant ciblée pour ce groupe d'individus, en particulier pour ceux ayant un niveau de besoin. Un autre problème qui se pose dans ce contexte est l'échec des professionnels de la santé charge de surveiller et de promouvoir Le bien-être de ces groupes d'individus.

Bien que les difficultés de mobilité des personnes âgées et les handicapés peuvent être réduits par l'utilisation de fauteuils roulants traditionnels, manuels ou motorisés, dans certains cas, leur utilisation est indépendante reste un défi. Le développement de nouvelles technologies qui offrent une plus grande sécurité et le confort de ces personnes, il s'agit d'un problème scientifique (associant des scientifiques de différents domaines de la connaissance) et sociale (la mobilisation des hommes politiques et hommes d'affaires).

Ces dernières années, les robots mobiles, de type autonome d'un Fauteuil Roulant Intelligent (FRI) ont été étudiés et développés pour des applications dans les hôpitaux ou les maisons, dans le but de fournir un plus grand bien-être et l'autonomie des personnes âgées et/ou des personnes ayant mobilité réduite.

Dans ce contexte, cette thèse aborde l'étude, la conception et la mise en œuvre d'une plateforme de développement pour l'FRI, ce qui facilite le développement des systèmes sensoriels et du comportement, de nouvelles méthodes de navigation et de planification intelligente axée sur les FRI. Penser à ce sujet, dans ce travail ont été proposées par la plateforme IntellWheels (PIW), certains concepts importants qui peuvent être appliqués à une nouvelle génération de FRI.

Un des plus importants de ces concepts est la structure générique et modulaire pour le développement de l'FRI sur Systèmes Multi-Agents (SMA). Pour le SMA développés, Il est l'application des concepts de l'agent micro et macro. Ainsi, chaque FRI représente un agent de la SMA global macro, en interaction avec d'autres agents (FRI, Port, agent adjoint, parmi d'autres). Toutefois, en interne, chaque FRI est un SMA formé par un ensemble de modules distribués agents autonomes – agents micro. Une plate-forme de communication pour les robots mobiles réels SMA mis en œuvre repose sur un concept de coopération entre les agents de maintenir un système de communication de l'organisation, le renforcement de la

préoccupation constant des aspects critiques pour la sécurité dans le système de communication.

Un autre concept proposé est l'application de l'interface multimodale, flexible, permettant une interaction plus naturelle entre l'utilisateur et l'IRC par des séquences des entrées d'utilisateur différents pour contrôler le président.

Le concept de réalité mixte appliquée à IntellWheels simulateur mis au point vous permet de tester l'IRC et de ses utilisateurs dans des environnements plus complexes et plus proches de ceux auxquels ils seront confrontés dans le monde réel dans un monde plus sûr et adorable. Une autre utilisation possible de la réalité mixte est la réhabilitation des patients. Cette caractéristique distingue le simulateur de IntellWheels autres existants, puisque la plupart des simulateurs de s'appliquer à ce domaine n'est pas explorer ce concept.

Enfin, la description de l'évolution réelle de trois prototypes développés durant cette thèse révèle les améliorations de la plate-forme matérielle, ainsi que le contrôle logiciel amélioré. Essais et des expériences menées avec le prototype réel du FRI et présenté dans cette thèse a permis de valider la plate-forme IntellWheels, pour les termes de l'opération intégrée de ses modules et son intégration avec d'autres FRI et d'autres agents macro. Les résultats peuvent montrer l'applicabilité et l'efficacité de l'approche proposée et, par conséquence, la plate-forme développée.

# Capítulo 1

## Introdução

**E**mbora seja comum estabelecer uma relação entre cadeiras de rodas e deficiência, nem sempre é fácil perceber os diferentes níveis e necessidades de cada tipo de deficiência ou problema de mobilidade. Nos últimos anos, a taxa de pessoas idosas tem aumentado significativamente e nota-se que o envelhecimento social é cada vez maior. O simples processo natural de envelhecimento causa alterações físicas, neurológicas e psicossociais que acabam por limitar os aspectos motores dos indivíduos idosos. Verifica-se que muitos idosos apresentam dificuldades de locomoção. Por outro lado, as lesões físicas ocorrem frequentemente e são causadas por acidentes, na maior parte dos casos domésticos. Estas acabam por afectar as capacidades motoras destes indivíduos, além de outros possíveis danos. As lesões físicas também podem ser causadas por condições médicas e doenças, como por exemplo: paralisia cerebral, esclerose múltipla, respiratória e circulatória, doenças genéticas ou químicas e exposição a drogas. Normalmente, as deficiências físicas acabam por limitar o controlo de alguns músculos dos braços, pernas ou face. É muito difícil generalizar as deficiências físicas, uma vez que cada indivíduo apresenta sintomas diferentes e utiliza diferentes estratégias para lidar com elas.

Actualmente, testemunha-se o aumento do número de pessoas da população mundial que carregam algum tipo de deficiência física e possuem dificuldades de locomoção. Entretanto, as enfermeiras encarregadas de trabalhar para proporcionar o bem-estar social a estes grupos de indivíduos estão restritas a um número insuficiente. Este problema vem preocupando políticos, empresários e cientistas. Também com aumento das actividades sociais para pessoas deficientes, cada vez mais equipamentos que possam contribuir para o bem-estar social destes indivíduos têm sido alvo de investigação. Diante disso, é desejado o desenvolvimento de novas tecnologias que proporcionem uma maior segurança e comodidade a estes indivíduos.

Com o objectivo de responder aos inúmeros problemas de mobilidade de indivíduos idosos ou deficientes, vários projectos relacionados com cadeiras de rodas inteligentes foram

criados nos últimos anos (Simpson, 1998) (Parikh et al., 2005) (Gonzalez et al., 2006) (Jia et al., 2007) (Rebsamen et al., 2007). As Cadeiras de Rodas Inteligentes (CRI) apresentam-se como uma boa solução para ajudar indivíduos com deficiências graves que não são capazes de operar cadeiras de rodas eléctricas clássicas, por si mesmos, em suas actividades diárias. Isto seja em hospitais seja para proporcionar um maior bem-estar para pessoas idosas e/ou pessoas portadoras de deficiências físicas específicas.

Uma Cadeira de Rodas Inteligente pode ser definida como um dispositivo robótico desenvolvido a partir de uma cadeira de rodas motorizada comercial, sistemas sensoriais, dispositivos de processamento e dispositivos de actuação. Esta tecnologia é utilizada essencialmente por indivíduos que apresentam deficiências físicas e/ou idosos com problemas de mobilidade. As CRI podem ser utilizadas principalmente em ambientes hospitalares mas também nos lares dos indivíduos que necessitam deste tipo de equipamento. Podem deste modo trazer independência, segurança e comodidade nas tarefas de locomoção nas actividades diárias do quotidiano destes indivíduos.

Hoje em dia já é possível construir este tipo de dispositivo a partir de hardware e dispositivos de controlo existentes, embora, estudos contínuos ainda sejam necessários para aprimorar a tecnologia e oferecer um maior nível de segurança e bem-estar para os potenciais utilizados.

As CRI podem ser de extrema utilidade pois, em alguns casos, o utilizador não é capaz ou tem grande dificuldade em utilizar uma cadeira de rodas tradicional. Nestes casos, as CRI são projectadas para facilitar e proporcionar a segurança e comodidade na mobilidade destes indivíduos, graças à sua capacidade de navegação semi-automática/automática, desvio automático de obstáculos, planeamento automático e comportamentos cooperativo e colaborativo. Através de uma interface mais flexível, adaptável e amigável com o utilizador é possível capturar o desejo do mesmo e traduzi-lo num comando. Este comando pode ser representado por uma linguagem de alto nível e/ou por intermédio de um variado número de dispositivos de entrada (voz, joystick, expressões faciais, movimentos da cabeça, gestos e teclados). Após interpretar o comando dado pelo utilizador, a CRI pode planear suas acções e executá-las de modo a atingir os objectivos propostos.

Apesar de existirem inúmeros projectos de investigação relacionados com as CRI e protótipos de CRI desenvolvidos para testar e validar teorias, aplicar novas tecnologias da informática e da robótica no apoio à deficiência profunda, verifica-se ainda a inexistência de cadeiras de rodas com reais capacidades de planeamento inteligente de acções e navegação autónoma. Embora muitos destes trabalhos tentem resolver alguns dos problemas específicos dentro desta área, nenhum autor destacou a preocupação em desenvolver uma ferramenta flexível que facilite o estudo e desenvolvimento de novas tecnologias nesta área. É neste ponto que esta tese se destaca em relação a esses trabalhos. O objectivo geral desta tese é estudar uma arquitectura, projectar e desenvolver uma plataforma que permita de forma fácil

e flexível estudar e desenvolver novas tecnologias para as CRI, com base em diversas metodologias, tais como: linguagens de comando em alto-nível, navegação semiautomática e automática, desvio de obstáculos, planeamento inteligente de acções, comunicação com outros dispositivos inteligentes, aplicação do paradigma de sistemas multi-agente (SMA) neste tipo de ambiente, entre outras.

## 1.1 Motivação

O envelhecimento populacional já é um fenómeno mundial e está a ocorrer tanto em países desenvolvidos como, de modo crescente, em países em fase de desenvolvimento. Uma projecção mundial para o envelhecimento da população, em termos demográficos, está ilustrada na Figura 1.

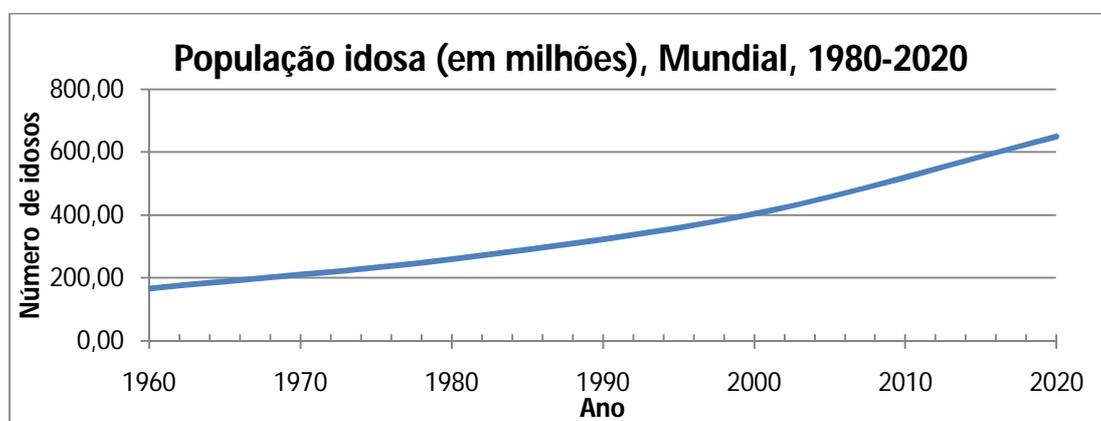


Figura 1: Crescimento mundial da população idosa

De acordo com estatísticas demográficas divulgadas pelo INE<sup>1</sup>, a evolução da população Portuguesa tem apresentado um continuado envelhecimento demográfico, em outras palavras, um incremento na taxa de idosos. Este aumento deve-se ao aumento da longevidade e declínio da fecundidade, ao aumento da eficácia da contracepção, ao prolongamento do período de estudos dos jovens, à entrada em massa das mulheres no mercado de trabalho, à crescente instabilidade das relações conjugais, entre outros factores. Nos últimos 20 anos o número de idosos duplicou em Portugal.

A Figura 2 apresenta o índice de envelhecimento em Portugal no período de 1981 à 2008 e uma projecção do índice até o ano 2050. Entende-se por índice de envelhecimento a relação existente entre o número de idosos e a população jovem. Este índice é expresso em número de residentes com 65 ou mais anos para cada 100 residentes com menos de 15 anos (de 0 a 14 anos).

<sup>1</sup> INE – Instituto Nacional de Estatística – Portugal. As estatísticas demográficas citadas foram publicadas em 10 de Novembro de 2008 e foram obtidos directamente do sítio <http://www.ine.pt>.

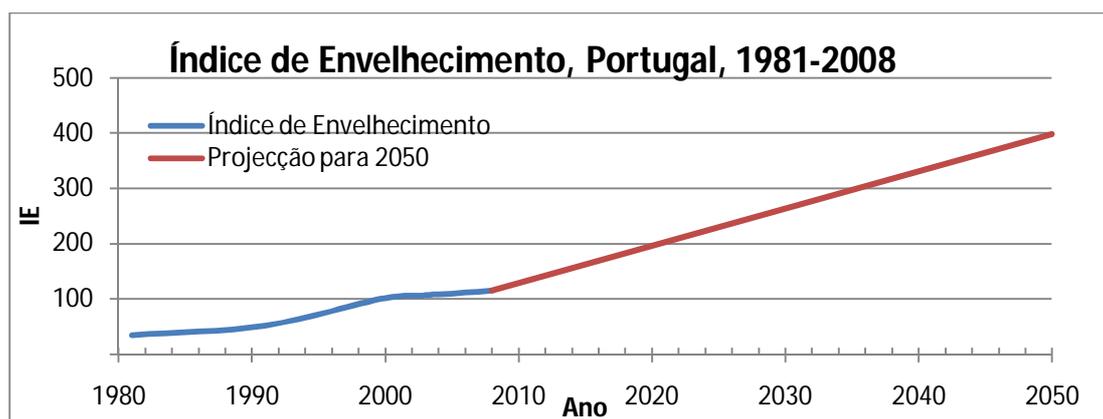


Figura 2: Índices de envelhecimento em Portugal

Ainda, segundo dados divulgados pelo INE, a população Portuguesa em 31 de Dezembro de 2008 era composta por 15,3% de jovens (com menos de 15 anos de idade), 17,6% de idosos (65 e mais anos de idade) e 67,1% de população em idade activa (dos 15 aos 64 anos de idade). Estes dados resultaram num índice de envelhecimento de 115 idosos por cada 100 jovens (114 em 2007).

O crescimento na taxa de idosos também é uma realidade noutros continentes. Segundo a comunicação social do IBGE<sup>2</sup>, em 27 de Novembro de 2008 o índice de envelhecimento no Brasil aponta para mudanças na estrutura etária da população. Em 2008, para cada grupo de 100 jovens existiam 24,7 idosos. Em 2050, os dados apontam para um quadro diferente no qual para cada 100 jovens existirão 172,7 idosos. Estes índices de envelhecimento da população Brasileira no período de 1980 a 2008 e com a projecção para 2050 estão ilustrados na Figura 3.

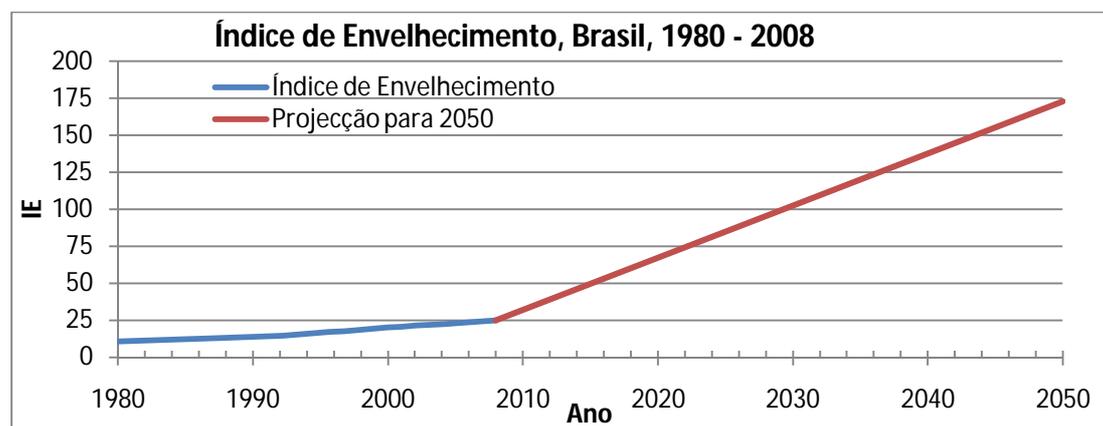


Figura 3: Índices de envelhecimento no Brasil

Embora a maioria dos governos estejam cientes de que o envelhecimento populacional é um processo natural, crescente, inevitável e de alcance global, poucos estão realmente preparados para lidar com os efeitos deste processo.

<sup>2</sup> IBGE – Instituto Brasileiro de Geografia e Estatística – Brasil.

O envelhecimento é biologicamente normal e evolui progressivamente, e não se dá, necessariamente, em paralelo ao avanço da idade cronológica. Com isso, o idoso acaba por ser forçado a modificar seus hábitos de vida e rotinas diárias passando a ocupar-se de actividades pouco activas e, assim, reduzir seu desempenho físico, suas habilidades motoras, sua capacidade de concentração, de reacção e de coordenação. Esses efeitos da diminuição do desempenho físico acabam dificultando a realização das actividades diárias e a manutenção de um estilo de vida saudável, gera apatia, auto desvalorização, insegurança e, conseqüentemente, leva o idoso ao isolamento social e à solidão.

Da mesma forma, alguns tipos de deficiências físicas reduzem as habilidades motoras. As deficiências físicas muitas vezes trazem dificuldades de locomoção por limitar o controlo de alguns músculos dos braços, pernas ou face. De acordo com os dados do recenseamento do INE em 12 de Março de 2001 foram contabilizadas 634.408 pessoas com deficiência, o que representava 6,1% da população residente em Portugal. A Figura 4 ilustra a taxa de incidência da população residente e a desagregação por tipos de deficiência.

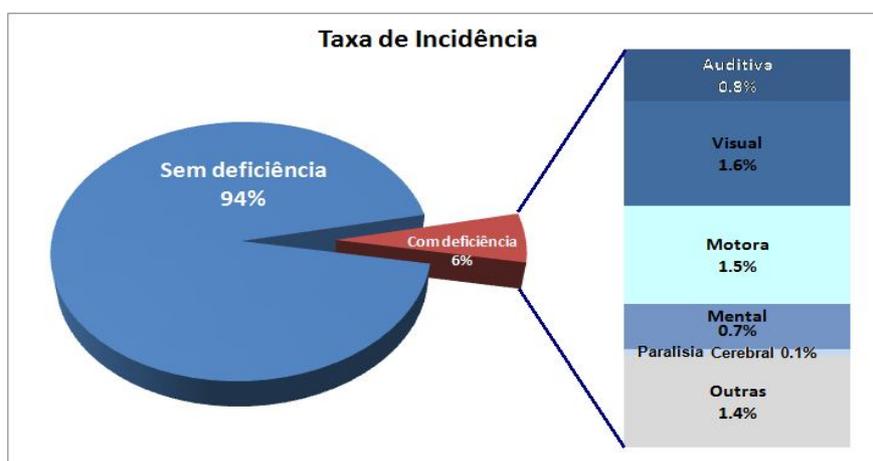


Figura 4: População sem deficiência e com deficiência, Portugal 2001

No Brasil, de acordo com dados de uma pesquisa no ano de 2000 realizada pelo IBGE verificou-se a existência de 25 milhões de pessoas portadoras de algum tipo de necessidade especial, o que representava 14,5% da população (IBGE, 2000). Ainda de acordo com o mesmo senso, existiam 955.287 pessoas tetraplégicas<sup>3</sup>, paraplélicas<sup>4</sup> ou hemiplégicas<sup>5</sup> permanentes. Segundo a Organização Mundial da Saúde (OMS), existem mais de 600 milhões de pessoas com necessidades especiais no mundo. Estatísticas apontam a existência de, aproximadamente, 5 milhões de indivíduos nos EUA que apresentam limitações físicas (Parikh et al., 2007).

As *Cadeiras de Rodas Inteligentes* podem auxiliar indivíduos com deficiências motoras severas bem como idosos com problemas de mobilidade na sua movimentação de uma forma

<sup>3</sup> A tetraplegia é quando uma paralisia afecta todas as quatro extremidades, superiores e inferiores, juntamente à musculatura do tronco.

<sup>4</sup> A paraplegia traduz-se na perda de controlo e sensibilidade dos membros inferiores.

<sup>5</sup> Hemiplegia é uma paralisia de metade do corpo.

segura, flexível e prática. Diversos estudos têm mostrado que crianças e adultos têm beneficiado do acesso aos meios independentes de mobilidade, o que inclui as cadeiras de rodas manuais e motorizadas, scooters, e “andadores” (Tefft et al., 1999), (Trefler et al., 2004). A independência na mobilidade aumenta oportunidades vocacionais e educacionais, reduz a dependência de enfermeiros e de familiares, o que acaba por resultar em sentimentos de autoconfiança por parte destes indivíduos. Para as crianças pequenas, a independência da mobilidade serve como base para a aprendizagem (Tefft et al., 1999).

Para adultos, a independência de mobilidade é um aspecto importante na auto-estima e representa um papel importante no caso de indivíduos idosos. Por exemplo, se idosos encontrarem cada vez mais dificuldade em andar ou no modo de locomoção, acabarão por ir com menor frequência à casa de banho ou até poderão começar a beber menos líquidos para tentar reduzir a frequência de urinar. As limitações da mobilidade são a causa principal de limitações funcionais entre adultos, com uma proporção estimada de 40/1000 pessoas com idade entre 18 e 44 anos e 188/1000 pessoas acima de 85 anos (Pope & Tarlov, 1991). As dificuldades na mobilidade são também fortemente evidenciadas nas *Actividades Diárias do Quotidiano* (ADQ).

Além disso, as dificuldades motoras resultam frequentemente na redução de oportunidades em socializar-se, o que conduz ao isolamento, à ansiedade, e à depressão. Por exemplo, o estudo (Iezzoni et al., 2001) mostrou que 31% das pessoas com dificuldades, principalmente de locomoção, relataram ter frequentemente depressão ou ansiedade, comparado com somente 4% das pessoas que não possuem dificuldades de mobilidade.

Embora as necessidades de muitos indivíduos com deficiência possam ser satisfeitas com as tradicionais cadeiras de rodas manuais ou motorizadas, alguns casos ainda encontram dificuldade ou impossibilidade de usar de forma independente este tipo de cadeira de rodas. Estes indivíduos frequentemente apresentam dificuldade em locomover-se de forma independente e necessitam de auxílio de outro indivíduo para movimentar a cadeira de rodas manual. A fim de aumentar o seu nível de conforto e independência, diversos investigadores têm sugerido o uso das tecnologias desenvolvidas originalmente para robôs móveis para criar CRI. As cadeiras de rodas inteligentes têm sido projectadas para fornecer o auxílio da navegação ao utilizador de diferentes maneiras, tais como assegurar um deslocamento livre de colisão, ajudar no desempenho das tarefas específicas (por exemplo, passagem através das entradas), e transportar o utilizador entre diferentes locais, de forma autónoma.

Um estudo recente indica que os clínicos têm um grande interesse nos serviços que uma cadeira de rodas inteligente pode oferecer (Fehr et al., 2000). Os resultados significativos deste estudo revelaram que:

- Os clínicos indicaram que 9% a 10% dos pacientes que recebem treino para o uso das cadeiras de rodas motorizadas frequentemente encontram grande dificuldade ou impossibilidade de usar este equipamento no desempenho das ADQ;

- Quando questionados especificamente sobre as tarefas de conduzir ou manobrar a cadeira de rodas inteligente, 40% dos pacientes relatam que estas tarefas são difíceis ou impossíveis;
- 85% dos clínicos entrevistados relataram que atendem alguns pacientes todos os anos que não podem utilizar uma cadeira de rodas motorizada porque lhes falta a habilidade motora requisitada, força ou precisão visual. Destes clínicos, 32% relataram atender, praticamente, um número de pacientes que não podem usar uma cadeira de rodas motorizada equivalente ao daqueles que podem.

Aproximadamente, metade dos pacientes é incapaz de controlar uma cadeira de rodas motorizada por métodos convencionais, de acordo com os clínicos que os tratam. Estes pacientes poderiam beneficiar-se directamente de um sistema de navegação automatizado. As cadeiras de rodas inteligentes são assunto de investigação desde o final dos anos 80 e recentemente tornaram-se assunto de interesse em todo o mundo.

## 1.2 Objectivos

O objectivo desta tese tem origem nas respostas às seguintes questões de investigação:

- Quais são os módulos necessários numa plataforma de desenvolvimento de CRI?
- Como poderá ser possível desenvolver uma CRI direccionada para um indivíduo com deficiências motoras severas?
- Como adequar as metodologias de planeamento tradicionais ao planeamento de sequências de acções em domínios dinâmicos e com conhecimento imperfeito do ambiente no qual a cadeira vai actuar?
- Que metodologias Multi-Agente são mais adequadas a fim de permitir à CRI interagir com outras cadeiras e dispositivos inteligentes?

Assim, procurando responder a estas questões, fixaram-se os seguintes objectivos:

- Desenvolver um conceito e projecto de uma plataforma de desenvolvimento de CRI;
- Construir um protótipo real de uma CRI, a partir de uma cadeira motorizada comercial com a adaptação de um sistema sensorial e de actuação adequado;
- Desenvolver uma interface que permita comandar a CRI de forma mais flexível e configurável;
- Estudar e desenvolver metodologias de planeamento adequadas à definição de sequências de acções de uma CRI;
- Desenvolver um simulador que permita testar as metodologias desenvolvidas;

- Criar uma modelação formal da plataforma;
- Implementar metodologias de comunicação e cooperação entre agentes nas CRI, de modo a permitir a sua correcta interacção com outros agentes inteligentes;
- Testar a plataforma e seus módulos desenvolvidos utilizando o protótipo real de CRI.

Em termos gerais, o foco principal desta tese consistiu em criar uma plataforma de desenvolvimento de CRI, tendo como objectivo principal a aplicação destas CRI em ambiente hospitalar com mapa previamente conhecido e em ambientes de alguma forma estruturados que possam facilitar a sua localização e de outros robôs de serviço que possam partilhar o mesmo ambiente.

De forma mais resumida, os objectivos mais específicos foram: investigar e desenvolver uma plataforma de desenvolvimento de cadeiras de rodas inteligentes que facilite a criação de novas metodologias de navegação, planeamento inteligente com o paradigma de sistemas multi-agente para resolver problemas associados às CRI. Espera-se que os resultados deste desenvolvimento possam trazer às cadeiras de rodas actuais reais capacidades de planeamento inteligente de acções e navegação autónoma e semi-autónoma. Tem-se ainda como objectivos específicos desenvolver algumas funções de operações tais como: controlo manual, semiautomático e automático. Estas funções irão determinar de que forma as acções, numa CRI, serão controladas. Para isso, será necessário propor soluções para os desafios relacionados com sistemas multi-agente, planeamento inteligente, representação e geração de planos, alocação de tarefas, coordenação e comunicação.

Finalmente, teve-se ainda como objectivos específicos a realização de um conjunto alargado de testes em simulação e também num protótipo de uma cadeira de rodas real de modo a validar os algoritmos implementados no mundo real.

### 1.3 Contribuições Científicas da Tese

Entre as principais contribuições desta tese é possível nomear:

- Conceito de uma plataforma de desenvolvimento de cadeira de rodas inteligente, nomeada por Plataforma IntellWheels (PIW). Embora exista um elevado número de projectos nesta área, até a escrita desse documento não foi possível encontrar referências de projectos que tenha por objectivo desenvolver uma ferramenta para facilitar estudar e desenvolver cadeira de rodas inteligente;
- Desenvolvimento de uma arquitectura baseada em sistemas multi-agente para a plataforma IntellWheels. Os poucos projectos que utilizaram esta tecnologia apresentam diferentes aspectos e de forma restrita. Neste projecto, procurou-se

uma aplicação mais ampla dos SMA com a utilização do conceito de micro e macro agente;

- Modelação do sistema multi-agente da plataforma IntellWheels. Ainda que existam alguns projectos que utilizam o paradigma de sistemas multi-agente nenhum destes apresenta uma modelação formal do sistema;
- Conceito modular da plataforma IntellWheels com os testes e validação dos módulos. Modularizar o sistema permitiu compreendê-lo melhor e facilitou a avaliação e o desenvolvimento de novos módulos;
- Estudo, desenvolvimento e testes de um controle multi-níveis distribuído;
- Conceito de micro e macro agente aplicado as cadeiras de rodas inteligente. Por um lado, uma cadeira de rodas inteligente pode interagir com outras cadeiras de rodas inteligentes e outros agentes inteligentes disponíveis no ambiente, este grupo forma macro agentes. Por outro lado, uma cadeira de rodas inteligente é composta por agentes (micro agentes);
- Criação e desenvolvimento da função de realidade mista para uso com as cadeiras de rodas inteligente. Esta funcionalidade é pouco explorada nesta área, a realidade mista permite testar a cadeira de rodas inteligente e treinar o utilizador de forma mais segura e económica;
- Estudo e desenvolvimento do módulo de Interface Multimodal nas cadeiras de rodas inteligentes. Uma subárea bem explorada dentro da área das cadeiras de rodas são as interfaces que permitem a interacção homem-máquina, mas o enfoque principal é desenvolver um método diferente de interacção (por gestos da cabeça, voz, reconhecimento da face, pensamento, etc). No entanto, o uso de uma interface multimodal, ou seja, a interacção através de múltiplas entradas e múltiplas saídas é uma novidade no âmbito das CRI.
- Desenvolvimento, implementação e teste de uma Plataforma de comunicação para o sistema multi-agente da Plataforma IntellWheels. As cadeiras de rodas inteligentes devem ser consideradas como um sistema crítico de segurança. No seu deslocamento num ambiente é muito possível que o mesmo venha a sofrer, por alguns instantes, problemas de conectividade com a rede de comunicação, o que resultaria em instabilidade em muitas plataformas de sistemas multi-agente. A plataforma de comunicação desenvolvida permite uma solução simples e eficaz para estes problemas.

## 1.4 Estrutura da Tese

Esta tese encontra-se estruturada em seis capítulos, onde o primeiro capítulo é formado por esta introdução ao trabalho que foi realizado.

O segundo capítulo apresenta algumas definições, premissas e restrições que servirão de proposições básicas para o desenvolvimento deste trabalho. Também neste capítulo será apresentada a fundamentação teórica da tese.

O estado da arte que contém uma síntese dos principais projectos de CRI está descrito no terceiro capítulo. Neste capítulo são discutidos as principais características em comum aos projectos mais relevantes nesta área.

O quarto capítulo descreve o conceito da PIW e a sua implementação, bem como, todos os módulos utilizados no desenvolvimento deste trabalho. Este capítulo está estruturado de forma semelhante ao segundo capítulo, uma vez que o segundo capítulo tem como objectivo fundamentar todas as implementações apresentadas no quarto capítulo.

No quinto capítulo é descrita a metodologia utilizada para os testes, a implementação dos protótipos de CRI reais e apresentados os resultados dos testes básicos de cada módulo desenvolvido. Neste quinto capítulo, são apresentados os cenários de teste e os resultados da avaliação nos três modos (simulado, real e realidade mista) para cada cenário.

O sexto e último capítulo apresenta as conclusões gerais deste trabalho, analisa os principais resultados obtidos, originalidades e discute as suas limitações. No final do sexto capítulo, algumas perspectivas de desenvolvimentos futuros são visionadas.

## Capítulo 2

# Definições e Fundamentos Teóricos

O objectivo deste capítulo é introduzir os diferentes temas científicos e tendências relevantes que estão relacionadas com os campos da robótica, inteligência artificial e mais precisamente vinculados às CRI. O capítulo está dividido em seis secções. Inicialmente é apresentando um pequeno resumo histórico da área de inteligência artificial com o enquadramento na área específica desta tese e em seguida algumas definições e premissas. Alguns dos conceitos específicos apresentados neste capítulo estão relacionados com as principais técnicas utilizadas em robótica inteligente, sistemas multi-agente e interfaces homem-máquina, nomeadamente no âmbito da robótica móvel com ênfase para a sua aplicação em cadeiras de rodas inteligentes.

### 2.1 Introdução

Desde o início dos anos 60, a inteligência artificial (IA) encontrou diversas aplicações industriais - na maior parte das vezes na área de decisão baseada em conhecimento para o projecto e a monitorização de produtos industriais ou processos. Esse facto foi realçado com avanços na informática, no advento de computadores pessoais e da Internet, desta forma muitas aplicações práticas de técnicas de inteligência artificial foram realizadas (Rezende, 2005).

Dada a impossibilidade de uma definição formal precisa para IA, visto que para tanto seria necessário definir primeiramente a própria inteligência foram propostas algumas definições operacionais, como por exemplo: “*uma máquina é inteligente se ela é capaz de solucionar uma classe de problemas que requerem inteligência para serem solucionados por seres humanos*” (McCarthy & Hayes, 1969). “*Inteligência Artificial é a parte da ciência da computação que compreende o projecto de sistemas computacionais que exibam características associadas quando presentes no comportamento humano a inteligência*” (Barr & Feigenbaum, 1981); ou ainda a “*Inteligência Artificial é o estudo das faculdades mentais*

*através do uso de modelos computacionais*” (Charniak & McDermott, 1985). Outros recusam-se a propor uma definição para o termo e preferem estabelecer os objectivos da IA: *“tornar os computadores mais úteis e compreender os princípios que tornam a inteligência possível”* (Winston, 1984). A inteligência artificial não se resume a raciocinar, decidir, aprender, planear, mas também em *“como integrar estes processos no seio de uma única entidade”* (Rezende, 2005).

Existem duas linhas principais de pesquisa para a construção de sistemas inteligentes: a linha conexionista e a linha simbólica. A linha conexionista visa à modelação da inteligência humana através da simulação dos componentes do cérebro, isto é, dos seus neurónios e das suas interligações. O modelo conexionista deu origem a área de redes neuronais artificiais (Bittencourt, 1998). A linha simbólica segue a tradição lógica, ou seja, modela a inteligência com o uso de formalismo lógico, e tem em McCarthy e Newell seus principais defensores (Bittencourt, 1998).

Além da linha conexionista e simbólica observa-se hoje o crescimento de uma nova linha de pesquisa em IA baseada na observação de mecanismos evolutivos encontrados na natureza, tais como, a auto-organização e o comportamento adaptativo. Nesta linha os modelos mais conhecidos são os algoritmos genéticos e os autómatos celulares (Bittencourt, 1998). Desta forma, pode-se sintetizar que as principais linhas de pesquisa para a construção de sistemas inteligentes são:

- Simbólica: utiliza formalismos do tipo lógico para simular o comportamento inteligente expresso através da linguagem. É a base para os sistemas periciais e para os sistemas baseados em lógica difusa.
- Conexionista: visa à modelação da inteligência humana através da simulação dos componentes do cérebro, isto é, de seus neurónios, e de suas interligações. É a base para as Redes Neuronais (RN).
- Evolutiva: simula a evolução natural para encontrar soluções para problemas complexos. É a base para métodos de optimização tais como os Algoritmos Genéticos e para a Vida Artificial.

Actualmente as principais áreas de pesquisa em IA simbólica são os sistemas periciais, aprendizagem, representação de conhecimento, aquisição de conhecimento, tratamento de informação imperfeita, visão computacional em robótica, controlo inteligente, modelação cognitiva, arquitecturas para sistemas inteligentes, linguagem natural e interfaces inteligentes.

Em relação a estas áreas de pesquisa a inteligência artificial distribuída aparece como um novo ponto de vista onde as técnicas tradicionais da IA combinadas com as funcionalidades dos sistemas distribuídos dão origem à noção de agente e, conseqüentemente, a de sociedade de agentes (Bittencourt, 1998).

A Inteligência Artificial Distribuída (IAD) é uma das áreas da IA que mais se desenvolveram nos últimos anos e apresenta um enorme potencial para aplicações (Durfée et al., 1989). A IAD estuda o conhecimento e os métodos de raciocínio que podem ser necessários ou úteis para que agentes computacionais participem de sociedades de agentes. Em termos de técnicas computacionais, a IAD reúne as de ambas as áreas: de sistemas distribuídos e as de IA (Bittencourt, 1998), sendo os principais problemas tratados:

- Permitir que agentes individuais raciocinem a respeito das acções, planos e conhecimentos dos outros agentes visando a coordenação e a cooperação entre os mesmos;
- Reconhecer e conciliar diferentes pontos de vista e interesses dos diversos agentes;
- Criar metodologias e projectar ambientes para a implementação de sistemas inteligentes distribuídos.

Desta forma, começaram a ser construídos sistemas multi-agente, em que vários elementos individuais colaboram no seu próprio interesse como uma equipa (por exemplo, jogos de futebol robótico (Reis & Lau, 2002) tipo RoboCup<sup>6</sup>), interação de acordo com certas normas, estabelecendo contactos, como na importante área do comércio electrónico, onde também existe interacção entre agentes de software e humanos. Nestas aplicações, cada agente tem de ter uma noção não só das crenças e intenções dos outros agentes, mas também do modelo e da política da própria estrutura/organização onde está inserido.

Alguns autores defendem que um agente artificialmente inteligente deve ser construído da base para o topo, pois a inteligência é uma propriedade emergente resultante da interacção de pequenas entidades com pouca inteligência (R.Murphy, 2000). Em muitas aplicações de sistemas multi-agente espera-se que inteligência emergja do sistema através da cooperação e colaboração dos agentes que compõem o sistema, mesmo que estes agentes não possam ser considerados individualmente como propriamente inteligentes. A *Cooperação* e *Colaboração* são, por vezes, usados como sinónimos, enquanto outros escolásticos usam esses termos distintamente de acordo com o grau de divisão do trabalho. Na cooperação os parceiros dividem o trabalho, resolvem sub-tarefas separadamente e então unem os resultados parciais num trabalho final. Na colaboração, os parceiros *juntos* efectuam o trabalho, podendo ocorrer uma divisão espontânea. Por exemplo: um parceiro pode responsabilizar-se pelos aspectos de baixo-nível da tarefa enquanto outro focaliza nos aspectos estratégicos (Miyake, n.d.). Na secção seguinte serão apresentadas algumas definições que servirá como conceitos base da tese.

As interacções entre os agentes e as organizações são modeladas a partir de conceitos normativos básicos, como obrigação, responsabilidade, autorização, poder e delegação. No entanto, devido ao não completo entendimento desses conceitos, as actuais técnicas de

---

<sup>6</sup> Maiores detalhes sobre a RoboCup pode ser encontradas em <http://www.robocup.org/>.

modelação organizacional, baseadas em geral em representações esquemáticas, falham em providenciar definições claras e precisas dos conceitos normativos, deixando-os à mercê da intuição dos utilizadores. Entretanto, tal facto não é recomendado, pela multiplicidade de significados que alguns desses conceitos podem ter, e ao seu carácter nuclear na definição das políticas organizacionais.

Algumas das principais aplicações de agentes inteligentes de software em desenvolvimento ou já desenvolvidas são:

- Manutenção e actualização de base de dados;
- Gestão da própria rede;
- Gestão administrativa e da produção, o onde utilizador defende as funções através de objectos e o agente realiza a execução e o controlo;
- Comércio electrónico, onde os agentes procuram especificações, preço, conveniência, entre outras características do produto ou serviço requerido pelo utilizador;
- Gestão de acesso móvel, onde os agentes processam a informação na fonte, evitando o congestionamento desnecessário da rede;
- Gestão de correio electrónico e de mensagens;
- Gestão de acesso à informação;
- Colaboração no processo de partilha de documentos;
- Adaptação da interface ao utilizador final, permitindo uma maior flexibilidade dos aplicativos em rede;

Embora as aplicações das metodologias de IA, Agentes Autónomos e Sistemas Multi-agente (SMA) sejam extremamente vastas, a sua utilização no auxílio de deficientes físicos é ainda muito reduzida. No entanto, esta é uma das áreas em que as metodologias da IA podem contribuir mais significativamente para incrementar a qualidade de vida dos seres humanos.

Em toda a história da humanidade, nunca as populações apresentaram expectativas de vida tão altas, fruto, principalmente de políticas de saúde pública e medicina preventiva, bem como dos avanços na área da pesquisa científica. Assim, o envelhecimento e a incapacidade funcional têm sido alvo de atenção e preocupação das pessoas.

Os deficientes físicos são aqueles indivíduos que, independente da idade, apresentam uma lesão que o impede definitiva ou temporariamente de executar suas funções usuais para a sua idade, que podem ser causadas por acidentes resultando em lesões na espinal medula (paralisia), amputações, lesões no cérebro, que afectam as capacidades motoras, etc. Podem

também ser provocadas por condições médicas tais como paralisia cerebral, esclerose múltipla, doenças circulatórias ou respiratórias, doenças genéticas ou exposição às drogas ou químicos.

A preocupação em permitir as pessoas, portadoras de deficiência, uma vida o mais normal possível vem crescendo muito, sendo uma preocupação actual para políticos, empresários e cientistas. Logo, diversos projectos para a construção de CRI foram iniciados em diversos pontos no mundo.

## 2.2 Definições, premissas e restrições

Devido a impossibilidade de uma definição formal precisa para muitos termos utilizados nesta tese seja por falta de um consenso entre autores da mesma área seja pela interpretação diferente de cada autor, nesta secção serão apresentadas algumas definições usadas como premissas na tese.

Inicialmente, define-se o conceito de cadeira de rodas inteligente (CRI). Tomando como base uma definição dada por (Hamagami & Hirata, 2004), define-se:

*Cadeira de rodas inteligente é um dispositivo robótico desenvolvido a partir de uma cadeira de rodas motorizada e dotado de sistemas sensoriais, processamento e de actuação. Supõe-se que a mesma deva apresentar pelo menos algumas características como: capacidades de navegação, de planeamento automático e capacidade ampliada de interacção homem-máquina; comportamento semi-autónomo com desvio de obstáculo, cooperativo entre um grupo de CRI, colaborativo entre utilizadores e outros sistemas.*

Inteligência, de forma simplista, pode ser definida como a capacidade de utilizar o conhecimento para resolver novos problemas e perceber como lidar com novas situações. Seguindo o mesmo raciocínio a Inteligência Artificial pode ser definida como a ciência para construir “máquinas” para fazer coisas que requerem inteligência quando feitas pelo homem.

A conceptualização, especificação e implementação de Agentes Autónomos é assunto essencialmente da área de IA, existem diversas definições sobre o conceito de Agente, mas não existe um consenso entre os autores da área sobre esta matéria, mas existe a noção de que a autonomia é essencial num agente. Com base no trabalho de Pattie Maes (Maes, 1996), onde uma simples definição de agente é: “*Um sistema computacional que habita um dado ambiente, sente e age nesse ambiente, e ao fazê-lo realiza um conjunto de objectivos ou tarefas para o qual foi projectado*”, e nesta noção central de autonomia e nos trabalhos de

Wooldridge e Jennings (Wooldridge & Jennings, 1995) (Russell & Norvig, 2002), iremos adoptar a seguinte definição de agente:

*“Um Agente é um sistema computacional, situado num dado ambiente, que tem a percepção desse ambiente através de sensores, tem capacidade de decisão, age de forma autónoma nesse ambiente através de actuadores, e possui capacidades de comunicação de alto-nível com outros agentes e/ou humanos, de forma a desempenhar uma dada função para a qual foi projectado.” (Reis, 2003).*

---

A partir da definição de agente, podemos estender a definição de modo a definir Sistemas Multi-Agente:

*Os SMA são sistemas formados por um conjunto de agentes autónomos (que possuem existência própria e independente da existência de outros agentes) que interagem entre si através de um ambiente compartilhado.*

---

Este paradigma enfoca o comportamento social, desta forma é preciso desenvolver mecanismos de interacção e coordenação entre os agentes. Pode-se citar duas características fundamentais nos SMA: capacidade de agir de forma autónoma nas suas decisões para alcançar seus objectivos; e capacidade de interagir com outros agentes através de protocolos de interacção que assemelham-se com as interacções sociais humanas (negociação, competição, coordenação, colaboração e cooperação).

Definir cooperação não é uma tarefa fácil, pois o seu conceito é mais complexo de que o de interacção e de colaboração, alguns autores e dicionários até definem cooperação e colaboração como sinónimos. Na cooperação pressupõe-se que existe uma relação de respeito mútuo e não hierárquica entre as entidades envolvidas, uma atitude de tolerância e convivência com as diferenças e um processo de negociação constante. Segundo Maçada (Maçada & Tijiboy, 1998) *“a diferença fundamental entre os conceitos de colaboração e cooperação está no facto de que para existir colaboração a entidade deve interagir com a outra existindo ajuda - mútua ou unilateral. Para existir cooperação deve haver interacção, colaboração mas também objectivos comuns, actividades e acções conjuntas e coordenadas”*.

De forma simplificada, assume-se Cooperação como sendo:

*Cooperação é o acto de um conjunto de agentes unir esforços através da divisão do trabalho para resolução de um assunto ou problema. Desta forma as entidades envolvidas resolvem subtarefas e então os resultados parciais são unidos num trabalho final*

---

E portanto, assume-se colaboração como sendo:

*Colaboração é a acção de um conjunto de agentes realizarem um dado trabalho em conjunto sem existir uma divisão em subtarefas.*

---

Foram definidas algumas restrições no desenvolvimento deste trabalho. A ter em conta, aspectos como praticabilidade das CRI e uma futura viabilidade comercial, foram impostas as seguintes restrições: simplicidade, requisitos de segurança, baixo custo, e baixo impacto visual na CRI. Entretanto, flexibilidade é uma característica importante para uma ferramenta de desenvolvimento (que é foco principal desta tese). Portanto, procura-se seguir as restrições impostas e manter a flexibilidade da ferramenta.

## **2.3 Robótica Inteligente**

Nesta secção são apresentados alguns conceitos, teorias e técnicas relacionadas com os algoritmos mais comumente utilizados na robótica inteligente e também utilizados nas CRI. A seguir, serão apresentados os conceitos: agentes reactivos, arquitectura de subsunção, modelação do mundo, algoritmos de pesquisa e planeamento automático.

### **2.3.1 Agentes Reactivos e Arquitectura de Subsunção**

Os agentes reactivos baseiam-se em arquitecturas reactivas. Neste tipo de arquitectura, os agentes não possuem representações internas e simbólicas do seu ambiente e nem utilizam mecanismos de raciocínio simbólico. Os agentes actuam seguindo um padrão de estímulo-resposta (Acção-Reacção) relativamente ao estado corrente do ambiente em que se encontram integrados. As principais características dos agentes e dos sistemas multi-agente reactivos são destacadas a seguir (Ferber & Gasser, 1991), (Wooldridge, 2001):

- Não há representação explícita do conhecimento: o conhecimento dos agentes é implícito (as suas regras de comportamento) e sua manifestação se evidencia através do seu comportamento e dos demais agentes;
- Não há representação do ambiente: o comportamento (resposta) de cada agente é baseado na sua percepção do ambiente (estímulo) a cada instante. Não há uma representação interna explícita do ambiente;
- Não há memória das acções: os agentes reactivos não mantêm nenhum tipo de histórico de suas acções, ou seja, o resultado de uma determinada acção passada não influencia directamente na decisão de uma acção futura;
- Organização etológica: a forma de organização dos SMA reactivos é similar à observada por animais que vivem em grandes comunidades, como por exemplo o comportamento das formigas e abelhas;

- Grande número de membros: em geral, os SMA reactivos possuem um grande número de agentes, com populações que podem chegar à ordem de milhares de membros.

Em suma, os agentes reactivos são muito simples e não possuem representação do seu ambiente. Suas reacções dependem unicamente de sua percepção deste ambiente. Uma forma usual de representar os comportamentos dos agentes reactivos é através de um conjunto de regras pré-definido.

O modelo de arquitectura reactiva foi proposto por Brooks para a construção de agentes autónomos, chamada Arquitectura de Subsunção (*Subsumption Architecture*) (Brooks, 1991). Esta arquitectura consiste em camadas de comportamentos, organizadas de forma hierárquica, concebida para realização de tarefas específicas. Os comportamentos competem uns com os outros de forma a influenciar nas acções finais do agente. Níveis inferiores da hierarquia representam os comportamentos mais elementares enquanto níveis mais altos representam comportamentos mais abstractos e elaborados.

### 2.3.2 Modelo do Mundo

De forma abrangente, uma maneira de estudar o deslocamento do robô é através da criação de um *Modelo do Mundo*. Neste pode-se utilizar mapas topológicos (representam o espaço de navegação numa rede de nós e arcos) ou métricos (representam o espaço de navegação total, num sistema de coordenadas globais). Nestes modelos os mapas são previamente construídos e armazenados para consulta durante o deslocamento do robô. O *mapa* é a representação das regiões contidas num ambiente, e deve incluir informações sobre as propriedades das regiões, os objectos contidos no ambiente. *Mapas métricos* ou baseados em grelhas representam o ambiente por meio de grelhas igualmente espaçadas, formando células. Cada célula indica a presença de um obstáculo na região correspondente (Borenstein & Koren, n.d.), (Thrun & Bücken, 1996).

Os mapas, para além de fornecerem uma visão global do ambiente, são utilizados pelo robô para se auto-localizar no ambiente. Um robô autónomo pode construir o mapa do ambiente ao mesmo tempo que se localiza. Esta técnica é vulgarmente designada por Localização e Mapeamento Simultâneos (*Simultaneous Localization And Mapping – SLAM*) (Leonard & Durrant-Whyte, 1991). Neste caso, a representação do ambiente é criada durante a navegação. Isso implica em que as informações contidas no mapa representem o ambiente num determinado tempo e pode não ser consistente com o estado actual do ambiente real. Entretanto, quanto mais explorado for o ambiente real, teoricamente, maior será a precisão das informações contidas no mapa.

### 2.3.3 Grafos

Grafo é o objecto básico de estudo da Teoria dos Grafos, que é uma área da matemática discreta, com suas raízes em jogos e recreações matemáticas. Actualmente, a Teoria dos Grafos tem sido aplicada em diferentes áreas (Informática, Investigação Operacional, Economia, Sociologia, Genética, etc.), pois um grafo constitui o modelo matemático que permite estudar as relações entre objectos discretos de qualquer tipo pertencentes a um determinado conjunto.

Um grafo  $G(V,A)$  é definido pelo par de conjuntos finitos  $V$  (conjunto não vazio de elementos chamados vértice ou nós do grafo) e  $A$  (conjunto de pares ordenados  $a=(v,w)$ , em que  $v$  e  $w$  pertencem a  $V$  e são chamados de arestas do grafo). Num grafo simples dois vértices  $v$  e  $w$  são adjacentes (ou vizinhos) se existe uma aresta  $a=(v,w)$  em  $G$ . Normalmente representa-se um grafo simples  $G(V,A)$  por um diagrama no qual os vértices são representados por pontos e as arestas por linhas unindo vértices adjacentes (Diestel, 2010),(Harary, 1969).

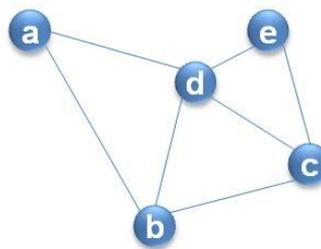


Figura 5: Diagrama que representa um grafo simples  $G$  composto por 5 vértices e 7 arestas

Por exemplo, no diagrama da Figura 5, o grafo  $G$  é definido por:

$$V = \{a, b, c, d, e\} \quad (\text{Eq. 1})$$

e

$$A = \{\{a, b\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{c, e\}, \{d, e\}\} \quad (\text{Eq. 2})$$

Em qualquer grafo simples existe, no máximo, uma aresta unindo cada par de vértice com uma relação simétrica, ou seja, as arestas que ligam os vértices não possuem qualquer orientação.

Na modelação de determinados problemas será conveniente considerar um sentido para as arestas. Para estes casos, é necessário a utilização de um grafo dirigido.

Um grafo dirigido (ou, abreviadamente, digrafo)  $D$  consiste num conjunto finito não vazio  $V$  de elementos chamados vértices, e num multi-conjunto finito  $A$  de pares ordenados de

elementos de  $V$ , chamados arcos. A relação definida por  $A$  não é simétrica, pois há uma orientação na relação. Um exemplo do uso de digrafos é a modelação de mapas de estradas com sentido único (Figura 6):

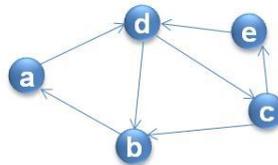


Figura 6: Diagrama que representa o grafo dirigido  $D$

Um outro exemplo de grafo dirigido é o grafo  $G_1$  (Figura 7) definido por  $V = \{p | p \text{ é uma pessoa da família Castro}\}$  e  $A = \{(v,w) | v \text{ é pai/mãe de } w\}$  que representa um conjunto de relações familiares.

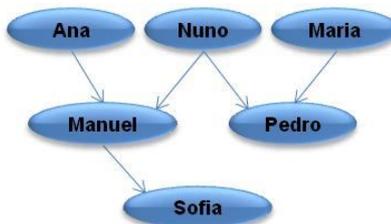


Figura 7: Grafo  $G_1$  que representa a relação das pessoas da família Castro

A Tabela 1 resume algumas propriedades dos grafos.

Tabela 1: Propriedades dos grafos	
Propriedades	Descrição
<b>Ordem</b>	A ordem de um grafo $G$ é dada pela cardinalidade do conjunto de vértices, ou seja, pelo número de vértices de $G$ . Nos exemplos acima: $ordem(G) = 5$ , $ordem(D) = 5$ e $ordem(G_1) = 6$ .
<b>Adjacência</b>	Num grafo simples (como por exemplo $G$ ) dois vértices $v$ e $w$ são adjacentes (ou vizinhos) se existe uma aresta $a=(v,w)$ em $G$ . Esta aresta é dita ser incidente a ambos, $v$ e $w$ . É o caso dos vértices $a$ e $b$ em $G$ . No caso de digrafo (a exemplo de $D$ ), a adjacência (vizinhança) é especializada em: <b>sucessor</b> – um vértice $w$ é sucessor de $v$ se há um arco que parte de $v$ e chega em $w$ . Em $G_1$ , por exemplo, diz-se que <i>Manuel</i> e <i>Pedro</i> são sucessores de <i>Nuno</i> ; <b>antecessor</b> – um vértice $v$ é antecessor de $w$ se há um arco que parte de $v$ e chega em $w$ . Em $G_1$ , por exemplo, diz-se que <i>Nuno</i> e <i>Maria</i> são antecessores de <i>Pedro</i> .
<b>Grau</b>	O grau de um vértice é dado pelo número de arestas que lhe são incidentes. Em $G$ , por exemplo: $grau(a) = 2$ e $grau(c) = 3$ . No caso de digrafos, a noção de grau é especializada em: <b>grau de emissão</b> – o grau de emissão de um vértice $v$ corresponde ao número de arcos que partem de $v$ . Em $G_1$ , por exemplo o $grauDeEmissao(Manuel) = 1$ , $grauDeEmissao(Nuno) = 2$ e $grauDeEmissao(Sofia) = 0$ ; <b>grau de recepção</b> – o grau de recepção de um vértice $v$ corresponde ao número de arcos que chegam a $v$ . Em $G_1$ , por exemplo o $grauDeRecepcao(Pedro) = 2$ , $grauDeRecepcao(Nuno) = 0$ e $grauDeRecepcao(Sofia) = 1$ .
<b>Fonte</b>	Um vértice $v$ é uma fonte se $grauDeRecepcao(v) = 0$ . É o caso dos vértices <i>Ana</i> , <i>Nuno</i> e <i>Maria</i> em $G_1$ .
<b>Destino</b>	Um vértice $v$ é um destino (sumidouro) se $grauDeEmissao(v) = 0$ . É o caso dos vértices <i>Sofia</i> e <i>Pedro</i> em $G_1$ .
<b>Laço</b>	Um laço é uma aresta ou arco do tipo $a=(v,v)$ , ou seja, que relaciona um vértice a ele próprio. Em $G_2$ há três ocorrências de laços para um grafo não orientado (ver Figura 8).

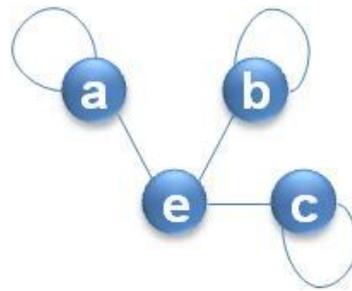
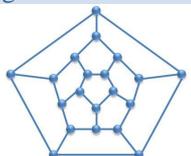
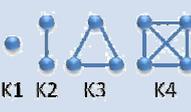
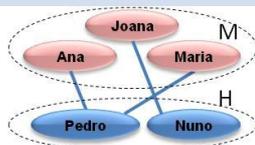
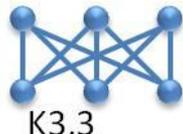


Figura 8: Grafo  $G_2$  não orientado com laços

Os grafos podem ser classificados de acordo com as relações entre os seus vértices (ver Tabela 2).

Tabela 2: Tipos de grafos		
Tipo	Descrição	Diagrama
<b>Grafo Regular</b>	Um grafo é considerado regular quando todos os seus vértices têm o mesmo grau. O grafo $G_3$ , por exemplo, é dito ser um grafo regular-3 pois todos os seus vértices tem grau 3.	 $G_3$ :
<b>Grafo Completo</b>	Um grafo é considerado completo quando há uma aresta entre cada par de seus vértices. Estes grafos são designados por $K_n$ , onde $n$ é a ordem do grafo. Um grafo $K_n$ possui o número máximo possível de arestas para um dados $n$ . Ele é, também regular-( $n-1$ ) pois todos os seus vértices tem grau $n-1$ .	 $K_1$ $K_2$ $K_3$ $K_4$
<b>Grafo Bipartido</b>	Um grafo é considerado bipartido quando seu conjunto de vértices $V$ puder ser particionado em dois subconjuntos $V_1$ e $V_2$ , tais que todas as arestas de $G$ unem um vértice de $V_1$ a outro de $V_2$ . Para exemplificar, sejam os conjuntos $H=\{h \mid h \text{ é um homem}\}$ e $M=\{m \mid m \text{ é um mulher}\}$ e o grafo $G(V,A)$ (ver o exemplo $G_4$ ) onde $V = H \cup M$ e $A=\{(v,w) \mid (v \in H \text{ e } w \in M) \text{ ou } (v \in M \text{ e } w \in H) \text{ e } \langle v \text{ foi namorado de } w \rangle\}$ O grafo $G_5$ é uma $K_{3,3}$ , ou seja, um grafo <b>bipartido completo</b> que contém duas partições de 3 vértices cada. Ele é completo pois todos os vértices de uma partição estão ligados a todos os vértices da outra partição.	 $G_4$ :  $K_{3,3}$ $G_5$ :
<b>Grafo Rotulado</b>	Um grafo $G(V,A)$ é considerado rotulado em vértices (ou arestas) quando, para cada vértice (ou aresta), estiver associado um rótulo. $G_5$ é um exemplo de grafo rotulado (além de ser um grafo bipartido).	
<b>Grafo Valorado</b>	Um grafo $G(V,A)$ é considerado valorado quando existe uma ou mais funções relacionando $V$ e/ou $A$ com um conjunto de valores. Para exemplificar (ver $G_6$ ), seja $G(V,A)$ onde $V=\{v \mid v \text{ é uma cidade com aeroporto}\}$ e $A=\{(v,w,t) \mid \langle \text{há linha aérea ligando } v \text{ a } w, \text{ sendo } t \text{ o tempo esperado de voo} \rangle\}$	 $G_6$ :

Os avanços mais importantes da Teoria dos Grafos têm sido motivados, quase sempre, pela tentativa de resolução de problemas práticos muito específicos — Euler e o problema das

pontes de Königsberg, Cayley e a enumeração de compostos químicos, Kirchoff e problemas de redes eléctricas, etc. Pode-se citar como problemas com importância na vida real:

- O problema do caminho mais curto – Utiliza-se grafo valorado para representa os possíveis caminhos ente diversos pontos (por exemplo cidades, ver  $G6$ ) e aplicando algum algoritmos de pesquisa sobre o grafo, procura-se encontrar o caminho mais curto entre dois pontos quaisquer representados neste grafo.
- O problema do carteiro chinês – Neste problema, um carteiro tem que distribuir cartas pelas casas de uma freguesia, voltando depois ao ponto de partida na estação dos correios. O problema é encontrar a menor distância que o carteiro terá de percorrer. Evidentemente terá de percorrer cada rua pelo menos uma vez, mas deverá evitar percorrê-las mais do que uma vez. Este problema impõe que se percorram todas as ARESTAS, pelo menos uma vez.
- O problema do caixeiro-viajante – Neste problema, um caixeiro-viajante pretende visitar várias cidades e voltar ao ponto de partida, percorrendo a menor distância possível. Este é um exemplo de problema que tem desafiado os investigadores, até hoje, na procura de um bom algoritmo. Este problema impõe que se percorram todos os VÉRTICES, pelo menos uma vez.

Vários problemas representados por um grafo podem ser resolvidos efectuando uma pesquisa nesse grafo. A pesquisa (ou procura) em grafo consiste em explorar o grafo, de forma sistemática ou com auxílio de alguma heurística.

### 2.3.4 Algoritmo de Pesquisa/Procura

A resolução de problemas, por métodos genéricos, é uma área da IA muito desafiadora e ocupa um papel importante na IA. Existem diferentes métodos e estratégias para a resolução de problemas com o uso de algoritmos de pesquisa (também designados por algoritmos de procura). Um algoritmo de pesquisa, de modo geral, é um algoritmo que recebe um problema como entrada e retorna a solução deste problema (em alguns casos, sob a forma de uma sequência de acções), geralmente após passar por diversas soluções. O conjunto de soluções válidas para um determinado problema é denominado de espaço de pesquisa (ou espaço de estados). Muitas estratégias de pesquisa utilizam grafos ou árvores para representar o espaço de estados do problema, ou seja, o modelo abstracto do problema. A solução é encontrada através duma pesquisa, guiada por alguma estratégica genérica, no espaço de pesquisa. No caso em que o importante é apenas a solução, então não será necessário manter guardada toda a árvore de procura, mas sim a sua fronteira. Caso o caminho para a solução seja importante, todos os caminhos que ligam a raiz da árvore aos nós na fronteira(Russell & Norvig, 2002) (Costa & Simões, 2008).

A saída de um algoritmo de pesquisa consiste no retorno do insucesso ou de uma solução. Embora alguns algoritmos possam ficar presos num laço de repetição infinito e nunca retornar uma saída. O desempenho dos algoritmos para resolução do problema pode ser analisado considerando três características, que se desejam atingir (Costa & Simões, 2008):

- **Seja completo:** se existir uma solução para o problema, ela será encontrada em tempo finito;
- **Seja discriminador:** caso exista várias soluções, que encontre a melhor;
- **Seja económico:** que encontre a solução em menos tempo (baixa complexidade temporal) e ocupe a menor quantidade de memória possível (baixa complexidade espacial).

As diferentes estratégias existentes podem ser agrupadas em função do grau de conhecimento sobre domínio do problema ou pelo tipo de solução: procura cega, procura heurística, procura local e procura estocástica.

#### 2.3.4.1 Procura Cega

Procura cega (também designada como procura sem informação) é quando não é utilizada nenhuma informação sobre o problema que possa ajudar no processo de geração e análise dos nós. Esta estratégia basicamente define um modo sistemático de navegar na árvore/grafos. A procura cega apresenta basicamente duas possibilidades, cada uma com variantes.

- **Primeiro em largura:** esta estratégia caracteriza-se por analisar primeiro todos os nós do mesmo nível. Uma vez não encontrada a solução, passa-se para o próximo nível. O processo de expansão e análise repete-se até alcançar o objectivo.
- **Custo uniforme:** nesta estratégia associa-se um custo às transições de estado. Este algoritmo é uma variante do em largura primeiro, que escolhe para expansão a cada momento o nó pertencente à fronteira cujo custo é o mais pequeno, este custo é calculado por uma função  $g(n)$ , que representa o custo para chegar do nó inicial ao nó corrente  $n$ .
- **Primeiro em profundidade:** este algoritmo segue uma filosofia completamente diferente da dos dois algoritmos apresentados anteriormente, este parte do nó raiz e vai expandindo um nó, escolhe um dos seus sucessores, expandi-lo por sua vez, continuando o processo até que encontre a solução ou o nó não possa ser mais expandido. Neste último caso, continua o processo com um irmão do último nó analisado, caso exista, ou regressa ao nível anterior para continuar o processo.
- **Profundidade limitada:** este algoritmo procura evitar o problema da incapacidade que o algoritmo de procura em profundidade primeiro apresenta quando lida com caminhos infinitos. Desta forma, este algoritmo fixa o nível máximo da procura.
- **Aprofundamento progressivo:** o algoritmo é a alteração do princípio da procura limitada, fazendo variar esse limite entre o nível 0 e infinito. Ele consiste na

chamada repetida do algoritmo de procura limitada para valores crescente do limite máximo. Ele combina aspectos positivos da procura em largura e da procura em profundidade.

### 2.3.4.2 Procura Heurística

Heurística é uma palavra provinda do grego e que significa encontrar. Segundo o standard ANSI/IEEE STD 100-1984, a heurística trata de métodos ou algoritmos exploratórios para solução de problemas. As soluções são procuradas por aproximações sucessivas, avaliando-se os progressos alcançados, até que o problema seja resolvido.

A procura heurística possui conhecimento para auxiliar na procura no espaço de pesquisa. Estes algoritmos utilizam informação adicional que torna-o capaz de estimar o custo do caminho do nó actual até ao nó solução. Esta informação é dada por uma função  $h(n)$ . Utiliza-se algoritmos de procura com heurística pelos seguintes motivos: evitar espaço de pesquisa exponencial; não dispersar recursos com caminhos irrelevantes; optimização da pesquisa.

Inúmeras variações e combinações destas técnicas são propostas na literatura. Enumera-se de seguida alguns algoritmos de procura que utilizam heurística:

- **Procura Gulosa:** consiste em escolher o nó na fronteira da árvore de pesquisa que parece ser o mais promissor de acordo com o valor de  $h(n)$ . Deste modo, o algoritmo limita-se a manter a fronteira da árvore de procura ordenada pelos valores de  $h(n)$ , sendo escolhido o nó que está teoricamente mais próximo da solução;
- **A\*:** este algoritmos combina as ideias da procura gulosa com a do custo uniforme para escolher a cada instante o melhor caminho passando pelo nó  $n$ , utilizando para tal a função  $f(n)=g(n)+h(n)$ . Devido a ampla utilização deste algoritmo na bibliografia e mesmo nesta tese, ele será detalhado na próximo subsecção;
- **IDA\*(interactive deepening A\*):** este algoritmo de pesquisa utiliza uma estratégia que auto-limita a memória que pode ser utilizada a cada momento. Trata-se de uma estratégia que utiliza a função  $f(n)=g(n)+h(n)$  e o aprofundamento é controlado pelos valores da função  $f(n)$ ;
- **SMA\*(Simplified Memory-Bounded A\*)(Russell & Norvig, 2002):** é um algoritmo que leva em consideração o problema da complexidade espacial do A\*. Para ganhar eficiência do ponto de vista da memória, o algoritmo abre mão do requisito de ser óptimo. A sua filosofia é simples: guardar o maior número possível de nós promissores e guardar informações sobre a qualidade dos nós abandonados, para evitar expandir repetidas vezes. De forma simplificada o algoritmo funciona mantendo numa fila, ordenados pelos valores de  $f(n)$ , a fronteira da árvore de procura. A dimensão da fila é fixa e é um parâmetro do algoritmo. Interactivamente, selecciona o elemento de menor custo,  $n$ , que, se for a solução

retorna com sucesso a procura. Caso contrário, calcula um dos seus sucessores  $s_n$  e determina o seu custo dado por  $f$ . Actualiza eventualmente o custo de  $f(n)$  e dos seus antecessores para guardar informações sobre a qualidade do caminho passando por  $n$ , no caso de todos os sucessores de  $n$  terem sido gerados. Se a memória esgotar-se, retira-se da fila o nó de maior custo e insere  $s_n$  na fila;

### 2.3.4.3 Procura Local

Os algoritmos apresentados até agora foram projectados para fazer a procura sistematicamente no espaço de pesquisa. A procura sistemática utilizada nestes algoritmos implica armazenar em memória um ou mais caminhos até o objectivo, e quando encontrado, o caminho faz parte da solução para o problema. No entanto, em muitos problemas, o caminho até o objectivo é irrelevante. Se o caminho não importa, pode-se considerar uma classe diferente de algoritmos, os que não se importam de forma alguma com os caminhos. Estes algoritmos de procura local utilizam apenas um único estado corrente e em geral movem-se apenas para os vizinhos deste estado. Embora estes algoritmos não sejam sistemáticos, eles têm duas vantagens: usam pouca memória (quase um valor constante); e frequentemente podem encontrar soluções razoáveis em grandes ou infinitos espaços de pesquisa para os quais os algoritmos sistemáticos são inadequados. Os algoritmos de procura local são úteis para resolver problemas de optimização.

Alguns algoritmos de procura local:

- **Subida da colina** (*Hill-Climbing*): Esta abordagem tem uma natureza tipicamente local e envolve ir melhorando progressivamente a solução candidata. Para além disso, o algoritmo descarta todos os vizinhos com menos valor. O maior ponto positivo desta abordagem é a sua simplicidade e a baixíssima complexidade espacial, porém não é um algoritmo completo e também não é óptimo. Alguns problemas conhecidos do algoritmo estão relacionados com os máximos/mínimos locais e planaltos que limitam o algoritmo não o deixando encontrar uma solução melhor;
- **Procura Tabu** (Glover, 1989): é um algoritmo heurístico, que procura melhorar progressivamente uma solução através de uma pesquisa local. O que distingue do algoritmo subida da colina é o uso de memória, designada normalmente por lista tabu, e que se destina a evitar ciclos explorar zonas promissoras ou forçar a visita a novas zonas de procura. Esse efeito é obtido devido a lista tabu conter ou as soluções que foram testadas ou os operadores que foram utilizados. A lista é utilizada principalmente para inibir o teste de soluções ou evitar operadores. Na procura tabu a memória pode ser utilizada segundo diferentes estratégias: para

intensificar a pesquisa na vizinhança de soluções de boa qualidade ou para diversificar a pesquisa para regiões do espaço de pesquisa ainda não visitados.

#### 2.3.4.4 Procura Estocástica

Ao contrário das metodologias apresentadas anteriormente, nas metodologias probabilísticas ou estocásticas são utilizadas variáveis aleatórias para guiar a pesquisa. Como por exemplo deste tipo de algoritmos temos:

- **Procura Aleatória:** este é um algoritmo cego que escolhe de forma aleatória qual o próximo nó a ser pesquisado e expandido. Este algoritmo salta de estado em estado sem nenhum critério. Para espaços de pesquisa muito grandes, a probabilidade de escolher um estado por onde já tenha passado é baixa, e em problemas que não se tem informação que possa ajudá-lo, pela sua simplicidade pode ser uma opção a considerar.
- **Arrefecimento Simulado** (*Simulated Annealing*): É um algoritmo semelhante ao subida da colina mas que procura evitar cair nos máximos locais ou nos planaltos. O algoritmo é baseado no processo térmico, designado por annealing ou recozimento, utilizado em metalurgia para obtenção de estados de baixa energia em sólidos. O processo consiste em duas fases: inicialmente a temperatura do sólido é aumentada para um valor elevado de fusão e seguidamente o arrefecimento é realizado lentamente até que o material se solidifique, o que faz com que os átomos que compõem o material se organizem numa estrutura uniforme com baixa energia. O algoritmo de arrefecimento simulado, de modo análogo, substitui a solução actual por uma solução vizinha tendo em conta a temperatura. Em cada momento é seleccionado aleatoriamente um sucessor do nó corrente. São utilizados dois factores na escolha do próximo nó: o valor da função heurística e o parâmetro de controlo  $T$  denominado temperatura. Este parâmetro, quando elevado, permite que o próximo estado escolhido possa ter valor heurístico pior, mas a medida que a temperatura reduz menor é a probabilidade de escolher um estado de pior qualidade. À medida que o algoritmo progride, o valor da temperatura é decrementado e o algoritmo converge-a para um óptimo local.
- **Algoritmos Genéticos:** esta metodologia de pesquisa lida com  $k$  estados por interacção, em vez de um. Este algoritmo baseia-se na selecção natural, ou seja, usa técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, selecção natural e recombinação. Os seus estados sucessores são gerados pela combinação de dois estados pais seleccionados de entre os indivíduos mais adaptados (com valor mais elevado da função objectivo), em vez de serem gerados pela modificação de um único estado. O algoritmo utiliza uma população de sucessões que vai evoluindo ao longo de diversas gerações. Na geração final, o indivíduo mais adaptado será a solução para o problema.

### 2.3.5 Algoritmo AStar (A\*)

Algoritmo A\* (lê-se: A Star) é um algoritmo para procura de um caminho num espaço de representação conhecida. Ele calcula o caminho num grafo de um vértice inicial até um vértice final ou através de uma representação por grelha. A\* é a combinação de aproximações heurísticas como do algoritmo de procura pela melhor escolha (*Best-first Search*) e da formalidade do Algoritmo de Dijkstra (Shapiro, 2000), ou seja, combina métodos heurísticos (métodos que utilizam algum tipo de informação sobre o problema para tomar decisões), e métodos formais (métodos que não possuem informações sobre o problema, mas analisam formalmente uma maneira de solucionar o problema).

O primeiro relato do algoritmo foi em 1968 por Peter E. Hart, Nils John Nilsson e Bertram Raphael (Hart et al., 1968). Este algoritmo quando utilizado com uma heurística apropriada, é considerado um algoritmo completo e óptimo (Costa & Simões, 2008) (Barr et al., 1986).

Assim, o A\* é muito utilizado em problemas de pesquisa, que vão desde a procura do melhor trajecto entre localidades até a resolução de problemas gerais, como por exemplo a resolução de um quebra-cabeças.

Na área da Robótica é bastante utilizado na navegação de um robô. Suponha-se uma rota do ponto *A* até o ponto *B*, a ser percorrida por um robô móvel. Basicamente o comportamento do método pode ser definido pela função  $f(x) = g(x) + h(x)$ , onde  $g(x)$  é uma função baseada em método formal que pode calcular a distância percorrida do ponto inicial (*A*) até o ponto em que o robô se encontra actualmente e a função  $h(x)$  é baseada num método heurístico e fornece a distância esperada do ponto actual até o destino (*B*). A soma destas duas funções resulta em  $f(x)$ , que seria a rota completa. Sendo as sub-rotas transformadas em nós dentro de um grafo, o objectivo deste método é a cada iteração procurar pelas sub-rotas de menor custo esperado, mantendo o custo do caminho até o estado actual. A Tabela 3 apresenta a descrição do algoritmo A\*.

**Tabela 3: Algoritmo A\***

```

Função A*(problema, InserirListaOrdenada, g+h): solução ou falha
1. Lista_nós ← FazListaOrdenada(EstadoInicial(problema))
2. Repete
  2.1. Se VaziaListaOrdenada(Lista_nós) Então
    2.1.1. Devolve falha
    Fim_de_Se
  2.2. nó ← RetiraListaOrdenada(Lista_nós)
  2.3. Se TesteObjectivo(nó) Então
    2.3.1. Devolve nó
    Senão
      2.3.2. InserirListaOrdenada(Lista_nós, g+h(Expansão(nó, Operadores(problema)))
    Fim_de_Se
  Fim_de_Repete
Fim_de_Função

```

Este método necessita de um mapa (ou grafo) bem definido do ambiente e não tem capacidade de planear rotas em ambientes com obstáculos móveis. O seu sucesso também está

bastante dependente da heurística utilizada para estimar o custo do caminho restante até ao destino.

### 2.3.6 Planeamento Automático

Esta crescente área de IA está presente em cenários como: planeamento de trajectória e movimentação de sistemas automáticos móveis; planeamento de percepção envolvendo acções de percepção de robôs para captação de informação do ambiente; planeamento de navegação que combinam a percepção e definições de trajectórias; planeamento de manipulação relacionado com movimentação de objectos como, por exemplo, montagem de peças, organização de contentores, entre outros (Ghallab et al., 2004).

Planear é o processo abstracto e deliberativo de escolha e organização de acções antecipando seus efeitos esperados. Por outras palavras, planeamento é a tarefa de encontrar uma sequência de acções para atingir um dado objectivo. Esse processo tem por objectivo atingir, da melhor forma possível, as metas pré-estabelecidas. Agentes autónomos e outros tipos de programas de software requerem fazer planeamento para executarem as suas tarefas. O Planeamento Automático (ou planeamento automatizado) é uma área da Inteligência Artificial (IA) que estuda este processo deliberativo por meio da computação. Para eles, dado o estado inicial em que eles estão, um conjunto de acções disponíveis e um conjunto de metas a serem satisfeitas deverão encontrar uma sequência de acções que permita satisfazer as respectivas metas. O problema de planeamento clássico inclui várias suposições que simplificam este problema: tempo atómico, não existência de eventos externos, efeitos de acções determinístico, omnisciência do conhecimento do agente e outros (Weld, 1999).

Um modo natural de resolver alguns problemas seria estudá-los, representá-los e desenvolver sistemas de planeamento de forma específica, ou seja, direccionados para um caso específico com fortes dependências do domínio. Porém, esta abordagem não interessa ao Planeamento Automático. Pelo contrário, esta área da IA está interessada em estudar e desenvolver sistemas capazes de planear sobre uma variedade de domínios distintos (sistemas independente do domínio), utilizando especificações e representações do problema, bem como conhecimento sobre o domínio. Evidentemente, o desenvolvimento de sistemas independente do domínio torna-se muito mais complexo.

A descrição de problemas em termos de enumeração de passos para a satisfação de um objectivo apareceu no final dos anos cinquenta, com problemas como a ida ao aeroporto (McCain et al., 2000) e os missionários e canibais (Lifschitz, 2000). O problema da ida ao aeroporto, por exemplo, foi definido por John McCarthy a partir da seguinte ideia: “Supondo que eu estou no escritório da minha casa e desejo ir ao aeroporto e considerando que o meu carro está na garagem da minha casa. A solução do problema seria: caminhar até ao carro e conduzir o carro até ao aeroporto”.

Nessa época, a dificuldade era centrada na correcta formalização do problema através de premissas que pudessem ser usadas para a geração automática da solução. Quando existiam, os algoritmos que usavam essas premissas na satisfação de objectivos eram complexos e específicos para cada problema. Alguns anos mais tarde, o próprio McCarthy propôs uma linguagem formal na tentativa de encapsular todas as componentes dos problemas temporais: o Cálculo Situacional (McCarthy, 1963). Este foi dos primeiros modos de lidar com o problema do planeamento e se utilizava do recurso da lógica. Nesta perspectiva, o objectivo, representado por meio de uma fórmula lógica, é encarado como um teorema que pretende-se provar. Os operadores utilizados são regras de inferência lógica do sistema. Esta abordagem, originada nos finais dos anos 1960, olha para o mundo como uma sequência de situações, que não são mais do que visões “instantâneas” do estado do mundo. Apesar da nova linguagem apresentar um avanço considerável em relação à representação de conhecimento, ainda faltava um algoritmo capaz de utilizar tal representação para a solução de problemas de planeamento.

Foi então que em 1971, Fikes e Nilsson lançaram o planeador STRIPS – *Stanford Research Institute Problem Solver* (Fikes & Nilsson, 1971). De formulação simples, o STRIPS marca o início do estudo de planeamento automático a partir da ideia dos operadores ADD-DEL-PRE.

Os operadores STRIPS permaneceram quase duas décadas como paradigma vigente, mas ainda assim, havia problemas difíceis de serem resolvidos por eles. A complexidade alta do seu algoritmo (pesquisa exaustiva no espaço de estados) fez com que muitos melhoramentos fossem propostos. A criação de novas linguagens de representação, específicas para planeamento, foi relegada a segundo plano – embora ainda hoje exista uma ampla comunidade de teóricos a trabalharem nesse sentido. Muitos outros planeadores utilizaram essa estratégia, compensando a falta de orientação na pesquisa com heurísticas de encadeamento (Fink & Veloso, 1994) ou linguagens mais expressivas (Nau et al., 1999). Um dos planeadores que teve um maior sucesso entre os planeadores baseados na ideia original do planeador STRIPS foi o PRODIGY.

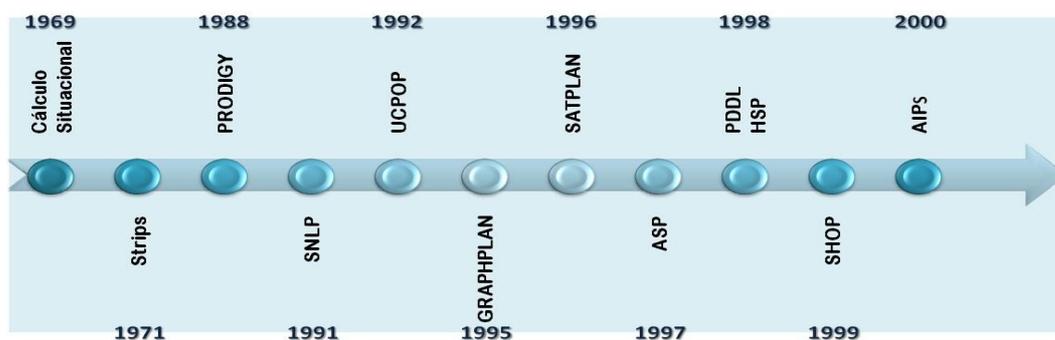


Figura 9: Linha do tempo no estudo de planeadores automáticos

A década de oitenta foi marcada por um desapontamento quanto à capacidade da geração automática de planos, e alguns, poucos, resultados só apareceram no final da década, como

pode ser observado na Figura 9. O planeador mais famoso do início da década de noventa foi o planeador PRODIGY (Minton et al., 1989),(Fink & Veloso, 1994) por um projecto homónimo da Universidade de Carnegie Mellon que adoptava uma hábil heurística de planeamento incluindo reordenação de operadores, aprendizagem automática (Minton, 1988) e busca em paralelo de sub-objectivos. Apesar desses esforços, os investigadores não conseguiam um mecanismo que fosse robusto o suficiente para resolver problemas genéricos.

Esse trabalho preserva a ideia inicial dos operadores STRIPS, representando os estados de um mundo a partir de um conjunto de literais e definindo uma série de operadores (pré-condição - acção - consequências) aplicáveis a esses estados. A geração de planos pelo PRODIGY é baseada em pesquisa de trás para frente (*backtracking*), realizada a partir de duas rotinas complementares: a simulação de planos e a regressão encadeada de estados. A simulação de planos é a aplicação de acções a partir do Estado Inicial, procurando atingir o Estado Objectivo (tal qual o STRIPS). A Regressão é uma heurística que gera estados intermediários, a partir do conjunto de pré-condições das acções que supostamente geraram o Estado Final. O projecto PRODIGY ficou famoso pelo pioneirismo no uso de aprendizagem automática, planeamento em tempo real e por tratar o planeamento automático como um problema de pesquisa. O sucesso do projecto pode ser creditado também ao grande número de investigadores envolvidos: Steven Minton, Jaime Carbonell, Jim Blythe, Xuemei Wang, Manuela Veloso, Dan Kahn, Oren Etzioni, Dan Kuokka e Daniel Borrajo, entre outros.

O desenvolvimento do PRODIGY teve muito impacto na comunidade de IA, servindo como base de comparação de desempenho com seus sucessores GRAPHPLAN e SATPLAN. Actualmente, a estratégia de pesquisa do PRODIGY persiste em planeadores que utilizam redes hierárquicas de tarefas para a geração de planos (Nau et al., 1999),(Smith et al., 1998).

Os anos noventa iniciaram com a tentativa de melhorar a velocidade de resposta na geração de planos e muitos estudos sobre linguagens e técnicas alternativas ao STRIPS. Nessa década, apareceram algumas metodologias válidas actualmente, como planeamento em ordem parcial, uso de variáveis na definição do esquema de acções, efeitos condicionais e linguagens baseadas em causalidade entre os operadores. SNLP (McAllester & Rosenblitt, 1991) e UCPOP (Penberthy & Weld, 1992) completavam com o PRODIGY o grupo de planeadores baseados em STRIPS de maior sucesso.

Em 1995, o planeamento automático sofreu novo estímulo quando uma metodologia de planeamento com base em grafos foi apresentada por Avrim Blum e Furst, o GRAPHPLAN (Blum & Furst, 1995) - um mecanismo de representação de intervalos de tempo e extracção de planos através de grafos. A simplicidade do novo planeador, combinado ao seu desempenho muito superior aos planeadores da época, estimulou o desenvolvimento de muitos outros planeadores a partir de sua ideia central (S. Kambhampati et al., 1997), (Smith & Weld, 1998), (Kautz & Selman, 1999), (Simpson & McCluskey, 1999). Tal sucesso

permitiu a recuperação de ideias associadas a planeamento baseado na satisfabilidade. Este planeador chamou a atenção da comunidade de IA por dois motivos:

- É um algoritmo simples e robusto, que apresenta uma velocidade de planeamento melhor que seus antecessores PRODIGY (Fink & Veloso, 1994) e UCPOP (Penberthy & Weld, 1992).
- Permitiu a automação da redução de problemas de planeamento ao problema da satisfabilidade proposicional (Kautz & Selman, 1992), (Kautz & Selman, 1999).

O GRAPHPLAN funciona em duas fases:

- *Expansão do grafo de planeamento*: estende o grafo progressivamente até atingir uma condição necessária (mas não suficiente) para um plano existir.
- *Extracção da solução*: faz uma pesquisa regressiva no grafo a procura de um plano solução; se não o encontrar, GRAPHPLAN volta a expandir o plano.

O grafo de planeamento é constituído de dois tipos de nós: os nós proposicionais e os nós que representam acções, distribuídos em níveis diferentes. Os nós proposicionais aparecem nos níveis pares e representam o conhecimento do mundo num determinado instante de tempo. Nos níveis ímpares, aparecem os nós que representam as acções cujos pré-requisitos estão presentes no nível anterior. Então, verifica se todos os literais do objectivo estão no nível actual com nenhuma ligação mutex (exclusão mútua) entre qualquer par delas. Se este fosse o caso, então uma solução pode existir dentro do grafo actual, assim o algoritmo tenta extrair essa solução. Se não, expande o grafo adicionando as acções para o nível actual e os literais do estado para o nível seguinte. O processo continua até que, ou uma solução esteja encontrada, ou se verificar que nenhuma solução existe.

No ano de 1996, Henry Kautz e Bart Selman lançaram o SATPLAN (Kautz & Selman, 1992), (Kautz et al., 1996) um planeador que traduzia o conhecimento do GRAPHPLAN ao problema da satisfabilidade. Tal transformação na forma de representar os problemas mostrou-se mais eficiente que o GRAPHPLAN na maioria dos problemas em que os dois sistemas eram comparados. Partindo deste princípio, Henry Kautz e Bart Selman propuseram em (Kautz & Selman, 1992) a formalização de problemas de planeamento automático como conjuntos de cláusulas proposicionais. A satisfação desse conjunto de cláusulas por um algoritmo chamado GSAT surpreendeu os autores pela velocidade superior aos demais planeadores da época. Desde então, muitos planeadores vêm explorando esta técnica, incluindo os conhecidos MEDIC (Ernst et al., 1998) e Blackbox (Kautz & Selman, 1998).

Em 1998, Hector Geffner propôs uma das técnicas mais rápida de planeamento conhecida até os dias de hoje: o planeamento por pesquisa heurística - HSP (Bonet & Geffner, 1999). A grande contribuição de Geffner foi a definição de uma função de custo que permitia verificar a distância dos estados intermediários ao estado objectivo.

Nessa mesma época, começaram a aparecer esforços para definir um modelo comum de testes que permita comparar os planeadores entre si, e assim surgiu uma linguagem de descrição de domínios em planeamento automático, denominada de PDDL – *Planning Domain Description Language* (McDermott, 1998),(Bacchus, 2000). A partir da PDDL, foram estabelecidos critérios e seleccionados problemas (*benchmarks* (Helmert, 2002)) para a realização de uma competição de planeadores automáticos AIPS – *Artificial Intelligence Planning Systems* (McDermott, 2000). Em 1998 foi fundada a competição bianual de planeamento automático, chamada IPC – *International Planning Competition*. Os primeiros passos dados pelos organizadores da IPC nas suas investigações foram guiados pelos destaques das competições de 1998 e 2000. Eles não estavam focados em identificar as ferramentas em si, mas sim as abordagens mais eficientes em relação aos critérios comparativos do AIPS:

- O número de problemas resolvidos por cada planeador;
- O tempo requerido para a obtenção de cada solução;
- A qualidade do plano final gerado.

Este terceiro critério, dito qualidade, está relacionado com o senso comum humano sobre as acções esperadas diante de um determinado problema. No caso do AIPS, os aspectos focados foram a redundância e o número de acções dos planos gerados. Outra contribuição das competições foi o estabelecimento de termos comuns para referenciar os elementos constituintes de problemas de planeamento e suas soluções.

Em 2002, o IPC forneceu uma extensão da PDDL (PDDL 2.1) para domínios de planeamento com expressão temporal. Os planeadores LPG (Gerevini & Serina, 2002), MIPS (Edelkamp & Helmert, 2001), TALPlanner (Doherty & Kvarnstrom, 2001) e TLPlan (Bacchus & Ady, 2001), entre outros, suportam o formato PDDL 2.1 e geram planos para domínios temporais. O planeador LPG-TD (Gerevini et al., 2004) suporta a extensão PDDL2.2.

### 2.3.6.1 STRIPS

STRIPS (*Stanford Research Institute Problem Solver*) permite especificar um problema de planeamento considerando algumas suposições sobre o problema e a sua representação. Um problema em STRIPS (Fikes & Nilsson, 1971) é definido por um tuplo  $P = \langle A; O; I; G \rangle$  sendo  $A$  um conjunto de átomos,  $O$  um conjunto de operadores, e  $I \subseteq A$  e  $G \subseteq A$  representam o estado inicial e as condições meta respectivamente. Os operadores  $op \in O$  são assumidos como instanciados. Cada operador tem listas de pré-condições, adição e eliminação,  $Prec(op) \subseteq A$ ,  $Add(op) \subseteq A$  e  $Del(op) \subseteq A$  respectivamente. O problema STRIPS pode ser mapeado a um problema de planeamento definido como  $S_P = \langle S, s_0, S_G, A, f, c \rangle$ , sendo:

- Os estados  $s \in S$  são colecções de átomos em  $A$ ;
- O estado inicial  $s_0$  é  $I$ ;

- Os estados meta  $s \in S$  são aqueles em que  $G \subseteq s$ ,
- As acções  $a \in A(s)$  são os operadores  $op \in O$  tais que  $Prec(op) \subseteq s$ ;
- A função de transição  $f$  mapeia estados sem estados  $s' = s - Del(a) + Add(a)$  para todo  $a \in A(s)$ ,
- Todos os custos de acções  $c(a,s)$  são 1.

Um estado é formado pelo conjunto de átomos que são verdadeiros nele. A suposição do mundo fechado, considerada num problema STRIPS, consiste em que os átomos não presentes num estado  $s$  são considerados negativos. Desse modo só é necessário conservar os átomos positivos em cada estado.

### 2.3.6.2 Linguagem PDDL

A linguagem de descrição de domínios em planeamento automático (PDDL) foi desenvolvida especialmente para a Competição de Planeadores, instituída a partir de 1998 (AIPS-98), onde o objectivo foi unificar a definição de domínios e problemas para sistemas de planeamento, para permitir compará-los de forma directa.

A PDDL é uma linguagem centrada na acção, inspirada pela formulação STRIPS de problemas de planeamento e ADL (*Action Description Language*). O seu núcleo é uma simples padronização de uma sintaxe para expressar semânticas familiares de acções, usando pré e pós-condições para descrever a aplicabilidade e os efeitos das acções. A sintaxe é inspirada na linguagem Lisp. A linguagem PDDL separa a descrição de acções parametrizadas, que caracterizam o comportamento do domínio, da descrição das condições iniciais e dos objectivos que caracterizam a instância do problema. Desta forma, o problema de planeamento é criado pela junção da descrição do domínio com a descrição do problema. Uma mesma descrição de domínio pode fazer par com várias diferentes descrições de problemas para produzir diferentes problemas de planeamento dentro de um mesmo domínio.

A extensão PDDL 2.1 foi desenvolvida de modo a representar domínios de planeamento com expressão temporal. Esta extensão permite modelar problemas do mundo real que envolvam tempo e restrições de recursos na geração de planos (Fox & Long, 2003). Já para o IPC-2004, uma nova extensão da PDDL (PDDL 2.2) foi criada para permitir a modelação de problemas de planeamento envolvendo eventos exógenos incondicionais determinísticos (isto é, representam factos que se tornam verdadeiros ou falsos em pontos de tempos determinados independentemente das acções no plano) e predicados derivados (Edelkamp & Hoffmann, 2003).

Para entender melhor a PDDL, será descrito o problema clássico do “Mundo dos Blocos - MDB” em PDDL. Na Tabela 4 é apresentado parte da definição do domínio e conjunto de acções que resolvem o problema. Como exemplo, será demonstrado apenas a acção “Pegar”, que significa pegar um bloco sobre a mesa.

Tabela 4: Definição do domínio do Mundo dos Blocos

```
(define (domain MDB)
  (:predicates (livre ?x) (sobre_mesa ?x)
               (braco_vazio) (seguro ?x) (sobre ?x ?y))
  (:action PEGAR
    :parameters (?x)
    :precondition (and (livre ?x) (sobre_mesa ?x) (braco_vazio))
    :effect (and(seguro ?x)(not(livre ?x))(not(sobre_mesa ?x))(not (braco_vazio)))
  ))
```

A acção “Pegar” contém em seus parâmetros (objectos a serem manipulados), suas pré-condições e seus efeitos. As demais acções para este problema são “Largar”, “Desempilhar” e “Empilhar”, todas contendo suas respectivas pré-condições, efeitos e parâmetros. A declaração do problema a ser solucionado dentro do domínio “MDB”, contém os objectos, estado inicial e estado objectivo, como ilustrado na Tabela 5.

Tabela 5: Definição do problema Mundo dos Blocos

```
(define (problem MDB1) ; Também conhecido como arquivo de fatos
  (:domain MDB)
  (:objects A B - blocos)
  (:init ; Declaração do estado inicial
    (sobre_mesa A) (sobre B A) (livre B) (braco_vazio))
  (:goal ; Declaração do estado objetivo
    (and (sobre_mesa A) (sobre E A) (sobre C E) (sobre_mesa B) (sobre D B)))
  )
```

### 2.3.7 Navegação e Controle

A navegação é capacidade de um robô móvel movimentar-se, dentro do seu ambiente, desde uma posição inicial até uma posição final (com base num ponto de referência), enquanto se desvia de obstáculos que podem estar no seu caminho. O problema da navegação foi resumido por Leonard e Durrant-White (Leonard & Durrant-Whyte, 1992) através de três questões: Onde estou? Para onde ir? Como chegar lá? (Borenstein et al., 1996).

A navegação de robôs móveis está envolvida na realização de quatro tarefas básicas:

- **Localização:** O robô encontra sua própria posição no ambiente, normalmente com base nas informações dos sensores e da representação interna do ambiente em forma de mapa;
- **Mapeamento:** É a construção de mapas, que representam o ambiente, através do sistema sensorial à medida que o robô atravessa o ambiente. A modelação do ambiente é realizada pela análise dos dados sensoriais para a construção e/ou modificação dos mapas.
- **Planeamento:** Determinar um caminho no ambiente entre a posição inicial e a posição final, sem que o robô colida com os obstáculos. Este planeamento é

realizado através da pesquisa de um caminho possível no mapa construído ou pré-armazenado.

- **Controlo:** É o método que controla o robô na sua condução através do caminho definido anteriormente. O movimento do robô é então controlado utilizando-se seu modelo cinemático e dinâmico. Durante o movimento do robô, o processo de percepção examina continuamente os dados sensoriais a fim de detectar potenciais colisões. Quando uma possível colisão é percebida, é iniciado um processo de desvio do obstáculo.

### 2.3.7.1 Métodos de Navegação na Robótica

Os métodos de navegação robótica podem ser divididos em dois grupos: medidas relativas de posição e medidas absolutas de posição (Borenstein et al., 1996).

#### Navegação baseada em medidas relativas de posição

**Navegação com Base em Odometria:** Este método baseia-se geralmente no uso de *encoders* para calcular o deslocamento do robô. Este efectua uma medição da rotação das rodas. Por outras palavras, este método integra a informação incremental do movimento ao longo do tempo. As vantagens deste método de navegação são a sua autonomia, custo, simplicidade e a capacidade de fornecer sempre ao robô a estimativa da sua posição. A grande desvantagem deste método é os erros acumulados que aumentam sem limites. Estes erros podem ser determinísticos (podem ser eliminados por calibração do sistema) ou não determinísticos (podem ser descritos por modelos geralmente estatísticos). Deve-se então utilizar em complemento uma referência independente de forma periódica para reduzir o erro, como por exemplo marcos artificiais. As principais fontes de erros sistemáticos desta técnica são: rodas desalinhadas; rodas com diâmetros diferentes; medida incorrecta da distância entre rodas. As principais fontes de erros não-sistemáticos são: variação do ponto de contacto da roda; escorregamento e derrapagem das rodas; intervenções externas.

**Navegação Inercial:** Esta técnica de navegação utiliza giroscópios e acelerómetros para fazer as medidas das rotações e acelerações do robô. Estas medidas são integradas com o objectivo de estimar a velocidade e posição do robô. A vantagem deste sistema de navegação é de ser autónomo. A principal desvantagem é o alto custo dos sensores e os erros acumulados pela integração. Os giroscópios modernos utilizam fibra óptica (também são conhecidos como giroscópios a laser), o que resulta numa elevada precisão.

#### Navegação baseada em medidas absolutas de posição

**Localização absoluta com base em faróis (*Active Beacons*):** Esta técnica calcula a posição absoluta do robô com base nas medidas da direcção de incidência de três ou mais sinais transmitidos (faróis). Os transmissores usualmente usam luzes ou frequências de rádio e são localizados em lugares conhecidos do ambiente. Entretanto, existem também

faróis/balizas passivos, que não emitem sinal. Por exemplo algo detectado por uma câmara. Existem ainda duas técnicas com faróis, a triangulação e a trilateração.

**Localização absoluta com base em marcos naturais** (*Natural Landmarks*): Esta técnica utiliza marcos com características próprias distintas do ambiente, o qual não precisa ser preparado. Entretanto, implica ter um grande conhecimento prévio a respeito do ambiente. A confiança nas informações obtidas com esta técnica é menor do que as obtidas utilizando técnicas baseadas em marcas artificiais.

**Localização absoluta com base em marcos artificiais** (*Artificial Landmarks*): Esta técnica utiliza marcos artificiais distintos os quais são localizadas em lugares conhecidos no ambiente de navegação. A vantagem destes marcos artificiais é que eles podem ser projectados para serem detectados e reconhecidos, mesmo em ambientes com condições adversas.

**Localização absoluta com base no ajuste de modelos** (*Model Matching*): Através desta técnica a informação obtida pela percepção do robô é comparada com um mapa modelo do ambiente. Se as características do mapa criado com base nos sensores do robô e o mapa modelo do ambiente coincidem, então a posição absoluta do robô pode ser estimada.

### 2.3.7.2 Planeamento do Movimento

Planear o movimento significa avaliar os caminhos livres de colisões desde a posição inicial até a posição final, a levar em consideração os aspectos geométricos, físicos e o tempo despendido nesta tarefa. O planeamento do movimento pode ser dividido em três aspectos: planeamento da trajectória (*Path Planning*); planeamento de manobras (*Manoeuvre Planning*); geração de trajectórias (*Trajectory Generation*).

Um caminho (*Path*) é uma rota entre o ponto de partida e de destino, ou seja, o conjunto de pontos no espaço geométrico por onde o robô deve passar para chegar no destino, enquanto a trajectória é um caminho no qual é especificada a aceleração e velocidade em cada ponto.

#### Planeamento da Trajectória

As metodologias de planeamento de trajectórias podem ser divididas em métodos globais e métodos locais. Os métodos globais permitem que o robô localize-se em termos absolutos (*global positioning system*) ou relacionado a um sistema de mapas, e mover-se para um ponto desejado. Metodologias como mapa de caminhos (*roadmap*) e decomposição celular (*cell decomposition*) fazem parte deste grupo. A metodologia dos campos potenciais (*potential fields*) pertence ao grupo dos métodos locais. Estes métodos possibilitam que o robô determine sua posição relativa a objectos próximos (dinâmicos ou não) e interaja com estes objectos (de Berg et al., 2008).

**Mapa de caminhos:** é um conjunto de posições, que um robô pode assumir, onde algumas delas (ou todas) estão interconectadas entre si. Cada conexão entre duas posições

representa um caminho válido, livre de colisão. As diferentes formas de obter um mapa de caminho são através de grafos de visibilidade, diagramas de Varonoi, caminhos livres (*freeway*) e silhueta (*silhouette*).

**Decomposição Celular:** este método divide o espaço contínuo livre do robô num número finito de simples regiões ou células levando a uma pesquisa discreta.

**Campos Potenciais:** é baseado na construção de campos de força (atractivo ou repulsivo), análogos aos campos magnéticos ou gravitacional. O robô é considerado como se fosse um objecto que sofre a influência de um campo de forças. O ponto de destino exerce uma força atractiva. Os obstáculos ou algo que o robô deva evitar exercem força repulsiva sobre ele.

### 2.3.7.3 Modelo Cinemático e Controlo do Movimento

A modelação cinemática de um robô móvel é um processo de baixo para cima, onde cada roda contribui para o movimento do robô e ao mesmo tempo impõe restrições sobre essa acção. As rodas são presas no chassis do robô com uma certa geometria e, portanto, suas limitações combinam-se para formar as restrições sobre o movimento global do robô.

Nesta subsecção serão descritas as restrições relacionadas ao movimento das cadeiras de rodas mais tradicionais, apresentada a modelação cinemática de uma cadeira de rodas inteligente e finalmente alguns controladores.

Na robótica, o termo holonómico está ligado às restrições cinemáticas dos robôs, sendo uma restrição chamada de holonómica quando essa é representada por uma única função explícita de variáveis de posição (Siegwart & Nourbakhsh, 2004). Considera-se uma restrição não holonómica quando somente as variáveis de posição não são suficientes para determinar um estado. Neste caso, a equação que representa o sistema não é integrável na posição final e é necessário um relacionamento diferencial (derivadas das variáveis de posição). Os robôs móveis diferenciais (como as CRI) são sistemas não holonómicos, pois as distâncias percorridas por suas rodas (variáveis de posição) não são suficientes para determinar a posição final do robô. É necessário conhecer o comportamento temporal das variáveis de posição, ou seja, como o movimento foi executado em função do tempo.

Ao analisar as CRI, cuja o movimento e orientação são realizadas por dois actuadores independentes, com modelação dada por objecto rectangular, movendo-se num espaço vazio com velocidade  $V$ . O estado do robô num plano cartesiano inercial  $\{0, x, y\}$  é completamente definido pelo vector:

$$(x_c, y_c, \theta, v_c, \omega_c)^T \quad (\text{Eq. 3})$$

Onde  $x_c$  e  $y_c$  são as coordenadas do ponto central do eixo das rodas,  $\theta$  o ângulo formado entre a base do robô  $\{c, x_c, y_c\}$  e a base inercial,  $v_c$  a velocidade linear no ponto  $C$  e  $\omega_c$  a velocidade angular (Figura 10).

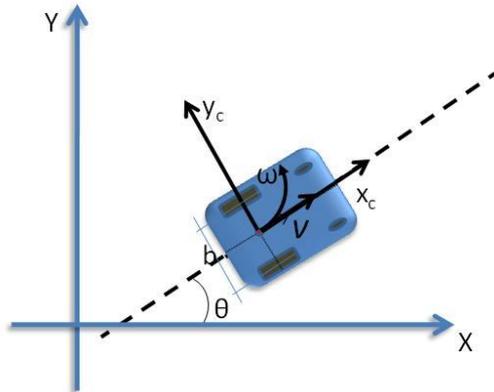


Figura 10: Representação de um robô não-holonômico

Sendo que as restrições não-holonômicas determinam que o robô móvel só pode se deslocar na direção normal ao eixo das rodas motoras e supondo que não existe, escorregamentos, o movimento fica restrito à relação (Yang & Kim, 1999):

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta = 0 \quad (\text{Eq. 4})$$

Com base nas informações obtidas pelos *encoders*, o cálculo da velocidade linear de cada roda é deduzido através de uma relação entre o número de impulsos dos *encoder*  $N$  e seu período de amostragem  $T$ :

$$v = \frac{N\pi D}{R_e T} \quad (\text{Eq. 5})$$

Onde  $v$  e  $D$  são respectivamente a velocidade linear e o diâmetro da roda, e  $R_e$  é a resolução do *encoder*. Uma possibilidade de representação das variáveis de estado é com base nas velocidades em relação ao ponto de contacto entre a roda direita ( $v_D$ ) e esquerda ( $v_E$ ) com o chão:

$$(x_c, y_c, \theta, v_D, v_E)^T \quad (\text{Eq. 6})$$

A escolha desta forma de representação dá-se essencialmente pela facilidade de medir estas grandezas através do sistema de odometria. A Figura 11 ilustra os vectores que representam as velocidades das rodas do robô ao realizar um movimento de rotação em torno de um ponto  $C$ , sendo  $b$  a distância entre os pontos de contacto das rodas,  $r$  o raio das rodas,  $L$  a distância do ponto  $C$  ao centro de rotação,  $\{\omega, \omega_D, \omega_E\}$  são as velocidades angulares (do centro de movimento da CRI, da roda direita e da roda esquerda, respectivamente).

Consideram para já o sistema contínuo, temos:

$$\begin{cases} v_D = \left(L + \frac{b}{2}\right)\omega = r_D\omega_D \\ v_E = \left(L - \frac{b}{2}\right)\omega = r_E\omega_E \end{cases} \quad (\text{Eq. 7}) \quad \begin{cases} v_D = v + \left(\frac{b}{2}\right)\omega \\ v_E = v - \left(\frac{b}{2}\right)\omega \end{cases} \quad (\text{Eq. 8})$$

Onde:

$$\begin{cases} v_D + v_E = 2\omega r = 2v \\ v_D - v_E = \omega L \end{cases} \quad (\text{Eq. 9})$$

Ou a partir das equações 5 e 6, chega-se a:

$$\begin{cases} v = \omega_D \frac{r_D}{2} + \omega_E \frac{r_E}{2} \\ \omega = \omega_D \frac{r_D}{b} + \omega_E \frac{r_E}{b} \end{cases} \quad (\text{Eq. 10})$$

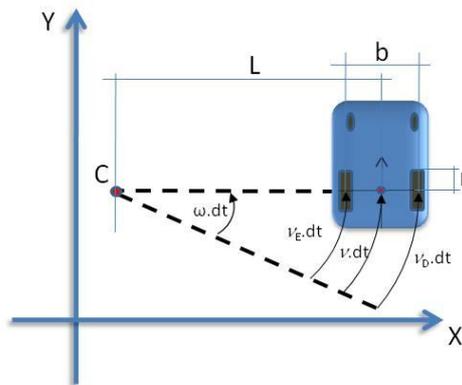


Figura 11: Representação dos vectores velocidades das rodas da cadeira

Desta forma, obtém-se a equação 9, que a partir das velocidades das rodas do robô pode-se calcular as velocidades lineares e angulares.

$$\begin{cases} v = \frac{v_D + v_E}{2} \\ \omega = \frac{v_D - v_E}{b} \end{cases} \quad (\text{Eq. 11})$$

A partir das velocidades da CRI, é necessário descobrir sua posição  $(x_C, y_C, \theta)$  no plano de referência, ou seja, resta conhecer como o resto do espaço de estados do robô evoluirá ao longo do tempo dadas as velocidades  $v_D$  e  $v_E$ .

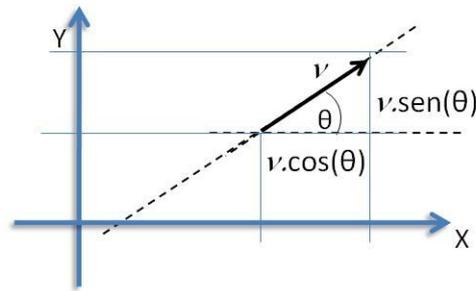


Figura 12: Representação dos vectores velocidade no centro de movimento  $C_m$

A Figura 12, uma representação em detalhe da Figura 10, apresenta os vectores de velocidade do robô em seu centro de movimento  $C_m$  de um robô móvel não-holonómico. Desta forma, considerando a condição de não escorregamento, pode-se descrever as equações cinemáticas de movimento do ponto  $C$  em relação a sua velocidade linear ( $v$ ) e angular ( $\omega$ ) (Bonci et al., 2005), (Indiveri, 1999), (Yang & Kim, 1999):

$$\begin{cases} \dot{x}_c = v \cos \theta \\ \dot{y}_c = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (\text{Eq. 12})$$

Ou em notação matricial:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{Eq. 13})$$

Importante salientar que a validade destas equações está restrita a robôs móveis não-holonómicos. Em casos em que não haja rolamento puro das rodas, existem diversas alternativas de modelação, entretanto, as mesmas não serão discutidas neste trabalho.

## 2.4 Interação e Interfaces Multi-Modais

Esta subsecção apresenta uma visão geral referente a área da interação homem-máquina com ênfase nas metodologias e ferramentas aplicáveis no contexto das CRI. Esta visão geral inclui definições básicas e terminologias, tecnologias envolvidas e ainda alguns projectos genéricos que utilizam Interface Multi-Modal (*multimodal interface* – MMI). Projectos que apresentam tecnologias de interação ligadas as CRI serão discutidas no Capítulo 3, sendo apresentado neste capítulo projectos mais genéricos.

Interação homem-máquina diz respeito ao estudo da interação entre as pessoas e as máquinas (computadores, equipamentos em geral). Nos últimos anos os avanços na informática criaram vários novos campos de estudo. Um deles é a Interação Humano - Computador. Esta é uma área multidisciplinar relacionada com engenharia informática, design, ergonomia, psicologia, sociologia, artes, linguística e áreas a fins. Esta interação dá-se através de interface do utilizador (ou interface homem-máquina – IHM), formada por software e hardware. De forma geral, uma interface é um elemento que estabelece a fronteira

entre as duas entidades. Sendo assim, a interface do utilizador é o sistema que possibilita aos utilizadores interagir com as máquinas, dispositivos e computadores. Isto ocorre através de dois métodos: entrada (permite ao utilizador manipular o sistema), e saída (permite ao sistema reproduzir os efeitos/respostas das acções do utilizador).

As interfaces com o utilizador gráficas tradicionais são baseadas no paradigma WIMP (*Window, Ícones, Menus, e um dispositivo apontador*), o qual usa o rato e o teclado como dispositivos de entrada físicos para interagir com a interface.

O avanço do campo de estudo da Interação Homem-Máquina não foi apenas na qualidade da interacção, também experimentou diferentes ramificações. Ao invés de apenas actualizar os projectos das interfaces tradicionais, os diferentes ramos desta área de investigação apresentaram diferentes enfoques, como por exemplo o conceito de multimodalidade – em oposição ao de unimodalidade. Este conceito evoluiu e passou a lidar com vários canais de interacção ao mesmo tempo resultando em interfaces mais adaptativas. Outra evolução está relacionada com o conceito de inteligência. Ao invés de interfaces baseadas em comando/acção esta área evoluiu para o conceito de interfaces activas que se adaptam ao utilizador. Todos esses enfoques têm por finalidade encontrar uma maneira de criar uma interacção mais natural com o utilizador.

A motivação no desenvolvimento de interfaces do utilizador deve-se à maioria das máquinas sofisticadas serem inúteis a menos que possam ser utilizadas adequadamente. Este argumento básico apresenta as condições principais que devem ser consideradas no projecto duma IHM: funcionalidade e usabilidade (Te'eni et al., 2007).

*Funcionalidade* dum sistema é definida pelo conjunto de acções ou serviços oferecidos aos seus utilizadores. No entanto, a importância da funcionalidade só é visível quando se torna possível de ser utilizada de forma eficiente pelo utilizador (Shneiderman & Plaisant, 2004). *Usabilidade* de um sistema com uma determinada funcionalidade é o grau e a escala pelo qual o sistema pode ser usado de forma eficiente e adequada para atingir os objectivos de um determinado utilizador. A real eficácia dum sistema é alcançada quando há um equilíbrio entre a sua funcionalidade e a sua usabilidade (Nielsen, 1994).

Tendo em conta os conceitos apresentados e a consideração de que os termos computador, máquina e sistema são muitas vezes utilizados de forma intercambiáveis neste contexto, os projectos de IHM devem possibilitar o ajuste entre o utilizador, a máquina e os serviços necessários para atingir um certo desempenho, tanto na qualidade como na optimização dos serviços (Te'eni, 2006).

A tecnologia disponível pode também afectar a forma como os diferentes tipos de IHM são projectados para a mesma finalidade. Um exemplo é o uso de comandos, menus, interfaces gráfica do utilizador (GUI), ou a realidade virtual para dar acesso as funcionalidades de um determinado equipamento.

## 2.4.1 Arquitecturas de Sistemas de Interação Homem-máquina

Um factor muito importante no projecto de uma interface homem-máquina é a sua configuração. As interfaces geralmente são definidas pelo número e variedade de entradas e saídas que oferecem. A arquitectura deste tipo de sistema define quem são as entradas e saídas e como elas interagem/trabalham, umas com as outras. Existem diferentes configurações e projectos na qual as interfaces são baseadas, como mencionado no início do capítulo pode-se classificar as interfaces de acordo com a modalidade:

### Sistemas Unimodais

São sistemas que possuem um único canal de comunicação. Por outras palavras, existe uma relação entrada/saída de um para um. Baseado na natureza do canal de interação, eles podem ser divididos em três categorias:

- Baseado em visão: esta categoria é a mais ampla na área de investigação em interação homem-computador. A considerar a grande quantidade de aplicações, variedade de problemas em aberto e abordagens, os investigadores que trabalham em diferentes aspectos da resposta humana que pode ser reconhecida com o sinal oriundo da visão. Pode-se citar como algumas das principais áreas de investigação, que tem por base a visão de máquina: a análise de expressões faciais; rastreamento dos movimentos do corpo humano; reconhecimento de gestos; detecção do olhar (rastreamento do movimento dos olhos).

- Baseado em áudio: a interação entre homem e máquina baseado em áudio é uma área importante nos sistemas de interação. Os sinais de áudio não são variáveis como os sinais obtidos dos sistemas de visão mas a informação obtida através do sinal de áudio são mais confiáveis e mais úteis, e em alguns casos são as únicas fontes de informação. A área de investigação relacionada com os sistemas baseados na interação por áudio pode ser dividida nas seguintes subáreas: reconhecimento da fala (Rabiner, 1993); reconhecimento do locutor; análise da emoção pelo som; Detecção de sons/ruídos produzidos pelo homem (respiração, suspiro, riso, choro, etc); interação musical.

- Baseado em sensores: esta categoria combina diversas áreas com um largo conjunto de aplicações. Estas diversas áreas possuem o sensor físico como ponto comum para possibilitar a interação homem-máquina. Estes sensores podem ser desde básicos aos mais sofisticados, como sejam: interação com base em caneta; rato e teclado; joysticks; sensores de movimento e digitalizadores; sensores de tacto; sensores de pressão; sensores de cheiro e gosto.

### Sistemas multimodais

Os sistemas multimodais combinam vários canais de comunicação (Jaimes & Sebe, 2007). Existe deste modo uma relação entrada/saída de  $n$  para  $m$ . A definição dos tipos de canais é

herdada dos tipos de comunicação humana, que são basicamente os seus sentidos: visão, audição, tacto, olfacto, e paladar. A possibilidade de interacção com a máquina inclui todos estes canais mas não está limitada unicamente a estes tipos. A interface multimodal actua como um facilitador de interacção entre o homem e o computador (ou máquina) via uma ou mais modos de entrada indo bem além dos tradicionais teclados e ratos. O número exacto de modos de entrada suportados, seus tipos e o modo no qual eles trabalham em conjunto pode variar muito de um sistema para outro. As interfaces multimodais podem conter diferentes combinações de modos de entrada como fala, gestos, expressões faciais, posição dos olhos e outros modos de entrada não convencional. Uma das combinações mais comuns é os gestos com a fala.

## 2.4.2 Projectos de Interface Multimodal

Um exemplo clássico de sistema multimodal é o “*Put That There*” (Bolt, 1980). Este sistema permitia mover um objecto de um ponto para outro de um mapa representado no ecrã apontando para o objecto e para o local destino desejado, em seguida dando o comando de voz “*Put That There*”. O número de projectos que utilizam o conceito de interface multimodal vem crescendo em diversas áreas de aplicação como, por exemplo, sistemas para auxiliar deficientes físicos a interagirem com computadores/máquinas/dispositivos, sistemas para reconhecer o estado emocional do utilizador, aplicações baseadas em mapas, interface homem-robô, interfaces na área da medicina.

A interface multimodal pode oferecer um número de vantagens muito maior que as interfaces tradicionais, mas a principal vantagem é oferecer uma experiência mais natural e amigável para o utilizador, como a interacção que o projecto Real Hunter (Chai et al., 2004) oferece. Neste projecto o utilizador aponta, com o dedo, para uma casa de interesse e faz perguntas. O sistema reconhece a fala e responde as perguntas relacionadas a casa indicada.

Alguns exemplos de aplicação que fazem uso de sistema multimodal são: sistema de análise automática em vídeo-conferência (McCowan et al., 2005); casas/escritórios inteligentes (Meyer & Rakotonirainy, 2003); monitorização de condutor (Smith et al., 2003); interface amigável para auxiliar pessoas com deficiência física (Kuno et al., 2003). Embora estas aplicações apresentem características de multimodalidade o foco principal, na maioria destes trabalhos, não está directamente ligado a interface homem-máquina. A Tabela 6 relaciona algumas aplicações mais importantes que utilizam o conceito de interface multimodal.

Tabela 6: Projectos de interface multimodal

Nome	Tipos de entradas	Descrição
QuickSet (Cohen et al., 1997)	Caneta, Reconhecimento da fala	È um aplicativo para treino militar, possibilita a interacção com aplicativos distribuídos: simulação de agente, tarefas baseadas em mapas, etc.
Assistive Multimodal System (Ronzhin & Karpov, 2005)	Reconhecimento da fala e movimento da cabeça	Um sistema para auxiliar pessoas com deficiências físicas para trabalhar com PC, a interacção realiza-se utilizando os movimentos da cabeça e comandos por voz.
MATCH (Johnston et al., 2001)	Caneta, Reconhecimento da fala	Aplicação MMI Móvel para informações de metro e restaurante em Nova York.
Real Hunter (Chai et al., 2004)	Ecrã táctil, Reconhecimento da fala	Utilizador selecciona objectos ou regiões com o toque (num ecrã táctil) enquanto pergunta por comando de voz, para obter informações relacionadas com o objecto ou região.
MATCHKiosk (Johnston & Bangalore, 2004)	Caneta, Reconhecimento da fala	È um quiosque de informações interativo. Ele permite comandos somente por voz, somente com uma caneta ou ainda com as duas entradas ao mesmo tempo.
<i>Multimodal Interface for an Interactive Simulated Vascular Reconstruction System</i> (Zudilova et al., 2002)	Gestos com a mão, Reconhecimento da fala, Manipulação de objectos virtuais em 3D	Simulação de um processo de reconstrução vascular e visualização dos resultados.
<i>Video based interface to textual information for the visually impaired</i> (Zandifar & Chahine, 2002)	Vídeo, Reconhecimento de escrita (OCR), Sintetizador de voz (TTS)	Acesso de informações textuais presentes em fontes não standard, tais como revistas ou jornais, para deficientes visuais
<i>A Multimodal 3D Healthcare Communication System</i> (Keskin et al., 7-9 May 2007)	Reconhecimento de gestos	Sistema de comunicação entre pacientes e médicos através de gestos.
<i>Automatic prediction of frustration</i> (Kapoor et al., 2007)	Expressões faciais, postura	Utiliza as expressões faciais em conjunto com a postura para indicar se utilizador está frustrado.
<i>Bi-modal emotion recognition</i> (Gunes & Piccardi, 2007)	Gestos com o corpo, expressões faciais	Integra as expressões faciais com gestos do corpo para classificar as emoções.
<i>Analysis of emotion recognition</i> (Busso et al., 2004)	Expressões faciais, reconhecimento da fala	Neste trabalho utiliza-se a fusão dos dados de reconhecimento facial e voz para classificar a emoção em neutra, triste, zangado ou feliz
<i>Naval Research Laboratory' interface</i> (Perzanowski et al., 2001)	Reconhecimento da fala, gestos com a mão	Esta interface permite controlar o robô apontando para um determinado sítio mais um comando de voz (“go over there”), também é possível utilizar a tela do PDA para interagir.

## 2.5 Sistemas Multi-Agente

Os Sistemas Multi-Agente pertencem a uma subárea da Inteligência artificial Distribuída, que por sua vez, é uma das áreas da Inteligência Artificial. SMA são sistemas formados por um número variado de agentes. Estas entidades computacionais, denominadas de agentes, possuem habilidades fundamentais como: decidir por si próprio o que devem fazer para

satisfazer seus objectivos (comportamento autónomo); interagir com outros agentes de forma social utilizando protocolos de interacção social inspirados nos humanos. Usualmente, cada agente possui um conjunto de capacidades comportamentais que definem sua competência, um conjunto de objectivos, e a autonomia necessária para utilizar suas capacidades comportamentais a fim de alcançar seus objectivos.

Um fundamento do paradigma de SMA é que um comportamento global inteligente pode ser alcançado a partir do comportamento individual dos agentes. Outra razão para utilizar SMA é que estes podem encontrar soluções de problemas que estão além das capacidades de um único agente.

Os SMA são compostos por diversos agentes que interagem ou trabalham em conjunto. Estes agentes podem ser homogêneos ou heterogêneos. Desta forma, eles podem apresentar diferentes capacidades de percepção e acção no ambiente, sendo assim, cada agente terá uma área de influência diferente sobre o ambiente, em outras palavras, terá capacidade de influenciar distintas partes do ambiente (ver Figura 13) (Jennings, 2000). Estes sistemas devem incluir pelo menos algumas das seguintes funcionalidades: coordenação, cooperação, competição e negociação. Para permitir que os agentes possam operar como parte do sistema, é fundamental a existência de uma infra-estrutura que permita a comunicação e/ou interacção entre os agentes que formam o SMA.

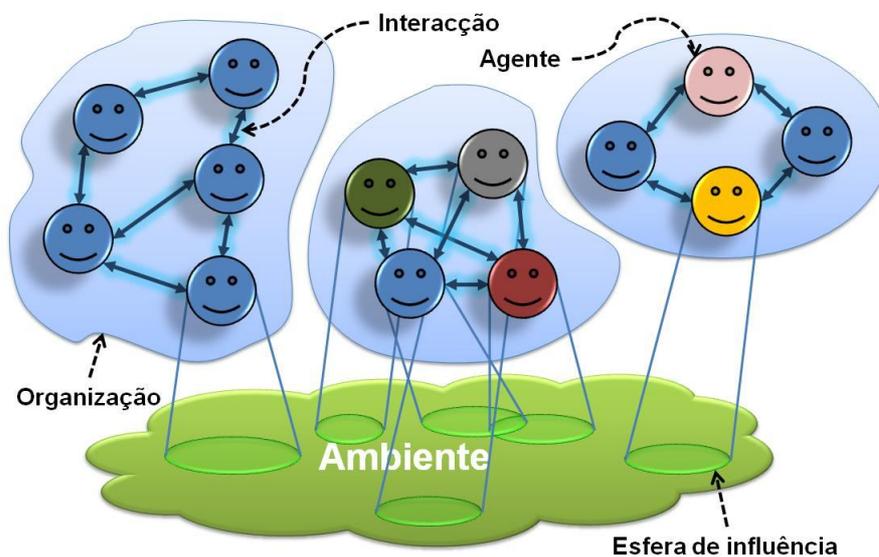


Figura 13: Estrutura de um Sistema Multi-Agente

No contexto de SMA, são utilizados alguns termos específicos, assim são apresentadas na Tabela 7, algumas terminologias mais utilizadas.

Terminologia	Descrição
<b>Papel (role)</b>	Cada agente pertencente ao SMA possui a responsabilidade de desempenhar um ou mais papéis para exercer na operação do sistema.
<b>Comportamento (Behavior)</b>	É o resultado das acções de um ou mais agentes.
<b>Funcionalidade (Functionality)</b>	O que o agente executa quando sofre um estímulo/evento do ambiente.
<b>Actividade (Activity)</b>	Organização funcional do código do agente que possibilita ao agente desempenhar os papéis a ele atribuídos.

Num SMA não é necessário que cada agente seja individualmente inteligente para alcançar um comportamento global inteligente (Alvares & Sichman, 1997) (Wooldridge, 2001)(Reis, 2003). Estes agentes possuem características que os diferenciam das outras aplicações de software. Porém, não é necessário apresentar todas as tais características (propriedades) ao mesmo tempo, mas em princípio, quanto maior o número de propriedades presentes no agente, maior o grau de inteligência este agente apresentará. As principais características que podem estar presentes nos agentes estão descritas na Tabela 8.

Propriedade	Descrição
Autonomia	Possibilidade que um agente tem de actuar e escolher as suas próprias acções, sem a intervenção humana ou de outros agentes, com base no seu conhecimento do mundo. O agente tem que ter a capacidade de tomar iniciativas próprias e possuir controlo sobre suas acções e estado interno (Wooldridge, 2001) (Khosla & Dillon, 1997).
Aprendizagem ou capacidade de adaptação	Os agentes alteram seu comportamento baseado em experiências passadas. Eles devem ser capazes de adaptar-se às novas condições do ambiente, e também aprender com as novas situações e não devem repetir os mesmos “erros” (Khosla & Dillon, 1997)(Franklin & Graesser, 1996).
Interação ou habilidade social,	Esta propriedade diz respeito a capacidade do agente em comunicar-se com outros agentes e com as pessoas, com o intuito de obter informações ou ajudar o utilizador (Khosla & Dillon, 1997)(Franklin & Graesser, 1996).
Colaboração ou cooperação	Capacidade do agente em trocar informações e colaborar com outros agentes para melhorar a qualidade de sua decisão. A habilidade de colaboração entre os agentes é uma extensão natural da capacidade de comunicação (Khosla & Dillon, 1997).
Competitividade	Ao contrário da característica anterior, agentes com estas características não querem colaborar/cooperar, entretanto, eles competem entre si para atender os seus objectivos ou tarefas.
Persistência ou continuidade temporal	É a habilidade do agente em manter-se activo continuamente através do tempo(Franklin & Graesser, 1996).
Reactividade	A literatura apresenta dois significados distintos: - Habilidade do agente em reagir de acordo com as mudanças do ambiente (Wooldridge, 2001)(Franklin & Graesser, 1996). - Capacidade dos agentes seleccionar as suas acções sem ter em conta informações do modelo interno do mundo, sendo sua acção escolhida com base nas informações actuais do ambiente(Russell & Norvig, 2002).
Pro-actividade ou orientado a metas	Característica dum agente em agir de acordo com um propósito, ele não deve agir simplesmente em resposta ao ambiente mas sim a procura de atingir uma meta (Wooldridge, 2001)(Franklin & Graesser, 1996).
Mobilidade	Capacidade para locomover-se de uma máquina (ambiente) para outra(Franklin & Graesser, 1996). Desta forma, os agentes podem ser: Móveis ou Fixo.

	(estacionários/situados).
Uniformidade	Refere-se a semelhança entre os agentes: homogéneos ou heterogéneos. Os agentes homogéneos utilizam a mesma arquitectura, desta forma esta é replicada para cada agente, enquanto os agentes heterogéneos utilizam arquitecturas distintas.
Modelação	Capacidade do agente em modelar o ambiente, seus estados ou dos outros agentes.
Raciocínio	Capacidade de analisar e actuar baseando-se no seu conhecimento actual e nas suas experiências. Formas de raciocínio dos agentes: <ul style="list-style-type: none"> <li>➤ Com base em regras – usa-se um conjunto de condições prévias para avaliar as condições do mundo.</li> <li>➤ Com base em conhecimento – possui grandes conjuntos de dados sobre cenários anteriores e acções resultantes, dos quais eles deduzem suas acções futuras.</li> </ul>

Nwana (Nwana, 1996), combinando as características de autonomia, cooperação e aprendizagem, define quatro tipos de agentes: agentes colaborativos, agentes colaborativos com capacidade de aprendizagem, agentes de interface e agentes verdadeiramente inteligentes (Figura 14). Deve-se ter em conta que as fronteiras desta classificação não devem ser consideradas como fronteiras rígidas bem definidas.

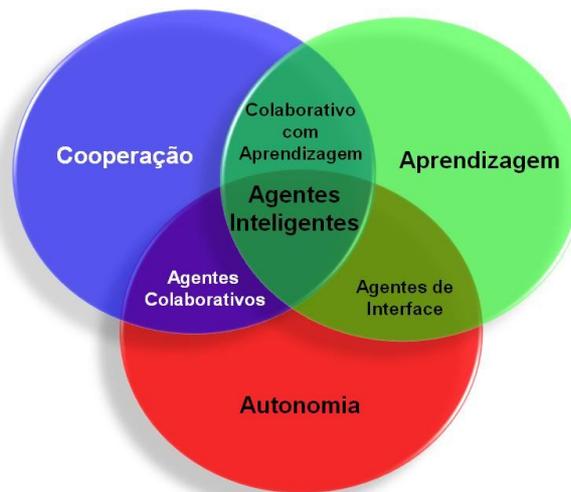


Figura 14: Topologia de agentes proposta por Nwana.

De forma geral, pode-se classificar os agentes em dois grupos: reactivos e deliberativos. A diferença está no facto que os agentes reactivos consideram as informações correntes do ambiente na tomada de decisões, este tipo de agente não possui "memória" para armazenar experiências, procedem de forma "instintiva". Um agente reactivo simples utiliza um conjunto de regras "situação-acção", onde a decisão é tomada em relação a percepção actual (Russell & Norvig, 2002), neste caso não considera e nem armazena as informações passadas. A sua arquitectura é representada pelo diagrama da Figura 15 e seu pseudo-código descrito na Tabela 9.

Tabela 9: Pseudo-código agente reactivo simples

```

função agente_reactivo_simples(percepção) retorna acção
estática: regras //conjunto de regras si tuação-acção
estado ← interpreta_input(percepção)
regra ← selecciona(estado, regras)
acção ← conclusão_regra(regra)
retorna acção
    
```



Figura 15: Diagrama esquemático de um agente reactivo

Enquanto os agentes deliberativos (também conhecidos por agentes cognitivos) podem aprender com as suas experiências e usam este conhecimento adquirido para auxiliar nas tomadas de decisões. A levar em consideração o conhecimento adquirido podemos considerar quatro tipos de arquitectura de agentes (Russell & Norvig, 2002):

- **Agentes que memorizam o mundo:** A sua decisão leva em consideração o conhecimento prévio do mundo (Figura 16).

Tabela 10: Pseudo-código agente Memorizando o mundo

```

função agente_reactivo_com_memoria(percepção) retorna acção
estática: estado //descrição do estado corrente do mundo
          regras //conjunto de regras si tuação-acção
estado ← actualiza_estado(estado, percepção)
regra ← selecciona_regra(estado, regras)
acção ← conclusão_regra(regra)
estado ← actualiza_estado(estado, acção)
retorna acção //mesma percepção pode gerar acções diferentes dependendo do estado do mundo
    
```



Figura 16: Diagrama esquemático de um agente reactivo com estado interno

- **Agentes geridos por objectivos:** Estes agentes utilizam a descrição do estado corrente e a informação sobre os objectivos, isso implica em pesquisar e planear. É mais flexível pois diferentes comportamentos são obtidos para o mesmo estado do mundo (Figura 17).



Figura 17: Diagrama esquemático de um agente gerido por objectivos

- **Agentes baseados em utilidade:** Para o agente as utilidades são medidas de “satisfação” relativa aos estados. As utilidades podem ser utilizadas para decidir entre objectivos em conflito ou ainda para medir a probabilidade de atingir o objectivo nos casos que existem incerteza nas acções. Agentes que utilizam função de utilidade são mais racionais (Figura 18).



Figura 18: Diagrama esquemático de um agente baseado em utilidades

- **Agentes adaptativos:** São agentes que alteram (melhoram) a sua capacidade de agir ao longo do tempo de acordo com as suas experiências. Este tipo de agente pode ser dividido em quatro componentes conceituais (Figura 19):
  - **Executor:** É equivalente a um agente com uma das arquitecturas descritas anteriormente. É responsável por seleccionar a acção.
  - **Crítico:** Mede o desempenho do agente considerando os standards fornecidos para a avaliação.

- Aprendizagem: Propõe modificações na estratégia de selecção das acções. Utiliza o feedback do Crítico e determina a maneira que as acções do Executor podem ser alteradas para melhorar o desempenho do sistema.
- Gerador de Estados: Sugere novas acções para explorar novos estados que podem vir a ser benéficos a longo prazo.

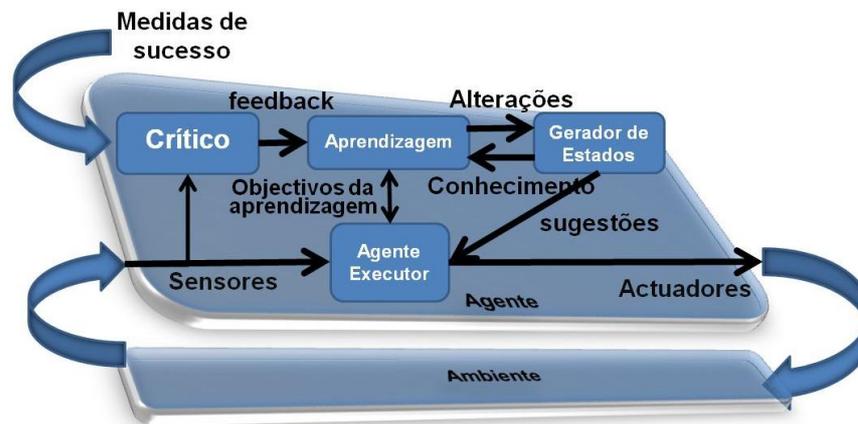


Figura 19: Diagrama esquemático de um agente adaptativo

A motivação para utilizar SMA está relacionada principalmente pelo facto de grande parte dos problemas vulgarmente encontrados, serem inerentemente distribuídos. Pode-se citar outras motivações relacionadas:

- Problemas reais são, muitas vezes, grandes e complexos para serem resolvidos por um único agente monolítico;
- Possibilitar a interoperação e interconexão de múltiplos sistemas legados (“*legacy*”), ou seja, sistemas de geração anteriores cuja manutenção do código já não é possível.
- Agentes individuais são limitados por seu conhecimento, recursos computacionais e suas perspectivas, os SMA pode fornecer soluções para problemas em que os peritos, os conhecimentos ou as informações necessárias, se encontram distribuídos;
- Promover uma solução natural para problemas distribuídos (geográfica e/ou funcionalmente);
- Sistemas multi-agente são naturalmente modulares, oferecem maior clareza e simplicidade conceptual do projecto;
- Facilitar a criação duma interface cooperativa homem-máquina mais natural, em que ambos funcionam como agentes no sistema.

Em problemas onde o conhecimento ou as tarefas são distribuídas, os SMA podem oferecer uma eficiência maior na utilização dos recursos. Sendo assim, várias razões adicionais para o uso de um SMA podem ser citadas(Stone & Veloso, 1996):

- Exigência do domínio do problema – Devido à distribuição espacial dos intervenientes;
- Eficiência computacional – Com o uso do paralelismo, é possível atribuir diferentes tarefas a diferentes agentes para que a execução seja mais rápida;
- Robustez – Possibilidade de utilizar diferentes agentes para a mesma tarefa, não existindo desta forma um ponto único de falha no sistema;
- Escalabilidade – Permite o aumento dos agentes intervenientes num determinado sistema aberto;
- Simplificação das tarefas individuais de programação – Nos SMA o problema global pode ser dividido em vários subproblemas;
- Interoperacionalidade entre os agentes dum SMA – Torna possível estudar a inteligência individual e o comportamento social dos agentes;
- Manutenção da privacidade da informação e conhecimentos individuais de cada agente.

Diversos benefícios (Tabela 11) podem ser obtidos pelo uso de SMA na resolução de problemas de IA, sobretudo uma maior eficiência na utilização de recursos para problemas onde o conhecimento ou actividade são distribuídos.

<b>Benefício</b>	<b>Motivo</b>
Resolução mais rápida de problemas	Devido ao processamento concorrente
Redução na comunicação	Devido ao processamento estar localizado junto à fonte de informação e a comunicação ser realizada a alto-nível
Aumento da flexibilidade e escalabilidade do sistema	Devido a possibilidade de interconexão de múltiplos sistemas com arquitecturas distintas
Aumento da fiabilidade	Devido à inexistência de um ponto singular de falha
Aumento da capacidade de resposta	Sensores, sistemas de processamento e actuadores estarem localizados em conjunto, no interior dos agentes
Facilidade ampliada no desenvolvimento de sistemas	Modularidade resultante da decomposição dos problemas e da decomposição dos sistemas em agentes semi-autónomos

Projectar e desenvolver SMA usando apenas linguagens de programação e técnicas tradicionais de engenharia de software não é uma tarefa fácil. Assim, para facilitar o seu desenvolvimento, algumas ferramentas foram desenvolvidas, tais como: plataformas de SMA, metodologias, ferramentas computacionais.

### 2.5.1 Plataforma de Comunicação

A comunicação é uma propriedade fundamental dos agentes para permitir que possam colaborar, cooperar e negociar. Em SMA, é inevitável que a comunicação siga regras para que os objectivos sejam alcançados, de forma efectiva e eficiente. Deste modo torna-se necessário o uso de uma linguagem compreensível por todos agentes presentes no ambiente. Desta maneira, a comunicação aqui é tratada a um nível mais elevado do que nas outras áreas da engenharia informática. O foco está na comunicação de alto-nível, utilizando linguagens de comunicação próximas da linguagem utilizadas na comunicação humana.

Na área dos SMA são apresentadas diversas maneiras dos agentes trocarem informações (Huhns & Stephens, 1999). Alguns autores chamam de *comunicação directa* o acto dos agentes trocarem mensagens de forma directa (Figura 20-a). Outra forma de comunicar é através de um agente “facilitador” que utiliza algum sistema ou agente especial para coordenar as actividades de comunicação (Figura 20-b). Os agentes também podem utilizar uma comunicação por *difusão de mensagens (broadcast)* (Figura 20-c), e até utilizar o modelo de comunicação através de memória partilhada ou quadro-negro (*blackboard*) (Figura 20-d).

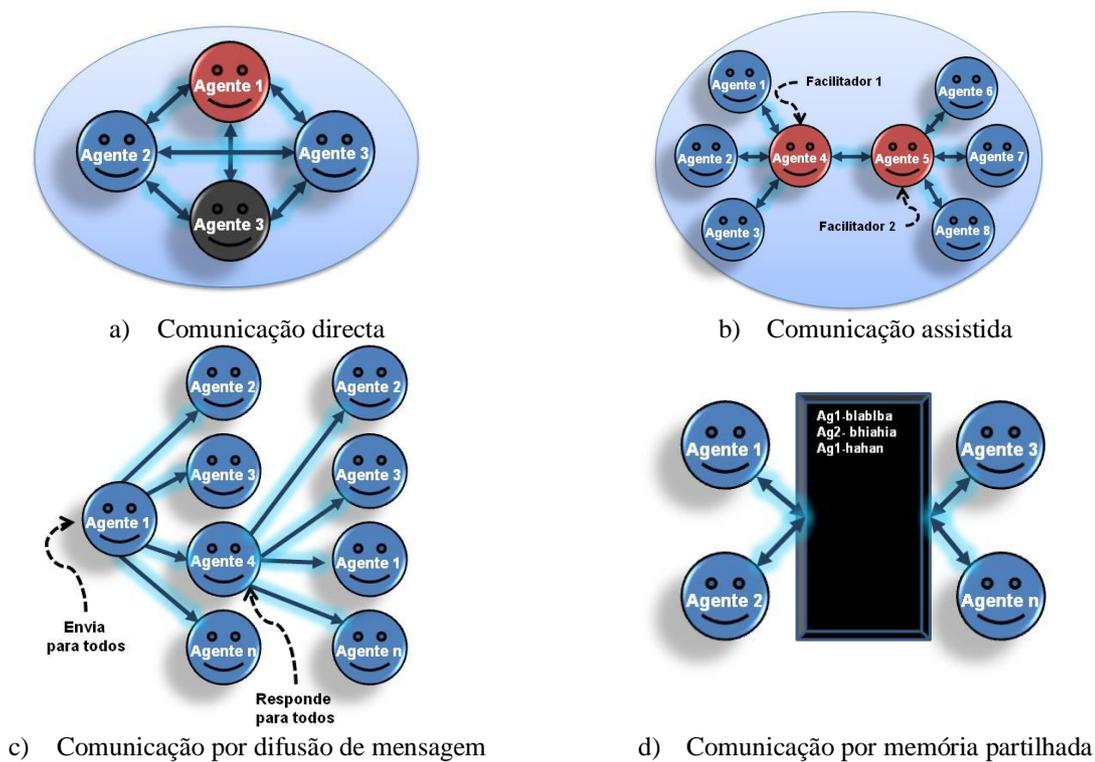


Figura 20: Formas de comunicação entre agentes

A capacidade de comunicação de cada agente é definida pelas linguagens de comunicação e a sua expressividade. A linguagem de comunicação deve ser comum e partilhada por todos os agentes, ser precisa e ter um número limitado de primitivas de comunicação. A

comunicação humana e a teoria dos actos comunicativos são utilizadas como modelo para a comunicação de agentes, sendo usado o conceito de performativa. Existem várias linguagens definidas no âmbito da comunicação em SMA: KQML (*Knowledge Query and Manipulation Language*), KIF (*Knowledge Interchange Format*), ACL (*Agent Communication Language*), entre outras. Dentre as mais utilizadas destaca-se o FIPA-ACL, sendo assim será detalhada na subsecção seguinte.

KQML é uma linguagem e um protocolo de comunicação de alto nível para troca de mensagens independente de conteúdo e da ontologia aplicável. A KIF é uma linguagem que se destina explicitamente a representar o conhecimento sobre um domínio de discurso específico. Foi desenvolvido primariamente como forma de definir o conteúdo de mensagens expressas em KQML. Foram desenvolvidas pelo KSE (*Knowledge Sharing Effort*), financiado pelo DARPA<sup>7</sup> no início dos anos 90, para servir ao mesmo tempo como um formato de mensagem e um protocolo de para gerir mensagens.

No desenvolvimento de SMA, para além de uma linguagem de comunicação torna-se necessário o uso duma ferramenta de infra-estrutura de agentes (plataforma) para reduzir a complexidade no desenvolvimento dum sistema de agentes. Algumas características comuns entre as plataformas de SMA encontram-se apresentadas na Tabela 12 (Weiss, 1999):

Tabela 12: Características encontradas nas plataformas de SMA

Propriedade	Valores
<i>Tipos de Agentes</i>	<i>São compostos por agentes autónomos e distribuídos, e podem apresentar comportamentos de colaboração ou competição.</i>
<i>Projecto do SMA</i>	<i>Normalmente são abertos, ou seja, desenvolvidos por mais de um projectista, não são centralizados</i>
<i>Ambiente</i>	<i>Fornecem uma infra-estrutura e especifica protocolos de comunicação e interacção</i>
<i>Autonomia de Projecto</i>	<i>Plataforma/Protocolo de Interação/Linguagem/Arquitectura Interna</i>
<i>Infra-estrutura de Comunicação</i>	<i>Memória Compartilhada (blackboard) ou Baseada em Mensagens, Orientada à Conexão ou Não Orientada à Conexão(email), Ponto-a-Ponto Multicast ou Broadcast, Síncrono ou Assíncrono</i>
<i>Directório de Serviço (Directory Service)</i>	<i>Páginas brancas (White pages), Páginas amarelas (Yellow pages)</i>
<i>Protocolo de Mensagens</i>	<i>KQML, FIPA-ACL, XML, HTTP ou HTML, OLE, CORBA</i>
<i>Serviço de Mediação</i>	<i>Baseado em Ontologia ou Transacções</i>
<i>Serviço de Segurança</i>	<i>Timestamps/Autenticação</i>
<i>Operações de Suporte</i>	<i>Armazenamento/Redundância/Restauração/Accounting</i>

Existem muitas plataformas de SMA disponíveis. Algumas delas podem ser utilizadas gratuitamente (“*open-source*”), outras são comerciais e variam em preço e capacidades (Tabela 13).

<sup>7</sup> DARPA – Defense Advanced Research Projects Agency, EUA

Tabela 13: Lista de plataformas de sistemas multi-agente

Plataforma	Licença	Criadores/Proprietário	Descrição
<b>ABLE</b>	Licença IBM	IBM	Plataforma e biblioteca, em Java, para a construção de agentes inteligentes usando aprendizagem automática ( <i>machine learning</i> ) e raciocínio ( <i>reasoning</i> )
<b>Agent Builder</b>	Comercial	Acronymics, Inc.	SMA de propósito geral, linguagens Java, C, C++
<b>Aglets</b>	Código aberto	IBM	Desenvolvimento de agentes móveis
<b>Bee-gent</b>	Comercial, Gratuito para teste	Toshiba Corporation	Plataforma de desenvolvimento de agentes. Bee-gent permite aos desenvolvedores construir sistemas flexíveis distribuídos.
<b>Cougaar</b>	COSL modificação da licença BSD	Raytheon BBN Technologies	Plataforma para SMA altamente distribuído, escalável e confiável, independente do domínio, agentes distribuídos em grande escala, complexos, grande quantidade de dados (pode ser configurado para aplicações de pequena escala e aplicações embarcadas), linguagem de programação em Java.
<b>CybelePro</b>	Comercial	Intelligent Automation, Inc	CybelePro é a versão paga do OpenCybele e oferece recursos avançados para o desenvolvimento rápido em grande escala, alta performance de sistemas baseados em agentes
<b>Jack</b>	Comercial	Agent Oriented Software Pty. Ltd. (AOS)	Plataforma em Java para o desenvolvimento de SMA, utiliza modelo BDI. Linguagem e ferramenta gráfica para planeamento.
<b>Jackal</b>	Código aberto	Consortium for Intelligent Integrated Manufacturing Planning-Execution (CIIM-PLEX)	Ferramenta para desenvolvimento de agentes baseada em Java que utiliza a linguagem de comunicação KQML. Algumas características são gestor facilitador de comunicação, interface estilo quadro-negro e facilidade de integração.
<b>JADE</b>	LGPL	Telecom Italia Lab (TILAB)	Plataforma para desenvolver SMA em conformidade com o padrão FIPA, linguagem de programação em Java.
<b>JADEX (JADE Extension)</b>	Código aberto	<i>Pokahr, Alexander e Braubach, Lars</i> – University of Hamburg	Fornecer ao JADE capacidade de modelação de agentes inteligentes, baseado nas noções de agentes com estados mentais e orientação a objectivos. Crenças, objectivos e planos podem ser criados e manipulados como classes JAVA
<b>JASON</b>	Código aberto	Bordini, Rafael H. – University of Durham e Hübner, Jomi F. – Universidade Regional de Blumenau	Plataforma que implementa <i>AgentSpeak(L)</i> , uma linguagem abstracta baseada na arquitectura BDI, linguagem de programação em Java.
<b>JATLite</b>	Código aberto	Universidade de Stanford	Conjunto de classes escrito em linguagem Java que fornece uma arquitectura básica para a construção de agentes, utiliza o protocolo de rede TCP/IP e o protocolo de comunicação KQML.
<b>JATLiteBean</b>	GPL	Universidade de Otago, desenvolvido por Bryce McKinlay e Emanuela Moreale	É um componente encapsulado JavaBean que estende as funcionalidades do JATLite
<b>JINI</b>	Licença Apache, V2	Sun Microsystems, Inc.	Plataforma para sistemas distribuídos, que assegura a fiabilidade e disponibiliza serviços complementares de interacção e comunicação, linguagem de programação em Java.
<b>OpenCybele</b>	GPL	Intelligent Automation, Inc	Infra-estrutura de agentes que tem como características capacidade <i>plug-and-play</i> de agentes de serviço, programação centrada em

			actividades, suporte multi-thread, linguagem de programação em Java.
<b>Zeus</b>	Código aberto	British Telecommunications plc (BT)	Ferramenta para o projecto e desenvolvimento de agentes distribuídos. Possui um ambiente gráfico, motor de inferência, planeador, <i>Web Services</i> , linguagem de programação em Java. As suas três componentes principais são: <i>The Agent Component Library/The Agent Building Tools / The Visualisation Tools</i>

### 2.5.1.1 Linguagem de Comunicação FIPA-ACL

A FIPA<sup>8</sup> define a linguagem FIPA-ACL para a comunicação entre os agentes, baseada em acções de fala, especificando a descrição da mensagem, seu modelo semântico e os protocolos de interacção. Define essencialmente a estrutura exterior da mensagem. Para se expressar o conteúdo de uma mensagem ACL pode-se utilizar qualquer linguagem, como por exemplo Prolog, objecto serializado ou XML. A FIPA sugere o uso da linguagem FIPA SL, que define o conteúdo em expressões do tipo proposição, acção ou expressão identificadora de referência.

A Tabela 14 apresenta a estrutura duma mensagem básica em FIPA-ACL, com as partes que a compõem.

Tabela 14: Formato duma mensagem FIPA-ACL
(communi cati ve act : sender <val or> : recei ver <val or> : content <val or> : l anguage <val or> : ontol ogy <val or> : conversati on-i d <val or> ... )

A mensagem FIPA-ACL possui vários parâmetros, tais como descritos na Tabela 15.

<sup>8</sup> FIPA (*Foundation for Intelligent Physical Agents*) é uma fundação, sem fins lucrativos, focada à produção de standards para a interoperabilidade de agentes heterogéneos e interactivos e sistemas com base em agentes. Sua missão oficial é: “A promoção de tecnologia e especificações de interoperabilidade que facilitem a comunicação entre sistemas de agentes inteligentes no contexto comercial e industrial.” (FIPA, 2009).

Tabela 15: Parâmetros contidos nas mensagens FIPA ACL

Parâmetro	Descrição
<b>Performativa (<i>Performative</i>)</b>	Conhecida como actos de comunicação ( <i>Communicative Acts</i> ), define uma acção que descreve a mensagem. É o único parâmetro obrigatório.
<b>Remetente (<i>Sender</i>)</b>	Remetente da mensagem.
<b>Destinatário (<i>Receiver</i>)</b>	Destinatário da mensagem.
<b>Responder Para (<i>Reply-To</i>)</b>	Destinatário de mensagem de resposta.
<b>Conteúdo (<i>Content</i>)</b>	Conteúdo da mensagem.
<b>Linguagem (<i>language</i>)</b>	Linguagem utilizada para expressar o conteúdo.
<b>Codificação (<i>encoding</i>)</b>	Codificação utilizada para o conteúdo.
<b>Ontologia (<i>Ontology</i>)</b>	Descrição do conteúdo.
<b>Protocolo (<i>Protocol</i>)</b>	Controle de conversação (protocolo de interacção).

A FIPA-ACL tem uma sintaxe semelhante ao formato da linguagem KQML. No entanto, o número de performativas ACL é bastante inferior ao número de performativas KQML. A linguagem ACL define unicamente 20 performativas para definir a interpretação desejada para cada tipo de mensagem. Os actos de comunicação podem descrever a acção das mensagens ACL (ver Tabela 15).

Tabela 16: Performativas providenciadas pela linguagem FIPA ACL.

Performativa	Significado
<b>Aceitação de Proposta (<i>accept-proposal</i>)</b>	Aceitação de proposta numa negociação
<b>Aceitação (<i>Agree</i>)</b>	Aceitação de desempenhar uma dada acção
<b>Cancelamento (<i>Cancel</i>)</b>	Cancelamento da execução de uma dada acção
<b>CFP (<i>Cfp</i>)</b>	<i>Call for proposals</i> . Utilizada para iniciar uma dada negociação
<b>Confirmação (<i>Confirm</i>)</b>	Confirmação da veracidade de uma dada mensagem
<b>Desconfirmação (<i>Disconfirm</i>)</b>	Inverso da mensagem anterior
<b>Falha (<i>Failure</i>)</b>	Uma tentativa de executar uma dada acção (usualmente requisitada por outro agente) que falhou
<b>Informação (<i>Inform</i>)</b>	Uma das performativas mais importantes da FIPA-ACL. Permite comunicar informação aos outros agentes
<b>Informação de veracidade (<i>Inform-if</i>)</b>	Informação sobre a veracidade de determinada informação
<b>Informa-Ref (<i>Inform-ref</i>)</b>	Informação sobre um dado valor
<b>Não-Percebido (<i>not-understood</i>)</b>	Indicação de que uma dada mensagem não foi percebida
<b>Propagação (<i>Propagate</i>)</b>	Pedido de propagação a um conjunto de agentes de uma dada mensagem
<b>Propor (<i>Propose</i>)</b>	Envio de proposta, por exemplo, como resposta a uma mensagem <i>cfp</i>
<b>Proxy (<i>Proxy</i>)</b>	Permite enviar uma mensagem que vai ser reenviada a um conjunto de agentes
<b>Solicitação de veracidade (<i>Query-if</i>)</b>	Pedido de informação sobre a veracidade de determinada informação
<b>Solicitação de valor (<i>Query-ref</i>)</b>	Pedido de informação sobre um dado valor
<b>Rejeição (<i>Refuse</i>)</b>	Recusa de executar determinada acção
<b>Rejeição de proposta (<i>Reject-proposal</i>)</b>	Recusa de uma proposta efectuada no contexto de uma dada negociação
<b>Pedido (<i>Request</i>)</b>	Uma das performativas mais importantes da FIPA-ACL. Consiste num pedido a um dado agente para executar determinada acção
<b>Pedido quando (<i>Request-when</i>)</b>	Pedido para executar uma dada acção quando uma determinada condição for verdadeira.
<b>Pedido sempre (<i>Request-whenever</i>)</b>	Pedido para executar uma dada acção sempre que uma determinada condição seja verdadeira.
<b>Notificação (<i>Subscribe</i>)</b>	Pedido para ser informado acerca das alterações relacionadas com determinado facto ou informação

A diferença fundamental entre a FIPA-ACL e o KQML está relacionada com as performativas providenciadas por cada linguagem. A linguagem FIPA-ACL procura promover uma semântica mais inteligível do que a linguagem KQML. Uma das vantagens mais significativas da FIPA-ACL, relativamente ao KQML, consiste na disponibilização de performativas mais adequadas à execução de processos de negociação.

O JADE utiliza a linguagem FIPA-ACL e protocolos de interacção definidos pela FIPA. Estes protocolos são usados para definir os papéis que cada agente deverá desempenhar, a sequência das mensagens trocadas e o tipo de mensagem que deve ser enviada e recebida em cada papel. O JADE é uma das plataformas de SMA mais divulgadas no meio científico que possui muitos atributos e características que facilitam muito a implementação de agentes em SMA e segue as especificações FIPA. Desta forma a plataforma JADE será apresentada com maiores detalhes, na próxima secção.

### 2.5.1.2 Plataforma JADE

JADE (*Java Agent DEvelopment framework*) é uma plataforma de desenvolvimento de aplicações ponto a ponto (“*peer-to-peer*”) baseadas no paradigma de agentes. Utiliza a especificação FIPA<sup>9</sup> para interoperabilidade entre SMA implementado em Java, que tem sido desenvolvida desde 2001.

Esta plataforma conceptualiza um agente como um processo independente e autónomo com uma identidade, persistente, que demanda de comunicação com os demais agentes do sistema para atingir seus objectivos. Apenas, então, define capacidades relacionadas à interacção e a comunicação entre agentes. No que diz respeito ao desenvolvimento de agentes inteligentes, mostra-se uma plataforma neutra e sugere a integração com alguma plataforma que proporcione aos agentes capacidade de “raciocínio”, como por exemplo o JESS<sup>10</sup>.

O principal objectivo JADE é a interoperabilidade entre SMA através de um amplo conjunto de serviços de sistema, os quais tanto facilitam como possibilitam a comunicação entre agentes, em conformidade com as especificações da FIPA. Pode-se citar: serviço de nomes (*naming service*) e páginas amarelas (*yellow-page service*), gestor de agentes (AMS – *Agent Management System*), directório facilitador (DF – *Directory Facilitator*), canal de comunicação de agentes (ACC – *Agent Communication Channel*), transporte de mensagens, serviços de codificação e decodificação de mensagens, linguagem FIPA-ACL e uma biblioteca de protocolos de interacção (padrão FIPA - request, FIPA-Contract-Net e FIPA-

---

<sup>9</sup> Uma descrição mais detalhada sobre a especificação FIPA pode ser encontrada em <http://www.fipa.org>.

<sup>10</sup> JESS é uma ferramenta orientada a regras, desenvolvida em Java. Ela oferece uma programação baseada em regras adequadas para automação de sistemas especialistas e no desenvolvimento de agentes inteligentes. JESS foi desenvolvida por Ernest Friedman-Hill do Sandia National Labs e começou a ser divulgada no final de 1995. Ver em <http://www.jessrules.com>.

Subscribe). Toda a comunicação entre agentes é feita via troca de mensagens, cuja estrutura de mensagens baseada em acções de fala (*speech act*) e serviço de ontologia.

JADE fornece classes intuitivas que devem ser herdadas ou instanciadas para a criação do agente, acesso e envio de mensagens, criação dos comportamentos, também fornece esqueletos para padrões de interacção (negociações, leilão e delegação de tarefas), dentre outros. Ela ocupa-se de todos os aspectos que não fazem parte do agente, propriamente dito, e são independentes das aplicações tais como transporte de mensagens, codificação e interpretação de mensagens, e ciclo de vida dos agentes.

Muitas informações sobre o funcionamento da ferramenta podem ser encontradas numa lista de discussão e numa série de documentos disponibilizados no sítio Web<sup>11</sup> da plataforma. Porém é difícil encontrar documentação referente ao desenvolvimento de ambientes.

Seguem abaixo algumas características da plataforma JADE na programação de SMA:

- **Plataforma distribuída de agentes:** JADE pode ser dividida em várias máquinas (*hosts*) – neste caso, é necessária uma conexão via RMI (*Remote Method Invocation*). Apenas uma aplicação Java e uma máquina virtual Java é executada em cada máquina. Os agentes são implementados como *threads* Java e inseridos dentro de repositórios de agentes chamados de *containers* (*Agent Containers*) que fornecem todo o suporte para a execução do agente. Cada *containers* é um objecto RMI que gere um conjunto de agentes localmente, controla o ciclo de vida dos agentes (criação, suspensão, destruição);
- **Escalonável:** permite desenvolver SMA tanto em pequena como em média escala;
- Oferece uma GUI (*Graphical User Interface*), interface visual para gerir vários agentes e *containers* de agentes ao mesmo tempo;
- **Permite o desenvolvimento de aplicações multi-domínio:** a interface simplifica o registo de agentes num ou mais domínios;
- **Ferramentas de *Debugging*:** auxiliam no desenvolvimento e depuração de aplicações multi-agente;
- **Possui um mecanismo interno de transporte de mensagens no formato FIPA-ACL;**
- **Fornecer suporte a execução de múltiplas, paralelas e concorrentes actividades de agentes através dos modelos de comportamentos (*Behaviours*);**

O conjunto de contentores (*containers*) activos é vinculado numa plataforma. Em especial, o contentor principal (*main container*) precisa estar sempre activo numa plataforma e todos os *containers* devem ser registados a este. Além disto, o *main container* é responsável pelo sistema de gestão de agentes e o serviço de páginas amarelas, através dos agentes AMS (*Agent Manager System*) e DF (*Directory Facility*), respectivamente. O registo RMI também encontra-se no contentor principal, sendo que ele nada mais é do que um servidor de nomes

---

<sup>11</sup> Uma documentação mais completa sobre JADE pode ser encontrada em <http://jade.tilab.com>.

que o Java utiliza para registar e recuperar referências a objectos através do nome. Desta forma, é o meio que JADE usa em Java para manter as referências aos outros *containers* de agentes que se conectam a plataforma. A Figura 21 ilustra a arquitectura JADE com a sua estrutura de *containers*.

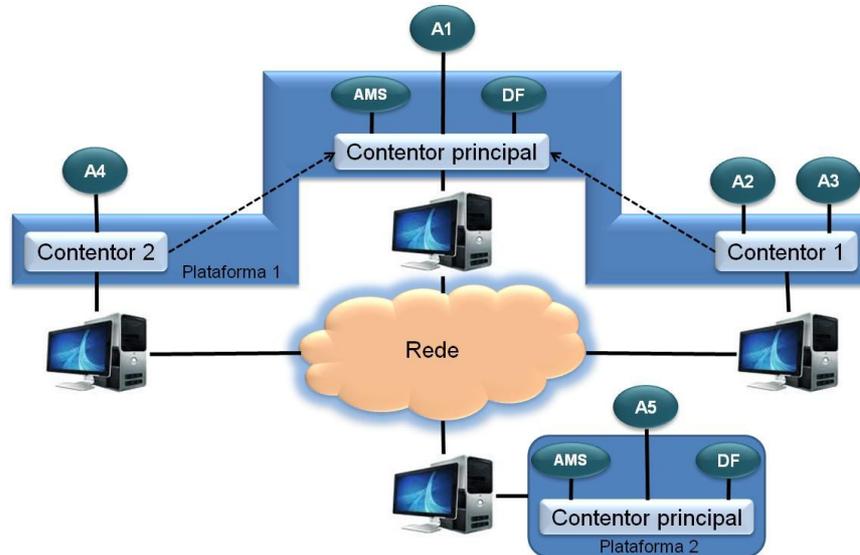


Figura 21: Arquitectura JADE

O AMS é o agente que coordena o acesso e o uso da plataforma de agentes. Apenas um AMS irá existir numa plataforma. Este agente gere o guia de endereços (*whitepages*) e controlo do ciclo-de-vida. Mantém o directório de identificadores de agentes (*Agent Identifier* - AID) e estados dos agentes. É também responsável pela autenticação de agentes e pelo controlo do registo. Cada agente tem que se registar no AMS para obter um AID válido.

O DF é o agente que gere o serviço de páginas amarelas (*yellow-pages*) dentro da plataforma. O sistema de transporte de mensagens (*Message Transport System*), também conhecido como canal de comunicação dos agentes (ACC). Ele é o agente responsável por gerir toda a comunicação entre agentes dentro e fora da plataforma. Todos os agentes, inclusive o AMS e o DF, utilizam esse canal para a comunicação.

JADE fornece um conjunto de ferramentas de suporte para gerir os SMA, as quais simplificam a gestão da plataforma. Cada ferramenta é representada por um agente:

- **Agente de Supervisão Remota:** (RMA – *Remote Management Agent*) é uma interface gráfica para controlo e gestão da plataforma. JADE mantém coerência entre múltiplos RMAs. O console RMA permite iniciar outras ferramentas JADE;
- **Agente Dummy:** é uma ferramenta gráfica de monitorização e depuração (*debugging*). Usando a interface gráfica é possível gerar mensagens ACL e enviar a outros agentes, mostrar uma lista de mensagens ACL enviadas e recebidas;

- **Agente Sniffer:** pode interceptar mensagens ACL enquanto elas estão a ser transmitidas, e mostra-as numa interface gráfica (notação similar ao diagrama de sequências UML). É útil para testar e estudar a interacção na sociedade de agentes;
- **Agente Introspector:** permite monitorar o ciclo de vida de outros agentes, tal como a troca de mensagens e comportamentos em execução;
- **Agente LogManager:** permite registar informação de *log* em execução;
- **Agente SocketProxy:** actua em modo bidireccional entre a plataforma JADE e a conexão TCP/IP.

Conclui-se que uma ferramenta adequada para auxiliar no desenvolvimento de SMA é essencial. O JADE oferece muitos atributos e características que facilitam este trabalho. Além de ser uma plataforma robusta e concisa, abstrai, para o desenvolvedor de agentes, a necessidade de preocupar-se com a implementação dos serviços básicos de um SMA, como por exemplo a comunicação, ferramentas de gestão, depuração e muitos outros atributos que um SMA necessita. É importante realçar a grande preocupação que o JADE tem tido em sempre manter os padrões especificados pela FIPA, resultando num alto grau de interoperabilidade e escalabilidade do ambiente.

## 2.5.2 Metodologias de Modelação de SMA

O uso crescente de aplicativos e tecnologia baseada em SMA criou a necessidade de desenvolver metodologias que possam auxiliar os projectistas não só no desenvolvimento e implementação, mas também na análise preliminar e as fases de concepção de um projecto. Este processo é cada vez mais semelhante ao que ocorre no desenvolvimento de projectos de software mais convencionais, onde são utilizadas técnicas tradicionais de engenharia de software, como por exemplo o uso do paradigma de orientação a objecto (Iglesias et al., 1998). Por outras palavras, somente uma boa plataforma (infra-estrutura) não é suficiente para desenvolver um SMA eficiente, também é preciso uma boa metodologia que permita organizar as informações de modo a usar certos princípios de engenharia.

Várias metodologias de Engenharia de software orientada a agentes (*Agent Oriented Software Engineering* - AOSE) foram propostas ao longo dos anos para auxiliar no desenvolvimento e modelação de sistemas multi-agente, alguns decorrentes das metodologias tradicionais existentes na engenharia de software (abordagens normalmente orientada a objecto), outros com uma origem mais inovadora. Estas metodologias são diferentes em termos do suporte à fase de desenvolvimento, desde o levantamento de requisitos até à implementação.

Nas subsecções seguintes, serão apresentados, de forma resumida, algumas das metodologias AOSE mais conhecidas, surgidas nos últimos anos (a referência (Bergenti et al., 2004) ou o capítulo 7 de (Sterling & Taveter, 2009) fornecem maiores informações para uma análise mais completa e detalhada das metodologias existentes).

### 2.5.2.1 MaSE

Em 2000 foi introduzida a metodologia para Engenharia de Sistemas Multi-agentes, denominada MaSE (*Multiagent Systems Engineering*). Esta metodologia é mais abrangente, uma vez que inclui as fases de análise e de projecto (Wood & DeLoach, 2000), como ilustrada na Figura 22. MaSE usa modelos gráficos para descrever os objectivos, comportamentos, tipos de agente e interfaces de comunicação dos agentes. Fornece também a definição detalhada do projecto da estrutura interna do agente (DeLoach et al., 2001).

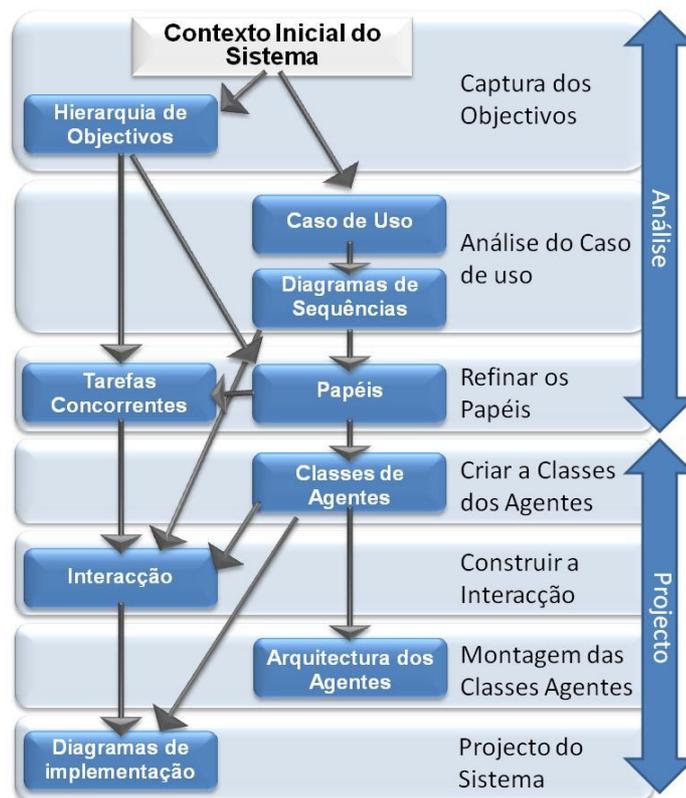


Figura 22: Fases da metodologia MaSE

Na primeira fase de análise, a partir de uma especificação inicial do sistema os objectivos são determinados e é criada uma estrutura hierárquica com o conjunto de objectivos obtidos. A segunda fase de análise está centralizada em torno dos casos de uso, ou seja a partir das especificações do sistema deve-se detectar os papéis, casos de uso e cenários dos casos de uso. A terceira fase de análise, tem por finalidade refinar os papéis identificados e produzir uma descrição mais detalhada de cada papel com seus respectivos objectivos e as interacções com outros papéis.

Na fase de projecto, os papéis são mapeados e especificadas as classes dos agentes; os protocolos de comunicação entre as classes de agente são detalhados; são definidos os detalhes internos de cada classe de agente; e finalmente um diagrama de implementação do sistema é criado. Uma ferramenta chamada *agentTool* foi desenvolvida para dar suporte a metodologia de MaSE (e mais recentemente, o MaSE com baseada em organização ou O-

MaSE). O sistema é modelado graficamente a partir da especificação inicial do sistema até a sua implementação por um conjunto de modelos gráficos inter-relacionados (DeLoach & Wood, 2001).

### 2.5.2.2 Tropos

A metodologia Tropos foi introduzida em 2002 como uma metodologia AOSE abrangente, englobando todas as fases de concepção do projecto, desde levantamento de requisitos iniciais até a implementação (Giunchiglia et al., 2002). Os seus conceitos-chave incluem o actor, objectivos, plano, recursos, dependência, capacidade e crença (Bresciani et al., 2004). Os actores têm objectivos a atingir através da execução de actividades e com o auxílio de recursos. O produto da fase de análise é chamado de modelo de dependência estratégica, um grafo onde cada nó representa um actor e, os arcos, as dependências entre actores. Durante o levantamento inicial de requisitos, actores e objectivos, através de uma análise orientada ao objectivo, são identificados os intervenientes e seus objectivos. As dependências entre os intervenientes e objectivos também são identificadas.

Na fase levantamento de requisitos, todos os requisitos funcionais e não funcionais do sistema são especificados detalhadamente. Nesta fase, o sistema é considerado como um único actor, enquanto entidades externas presentes no ambiente são consideradas como agentes de interacção.

A fase de projecto arquitectural produz um modelo de arquitectura do sistema, e descreve como componentes trabalham em conjunto. Durante a fase que detalha o projecto, são produzidos modelos detalhados de cada componente, mostrando como os objectivos são atendidos pelos agentes. Nesta fase, detalhes como linguagem de comunicação de agente e protocolos são especificados usando uma linguagem de modelação mais detalhada, como UML (Giorgini et al., 2003).

### 2.5.2.3 Prometheus

Prometheus foi introduzido em 2002 como resultado dos conhecimentos da indústria em conjunto com os conhecimentos académicos (Padgham & Winikoff, 2002). Esta metodologia foi idealizada para se obter um processo que pudesse ser utilizado no desenvolvimento de agentes inteligentes, ela é composta por um conjunto de conceitos, processos, e métodos para obtenção dos requisitos e projecto do sistema. Neste sentido, ela fornece um processo detalhado e interactivo para a especificação, projecto, implementação e teste de sistemas de software orientado a agentes. Prometheus suporta o desenvolvimento de agentes inteligentes que usam arquitectura BDI, desta forma inclui alguns conceitos tais como crenças (*Beliefs*), desejos (*Desires*) e intenções (*Intentions*), eventos, planos e objectivos.

A metodologia inclui três fases: a fase de especificação do sistema, que se concentra no sistema como um todo, identificação de objectivos, funcionalidades, cenários de caso de uso e

especificar a interface do sistema (interacção sistema/ambiente); a fase do desenvolvimento da arquitectura, que usa os modelos produzidos na fase anterior para determinar os tipos de agentes que estarão presentes no sistema, como eles irão interagir uns com os outros e reagir as mudanças do ambiente; e a fase de projecto detalhado, que produz diagramas detalhados de funcionalidades de cada agente (capacidades, eventos, planos e dados) e recursos, bem como vários outros detalhes de implementação (Winikoff & Padgham, 2004). Uma ferramenta de projecto denominada PDT (*Prometheus Design Tool*) foi desenvolvida para oferecer suporte à metodologia Prometheus na concepção de sistemas de agente (Padgham et al., 2007). Esta ferramenta permite a validação e geração de código da estrutura do agente (Padgham & Winikoff, 2004).

#### 2.5.2.4 GAIA

A metodologia de GAIA original foi proposta em 2000, por Wooldridge, Jennings e Kinny (Wooldridge et al., 2000). Na sua proposta original consistia de duas fases: análise e projecto. Por outro lado, não se preocupa com as fases de levantamento de requisitos e de implementação (Wooldridge et al., 2000). Esta metodologia parte de um conjunto de requisitos para um determinado projecto que foi detalhado de forma adequada para ser implementado e durante o processo de modelação ele é refinado de forma sucessiva. GAIA apresenta dois conceitos principais: conceito abstracto – utilizados para conceituar o sistema durante a fase de análise; conceito concreto – utilizados no processo de projecto do sistema e corresponde determinar os elementos que estarão presentes na implementação. Na fase de análise são apresentados:

- **Modelo de papéis:** que descreve preliminarmente os papéis existentes no sistema, sendo esta uma descrição abstracta da função pretendida por uma entidade. Neste modelo também são indicadas as permissões e direitos associados a cada papel, que relacionam o tipo e a quantidade de recursos que podem ser explorados assim como as responsabilidades do papel, o que resulta na determinação da funcionalidade do agente no ambiente. É importante salientar que vários agentes podem desempenhar um mesmo papel, assim como um agente pode desempenhar vários papéis;
- **Modelo de interacção:** é desenvolvido um conjunto de definições de protocolo, um para cada tipo de interacção entre papéis, que consiste na definição dos atributos propósito (uma descrição sucinta da natureza da interacção), iniciador (quem tem o papel responsável por iniciar a interacção), interlocutor (papel com o qual o iniciador interage), entradas (é a informação recebida e utilizada pelo iniciador enquanto executando o protocolo), saídas (é a informação fornecida pelo/para o respondedor durante o curso de interacção).

Na fase do projecto são desenvolvidos os modelos de agente (define os tipos de agente que irão fazer parte do sistema, e as instâncias de agente que serão usadas a partir destes), modelos de serviços (derivados de actividades e protocolos de cada papel, define os principais

serviços que serão atribuídos a cada tipo de agente) e modelos de comunicação/relação (do inglês *acquaintance model*, que define os elos de comunicação entre os tipos de agente). Na Tabela 17, é apresentado um resumo de algumas extensões propostas para a metodologia de GAIA (uma análise mais detalhada de algumas dessas extensões pode ser encontrada em (Cernuzzi et al., 2004)).

**Tabela 17: Principais extensões da metodologia GAIA**

<b>Extensão ROADMAP</b>	As extensões ROADMAP ( <i>Role Oriented Analysis and Design for Multi-Agent Programming</i> ) para GAIA foram propostas em 2002, por Juan, Pearce e Sterling (Juan et al., 2002) e posteriormente estendida. ROADMAP introduz novos recursos para a metodologia de GAIA, a fim de eliminar ou atenuar algumas deficiências identificadas: apoio à colecta de requisitos (através da introdução de um modelo de casos de uso); novos modelos para descrever o conhecimento e domínio do ambiente (modelo de conhecimento e modelo de ambiente); níveis de abstracção que permite a decomposição interactiva do sistema, dos modelos e representações dos aspectos sociais e características individuais; e modelação de reflexão em tempo de execução, com objectivo de permitir mudanças de aspectos sociais e individuais em tempo de execução, como a permissão dos papéis de ler, escrever e alterar as permissões dos mesmos, seus atributos (tais como protocolos) ou um membro de um atributo (um protocolo específico, por exemplo).
<b>GAIA V. 2.0</b>	As extensões oficiais de GAIA (referida como GAIA v.2) foram introduzidas em 2003, por Zambonelli, Jennings e Wooldridge, para enriquecer a metodologia original (Zambonelli et al., 2003). A fase de análise foi expandida para incluir um modelo organizacional (decompondo o sistema em sub-organizações), um modelo de ambiente (que descreve o ambiente no qual o SMA está situado), e os modelos organizacionais (contendo regras organizacionais globais que o sistema deve respeitar e aplicar). A fase de projecto foi dividida em fases de concepção e projecto da arquitectura. Na primeira, as funções e diagramas de interacção são desenvolvidos, e um modelo de estrutura organizacional é introduzido (contendo a estrutura, a topologia e o controle do sistema). Na segunda, os modelos de agente e serviço são criados (como na metodologia original). A Figura 23 mostra esses modelos e as suas relações detalhadas na metodologia GAIA v.2
<b>AUML Extensions</b>	Agente UML (AUML5) foi introduzida no ano 2000 como um conjunto de UML idiomáticas e extensões para lidar com agentes (Odell et al., 2001),(Bauer et al., 2001). Em 2004, Cernuzzi e Zambonelli propoem que o protocolo de interacção agente (AIP) (parte principal do AUML) fosse usado em conjunto com GAIA para fornecer uma notação mais rica, compacta e formal para interacção dos agente, com o objectivo de reduzir a ambiguidade e permitir a especificação das várias possibilidades de escolha do agente (Cernuzzi & Zambonelli, 2004). Embora isso já tinha sido sugerido anteriormente (por exemplo, em (Juan et al., 2002) na página 6 ou (Zambonelli et al., 2003) na página 348) mas nunca tinha sido detalhado.

### **Other Extensions**

Em (Castro & Oliveira, 2008), Castro e Oliveira utilizam a metodologia de GAIA para modelação de um centro de controlo de operações de uma empresa aérea e propõem alguns complementos (e substituições), a alguns dos seus modelos. No seu trabalho propõem a substituição das tabelas de protocolo por diagramas de interacção de UML 2.0; a notação formal da estrutura organizacional com um diagrama UML 2.0; o modelo de agente com um diagrama de classes UML 2.0; e o modelo de serviço com um diagrama de classes UML 2.0.

Também é sugerido usar, em conjunto, uma representação UML 2.0 dos papéis e diagrama de interacção para ajudar a visualizar melhor os papéis, actividades e protocolos; e algumas representações gráficas combinadas para complementar o modelo de papéis preliminares, modelos de interacção e a estrutura organizacional.

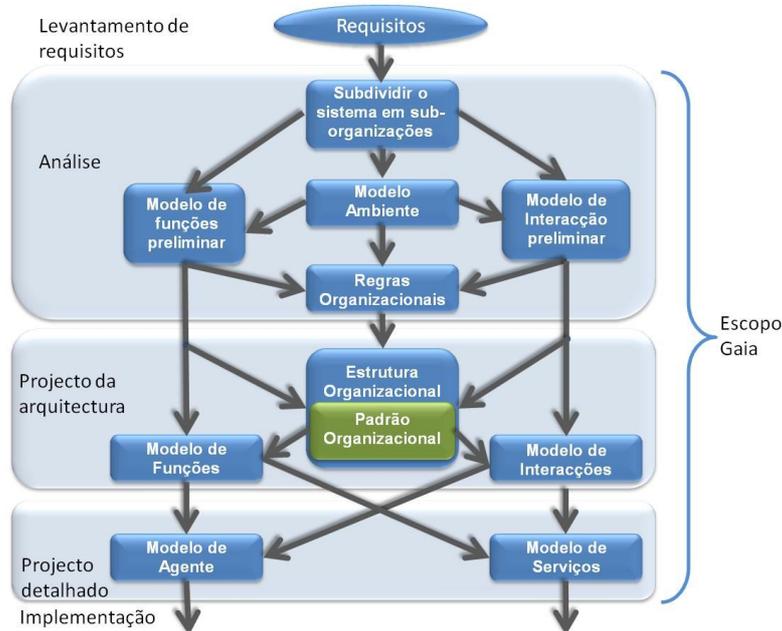


Figura 23: Modelo GAIA v.2 (adaptado de(Zambonelli et al., 2003))

Em (Gonzalez-Palacios & Luck, 2008), Gonzalez-Palacios e Luck estenderam a metodologia GAIA com a introdução de uma fase de projecto de agente e reforçar o processo metodológico com a utilização de iterações. A fase de projecto de agentes segue a fase de projecto detalhado da metodologia GAIA e produz uma especificação baseada em objecto do qual a implementação pode ser derivada. Esta abordagem não depende de uma arquitectura específica de agente, e ela permite escolher a melhor arquitectura para um determinado agente. A utilização de iterações fornece a metodologia GAIA um processo metodológico mais flexível que facilita o desenvolvimento de grandes sistemas, devido a decomposição do sistema em iterações.

Diversas publicações podem ser encontradas comparando algumas dessas metodologias, como por exemplo em (Iglesias et al., 1999), (Bayer & Svantesson, 2001) (apresentam uma análise detalhada da metodologia GAIA e a metodologia MAS-CommonKADS), (Dam & Winikoff, 2003) (os autores apresentam uma profunda análise de MaSE, prometheus e Tropos), ou (Sturm & Shehory, 2004) (os autores apresentam uma comparação abrangente entre GAIA, Tropos, MaSE de acordo com as várias características, agrupados por categorias), entre outros.

As metodologias orientadas a agentes continuam a ser desenvolvidas, e investigação nesta direcção vai procurar determinar quais metodologias orientadas a agente são mais adequadas

para auxiliar no desenvolvimento de determinados projectos ou sistemas. A metodologia GAIA usa uma visão organizacional para construir SMA, sendo reconhecida como uma valiosa metodologia para o desenvolvimento de sistemas complexos baseados na abordagem multi-agente abertas. No entanto, para ser utilizada no desenvolvimento de sistemas do mundo real, esta metodologia precisa ser estendida em vários aspectos (Gonzalez-Palacios & Luck, 2008).

### 2.5.2.5 Metamodelo com SPEM

Nesta subsecção, será apresentado o conceito de SPEM (*Software Process Engineering metamodel*) desenvolvido pelo OMG (*Object Management Group*), que em Novembro de 2002 oficializou como um padrão. O SPEM foi utilizado para obter uma versão estendida da metodologia GAIA. SPEM é um metamodelo para a descrição de um processo concreto de desenvolvimento de software ou uma família relacionada de processos de desenvolvimento de software e utiliza UML (*Unified Modeling Language*) como notação. Segundo a proposta inicial da OMG, SPEM limita-se a definir o conjunto mínimo de processos, o qual modela os elementos necessários para descrever qualquer processo de desenvolvimento de software.

Processo de software é um conjunto de actividades executadas para desenvolver e manter software, através de um fluxo finito de actividades inter-relacionadas que produzem produtos necessários às actividades posteriores. Um processo de software tem elementos comuns como: actividades, papéis e artefactos. Os objectivos da modelação de processos de software são facilitar o entendimento, a adaptação e gerência do processo.

O SPEM modela os processos através de diagramas UML estereotipados, onde define estereótipos para seus elementos de modelação, baseado na ideia que um processo de desenvolvimento de software é uma colaboração entre entidades abstractas activas, nomeadas de papéis (*roles*), que realizam operações, chamadas actividades (*activities*), sob entidades concretas e tangíveis, denominada de artefactos ou produtos de trabalho (*workproduct*). Um *process* pode ser considerado como uma colaboração entre papéis para atingir uma meta ou objectivo. Para direccionar a sua execução, ordena-se as actividades pela ordem, em que devem ser, ou podem ser, executadas. Também é necessário definir o modelo do processo no tempo (estrutura *Lifecycle* em termos de fases e interacções). *Lifecycle* de processo é definido como uma sequência de fases para atingir uma meta específica, e resulta na definição do comportamento completo de um processo a ser executado num certo projecto ou programa.

A descrição do trabalho executado no processo é um tipo de *Operation*, chamada de *Workdefinition*, e suas subclasses são *Activity*, *Phase*, *Iteration* e *Lifecycle*. Alguns dos estereótipos mais utilizados neste trabalho estão descritos na Tabela 18.

Tabela 18: Alguns estereótipos de SPEM 2.0

Representação	Estereótipo	Descrição
	<i>WorkProduct</i> (Artefacto ou produto do trabalho)	É qualquer elemento produzido, consumido ou modificado por um processo. Ex: modelos, planos, documentos, código fonte, etc. É a descrição de uma classe de produtos do trabalho produzidos num processo e uma categoria de produto do trabalho
	<i>UML/Formal Model</i> (Modelo Formal em UML)	É uma especialização de <i>WorkProduct</i> .
	<i>RoleDefinition</i> (Definição do Papel)	Descreve os papéis, responsabilidades e competências que um determinado indivíduo tem dentro do processo
	<i>TaskDefinition</i> (Definição da Tarefa)	É um tipo de operação que descreve o trabalho realizado num processo.
	<i>Activity</i> (Atividade)	É a principal subclasse de <i>WorkDefinition</i> , descreve um trabalho (tarefas, operações e ações) realizado por um <i>ProcessRol</i> , em outras palavras, atividade descreve uma determinada atividade que um papel realiza dentro de um processo.
	<i>Phase</i> (Fase)	É uma especialização de <i>WorkDefinition</i> em que a <i>precondition</i> define o critério de entrada da fase e o <i>goal</i> define o critério de saída da fase. Uma fase é definida como um seqüenciamento de <i>goals</i> num espaço de tempo
	<i>ProcessPackage</i> (Pacote de processos)	É um agrupamento coerente de elementos do processo (produtos, papéis, atividades) cujas atividades são organizadas segundo algum ponto de vista ou tema comum (Ex: Análise e Projeto, teste, implementação, etc.)

Em (Garro & Turci, 2003) é apresentado um metamodelo SPEM para a versão original do GAIA. Este modelo foi produzido pelo comité técnico de metodologia da FIPA usando a versão anterior do SPEM, versão 1.1 (Group, 2005). Em outros trabalhos como (García-Ojeda et al., 2005) os autores usam SPEM para descrever a integração de GAIA com AUML e (Moraitis & Spanoudakis, 2006) o SPEM é usado para descrever a integração de GAIA com a plataforma de JADE.

## 2.6 Sumário

Este capítulo apresentou os conceitos teóricos da área de inteligência artificial e robótica inteligente que foram utilizados no desenvolvimento deste trabalho. Inicialmente, como introdução do capítulo, foi apresentado um resumo histórico da área da IA e de que forma o tema do trabalho está relacionado com a IA.

Na segunda secção foram apresentadas algumas definições, premissas e restrições impostas para o desenvolvimento do trabalho. Na comunidade científica algumas definições não são muito clara ou aceitas por toda a comunidade, justificando assim, apresentação destas definições e premissas admitidas neste trabalho.

Alguns dos conceitos específicos apresentados neste trabalho foram:

- Robótica inteligente: foi apresentado nesta subsecção o conceito de agentes reactivos e arquitectura de subsunção; na seguinte, uma breve discussão de como é feita a modelação do mundo na robótica; logo em seguida, foi apresentada uma parte da teoria de grafos que são usados como ferramenta na área da IA que estuda métodos genéricos de resolução de problemas e métodos de planeamento automático; e finalizou com métodos de navegação e controlo.
- Interfaces homem-máquina, suas arquitecturas e projectos de interfaces multimodais.
- Sistemas multi-agente: resumo do paradigma de SMA, plataforma de SMA e métodos de modelação de SMA.

No próximo capítulo será apresentado o estado da arte sobre CRI, com os principais projectos desenvolvidos. Será também apresentada uma discussão relativa aos aspectos em comum. Os conceitos fundamentais, técnicas/teorias apresentados neste capítulo estarão presentes no desenvolvimento desta tese apresentado nos Capítulos 4 e 5.

# Capítulo 3

## Estado da Arte

O objectivo deste capítulo é apresentar os principais trabalhos de investigação desenvolvidos relacionados com a área das cadeiras de rodas inteligentes. O capítulo está dividido em quatro secções. Inicialmente é apresentada uma pequena introdução referente à quantidade de trabalhos já publicados e um resumo de cada um dos principais projectos publicados. As secções dois e três apresentam, respectivamente, uma lista alargada dos projectos e uma síntese das características comum entre eles. Finalmente, a última secção, contém o sumário do capítulo.

### 3.1 Principais projectos de Cadeiras de Rodas Inteligentes

Nos últimos anos, um grande número projectos de cadeiras de rodas inteligentes tem sido desenvolvido. A Figura 24 ilustra o número de publicações disponíveis, em algumas das principais bases de dados científicas, relacionadas com o tema das cadeiras de rodas inteligentes. Foram utilizados seis motores de pesquisa/base de dados, e realizadas duas pesquisas distintas utilizando o termo cadeira de rodas inteligente em inglês. Na primeira pesquisa foi utilizado como parâmetro de pesquisa as palavras juntas, ou seja, “*intelligent wheelchair*” e na segunda pesquisa, os parâmetros separados ou seja “*intelligent*” e “*wheelchair*”. Os resultados do Google académico para a pesquisa com as palavras separadas foram retirados do gráfico pois o número encontrado foi superior a 16.000, sendo que muitos destes resultados não correspondem a área de interesse.

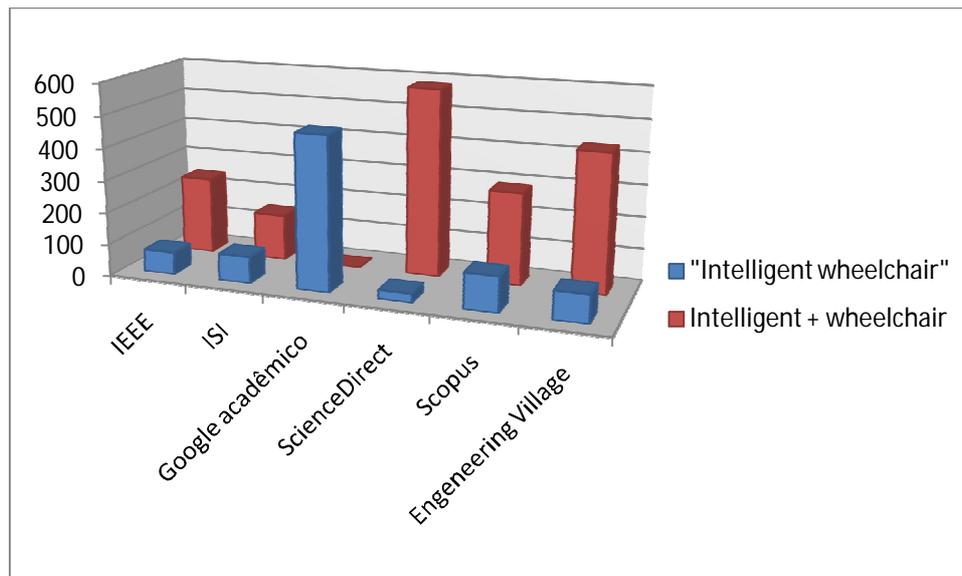


Figura 24: Resultados da pesquisa pelo tópico "Cadeira de rodas inteligente"

As descrições dos projectos seguintes são o resultado de uma revisão bibliográfica referente a projectos de cadeira de rodas inteligente (CRI) com o foco na concepção, implementação, interface e uso dos sistemas de comunicação.

### 3.1.1 Cadeira de Rodas Autónoma da Universidade do Estado do Arizona

Um dos primeiros projectos com o conceito de uma cadeira de rodas autónoma para deficientes físicos foi proposto por Madarasz et al. (Madarasz et al., 1986), em 1986, onde apresentaram uma cadeira de rodas equipada com um microcomputador, uma câmara digital, e um scanner ultra-som, o protótipo apresentado neste projecto é ilustrado na Figura 25. O seu objectivo foi desenvolver um veículo capaz de operar sem a intervenção humana em ambientes populosos com poucas ou nenhuma colisão com os objectos ou pessoas nele contidos. Neste projecto, a câmara foi utilizada para reconhecer marcos de referência e objectos móveis, assim como, localizar, reconhecer objectos anteriormente catalogados (tal como número da sala e elevadores) e também para conduzir a cadeira no centro do corredor. Os sensores de ultra-som foram utilizados para determinar as distâncias da cadeira em relação aos objectos e para orientar a cadeira em relação às paredes e aos corredores. Em algumas situações foram utilizadas as informações do ultra-som em conjunto com a câmara, como por exemplo para verificar se a porta do elevador está aberta ou fechada.

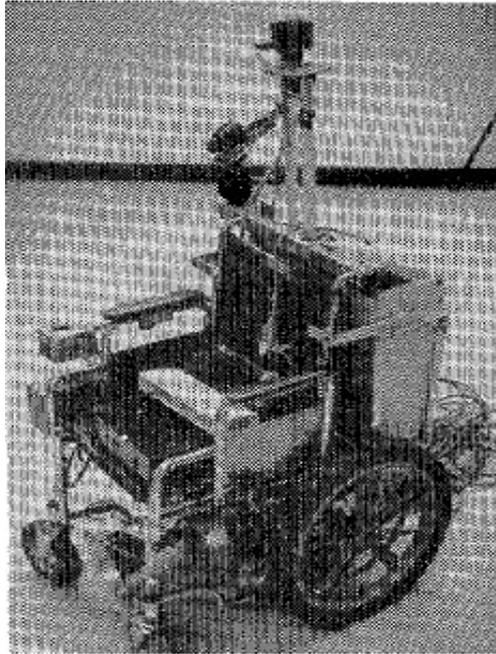


Figura 25: Protótipo desenvolvido por Madarasz

### 3.1.2 CALL Centre

De 1990 a 1994, foram executados dois projectos relacionados com CRI, pelo grupo CALL Centre da Universidade de Edimburgo na Escócia (Odor, 1995). O primeiro destinava-se a desenvolver uma cadeira de rodas inteligentes e, em seguida, o segundo projecto destinava-se a realizar a sua avaliação. Uma investigação para a avaliação qualitativa sobre o uso eficaz de cadeiras de rodas inteligentes para emergir mobilidade, comunicação, educação e desenvolvimento parte das crianças utilizadoras deste recurso.

A CRI chamada CALL Centre Smart Wheelchair originalmente foi desenvolvida para crianças com diversas deficiências graves que não podem utilizar os sistemas de mobilidade tradicionais, o protótipo apresentado neste projecto é ilustrado na Figura 26. Com a CRI, as crianças podem experimentar novas oportunidades de comunicação, de aprendizagem, exploração, brincadeiras e adquirir algum grau de mobilidade independente. O projecto concentrou-se no desenvolvimento da CRI e ferramentas para avaliar a sua utilização. Doze protótipos de CRI foram desenvolvidas e testadas em três escolas especiais em Edimburgo.

Os principais objectivos na avaliação consistiam em: Ajudar a equipa do projecto a analisar o valor do investimento em cadeiras; Explorar os efeitos da motivação, mobilidade e as oportunidades de comunicação para crianças com idades e deficiências variadas e numa variedade de configurações;

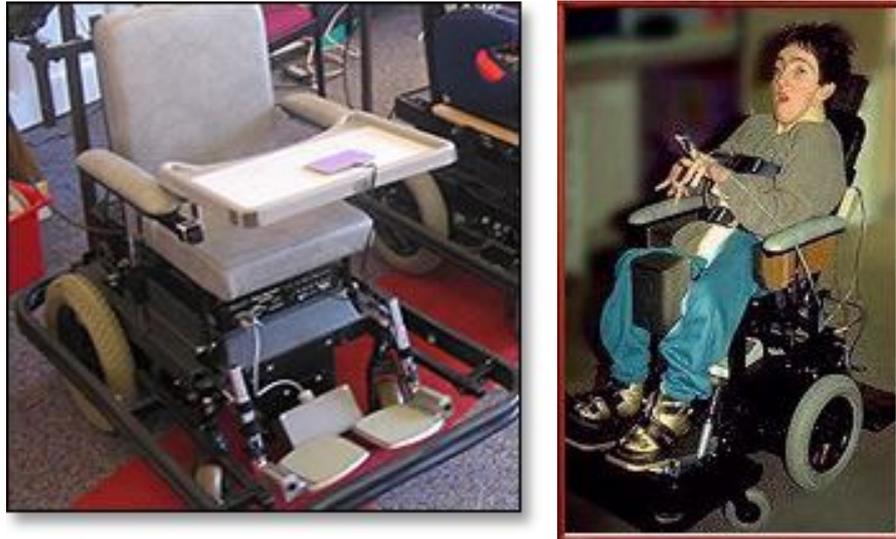


Figura 26: CRI – CALL Centre da Universidade de Edimburgo

Esta CRI pode ser conduzida por um único ou vários dispositivos: um selector de direcção, joystick proporcional e/ou um computador portátil.

A base da CRI baseia-se numa cadeira de rodas motorizadas comercial, controlada por um computador. As cadeiras destinam-se a ser usadas por crianças que não têm as habilidades físicas, de percepção ou cognitivas para controlar uma cadeira motorizada normal, e portanto, há uma série de recursos para proteger o utilizador e o ambiente e aumentar o número de actividades possíveis com a cadeira.

Sensores de colisão podem parar a cadeira em contacto com um obstáculo, voltar ou desviar do objecto. Sensores de ultra-som foram utilizados para reduzir a velocidade antes de colisões e para auxiliar a passagem em corredores e portas. A função de seguir linha permite que a cadeira siga uma linha feita no chão. Estas são utilizadas para auxiliar na navegação de uma sala para outra sala ou situações difíceis (como passagem por portas). A cadeira confirma as instruções e anuncia a acção para o utilizador por meio de um sintetizador de voz ou outro recurso.

Neste projecto foi dada ênfase a criar uma CRI que complementasse o utilizador, ou seja, o objectivo não foi desenvolver um veículo autónomo que teria pouco valor educativo ou terapêutico. O controlo da cadeira foi visto como uma parceria simbiótica entre o utilizador e a cadeira. Cada cadeira é desenvolvida com características específicas, dispositivos e controlos escolhidos para satisfazer objectivos educativos e terapêuticos para cada criança individualmente, de acordo com as habilidades do utilizador e o tipo de ambiente (Nisbet, 2002).

### 3.1.3 OMNI

Hoyer e Holper (Hoyer & Hölper, 1993) apresentaram, em 1993, uma arquitectura de um controlo modular para uma cadeira de rodas omnidireccional onde o seu objectivo foi desenvolver um sistema de grande funcionalidade com alta flexibilidade, o protótipo apresentado neste projecto é ilustrado na Figura 27. Deste modo, estes investigadores desenvolveram uma arquitectura aberta de controlo com uma estrutura modular composta por diversas unidades, providas de uma inteligência local, sendo que os módulos são independentes entre si (Hoyer et al., 1997). Na arquitectura apresentada o sistema foi dividido em controlo de baixo nível (incluindo a unidade de movimentação, módulo de sensores e braço robótico), funcionalidades de alto nível (abrange módulo de planeamento de trajectória, planeador de tarefas) e módulo de interface (controlo por voz, terminal e joystick).



Figura 27: Cadeira de rodas omnidireccional do projecto OMNI

Na arquitectura inicial apresentava um braço robótico MANUS. A continuidade deste projecto deu-se vinculada ao projecto OMNI (*Office Wheelchair with High Manoeuvrability and Navigational Intelligence for People with Severe Handicap*) (Buhler et al., 1995). O protótipo OMNI apresentou funcionalidades interessantes de navegação tais como o desvio de obstáculo simples, algumas funções específicas (seguir parede, passagem por porta) e possibilidades de navegação autónoma.

### 3.1.4 NavChair

A ideia do projecto NavChair foi concebida inicialmente por Simon Levine (Director de reabilitação física no hospital da Universidade de Michigan, USA). O projecto iniciou em 1991 com um subsídio de US\$ 330 mil e duração de 3 anos.

NavChair é uma CRI com base numa cadeira de rodas motorizada comercial acrescida de um computador, sensores de ultra-som, módulo de interface com joystick e módulo de potência da cadeira de rodas (Simpson, 1998) (Bell et al., 1994) (Levine et al., 1997)(Figura 28). As funções disponíveis que caracterizam esta cadeira de rodas são: desvio de obstáculos; seguir paredes; passar por portas. Neste projecto foi desenvolvido e testado um sistema de controlo compartilhado, onde o utilizador e a CRI dividem o controlo com o objectivo de uma navegação mais segura.



Figura 28: Protótipo da CRI - NavChair

### 3.1.5 Tin Man

Miller e Slak (Miller & Slack, 1995)(Miller, 1998) investigadores norte americanos, vinculados ao MITRE Corporation e KISS Instituto para Robótica Prática - KIPR, projectaram o sistema Tin Man I (Figura 29- a) com o objectivo de criar um equipamento de baixo custo, onde o mesmo foi desenvolvido com base numa cadeira de rodas motorizada comercial acrescida de uma variedade de sensores (*encoders*, sensores de contacto, Infravermelhos, sonares e bússola) e um microprocessador com a finalidade de auxiliar o utilizador na operação da cadeira de rodas motorizada. Nesta primeira etapa, foram apresentados três modos de operação: Utilizador a conduzir com desvio automático de obstáculo; Mover-se ao longo de uma trajectória; Mover para um ponto (x,y).



a) Tin Man I

b) Tin Man II

Figura 29: Sistema Tin Man

A evolução do projecto resultou na construção do protótipo Tin Man II com os seguintes objectivos: modificar a interface com o utilizador; aumentar a velocidade de operação; reduzir a dependência dos sensores de contacto, desta forma, foram instalados 20 sensores de infravermelhos.

Nesta nova versão, Tin Man II (Figura 29- b) apresentou mais algumas funções como: armazenar informações de viagens; retornar ao ponto de partida; seguir parede; passar por portas; ir recarregar a bateria.

### 3.1.6 Wellman

Wellman et al. (Wellman et al., 1994) propuseram uma cadeira de rodas híbrida que foi equipada com duas pernas adicionais às suas quatro rodas, para permitir a mesma subir degraus e mover-se em terrenos acidentados. Este projecto foi desenvolvido no Laboratório GRASP (*General Robotics and Active Sensory Perception*) da Universidade da Pensilvânia, Filadélfia, o protótipo apresentado neste projecto é ilustrado na Figura 30.

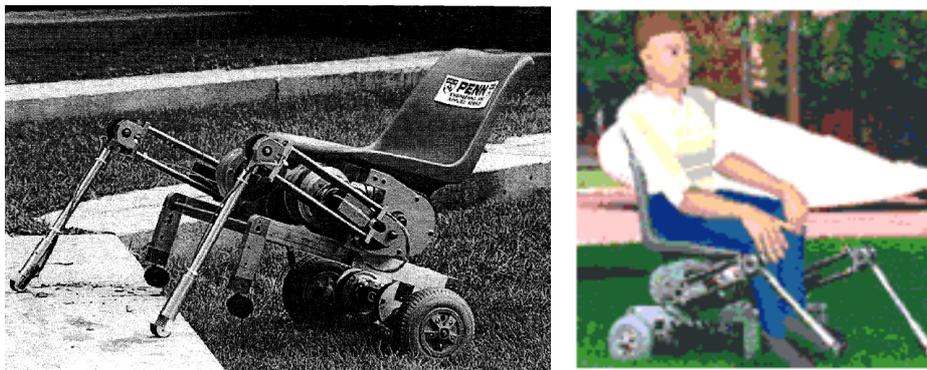


Figura 30: Cadeira de rodas equipada com duas pernas

### 3.1.7 FRIEND:

O projecto FRIEND (*Functional Robot arm with user-frIENdly interface for Disabled people*) desenvolvido pelo Instituto de automação da Universidade de Bremen, é um robô para reabilitação cujo objectivo é auxiliar deficientes físicos que possuem dificuldades de locomoção e também deficiência motora nos membros superiores (Martens et al., 2001), os protótipos apresentados neste projecto são ilustrados na Figura 31. Na sua primeira versão era composto por uma cadeira de rodas motorizada e um manipulador MANUS com seis graus de liberdade (Borgerding et al., 1999), ambos os sistemas, cadeira e manipulador, eram controlados por uma arquitectura complexa de controlo. O utilizador pode interagir com o sistema através de uma interface homem-máquina que consiste numa tela plana e comandos de voz. Neste caso, o controlo interpreta os comandos de voz transformando-os em ordens executáveis. Os comandos de voz em linguagem natural são captados pelo microfone e reconhecidos através do software ViaVoice Gold da IBM® e, após traduzidos em palavras específicas, estas são enviadas para um interpretador de comandos.

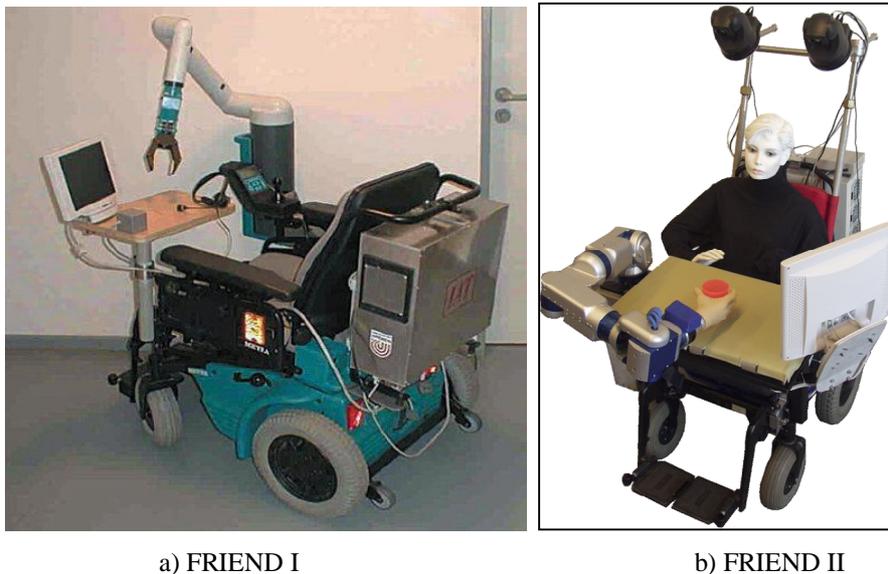


Figura 31: Projecto FRIEND

Em 2005 o sistema evoluiu para uma segunda geração, denominada de FRIEND II. Além de estender o hardware anterior, foi implementada uma nova estrutura de software em várias camadas (Prenzel et al., 2005). Sua finalidade era poder usar e interagir com dispositivos inteligentes presentes num ambiente doméstico através da adaptação ou a geração de novas sequências de acções. Essas sequências de acções devem ser utilizadas para controlar automaticamente o sistema. O Framework foi dividido em 3 partes: camada de hardware, camada de habilidades e camada controle sequencial. O que resultou na possibilidade de executar diversas habilidades simultaneamente. Para permitir isso, foram refeitos a forma de comunicações entre os módulos do sistema para usar CORBA e seus mecanismos específicos. A modularidade e expansibilidade do sistema deve-se à implementação deste método de

comunicação. Além disso, o uso de CORBA permitiu o encapsulamento de recursos de comunicação e uma gestão avançada de dados.

### 3.1.8 Smartchair

Smartchair é uma CRI desenvolvida no GRASP, laboratório robótica e automação da Universidade da Pensilvânia nos EUA. A cadeira é composta por duas câmaras (uma omnidireccional e outra tradicional para a interacção com o utilizador), vídeo projector, sensores infra-vermelho, scanner laser, placa de processamento embarcada e controlo inteligente adaptado ao utilizador, o protótipo apresentado neste projecto é ilustrado na Figura 32. A CRI disponibiliza vários modos de operação, incluindo o modo "viagem para o destino" que usa uma interface adaptável de acordo com circunstância de utilização, navegar em corredor, passagem por porta e desvio de obstáculo (Parikh et al., 2005), (Parikh et al., 2004), (Parikh et al., 2007). Este projecto utiliza fusão sensorial das informações da visão por computador com o scanner laser para calcular as informações de profundidade.

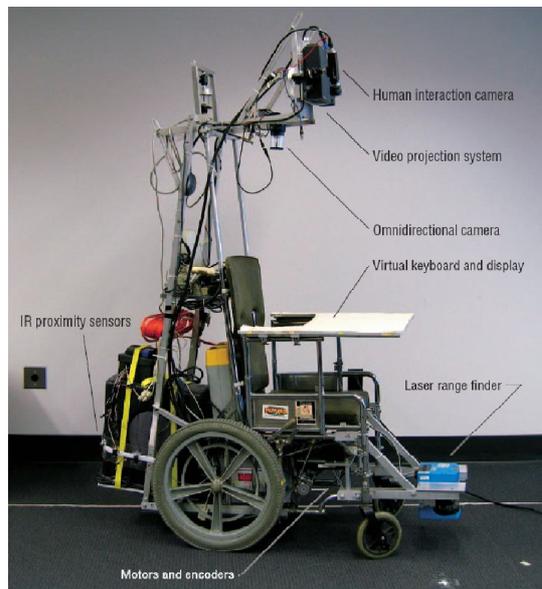


Figura 32: Protótipo Smartchair

No projecto Smartchair foram utilizados *encoders* e a câmara com o reconhecimento de marcos de referência para a localização e orientação da cadeira em ambientes fechados. Testes preliminares em ambientes externos também foram efectuados utilizando GPS.

### 3.1.9 VAHM

As primeiras referências bibliográficas do projecto *Véhicule Autonome pour personnes Handicapés Moteurs* (VAHM), foram publicadas a partir de 1998 (Bourhis & Agostini, 1998). O foco do projecto era definir uma interface homem-máquina como um caminho para controlar de forma mais eficiente as cadeiras eléctricas. VAHM foi desenvolvida no Laboratório de Automação e Sistemas Co-operativo (LASC) da Universidade de Metz em

França. Actualmente, já está no seu terceiro protótipo, sendo que o primeiro protótipo foi construído a partir de um robô móvel. Com a evolução do projecto, os protótipos seguintes foram construídos com base em cadeiras de rodas motorizadas (Figura 33). Este projecto apresenta algumas características como: a navegação autónoma através de mapas internos, navegação semi-autónoma com desvio de obstáculos e a função de seguir paredes.



Figura 33: Protótipos do projecto VAHM

Por volta do ano 2000 um projecto de um módulo simulador foi iniciado e motivado pela necessidade de avaliar o sistema desenvolvido em colaboração com pacientes deficientes, este simulador recorre à realidade virtual com o objectivo de permitir que os pacientes reais pudessem testar a cadeira de forma segura com algum realismo (Niniss & Nadif, 2000). Em 2002 a arquitectura de controlo do projecto evoluiu para uma estrutura que levaria em conta o ambiente circundante (Pruski et al., 2002). A nova arquitectura de controlo foi dividida em três partes essenciais: planeamento, comportamento e coordenação. Eles foram implementados por meio de três classes de agentes:

- Agentes de comportamento – fornecem informações de controlo de baixo nível;
- Agentes cognitivos – responsáveis pela recolha dos dados fornecidos pelos agentes de comportamento e interpretá-lo. Estes agentes criam um modelo do ambiente que os rodeia;
- Agentes de ambiente – responsáveis pela interacção da cadeira de rodas com o ambiente externo.

Esses agentes comunicam e interagem uns com os outros, fazendo uso de um sistema multi-agente que com uso de memória compartilhada como facilitador de comunicações.

A utilização de um sistema multi-agente para fazer a gestão da comunicação num ambiente de cadeira de rodas trouxe inúmeras vantagens para o sistema (Pruski et al., 2002). Uma delas foi a robustez do sistema. A implementação dos agentes acima mencionados,

tornou possível implementar um sistema de controlo redundante. Sempre que um agente para de funcionar devido a uma falha do sensor ou por problemas funcionais, ele é imediatamente substituído por um novo.

### 3.1.10 Robô SENA

O robô SENA está actualmente em desenvolvimento na Universidade de Málaga, na Espanha. É um dos poucos projectos de CRI na literatura que tratam do sistema de comunicação. Tem como base uma cadeira de rodas eléctrica comercial equipada com diversos sensores, bem como com uma câmara montada na parte superior e é controlada por um computador através de uma conexão USB ao micro-controlador que actua sobre de motores da cadeira (Gonzalez et al., 2006), o protótipo apresentado neste projecto é ilustrado na Figura 34.



Figura 34: Cadeira de Rodas SENA

A arquitectura de controlo da SENA foi inicialmente desenvolvida com base numa estrutura de 3 camadas denominada por ACHRIN (Galindo et al., 2006). Essas camadas funcionais foram chamadas de deliberativa, execução e controlo. O conjunto destas camadas facilitou a integração do utilizador em qualquer uma das operações robóticas da cadeira de rodas, incluindo a tarefa de deliberação e a execução do plano.

O último sistema desenvolvido evoluiu para um sistema multi-agente (Galindo et al., 2006), chamado de MARCA, pois o sistema inicial ACHRIN apresentava algumas deficiências. Entre estas deficiências podem-se citar:

- O modelo rígido de comunicação cliente-servidor;
- Redundância do sistema tinha que ser executada manualmente, ou seja, o sistema não possuía um mecanismo que permitisse executar rotinas ou funções que fossem executadas automaticamente para trazer a redundância do sistema.

A metodologia de Sistema Multi-Agente (SMA) foi escolhida devido à maturidade dos SMA, robustez, escalabilidade, bem como a autonomia e comportamento racional que é dada aos agentes. O Sistema implementado anteriormente foi adaptado para essa nova arquitectura com base na utilização de agentes. Foi criado um modelo agente padrão que serviu como base para todas as implementações dos agentes e este foi nomeado estrutura comum de agente (CAS).

A comunicação inter-agente foi projectada para usar passagem de mensagens e seu formato foi baseado no padrão FIPA-ACL (FIPA, 2009). Além disso, um sistema de aprendizagem foi incorporado na máquina de estado dos agentes com um sistema de recompensa.

### 3.1.11 RoboChair

RoboChair é um projecto desenvolvido pela Universidade de Essex no Reino Unido em conjunto com Instituto de Automação da Academia Chinesa de Ciência (CAS) em Beijing-China. O objectivo principal neste projecto foi desenvolver uma cadeira de rodas inteligente com um alto desempenho e baixo custo para auxiliar na mobilidade dos idosos e deficientes físicos. Dentro deste contexto, desenvolveu-se uma interface amigável entre o utilizador e a cadeira, capacidade de evitar colisão e planeamento de trajectória automático (Figura 35). Esta CRI está equipada com sistema de visão e sistema de comunicação sem fios para permitir que médicos, enfermeira ou outra pessoa responsável pelo paciente possam monitorar e telecomandar a cadeira, caso seja necessário.

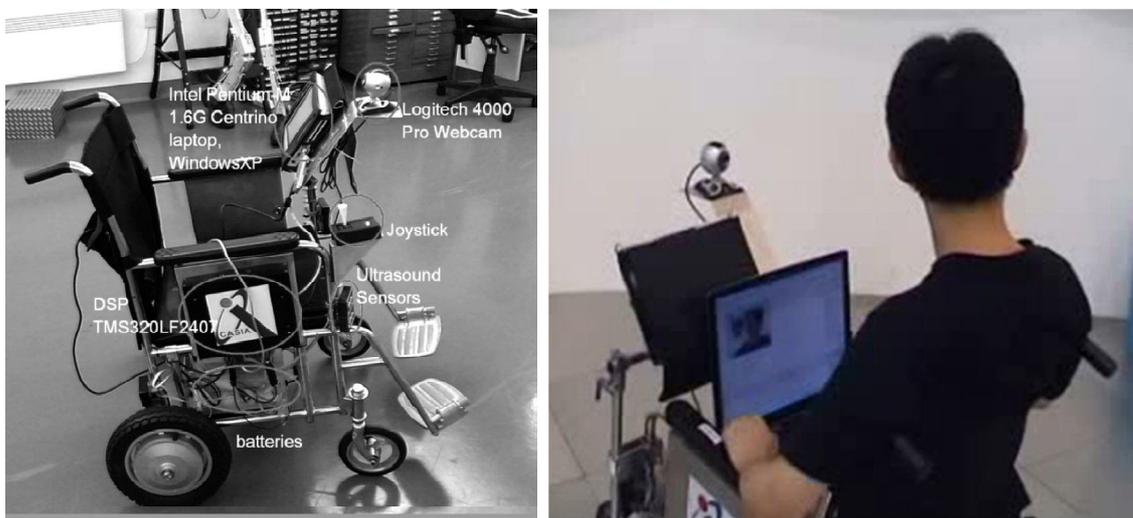


Figura 35: RoboChair: Cadeira de rodas com comando baseado em gestos com a cabeça

Neste desenvolvimento, com a finalidade de ter maior precisão na detecção do rosto foram combinados dois algoritmos (*Adaboost Face Detection* e *Camshift object tracking*) o que resultou num sistema de controlo de mãos-livres por rastreamento do rosto e reconhecimento de gestos em tempo real (Jia et al., 2007).

Também neste projecto é utilizada visão por computador para identificar marcos de localização, assim como o desenvolvimento de vários modos de operação que inclui o seguimento de parede, evitar colisão e navegação autónoma para algum sítio marcado no mapa.

### 3.1.12 ACCoMo

Este protótipo foi desenvolvido na divisão de inteligência artificial e departamento de engenharia eléctrica e electrónica em Chiba-shi no Japão, com o nome ACCoMo (sigla do inglês: *intelligent wheelchair as Autonomous, Cooperative, COLlaborative MOBILE robot*) (Hamagami & Hirata, 2004), o protótipo apresentado neste projecto é ilustrado na Figura 36.



Figura 36: Protótipo ACCoMo

ACCoMo é um protótipo de CRI projectado para permitir a movimentação segura de deficientes físicos em ambientes fechados. Este protótipo foi construído a partir de uma cadeira de rodas motorizada comercial, sensores, computador portátil, ecrã táctil e etc. A proposta do projecto ACCoMo é ser um agente com comportamento autónomo com desvio de obstáculos (Hamagami & Hirata, 2005), cooperativo com outras cadeiras e colaborativo com o utilizador. A inteligência desta CRI advém de algoritmos de aprendizagem por reforço, redes neurais e algoritmos genéticos. Espera-se que um nível maior de inteligência possa emergir com a interacção entre agentes e o ambiente. A navegação autónoma baseia-se em mapas de ambientes internos, e a localização com Identificação por Rádio Frequência (RFID, do inglês *Radio Frequency IDentifier*). A interacção com utilizador é efectuada através de uma interface com o ecrã táctil e a interacção com outros dispositivos é realizada através de um sistema de comunicação e o auxílio do RFID para conhecer a posição dos dispositivos. Um exemplo de comportamento cooperativo apresentado (Hamagami & Hirata, 2004) baseou-se numa situação de passagem por um corredor estreito onde duas cadeiras desejam passar ao mesmo tempo num dado ponto. A solução apresentada consiste em que, quando as CRI se encontram neste ponto estreito elas comunicam e emerge uma concessão de uma das CRI. Desta forma uma delas recua permitindo a passagem da outra.

### 3.1.13 RobChair:

O RobChair foi um projecto desenvolvido pelo Instituto de Sistemas e Robótica da Universidade de Coimbra em Portugal (Pires & Nunes, 2002) (ver Figura 37). RobChair foi definido como um robô móvel construído a partir de uma cadeira de rodas motorizada comercial, dotada de sensores, um modo diferenciado de interacção com o utilizador (Pires et al., 1998). Este projecto teve por finalidade desenvolver um protótipo com vista em desenvolver e testar novas plataformas de controlo que possam melhorar as condições de locomoção das pessoas (Solea & Nunes, 2008). Para permitir uma maior capacidade de autonomia aos utilizadores procuraram-se capacidades de navegação autónoma, neste sentido foram desenvolvidos o modo de operação semi-autónomo e autónomo.



Figura 37: Robô móvel RobChair

Robchair apresentou uma arquitectura baseada num sistema distribuído e modular. Os diversos módulos de hardware que compõem esta CRI estão interligados sobre um barramento de comunicação CAN (*Controller Area Network*). O módulo de controlo implementado neste projecto segue a arquitectura de subsunção (*subsumption architecture*) proposta por Rodney Brooks (Brooks, 1986) onde vários níveis de funcionalidades são representados, como por exemplo: navegação segura (comportamento de evitar colisão e controle de velocidade); e comportamento de desvio de obstáculos.

O comportamento de evitar colisão representa a habilidade de não colidir com os obstáculos, enquanto o comportamento de desvio de obstáculos representa a capacidade de contornar os obstáculos com eficiência. Estes são métodos reactivos que utilizam as informações dos sensores e actuam nos motores em tempo-real. Para implementar estes comportamentos foi utilizado o método do campo potencial que gera uma força repulsiva artificial associada aos obstáculos.

### 3.2 Síntese dos projectos de CRI

A Tabela 19 apresenta uma lista alargada dos principais projectos de CRI relatados na literatura (Simpson, 2005). Nela são apresentados o nome e localidade do projecto, assim como, os tipos de sensores que foram utilizados, tipos de entradas do utilizador, a partir de que base a CRI foi construída, métodos de controlo, forma de representar o ambiente, modos de operação e quem participou dos testes.

Tabela 19: Projectos de cadeiras de rodas inteligentes

Cadeira de Rodas Inteligente	Sensores							Entradas				Factor-Forma		Controlo											
	Sonar	Infravermelhos	Scanner laser	Laser stripe	Visão Computacional	colisão/contacto	Odometria	Sequência	Bussola	GPS	Giroscópio	Voz	EOG	Force-Feedback Joystick	Orientação pela face/cabeça	Reconhecimento de gestos	Deictic	EMG	Robo Modificado	CR Motorizada Modificada	Add-On Unit	Omnidireccional	Sub assumção	Redes Neurais	Multi-níveis
1 ACCoMo, Universidade Chiba, Chiba, <b>Japão</b>	X		X									X							X				X	X	X
2 Automated-Guided Wheelchair—NEC Corporation, <b>Japão</b>	X					X														X*					
3 Autonomous Wheelchair—Arizona State University, <b>USA</b>	X				X	X														X					
4 CALL Centre, Universidade of Edinburgh, <b>Escócia</b>					X	X						X								X					
5 CCPWNS—University of Notre Dame, <b>USA</b>					X	X														X*					
6 CHARHM—Advanced Technologies Development Center, <b>Argélia</b>					X															X					
7 COACH—French Atomic Energy Comission, <b>França</b>	X	X																		X					
8 CWA—National University of Singapore, <b>Singapore</b>						X														X					
9 CWA (Power)						X														X					
10 Deictic Wheelchair, Northeastern University, Boston, <b>USA</b>	X			X	X							X								X					
11 EASY, Universidade de Bremen, <b>Alemanha</b>					X	X						X	X							X					X
12 FRIEND, Universidade de Bremen, <b>Alemanha</b>					X	X					X	X								X					
13 Hephaestus—TRAC Labs, <b>USA</b>	X				X															X*					
14 INCH—Yale University, <b>USA</b>	X	X			X	X														X					
15 INRO—FH Ravensburg-Weingarten, <b>Alemanha</b>	X	X			X	X		X												X					
16 Intelligent Wheelchair—University of Texas at Austin, <b>USA</b>	X	X			X	X	X																		
17 Intelligent Wheelchair System—Osaka University, Japan (protótipo 1)	X				X	X						X	X							X					
18 Intelligent Wheelchair System (protótipo 2)	X				X	X						X								X					
19 IntellWheels, Universidade do Porto, FEUP, <b>Portugal</b>	X	X		X	X	X					X	X	X	X					X			X	X		
20 Luoson III—National Chung Cheng University, <b>Taiwan</b>	X				X	X	X		X		X	X								X					
21 MAid—RIAKP, <b>Alemanha</b>	X	X	X			X				X										X					X
22 Mister Ed—IBM, <b>USA</b>	X	X			X	X														X			X	X	
23 Mr. HURI—Yonsei University, <b>Korea</b>	X				X						X		X	X						X					
24 NavChair—University of Michigan, <b>USA</b>	X				X	X					X									X					
25 NLPR Robotized Wheelchair—Chinese Academy of Sciences, <b>China</b>	X				X	X					X									X			X	X	
26 OMNI—University at Hagen, <b>Alemanha</b>	X	X			X	X														X	X			X	
27 Orpheus—National Technical University of Athens, <b>Grécia</b>	X					X														X					
28 Phaeton—Northeastern University, <b>USA</b>	X	X		X	X	X								X						X					
29 RobChair—University of Coimbra, <b>Portugal</b>	X	X			X						X									X			X		
30 RoboChair, Universidade de Essex e CAS, <b>UK e China</b>	X				X						X	X	X	X						X					
31 Robotic Wheelchair—FORTH, <b>Grécia</b>	X				X	X														X					
32 Rolland—University of Bremen, <b>Germany</b> (protótipo 1)	X	X		X	X	X														X				X	
33 Rolland (protótipo 2)	X				X															X				X	
34 Rolland (protótipo 2 com laser)		X			X															X					
35 SENA, Universidade de Málaga, <b>Espanha</b>	X	X	X		X	X					X	X								X					
36 SENARIO—TIDE, Finlândia	X	X			X	X					X									X					
37 Siamo—University of Alcalá, <b>Espanha</b>	X	X		X	X	X	X				X	X	X	X						X*					
38 SIRIUS—University of Seville, <b>Espanha</b>	X				X															X					
39 Smart Alec—Stanford University, <b>USA</b>	X				X								X							X					

		Mapas				Modos de Operação											Avaliação								
		Topológico	Métrico	Greha de ocupação	Referências Naturais	Referências Artificiais	Navegação Autônoma	Evitar colisões	Seguir parede	Passar pela Porta	Retornar para Doca	Seguir trajetória Salva	Retornar mesma trajetória	Perseguir Objectivo	Seguir linha	Dirigir-se a um sitio	Three-point Turn	Bump and Backup	Modo Automático	Seleção de Modo Manual	ninquem não tripulado	em boa forma física	Deficientes físicos		
1	ACCoMo, Universidade Chiba, Chiba, <b>Japão</b>	X					X															X			
2	Automated-Guided Wheelchair—NEC Corporation, <b>Japão</b>												X										X		
3	Autonomous Wheelchair—Arizona State University, <b>USA</b>	X		X				X															X		
4	CALL Centre, Universidade of Edinburgh, <b>Escócia</b>											X	X											X	
5	CCPWNS—University of Notre Dame, <b>USA</b>			X						X												X	X		
6	CHARHM—Advanced Technologies Development Center, <b>Argélia</b>	X					X															X			
7	COACH—French Atomic Energy Comission, <b>França</b>							X	X																
8	CWA—National University of Singapore, <b>Singapore</b>										X											X			
9	CWA (Power)										X											X			
10	Deictic Wheelchair, Northeastern University, Boston, <b>USA</b>							X		X															
11	EASY, Universidade de Bremen, <b>Alemanha</b>			X	X		X	X		X															
12	FRIEND, Universidade de Bremen, <b>Alemanha</b>																								
13	Hephaestus—TRAC Labs, <b>USA</b>							X															X	X	
14	INCH—Yale University, <b>USA</b>							X															X		
15	INRO—FH Ravensburg-Weingarten, <b>Alemanha</b>	X					X				X														
16	Intelligent Wheelchair—University of Texas at Austin, <b>USA</b>	X	X				X															X			
17	Intelligent Wheelchair System—Osaka University, Japan (protótipo 1)	X						X	X			X				X							X		
18	Intelligent Wheelchair System (protótipo 2)							X	X			X					X						X	X	
19	IntellWheels, Universidade do Porto, FEUP, <b>Portugal</b>	X	X		X		X	X	X		X	X	X	X	X	X	X	X	X				X		
20	Luoson III—National Chung Cheng University, <b>Taiwan</b>											X												X	
21	MAid—RIAKP, <b>Alemanha</b>	X	X	X			X				X						X						X		
22	Mister Ed—IBM, <b>USA</b>							X	X	X		X					X					X			
23	Mr. HURI—Yonsei University, <b>Korea</b>							X															X		
24	NavChair—University of Michigan, <b>USA</b>		X					X	X	X							X							X	
25	NLPR Robotized Wheelchair—Chinese Academy of Sciences, <b>China</b>	X	X				X	X	X									X							
26	OMNI—University at Hagen, <b>Alemanha</b>							X	X	X	X	X	X	X										X	X
27	Orpheus—National Technical University of Athens, <b>Grécia</b>	X					X	X																X	
28	Phaeton—Northeastern University, <b>USA</b>							X			X														
29	RobChair—University of Coimbra, <b>Portugal</b>							X	X								X						X		
30	RoboChair, Universidade de Essex e CAS, <b>UK e China</b>	X	X				X	X	X																
31	Robotic Wheelchair—FORTH, <b>Grécia</b>							X			X						X						X		
32	Rolland—University of Bremen, <b>Germany</b> (protótipo 1)		X	X				X	X	X	X					X							X		
33	Rolland (protótipo 2)	X	X	X				X	X	X	X			X		X	X	X					X		
34	Rolland (protótipo 2 com laser)	X						X															X		
35	SENA, Universidade de Málaga, <b>Espanha</b>	X						X	X	X	X			X									X		
36	SENARIO—TIDE, Finlândia	X	X	X				X	X															X	
37	Siarno—University of Alcalá, <b>Espanha</b>	X			0		X	X	X	X	X		X				X						X		
38	SIRIUS—University of Seville, <b>Espanha</b>							X			X												X		
39	Smart Alec—Stanford University, <b>USA</b>							X	X			X					X						X	X	

Cadeira de Rodas Inteligente		Sensores							Entradas				Factor-Forma		Controlo										
		Sonar	Infravermelhos	Scanner laser	Laser stripe r.	Visão Computacional	colisão/contacto	Odometria	Seguir linha	Bussola	GPS	Giroscópio	Voz	EOG	Force-Feedback joystick	Orientação pela face/cabeça	Reconhecimento de gestos	Deictic	EMG	Robo Modificado	CR Motorizada Modificada	Add-On Unit	Omnidirecional	Sub assumption	Redes Neurais
41	CALL Center, UK	X				X	X													X					
42	Chinese University of Hong Kong, China	X																			X				X
43	Kanazawa University, Japão					X																			
44	Toyohashi University, Japão	X	X			X						X								X	X				
45	University of Ancona, Italia	X				X														X					
46	University of Plymouth, UK	X			X	X				X										X					X
47	University of Portsmouth, UK	X																		X*					
48	SmartChair—University of Pennsylvania, UK		X	X		X	X							X					X						X
49	SPAM—AT Sciences, USA	X	X			X	X													X*					
50	SWCS—AT Sciences, USA	X	X			X														X					
51	TAO—Applied AI Systems, Inc., Canadá	X				X	X													X*					X
52	TAO-2	X			X	X														X*					X
53	TetraNauta—University of Seville, Espanha	X	X			X	X				X									X*					
54	TinMan—KIPR, USA	X	X			X	X	X												X					
55	VAHM—Universit�e de Metz, Frana (modified wheelchair)	X				X														X				X	X
56	VAHM (robot base)	X	X			X	X													X				X	X
57	Voice-cum-Auto Steer Wheelchair—CEERI, �ndia	X				X	X				X									X					
58	WAD Project—Bochum University, Alemanha	X				X														X					
59	Watson—NAIST, Japão		X	X								X								X					
60	Wheelesely—Massachusetts Institute of Technology, UK	X	X		X	X	X				X									X					

Cadeira de Rodas Inteligente		Mapas				Modos de Operao										Avaliao									
		Topologico	Metrico	Grelha de ocupao	Referencias Naturais	Referencias Artificiais	Navegao Autonomia	Evitar colises	Seguir parede	Passar pela Porta	Retornar para Doca	Seguir trajetria Salva	Retornar mesma trajetria	Perseguir Objectivo	Seguir linha	Dirigir-se a um s�to	Three-point Turn	Bump and Backup	Modo Autom�tico	Seleo de Modo Manual	ninguem	no tripulado	em boa forma f�sica	Deficientes f�sicos	
41	CALL Center, UK													X											X
42	Chinese University of Hong Kong, China										X							X							X
43	Kanazawa University, Japão	X			X	X																			X
44	Toyohashi University, Japão		X				X																		X
45	University of Ancona, Italia		X				X																		X
46	University of Plymouth, UK				0					X															X
47	University of Portsmouth, UK		X						X																X
48	SmartChair—University of Pennsylvania, UK						X	X	X	X		X	X		X			X							X
49	SPAM—AT Sciences, USA							X																	X
50	SWCS—AT Sciences, USA						X																		X
51	TAO—Applied AI Systems, Inc., Canadá	X		X	X		X	X	X	X								X							X
52	TAO-2	X			0		X	X	X	X								X							X
53	TetraNauta—University of Seville, Espanha	X					X																		X
54	TinMan—KIPR, USA	X					X	X											X						X
55	VAHM—Universit�e de Metz, Frana (modified wheelchair)	X	X				X	X	X										X						X
56	VAHM (robot base)	X	X		X		X	X	X										X						X
57	Voice-cum-Auto Steer Wheelchair—CEERI, �ndia	X					X						X												
58	WAD Project—Bochum University, Alemanha						X	X															X		
59	Watson—NAIST, Japão	X																							X
60	Wheelesely—Massachusetts Institute of Technology, UK						X	X											X						X

Apesar de uma longa hist ria de investigao em cadeiras de rodas inteligentes, poucas cadeiras de rodas com alguma intelig ncia so actualmente comercializadas. Pode-se verificar na Tabela 20 algumas cadeiras comerciais que possuem algum grau de intelig ncia. Nesta

tabela encontram-se informações como o distribuidor, preço, sensores utilizados e modo de operação.

Tabela 20: Cadeiras de rodas inteligentes comerciais

Especificação	Smart Wheelchair	Smart Box	TAO-7	WheelChair Pathfinder	Robotic Chariot
Distribuidor	Smile Rehab, Ltd.	Smile Rehab, Ltd.	Applied Systems, Inc.	Nurion Industries	ActivMedia
Preço	\$14200	\$5000	\$37400	\$4500	\$36490
Sensor	sonar, sensor de colisão, detecção de linha	sensor de colisão, detecção de linha	sonar, sensor de distância por infravermelho, visão computacional	sonar, sensor de distância por laser	sensor de distância por infravermelho, encoders, sensor de colisão, GPS, visão computacional
Modos de Operação	se colisão parar, se colisão voltar, se colisão virar, seguir linha	se colisão parar, se colisão voltar, se colisão virar, seguir linha	vaguear aleatoriamente, navegação compartilhada, navegação autónoma	Vibrar quando encontrar obstáculo ou declive acentuado, Não controla a cadeira.	vaguear aleatoriamente, navegação compartilhada, navegação autónoma
Inclui Cadeira	sim	não	sim	não	sim
Utilizador	Crianças	Crianças e adultos	Investigadores	Crianças ou adultos com deficiência visual	Investigadores

Pode-se observar, nesta relação de CRI comercial, que o custo do equipamento é muito elevado e apresentam poucas habilidades desejáveis.

### 3.3 Características gerais dos projectos

De entre os diversos projectos sobre cadeiras de rodas inteligentes pode-se observar que a grande maioria destes projectos preocupa-se com a adaptabilidade ao indivíduo e com os requisitos de segurança.

Para obter a aceitação dos utilizadores, as cadeiras de rodas inteligentes devem adaptar-se às necessidades de cada indivíduo (Mazo, 2001), (Bergasa et al., 1999), (Ju et al., 2009). Tendo-se em conta que o objectivo destes projectos são o apoio a pessoas deficientes e idosos, o foco deve ser em como complementar as habilidades destes utilizadores. Desta forma, as investigações e os desenvolvimentos não se concentram em criar apenas sistemas completamente autónomos, mas sim, cadeiras de rodas inteligentes que operem de forma semi-autónoma. Portanto, as CRI devem ter capacidade de executar algumas tarefas de forma autónoma, mas outras tarefas terão que contar com as habilidades e experiência do utilizador. Por outras palavras, uma CRI é um sistema que deve ser muito interactivo, onde o controlo do sistema é efectuado em conjunto entre o paciente e o software da CRI. Por esta razão o projecto de uma Interface Homem-Máquina (IHM) é um dos principais pontos a serem abordados no desenvolvimento de uma CRI. Para tornar mais flexível a interacção, alguns projectos apresentam soluções diferentes. Como por exemplo, alguns projectos voltados para pessoas tetraplégicas (Figura 38), estão a utilizar o reconhecimento das expressões faciais (Ju et al., 2009) e movimentos com a cabeça (Jia et al., 2007) (Adachi et al., 1998) (Ng & De Silva, 2001) para guiar a cadeira de rodas. O projecto SIAMO apresenta cinco maneiras diferentes de interacção, onde pode-se destacar a eletroculografia (EOG). EOG é uma técnica

que mede o potencial de repouso entre a córnea e a retina, uma das principais aplicações é no registo dos movimentos oculares. Desta forma este projecto propõe controlar a cadeira de rodas inteligente com o movimento dos olhos. Outra forma apresentada para controlar as CRI pode ser com o “pensamento”. Esta tecnologia utiliza sensores que captam as ondas electromagnéticas do cérebro (Lakany, 2005) (Rebsamen et al., 2007) (Iturrate et al., 2009).



a) Projecto SIAMO (Mazo, 2001)



b) Projecto BCW (Rebsamen et al., 2007)



c) BCI a controlar CRI (Iturrate et al., 2009)

Figura 38: CRI que são controladas sem usar as mãos

Os robôs de serviço geral e robôs de reabilitação devem ser considerados como sistemas críticos de segurança, pois o funcionamento incorrecto deste tipo de equipamentos, em contacto directo com os seres humanos, pode causar acidentes graves aos utilizadores. No caso das CRI o correcto funcionamento é ainda mais crítico, pois elas transportam pessoas que normalmente são muito dependentes do comportamento correcto do sistema. Por exemplo, se um deficiente está a operar uma CRI e dá instruções de ir a casa de banho, a execução confiável do comando deve ser considerado como um sistema crítico pois as falhas não podem ser são aceites. Poucos projectos se preocupam com a questão de como projectar uma cadeira de rodas e ter em conta os aspectos de segurança crítica. O projecto Rolland é um dos poucos que busca aplicar métodos formais como técnicas de análise de risco (Lankenau et al., 1998) (Röfer & Lankenau, 2000) e modelos de verificação (para definir requisitos de segurança do sistema e provar que estes requisitos foram atendidos) e aplicar no sistema de controlo compartilhado (Borgolte et al., 1998).

### 3.3.1 Funções Presentes nos Projectos de CRI

Um número grande de diferentes funções são encontradas nos diversos projectos apresentados na bibliografia, o número de funções é tão grande quanto a variedade de deficiências. Supõem-se que as CRI devam trabalhar de forma segura e robusta num ambiente comum (com nenhuma ou pouca alteração do ambiente para melhorar o desempenho da mesma) operadas por pessoas sem conhecimentos técnicos. As CRI devem também ser adaptáveis às diferentes deficiências do utilizador. Um outro aspecto relaciona-se com a

configuração e pequenas manutenções, desta forma esta interface deve ser simples e intuitiva de modo a possibilitar à equipa de reabilitação a realização da configuração e pequenas manutenções. A seguir são apresentadas algumas funções relevantes, que estão presentes em diversos projectos analisados.

### 3.3.1.1 Detecção de Obstáculos

A qualidade da detecção de obstáculos está associada ao tipo, qualidade e quantidade de sensores utilizados, mas por outro lado, a interpretação, representação e processamento dos dados fornecido também tem influência directa na qualidade da detecção.

Praticamente todos os projectos utilizam codificadores incrementais (*encoders*), acoplados ao eixo das rodas, para processar a velocidade e direcção do movimento. Desta forma é possível rastrear o deslocamento da CRI. O emprego dos sensores de proximidade variam significativamente em número, tipo e modelo nestes projectos. O tipo de sensor de proximidade mais comum é por ultra-sons. Estes sensores são geralmente montados em anel ao redor da CRI (por exemplo, SENARIO (Katevas et al., 1997) e Rolland (Röfer & Lankenau, 2000))(Figura 39). No entanto, alguns projectos só cobrem a parte frontal da CRI utilizando estes sensores (NavChair, INRO(Schilling et al., 1998)). Outro sensor de proximidade também muito utilizado é o sensor de infra-vermelhos (RobChair, Wheellesly). A aplicação de scanners laser já é relativamente rara em projectos de CRI devido ao seu custo elevado. SENARIO e MAid (Prassler et al., 2001) são exemplos de projectos que utilizam scanner laser.



a) Projecto Rolland



b) SENARIO



c) CRI MAid

Figura 39: Exemplo de projectos de CRI que utilizam sensores proximidade

Um outro tipo de sensor que já teve sua relevância, mas que recentemente, teve sua importância diminuída foi o do tipo “*bumper*” desenvolvido a partir de sensores de toque, o qual retornam um sinal digital que informa se a CRI está em contacto com um obstáculo. Entre os projectos que utilizam este tipo de sensor, destacam-se: Deictic (Crisman, 1998), Wheellesly (Yanco, 1998) e TAO (Gomi & Griffith, 1998). Actualmente este tipo de sensores não é muito utilizado dado que os sensores de proximidade permitem detectar a aproximação

exagerada e colisão com objectos. As câmaras estéreo podem também ser utilizadas como sensor de proximidade, usadas para estimar as distâncias dos objectos que se encontram ao redor da CRI (TAO, Deictic). As Câmaras também podem ser utilizadas para detectar buracos ou degraus (INRO, SENARIO).

A quase totalidade dos projectos utiliza algum método para evitar colisões. O princípio básico é permitir que o software de controlo, com base nas informações dos sensores de proximidade, pare a CRI antes da colisão e simultaneamente procure manter uma distância segura aos objectos. O projecto TAO utiliza uma acção directa do sensor e as informações dos sensores não são armazenadas. A maioria dos outros projectos mantém armazenado um mapa local de obstáculos criado a partir das informações obtidas pelos sensores (NavChair, SENARIO, Rolland).

### 3.3.1.2 Desvio de Obstáculos

Uma das características mais importante das CRI é transportar os pacientes de forma segura. Neste sentido uma CRI deve possuir a habilidade de desvio de obstáculos robusta e confiável. Entretanto, na bibliografia encontram-se várias interpretações para o termo “desvio de obstáculos”. Uma abordagem é o desvio de obstáculo de forma puramente reactiva, como por exemplo no projecto TAO. Neste caso os motores são accionados directamente pela leitura dos sensores. Caso o utilizador não esteja de acordo com a decisão do sistema ele pode sobre enviar um comando através de um movimento contrário com o joystick.

O desvio de obstáculos com base em mapas é uma abordagem comum a muitos projectos. Neste caso utiliza-se o armazenamento de informações recentes recebidas dos sensores para fazer uma detecção de objectos de forma mais confiável. Outra abordagem, conhecida como “Vector Field Histogram” (apresentada no Projecto NavChair), encontra um ajuste entre a direcção do comando do utilizador e a melhor direcção livre de colisão.

O problema do controlo compartilhado está em fundir os comandos do utilizador com os comandos automáticos de desvio de obstáculos. Os projectos NavChair e Rolland utilizam uma abordagem de desvio de obstáculo que considera a intenção do utilizador como a tendência de direcção de deslocamento. Por exemplo no projecto Rolland a direcção indicada através do joystick é indicada no mapa local de obstáculos com a finalidade auxiliar na decisão de qual lado do obstáculo a CRI deverá passar.

## 3.3.2 Comportamento Baseado em Habilidades

Através de uma interface mais sofisticada é possível dar instruções significativamente mais abstractas, ou seja, comandos de alto nível que disparem acções complexas, como por exemplo um comando que gere a acção de ir à casa de banho. Diversos projectos apresentam várias habilidades locais de navegação, como seguimento de parede, rastreamento de objectos, rodar sobre seu eixo e passagem por portas, entre outras. No projecto Deictic é

apresentada uma habilidade de seguir algum objecto próximo da CRI que foi indicado pelo utilizador através de um comando abstracto. Para rastrear o tal objecto (que foi indicado com o auxílio de uma imagem de vídeo) é utilizado um sistema estéreo de visão.

### 3.3.3 Navegação

Técnicas de auto-localização são um requisito básico para a navegação das CRI. Um grande desafio para os grupos de investigação nesta área é desenvolver métodos de auto-localização que trabalhem em ambientes que não sejam necessariamente conhecido anteriormente.

Com o objectivo de facilitar a adaptação da CRI a ambientes diferentes, muitos projectos utilizam a abordagem de aprendizagem computacional. A CRI deve ser treinada para operar num certo ambiente pela equipa de apoio. Após isso, ela deverá estar apta a realizar tarefas de navegação naquele ambiente. Durante o processo de treino a CRI constrói um mapa do ambiente que, posteriormente, será utilizado com as técnicas de auto-localização. Entre os projectos apresentados existem alguns que empregam mapas topológicos (TAO) e outros que utilizam uma combinação dos mapas topológicos e métricos (Rolland, SENARIO). Em ambientes externos, a navegação faz uso de sistema de posicionamento global por satélites – GPS. Como exemplo de projectos que se utilizam da técnica de navegação com GPS pode-se citar: INRO, NavChair e TAO.

### 3.3.4 Interface Homem-máquina

Um ponto importante nos projectos de CRI é o desenvolvimento de Interfaces Homem-Máquina (IHM), estas devem permitir que o utilizador comande a CRI num nível significativamente mais abstracto comparado com as cadeiras de rodas eléctricas tradicionais. Um exemplo é uma interface que possibilite dar instruções à CRI através do reconhecimento da face, neste caso o utilizador pode determinar um comando de alto nível através de uma sequência de expressões.

Um grupo grande de projectos utiliza joysticks normais como dispositivo de entrada e nenhum dispositivo especial de saída, com excepção de simples ecrãs. Entretanto, outros projectos (RobChair, SIAMO) utilizam sistema de reconhecimento da fala para habilitar o utilizador a executar comandos através da voz. Em Wheelesley, o utilizador controla a CRI com comandos de alto nível através de uma interface gráfica no ecrã de um computador portátil (Yanco, 1998). O projecto SIAMO utiliza outros dispositivos de entrada como o controlo através de um dispositivo de sopro e o uso de uma câmara CCD para permitir receber instruções através da detecção da face do utilizador (Bergasa et al., 1999). Como apresentado no início desta secção, existem outros projectos que utilizam também a detecção de expressões faciais como IHM (Jia et al., 2007)(Pei Chi & de Silva, 2001)(Adachi et al., 1998),

assim como alguns que apresentam um método de controlo que utiliza o “pensamento” para controlar a CRI (Lakany, 2005) (Rebsamen et al., 2007).

A plataforma experimental do projecto FRIEND é equipada com um computador e um manipulador robótico. O tópico principal do projecto é o controlo do manipulador e a sua interface homem-máquina. Tanto o projecto INRO quando RobChair empregam sinais de rádio para realizar a comunicação da cadeira de rodas com uma estação remota com o propósito da teleoperação.

### **3.3.5 Sistema Multi-agente e Cadeira de Rodas Inteligente**

Poucos projectos consideram que um grupo de CRI e suas interações podem resultar em comportamento de alto nível, e os poucos projectos que utilizam o paradigma de SMA apresentam concepções bem distintas nas suas aplicações. Entretanto, pode-se questionar algumas destas implementações pelo facto de seus sistemas apresentarem mais características de um software modular do que propriamente um SMA.

Galindo (Galindo et al., 2006) propõe uma arquitectura de controlo baseado no paradigma dos sistemas multi-agente para robôs de serviço e aplica-a a uma cadeira de rodas. Este sistema de controlo é composto por vários agentes tais como: agente agenda; agente planeador; agente sentinela; agente sequenciador de plano; agente modelador do mundo. Este sistema promete facilitar também a interação homem-máquina para interagir não só no controlo manual mas também no planeamento. Neste projecto é utilizada comunicação com base na linguagem ACL e aprendizagem para determinar qual é a acção mais adequada de acordo com o ambiente e as características do robô. De forma semelhante, Kuo (Kuo et al., 2003) também propõe um controlo para cadeiras de rodas utilizando o paradigma de SMA. Este sistema consiste num SMA com funções de planeamento de trajectória, navegação, desvio de obstáculos e outras funções. No entanto, neste trabalho, foram apresentados apenas resultados de simulação. Georgios (Lidoris & Buss, 2006) apresenta um SMA com uma arquitectura modular para plataforma de robôs móveis desenvolvida em JADE, sendo que o foco da sua aplicação é as cadeiras de rodas inteligentes. Neste trabalho ele salienta que o domínio da CRI deve ser naturalmente considerado como sistema crítico de segurança, mas os resultados apresentados são apenas de simulação da interação dos agentes para provar o conceito.

ACCoMo (Hamagami & Hirata, 2004) (Hamagami & Hirata, 2005) é um dos poucos projectos de CRI apresentado na literatura que aborda a CRI como um agente que pode adquirir um comportamento autónomo, cooperativo e colaborativo. O comportamento cooperativo emerge da interação com outras CRI de forma dinâmica, esta função é utilizada para passar em lugares apertados onde as cadeiras comunicam sua posição e através de uma função de satisfação, resulta numa acção de retrocesso de uma das cadeiras, para permitir que

a outra passe. Neste projecto não é utilizado nenhum planeamento prévio ou negociação para evitar esta situação. Embora estas características (comportamento autónomo, comunicação, colaboração, cooperação) sejam comuns a sistemas multi-agente este projecto não aborda a solução desta forma, talvez por não formalizar uma plataforma de comunicação e nem uma arquitectura de sistema multi-agente.

### 3.4 Sumário

Baseado nos principais requisitos para tornar uma cadeira de rodas motorizada numa CRI apresentados no capítulo 2, neste capítulo foram apresentadas as diferentes abordagens adoptadas pelos principais grupos de investigação na área. Também foi apresentada uma tabela estendida reunindo os principais projectos de CRI, contendo um resumo dos tipos de sensores utilizados e o foco de cada trabalho de investigação na área.

Embora existam imensos projectos na área das cadeiras de rodas inteligentes (e até alguns modelos comerciais de cadeiras de rodas com algumas características de CRI) na utilização das novas tecnologias de informática e robótica no apoio à deficiência profunda, verifica-se ainda a inexistência de cadeiras de rodas com reais capacidades de planeamento inteligente de acções e navegação autónoma. Torna-se assim cada vez mais necessário dotar cadeiras de rodas inteligentes de mecanismos que permitam realizar de forma semi-autónoma a execução dos desejos dos seus utilizadores, através de metodologias como: linguagens de comando de alto nível; navegação semi-automática; planeamento inteligente de acções; comunicação com outros dispositivos inteligentes; aplicação do paradigma SMA nestes ambientes; etc.

Da análise realizada aos principais projectos mundiais de investigação na área das CRI, pode-se verificar a inexistência de um projecto que seja capaz de adaptar uma cadeira de rodas eléctrica transformando-a numa verdadeira CRI. Para tal seria necessário realizar a adaptação com o princípio de causar o menor impacto sobre o aspecto original da cadeira de rodas, com modularidade de hardware que permita sua utilização em diferentes modelos com mínimas modificações e com uma plataforma de software que permita sua interacção com ambientes simulados e realidade aumentada. Este trabalho procura preencher algumas destas lacunas, neste sentido foca-se na criação de uma ferramenta que ajude a desenvolver equipamentos que atendam estas necessidades. O Capítulo 4 apresenta a arquitectura de hardware e software proposta para uma plataforma de desenvolvimento de CRI.

## Capítulo 4

# Conceito e Implementação da Plataforma IntellWheels

Este Capítulo apresenta o conceito, projecto e implementação da Plataforma de Desenvolvimento de Cadeira de Rodas Inteligente IntellWheels desenvolvida no âmbito deste trabalho. Como o objectivo deste trabalho consistiu em projectar uma ferramenta flexível, que facilite o desenvolvimento de novas tecnologias direccionadas para às Cadeiras de Rodas Inteligentes (CRI), pode-se considerar esta ferramenta como um Framework completo para permitir o desenvolvimento de CRI. O sistema foi projectado com características modulares e tem por base o paradigma dos sistemas multi-agente.

### 4.1 Conceito e Projecto

O projecto, *IntellWheels*, tem como objectivo principal a criação de uma plataforma de desenvolvimento para cadeiras de rodas inteligentes (Braga et al., 2008) (Braga et al., 2009), intitulada de *Plataforma IntellWheels (PIW)*. Esta tese tem foco principalmente na investigação e concepção de uma plataforma multi-agente, que permita a fácil integração de diferentes sensores, actuadores, dispositivos de interacção estendida com o utilizador, metodologias de navegação, técnicas de planeamento inteligente e metodologias de cooperação, para resolver problemas associados a cadeiras de rodas inteligentes. Espera-se que esta plataforma facilite o desenvolvimento e teste de novas metodologias e técnicas relativas às CRI. Dentro deste conceito, procurar-se-á criar um conjunto de ferramentas (software e hardware) que possa ser facilmente integrado em qualquer cadeira de rodas eléctrica, disponível comercialmente, com pequenas modificações. O projecto leva ainda em consideração questões como: manutenção de baixo custo e não se degradar o conforto e ergonomia da cadeira de rodas original.

Acredita-se que esta plataforma possa vir a facilitar a criação de novas tecnologias para as CRI e estas resultem em reais capacidades inovadoras e implementáveis. Pode-se citar, como capacidades relevantes, nas CRI: planeamento inteligente; navegação autónoma; navegação semi-autónoma através de comandos do utilizador (paciente) com uma linguagem de alto nível. Estas capacidades podem ser atingidas através de um sistema de controlo avançado, que vai desde um simples controlo compartilhado (no qual o sistema consegue evitar colisões quando o utilizador está a controlar a cadeira de forma manual) até o controlo de tarefas mais complexas com comandos de alto nível (neste caso comandos gerados através de uma interface multimodal, planeamento automático, condução autónoma e mapeamento do ambiente).

O sistema foi concebido com seis módulos básicos, ilustrados na Figura 40, nomeadamente: planeamento; interface; simulação; comunicação; navegação e hardware.

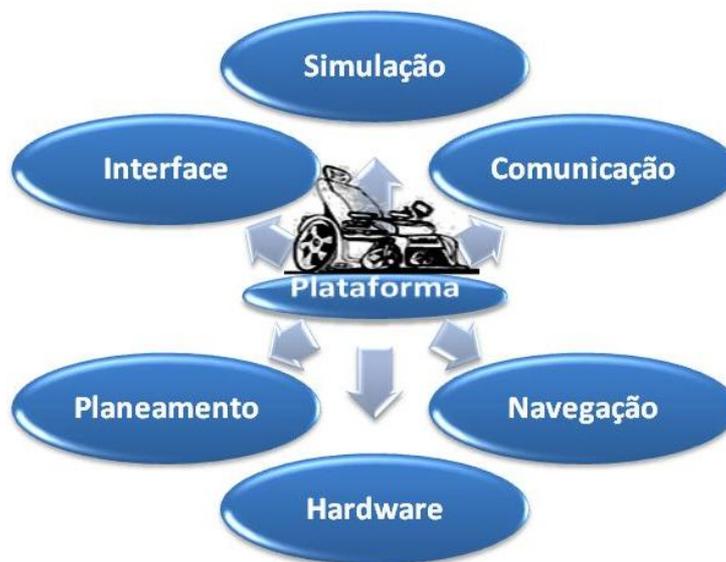


Figura 40: Módulos básicos da plataforma IntellWheels

A Figura 41 representa a arquitectura de software proposta neste trabalho. Esta arquitectura utiliza o paradigma de Sistemas Multi-Agente (SMA). A opção pelo uso de SMA deveu-se à sua flexibilidade, com o propósito de facilitar a adição de novos módulos e a interacção entre as cadeiras, entre outros dispositivos inteligentes. Nesta arquitectura podem-se observar quatro agentes básicos, que compõem uma única cadeira:

- **Agente Interface:** responsável pela interacção do utilizador com a CRI;
- **Agente de Inteligência:** responsável pelas acções de planeamento da CRI;
- **Agente de Percepção:** responsável por ler os sensores adequados em cada momento, localização e mapeamento do ambiente;
- **Agente de Controlo:** responsável pelas acções de controlo das acções básicas, controlo das rodas, desvio de obstáculos.

Estes agentes são heterogêneos e podem colaborar entre si ou com outros agentes de outra cadeira. Pode-se observar na arquitectura, que os agentes, descritos anteriormente, podem em conjunto controlar tanto uma cadeira real como uma cadeira simulada.

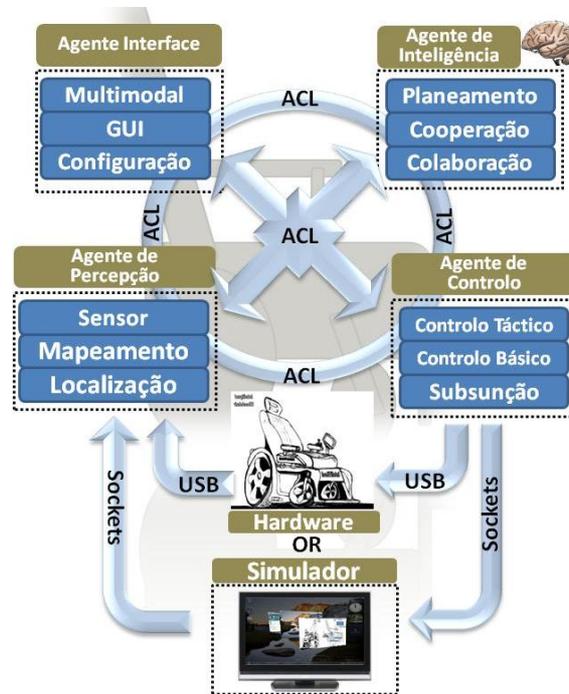


Figura 41: Arquitectura de software da plataforma IntellWheels

O ambiente de operação da CRI pode ser visto como um SMA composto por vários SMA, por outras palavras, este ambiente é subdividido em micro e macro agentes (Figura 42). Na visão micro, uma cadeira é composta por micro agentes e na visão macro a cadeira em si é um macro agente, assim como os demais agentes existentes no sistema (por exemplo, agente porta, agente de serviços, agente médico, etc.).

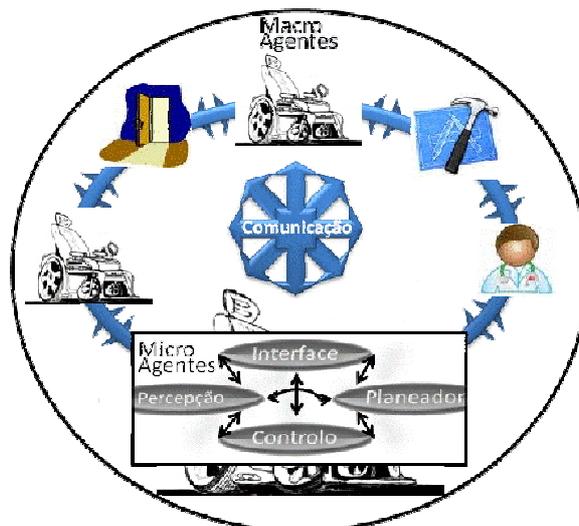


Figura 42: Micro e macro agentes

Um dos recursos mais inovadores da plataforma é o uso do conceito de realidade mista, onde é possível que CRI reais possam interagir com objectos e CRI virtuais. Essa possibilidade de interacção permite realizar testes de alta complexidade com número considerável de dispositivos e cadeiras de rodas, o que vem a resultar na redução de custos associados ao desenvolvimento de novas tecnologias, uma vez que não há necessidade de se construir um grande número de CRI reais com infra-estruturas complexas e caras. Com este recurso é possível analisar e estudar a interacção entre um grande número de CRI, como por exemplo, configurar um ambiente com uma cadeira real e várias outras virtuais. A interacção entre os agentes neste cenário resultaria na mesma interacção que num cenário só com cadeiras reais, pois os agentes envolvidos são os mesmos, a única diferença básica é o corpo do robô.

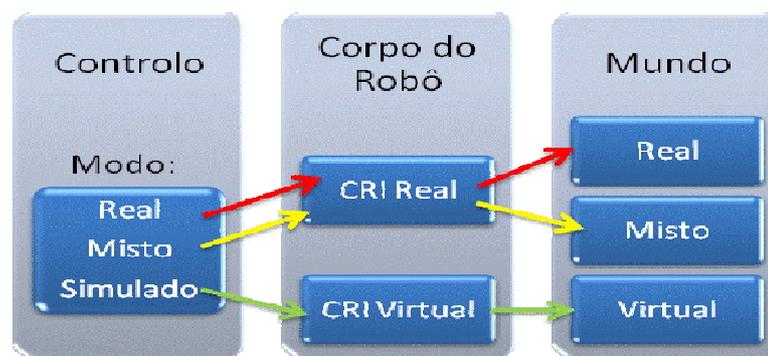


Figura 43: Modos de operação

Como ilustrado na Figura 43, esta plataforma permite que o sistema trabalhe em modo real (a CRI possui um corpo real), simulado (o corpo da cadeira é virtual) ou realidade mista (cadeira de rodas real com a percepção de objectos reais e virtuais). No modo real é necessário conectar o sistema (software) ao hardware real, no simulado o sistema é conectado ao simulador e na realidade mista, o sistema é conectado a ambos (hardware e simulador).

## 4.2 Hardware da Plataforma

Para que uma cadeira de rodas possa ser considerada inteligente é necessário que ela seja capaz de perceber o ambiente ao seu redor, planejar suas próximas acções, reagir às mudanças do ambiente e apresentar uma interface expandida, para uma melhor interacção com o utilizador. A interface das CRI deve permitir enviar comandos e indicar desejos do utilizador de forma mais flexível e adaptável para cada utilizador. Para atender tais características foi desenvolvido um kit de hardware projectado para ser flexível e de fácil adaptação às cadeiras de rodas comerciais. Esse conjunto de dispositivos de hardware pode ser classificado em três blocos, de acordo com a sua funcionalidade: dispositivos de interacção com o utilizador (*joystick* tradicional e USB, gestos com a cabeça, teclado, expressão facial e microfone para captar comandos de voz); sensores (sonar, *encoder*, *webcam*, infravermelho); outros

dispositivos (placa de controlo e aquisição de dados, modulo de potência, computador portátil).

A arquitectura de hardware desenvolvida (Figura 44) apresenta um conjunto de dispositivos básicos que podem ser facilmente expandidos. O desenvolvimento desta arquitectura procurou atender às capacidades desejadas, mas também manter o baixo custo, o conforto e a ergonomia da cadeira de rodas original, quando estes equipamentos forem adicionados à cadeira de rodas.

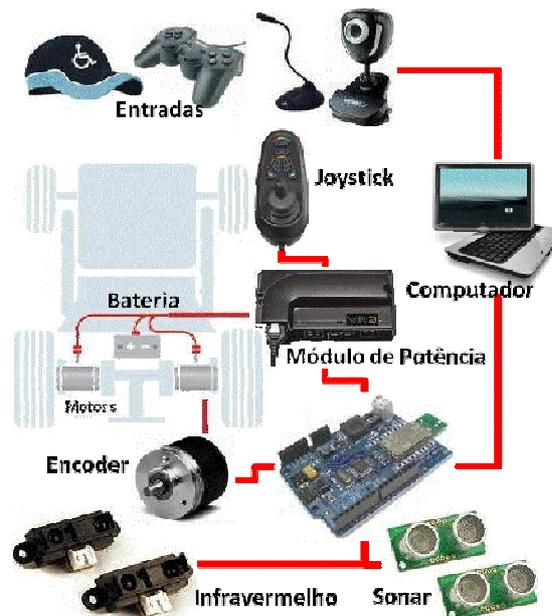


Figura 44: Arquitectura de Hardware do projecto IntellWheels

### *Dispositivos de Entrada*

Diversos tipos de dispositivos de entrada foram utilizados neste projecto para permitir que pessoas com diferentes tipos de deficiência e habilidades sejam capazes de guiar as CRI. O objectivo é permitir que o utilizador possa escolher o dispositivo de entrada mais confortável e seguro para controlar a cadeira. Além disto, estas múltiplas entradas de interacção com a CRI podem ser integradas com um sistema de controlo de decisão responsável por habilitar ou desabilitar algum tipo de entrada, caso seja observado algum conflito, ruído ou alguma situação de perigo. Por exemplo, num ambiente muito ruidoso, onde o reconhecimento de voz pode falhar, este tipo de entrada pode ser temporariamente desabilitada. Estes dispositivos vão desde joysticks tradicionais até comandos pelas expressões faciais, incluindo nomeadamente:

- **Joystick tradicional:** este tipo de dispositivo de entrada, presente nas cadeiras de rodas motorizadas comerciais, é uma forma robusta e simples de controlar a cadeira de rodas. No entanto, pessoas paraplégicas ou com paralisia cerebral não possuem condições ou

habilidades para controlar a cadeira de rodas motorizada com este dispositivo. Este dispositivo está presente na arquitectura de hardware devido sua simplicidade e robustez.

- **Joystick USB:** os joysticks USB são um pouco mais sofisticados que os joysticks originais. Este tipo de joystick é muito utilizado em videojogos onde o seu maior atractivo é a quantidade de botões e eixos que podem ser configurados para facilitar a navegação. Estes também podem apresentar a mesma desvantagem, dos joysticks tradicionais, se forem utilizados nas mesmas condições, por exemplo o controlo da cadeira com o “stick” analógico utilizado por uma pessoa paraplégica;

- **Gestos com a cabeça:** este dispositivo permite uma interacção amigável entre o utilizador e a cadeira de rodas através de gestos com a cabeça, desta forma é possível guiar a cadeira com gestos da cabeça. Para capturar os movimentos são utilizados acelerómetros instalados num boné e os dados são enviados para o computador através de comunicação *bluetooth*;

- **Teclado e ecrã de toque:** estes dispositivos são utilizados para configurar/parametrizar a CRI, como uma alternativa ao controlo manual;

- **Expressões Faciais:** ao fazer uso de uma *webcam* comum, este dispositivo reconhece algumas expressões faciais (tais como piscar um olho, abrir a boca ou franzir a sobancelha) que podem ser utilizadas, desde um comando básico (como por exemplo: seguir em frente, virar à direita, virar à esquerda, parar e etc.) até comandos de alto nível (ir para a casa de banho, ir para enfermaria e etc.).

- **Reconhecimento de fala:** este sistema utiliza um microfone comum para capturar o som da fala, que é interpretada por um motor de reconhecimento de voz. Para interpretar o comando por voz é possível utilizar um software comercial (IBM, 2009) ou a Interface para Programação de Aplicativos (*Application Program Interface* - API) desenvolvida pela Microsoft, designada *Speech Application Programming Interface* (SAPI), que permite o uso do reconhecimento da fala e o sintetizador de fala nas aplicações Windows. A SAPI é uma interface distribuída de forma gratuita para ser usada em qualquer aplicação do Windows. Muitas versões de reconhecimento de voz e de sintetizadores são distribuídas gratuitamente, embora nem todas. Inicialmente utilizou-se IBM via voice e posteriormente substituiu-se por SAPI, o que permitiu deixar esta funcionalidade integrada dentro do agente interface.

### **Sensores**

Para compor o conjunto do hardware e dotar a cadeira de rodas com capacidade de evitar obstáculos, seguir paredes, mapear ambiente, perceber buracos e desníveis no solo, foi projectada uma barra em forma de U, contendo um conjunto de oito sensores de ultra-sons e doze sensores infravermelhos. Neste conjunto também foram incluídos dois *encoders* (transdutores geradores de Impulsos), acoplados às rodas (para fornecer as informações que permitem medir a distância, velocidade e posição), assim como uma *webcam* para reconhecer

marcos artificiais, com a finalidade de refinar a odometria. É importante lembrar que o baixo custo e a manutenção da ergonomia e do conforto original da cadeira de rodas comercial foram considerados como requisitos do projecto. Deste modo foi excluída a possibilidade de utilização de sensores de elevado custo tais como laser *range-finders* ou câmaras de vídeo 3D. Foi também excluída a possibilidade de colocação de equipamentos em zonas da CRI que perturbem a sua normal utilização pelo paciente.

### ***Outros dispositivos de hardware***

Para complementar a plataforma de hardware foram necessários outros equipamentos mais genéricos, nomeadamente:

- Placa de controlo e aquisição de dados: Esta placa é utilizada para colectar informações dos sensores e enviar a referência de controlo dos motores para o módulo de potência. Esta placa (ArduinoMega<sup>12</sup> (Pardue, 2010)) é conectada ao computador da plataforma através de uma conexão USB.
- Módulo de potência: Converte o sinal de controlo em potência para os motores e oferece protecções de sobre corrente (módulo comercial produzido pela PG Drives Technology modelo VR2).
- Computador Portátil: Para executar o software da plataforma é usado um computador portátil (HP Pavilion tx1270EP, AMD Turion 64 X2 TI60), entretanto, outros computadores equivalentes ou de capacidade superior podem ser utilizados.

## **4.3 Simulador de Cadeira de Rodas**

Este módulo do sistema nomeado de *Simulador IntellWheels*, permite criar um mundo virtual onde é possível simular o ambiente de um recinto fechado (como por exemplo um pavimento de um hospital), assim como cadeiras de rodas e objectos genéricos (mesas, portas entre outros objectos). Os objectivos deste simulador são essencialmente dar suporte aos testes dos algoritmos, analisar e testar os módulos da plataforma e treinar os utilizadores da CRI de forma segura num ambiente simulado. Outro objectivo do simulador está relacionado com a realização de testes com um elevado número de cadeiras de rodas inteligentes, o que seria impossível com CRI reais devido ao seu custo e complexidade. Considerando que a realização de testes num ambiente real implica que a cada pequena modificação (nos algoritmos ou hardware) se torna muito dispendiosa em termos de tempo, dinheiro, mesmo sem contar com as situações de risco que o utilizador ou a CRI podem vir a serem submetidos

---

<sup>12</sup> Arduíno Mega é uma plataforma de hardware livre, projectada com um microcontrolador baseado no ATmega1280 com suporte a várias entradas/saídas. Ver <http://arduino.cc/>

nestes testes. Submeter a CRI a testes exaustivos em ambientes simulados, antes dos testes reais, é desejável e de grande importância.

O simulador IntellWheels é uma adaptação de um simulador com código aberto (*open-source*) chamado Ciber-rato (Lau et al., 2002). Este simulador foi desenvolvido na Universidade de Aveiro e é utilizado numa modalidade Ciber-rato do concurso Micro-rato destinada a concorrentes que apenas estão interessados em envolver-se no desenvolvimento de algoritmos de controlo de robôs móveis e autónomos, sem a preocupação com os problemas associados ao projecto, montagem e testes de robôs reais. O Simulador Ciber-rato cria um ambiente simulado contendo um labirinto e modeliza todos os componentes de hardware dos robôs. Os robôs simulados são controlados por agentes de software conectados ao simulador.

O simulador Ciber-rato apresenta várias características úteis para simulação de uma cadeira de rodas, tais como a simulação de ambientes diferentes, facilidade em configurar novos ambientes, robôs diferenciais e alguns sensores (por exemplo, bússola, sensores de proximidade e GPS) (Braga et al., 2008). Além do servidor de simulação, o ambiente de simulação também contém um visualizador de simulação em 2D utilizado nas competições do Ciber-rato (Lau et al., 2002).

O Simulador IntellWheels preserva a arquitectura conceitual do simulador Ciber-rato (ver Figura 45) que é do tipo cliente-servidor. O simulador funciona como servidor e os agentes de controlo dos robôs e o visualizador como clientes (Braga et al., 2009). Pode-se observar que a arquitectura do simulador é distribuída em vários tipos de aplicações que comunicam entre si.

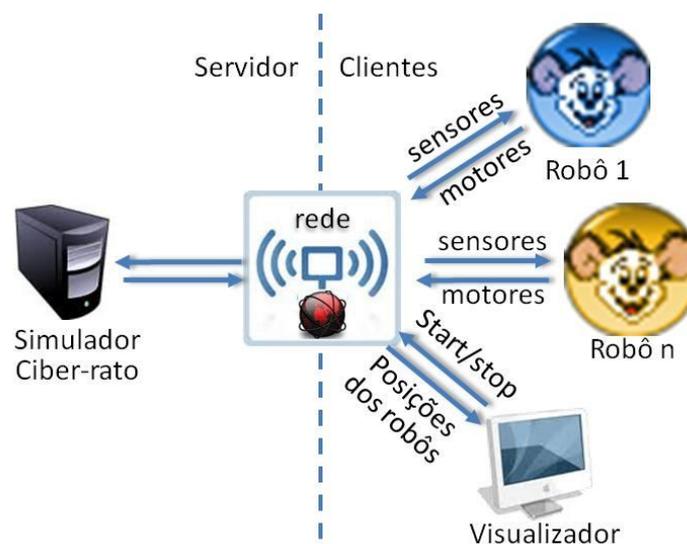


Figura 45: Arquitectura do Ciber-rato (adaptado de (Lau et al., 2002))

A arquitectura funcional do Simulador IntellWheels é conceptualmente ilustrada na Figura 46. De forma mais abstracta, este simulador consiste num servidor central de simulação onde cada agente (robô) se conecta a ele. Os agentes são aplicativos externos (estes podem ser

desenvolvidos em qualquer tipo de linguagem) que se conectam por meio de protocolo UDP. Esta arquitectura torna a estrutura do simulador modular.

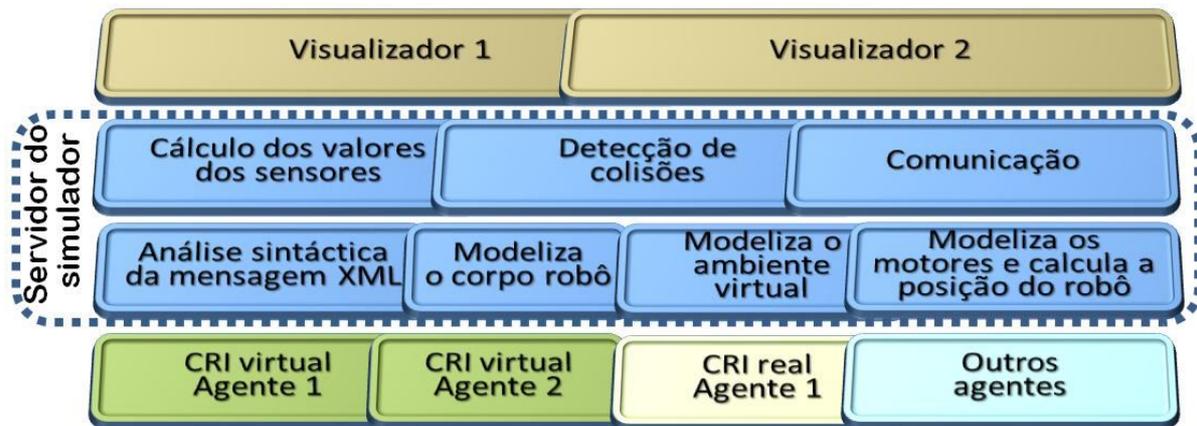


Figura 46: Arquitectura do Simulador IntellWheels

O servidor é responsável por realizar todos os cálculos referentes a simulação (deteção de colisão, cálculo da posição, emulação dos motores e valores dos sensores). Os agentes que compõem a CRI (Figura 41) podem ligar-se ao simulador em vez de se ligarem à CRI real. Deste modo, eles recebem as informações relativas à percepção dos sensores virtuais e enviam o sinal de controlo para os motores da CRI. Assim, todas as consequências de uma modificação nos algoritmos do sistema podem ser verificadas em questão de segundos no simulador.

As alterações introduzidas a este simulador para uso no projecto IntellWheels incluem também a introdução da noção de realidade mista. A Figura 47 ilustra as conexões entre o servidor de simulação e os demais clientes (por exemplo, agentes da cadeira de rodas real/virtual, agente que controla porta real/virtual, visualizadores da simulação). Esta conexão pode acontecer tanto em rede com fios, sem fios ou em ambas. A interação entre os mundos virtual e real acaba por criar um ambiente de realidade mista para a CRI.



Figura 47. Conexão com o Simulador IntellWheels

Conceptualmente, a Realidade Mista (ou Realidade Misturada) está relacionada com tecnologias que envolvem a fusão dos mundos reais e virtuais e contém dois subgrupos denominados de Realidade Aumentada e Virtualidade Aumentada. No ambiente real não existe nenhuma interação entre objectos criados pelo computador e objectos físicos. O sistema é composto exclusivamente por seus objectos reais (paredes, mesas, cadeiras, etc.) e percepções (leituras de sensor de sonar, câmaras VGA, etc.). No ambiente puramente virtual, o sistema é matematicamente modelado e a percepção do mundo está limitada ao que está contido nos dados virtuais. Todos os parâmetros que influenciam na simulação são calculados por um computador e os resultados são baseados exclusivamente nas informações inicialmente programadas. Por fim, a Realidade Aumentada e Virtualidade Aumentada tratam de um ambiente que envolve tanto Realidade Virtual como elementos do mundo real, dando origem a um ambiente misto em tempo real, embora que na Realidade Aumentada o ambiente predominante é o mundo real e na Virtualidade Aumentada o ambiente predominante é o mundo virtual. A transição gradual do real para o virtual é denominada Realidade/Virtualidade contínua. A Figura 48 apresenta um diagrama que mostra as possibilidades gradativas da sobreposição do real com o virtual e vice-versa.



Figura 48: Representação contínua do mundo real ao mundo virtual

A realidade mista estende recursos do simulador, o que torna possível realizar desde testes mais simples de algoritmos até testes mais complexos, com a CRI real a reagir num cenário mais dinâmico (obstáculos móveis, mapas complexos e outros agentes inteligentes) de forma segura e económica.

No modo de realidade mista, o agente CRI real calcula a sua posição (a partir de informações obtidas dos *encoders* da cadeira de rodas) e em seguida envia esta informação ao simulador. Uma vez que os dados são recebidos, o simulador posiciona o modelo da CRI, na posição equivalente, e retorna a percepção dos sensores virtuais. O agente cadeira, após receber as informações dos sensores (reais/virtuais), calcula a sua próxima acção e, desta forma, envia o sinal de controlo para os motores da cadeira de rodas real.

Foram realizadas alterações significativas no modelo do robô, assim como foi desenvolvido um novo algoritmo de detecção de colisões (ver Figura 49). Em relação ao modelo do robô, o corpo circular foi modificado por um modelo rectangular personalizável, o que resultou na necessidade de modificação do algoritmo de vários módulos, principalmente a detecção de colisão, a velocidade angular e cálculo do valor dos sensores.

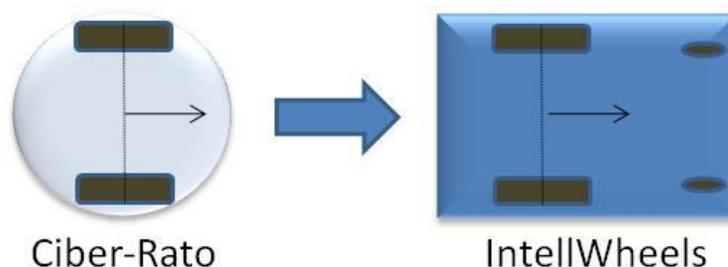


Figura 49: Modificação do corpo do robô de circular para rectangular

O Simulador IntellWheels pressupõe que todos os robôs têm uma forma rectangular, com uma largura e altura configurável. Além disso, uma cadeira de rodas tem o seu centro de

movimento dependente da posição de suas rodas em relação à sua estrutura física. Normalmente, o centro de movimento está próximo da parte traseira do robô, no centro do eixo das rodas de tracção. Na maioria das cadeiras de rodas disponíveis no mercado, o centro está na metade da sua largura e entre 70% a 90% do comprimento do robô, como exemplificado na Figura 50.

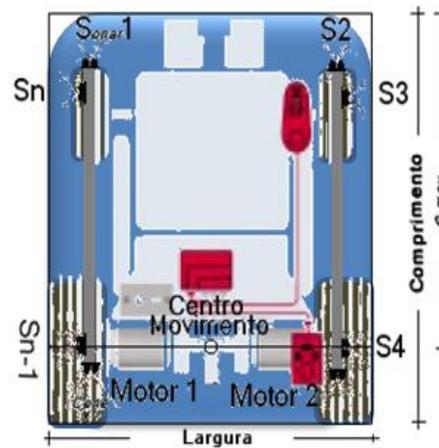


Figura 50: Agente robótico simulado (CRI)

Centro do movimento ( $C_{mov}$ ) pode variar entre 0 e 1,0 e define a posição do eixo das rodas do robô em relação ao seu comprimento. Por exemplo, um  $C_{mov} = 0,8$  define o eixo mais próximo da traseira do robô e um  $C_{mov} = 0,5$  define o eixo no centro do robô.

Outros parâmetros básicos de configuração da CRI simulada são a velocidade máxima e a característica dinâmica. Assim, cada CRI pode ter a sua própria velocidade máxima e sua curva de evolução da potência. A velocidade máxima por defeito é de um metro por segundo. Para controlo de aceleração, foi utilizada a (Eq. 14 para calcular a potência do motor, dado um valor de potência de entrada, o comportamento dinâmico da CRI foi aproximado por um modelo de primeira ordem.

$$output_n = (1 - curve) * input_n + curve * output_{n-1}, \quad (Eq. 14)$$

Onde  $output_n$  é a nova saída de potência do motor,  $curve$  é um valor entre 0 e 1,0 que define a inclinação da curva de aceleração, e  $output_{n-1}$  é o valor de energia do período anterior. A Figura 51 ilustra as curvas de evolução da potência resultante da (Eq. 14 com três valores diferentes do parâmetro  $curve$  (0,1; 0,5 e 0,8). Ambas as curvas usaram a constante entrada em 100% (i.e. a cadeira de rodas é acelerada utilizando a potência máxima dos motores).

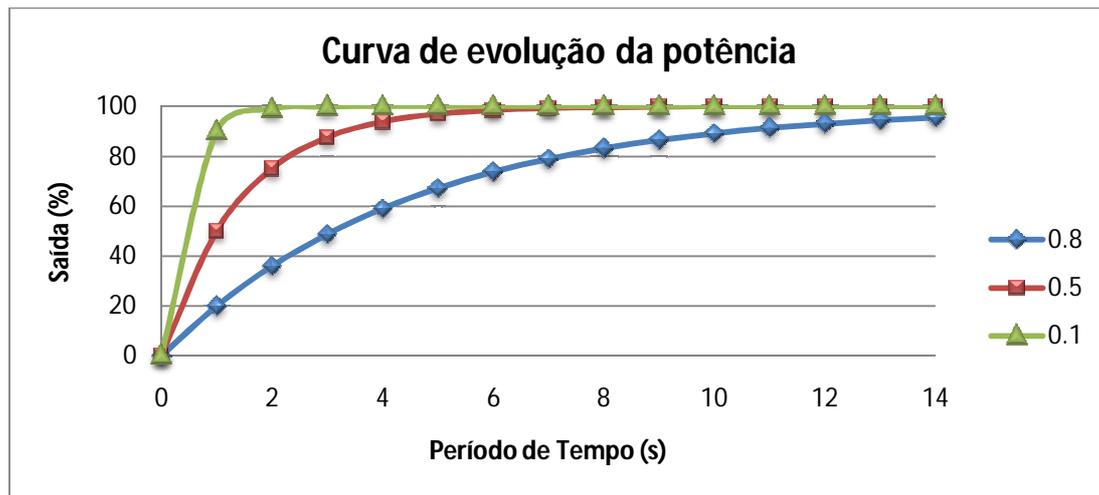


Figura 51: Curvas de evolução da potência com o parâmetro curve = (0,1, 0,5 e 0,8)

Originalmente, para detectar a colisão entre robôs, o simulador Ciber-Rato verificava se a distância que separa os centros dos robôs (através das coordenadas X e Y) é menor do que duas vezes o raio do robô (todos os robôs eram circulares com o mesmo raio). Esse algoritmo simples não é aplicável para robôs com raios diferentes e tão pouco para os robôs com forma rectangular. A posição e o tamanho das cadeiras de rodas para ser representada no mapa dependem da modelização feita por quatro parâmetros: centro de movimento, ângulo de orientação da cadeira de rodas, largura e comprimento. A partir destas informações pode-se calcular as coordenadas de todos os cantos do robô. Usando esta informação, o novo algoritmo de detecção de colisão segue os seguintes passos:

- Usando pares de cantos são definidos os pontos que representam o segmento de recta e encontra-se uma equação para cada um dos segmentos de rectas para cada robô.
- O ponto de intersecção das duas rectas é calculado. Se as linhas são paralelas nenhum ponto é calculado porque não existe nenhuma intersecção.
- Coordenadas X e Y são verificadas para encontrar se eles estão localizados dentro de cada segmento de linha do robô. Se assim for, então há uma colisão entre os dois robôs.
- Este processo é repetido para as 4 rectas de cada robô.

A Figura 52 ilustra o teste para verificar se existe a intersecção dos segmentos de recta de cada robô. A recta definida pelos cantos 0 e 1 do robô A está sendo verificada em relação a recta definida pelos cantos 0 e 2 do robô B. Pode-se observar que existe um ponto de intersecção que está dentro do segmento de recta do robô A mas está fora do segmento de recta do robô B, portanto, nenhuma colisão é detectada.

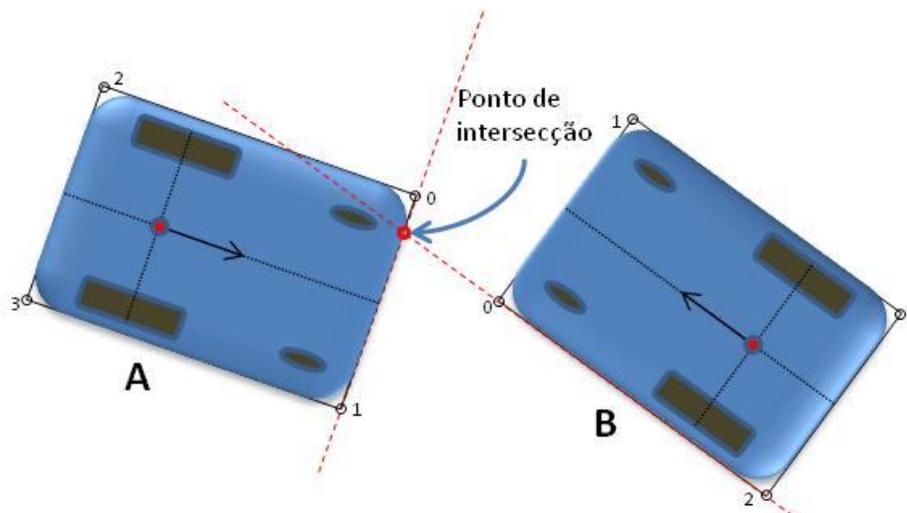


Figura 52: Verificação da intersecção dos segmentos de recta

No simulador, as informações das paredes são armazenadas numa matriz de paredes e cada parede, por sua vez, representada como uma matriz de cantos. Um canto é definido por suas coordenadas X e Y. Desta forma, o algoritmo de detecção de colisão com as paredes é semelhante do algoritmo de detecção de colisões entre robôs. Cada dois cantos consecutivos define um segmento de recta que será verificado com o segmento de recta dos robôs. Se o ponto de intersecção estiver dentro dos dois segmentos de recta, então existe uma colisão.

O posicionamento dos sensores de proximidade também foi adaptado, como pode ser visualizado na Figura 53. Originalmente, o simulador Ciber-Rato simula sensores de proximidade que só poderiam ser posicionados no perímetro do robô circular com um cone fixo e direcção fixa.

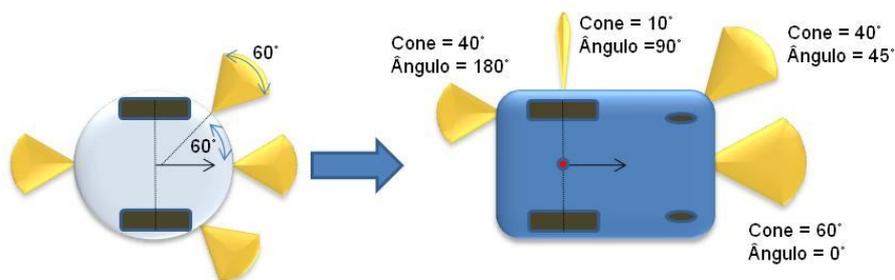


Figura 53: Modificações nos sensores de proximidade

Os sensores agora podem ser configurados através das coordenadas X e Y (relativo ao centro de movimento do robô), assim como também é possível configurar a abertura do cone e a direcção que este está apontado. Todos estes parâmetros agora são configuráveis no momento do registo do agente no simulador. Um cone mais amplo simula um comportamento mais semelhante a um sensor de ultra-sons e um cone menor assemelha-se a um sensor de infravermelhos.

### Visualizador do Simulador

Visualização é um aspecto importante para a compreensão humana, uma vez que os seres humanos processam as informação gráfica pré-conscientemente (Lau et al., 2002), (Rohrer, 2000). Tendo isso em mente, o Visualizador IntellWheels foi desenvolvido para permitir visualizar a representação das CRI e do ambiente (por exemplo, o mapa com suas paredes, portas, objectos) em duas e três dimensões. O visualizador é capaz de desenhar graficamente o mundo modelado através da definição do mapa e das posições dos agentes robóticos enviadas pelo simulador. O visualizador conecta-se ao simulador através do protocolo UDP e troca informações através de mensagens XML. Em cada período de tempo do simulador os novos estados do mundo são calculados, incluindo as informações de cada robô. A Figura 54 ilustra a janela 2D do visualizador, onde foi simulado um ambiente com sete salas, um corredor e uma cadeira.

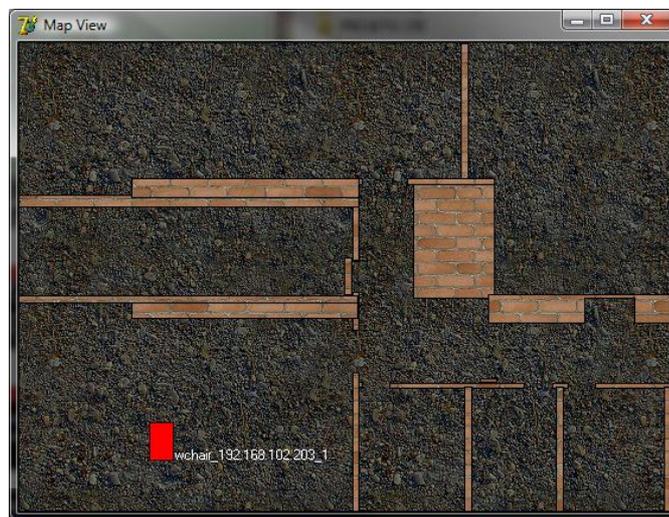


Figura 54: Janela 2D do Visualizador

Esta arquitectura permite que exista vários visualizadores conectados ao simulador. Para que os visualizadores possam actualizar a representação gráfica o simulador envia periodicamente todas as informações necessárias. A visualização 3D usa tecnologia OpenGL (Woo et al., 1999). Foram implementados dois modos de exibição em 3D: modo de exibição em primeira pessoa (esse modo permite uma visualização como se fosse o verdadeiro condutor da cadeira de rodas) e modo de câmara livre (neste é possível posicionar a câmara em qualquer localização).

A Figura 55 apresenta a janela do visualizador no modo de câmara livre (em uma simulação com três agentes cadeira de rodas, um agente mesa e um agente de porta) e no modo primeira pessoa (simulação da cadeira a navegar num corredor).

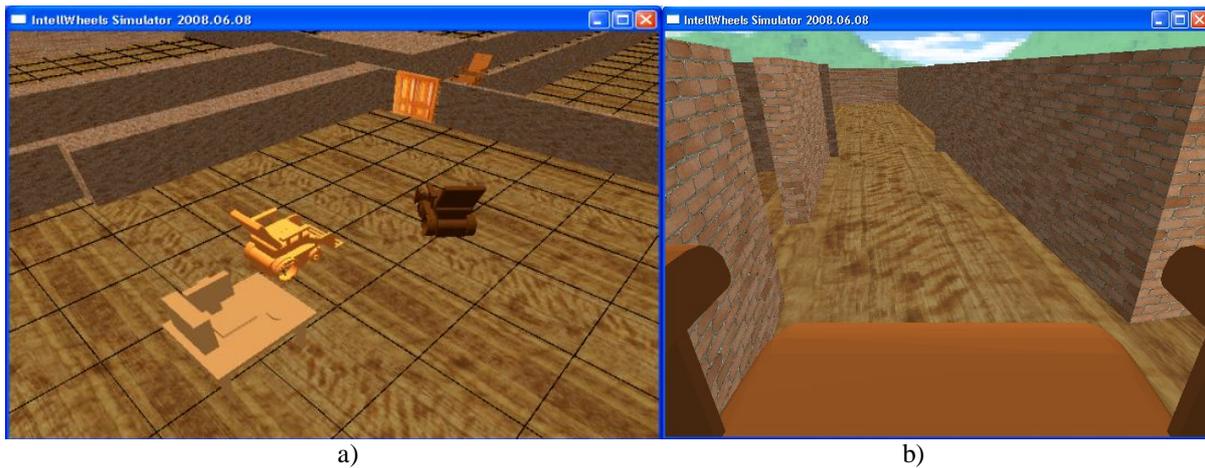


Figura 55: Visualizador 3D, a) modo câmara livre e b) modo primeira pessoa

## 4.4 Planeamento inteligente

Planeamento inteligente em inteligência artificial está relacionado com a realização de sequências de acções tipicamente executadas por agentes inteligentes, robôs autónomos ou veículos autónomos. Um planeador típico recebe três entradas: o estado inicial do mundo, uma descrição do objectivo e um conjunto de acções possíveis (e respectivas pré-condições e efeitos no mundo). Em domínios mais realistas é necessário decompor o plano. Desta forma, surge o planeamento hierárquico onde uma certa tarefa pode ser realizada por uma acção primitiva ou pode ser decomposta num conjunto de tarefas de mais baixo nível.

A complexidade do planeamento inteligente está na dependência do seu correcto funcionamento das suposições simplificadas que são tomadas, tais como: tempo discreto, espaço discreto, mundo determinístico e totalmente acessível, etc. Os planeadores clássicos realizam todas estas suposições. Na robótica, no entanto, onde os robôs se deslocam no mundo real e convivem com os humanos todas estas suposições deixam de fazer sentido. O tempo é contínuo, o espaço é contínuo, o mundo não é determinístico (não existindo qualquer garantia de execução correcta das acções) e não é totalmente acessível. No entanto, o planeamento de alto nível continua a ser extremamente necessário, como é o caso da cadeira de rodas inteligente. Um planeamento de trajectórias complexas é essencial principalmente para o mundo real e dinâmico, que normalmente é apenas semi-conhecido.

O objectivo actual deste módulo é avaliar as reais necessidades e requisitos de um planeador inteligente aplicado a CRI. Os planeadores desenvolvidos neste trabalho tiveram como objectivo principal servir como uma ferramenta didáctica e integrada ao sistema, ficando o desempenho do planeador e a qualidade dos planos gerados fora do âmbito deste trabalho. Baseado no estudo realizado, foi construída uma ferramenta gráfica de planeamento para servir de base a novos desenvolvimentos.

O módulo de planeamento é responsável por criar uma sequência de acções de alto nível necessárias para atingir a meta global. Este módulo compreende a camada estratégica da arquitectura de controlo multi-nível (ver subsecção 4.5 Navegação e Controlo) e é executada pelo Agente Inteligente (maiores detalhes na subsecção 4.7 Sistema Multi-agente) que compõe o Sistema Multi-agente da Plataforma IntellWheels (PIW).

O planeador utilizado neste trabalho foi inicialmente implementado com base no algoritmo STRIPS (Figura 56) e mais tarde substituído por uma metodologia de planeamento baseado em grafo (Figura 57). Planeamento com base em grafos é uma poderosa estrutura de dados que codifica a informação sobre que estados podem ser alcançáveis, ou em outras palavras, trata-se de uma sequência de nível que corresponde a um conjunto de estados ou de acções. Também foi utilizada uma linguagem para descrever o problema e o domínio, o *Planning Domain Definition Language* - PDDL (Edelkamp & Hoffmann, 2003).

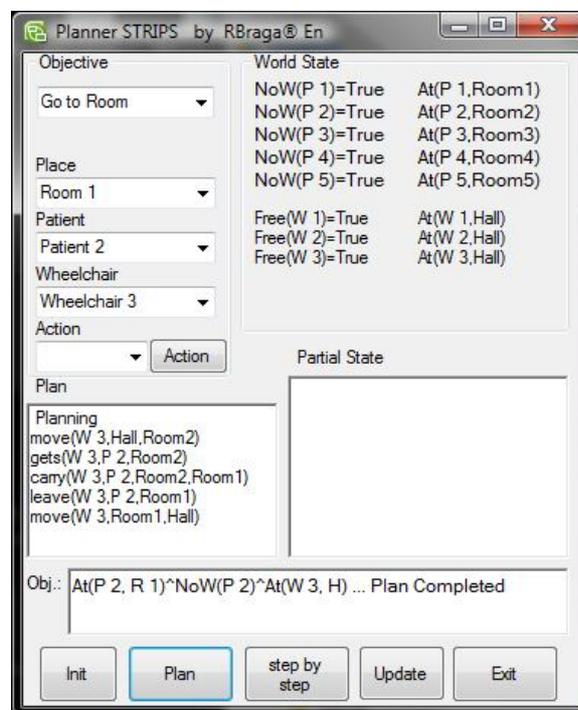


Figura 56: Planeador com base em STRIPS

O módulo de planeamento utilizou inicialmente o algoritmo de planeamento automático linear do tipo STRIPS com encadeamento inverso. Foram criados 5 objectivos diferentes: ir para algum quarto; ir para o jardim; ir à casa de banho; ir à enfermaria e ir ao refeitório. Cada um destes objectivos possui características diferentes, como por exemplo, ir para o quarto implica que a cadeira volte vazia para o hall. Esta abordagem utilizada inicialmente para o planeamento é muito rígida pois a implementação está muito voltada ao cenário de aplicação (neste caso, este foi desenvolvido para planear certas tarefas comuns num ambiente hospitalar), o que resultaria na necessidade de refazer-se a implementação para cada novo

cenário. Na maioria das situações, este planeador apresentou planos completos, mas nalgumas situações ele apresentou a anomalia de Sussman<sup>13</sup> (Sussman, 1975). Então, optou-se por utilizar um planeador com base em GraphPlan juntamente com uma linguagem padronizada para descrever o domínio e o problem. Isto permite não só o planeamento das acções da cadeira num ambiente hospitalar, a geração de planos para diferentes cenários, assim como, a geração de planos para as acções internas da CRI.

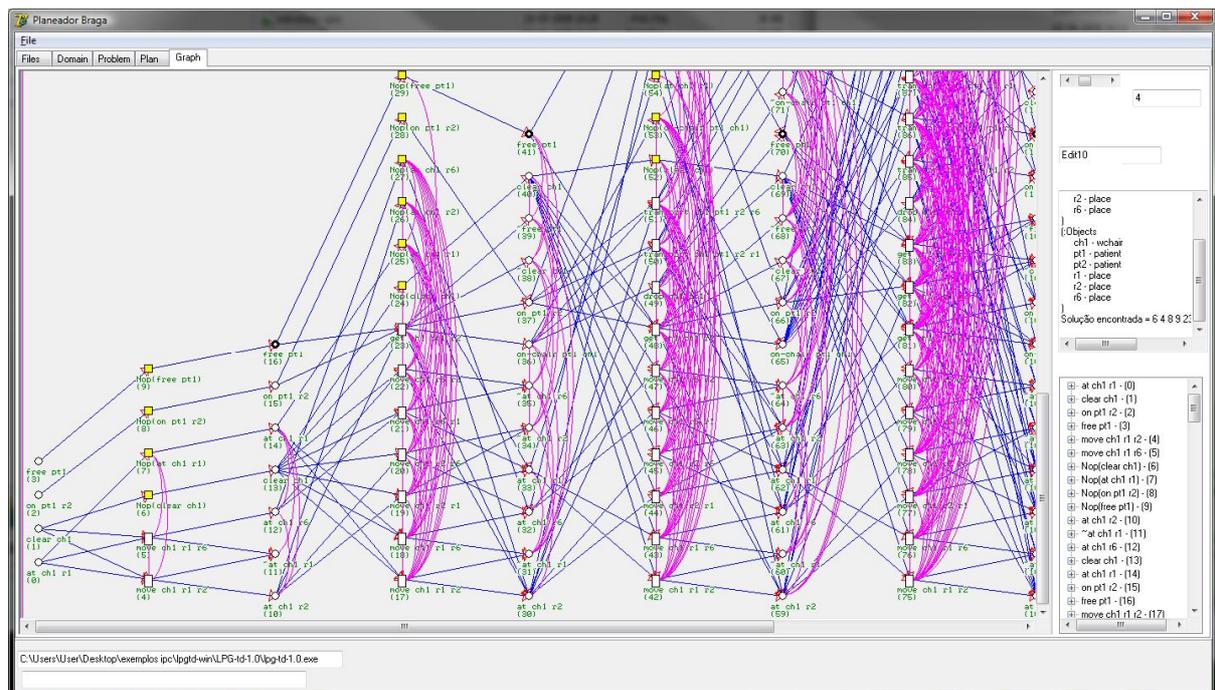


Figura 57: Planeador com base em GraphPlan

Este planeador foi implementado de forma gráfica com o objectivo de se tornar uma ferramenta que facilite o ensino e desenvolvimento de técnicas de planeamento com base em GraphPlan. Como mencionado no capítulo anterior, esta técnica de planeamento é dividida em duas etapas. A primeira etapa é a geração de um grafo que representa as acções e estados. O grafo gerado pode ser observado na Figura 57, onde pode observar-se os nós (que representam as proposições e as acções, distribuídos em níveis diferentes), os arcos (que representam as pré-condições ou efeito das acções) e também as relações *mutex* (conflitos entre acções ou entre predicados). A segunda etapa é a extracção do plano. Nesta etapa foi utilizado a busca em profundidade para procurar a solução.

<sup>13</sup> A Anomalia Sussman é um problema em IA, primeiramente descrito por Gerald Sussman, que ilustra uma falha em algoritmos usados em planeadores “não intercalados” que não conseguiam encontrar uma solução sem redundâncias, uma vez que, tratavam os sub-objectivos isoladamente, em vez de intercalá-los.

A Figura 58 apresenta as janelas de configuração do planeador com base em GraphPlan. Com isso, é possível criar, editar ou carregar a descrição do domínio e do problema com o uso da linguagem PDDL.

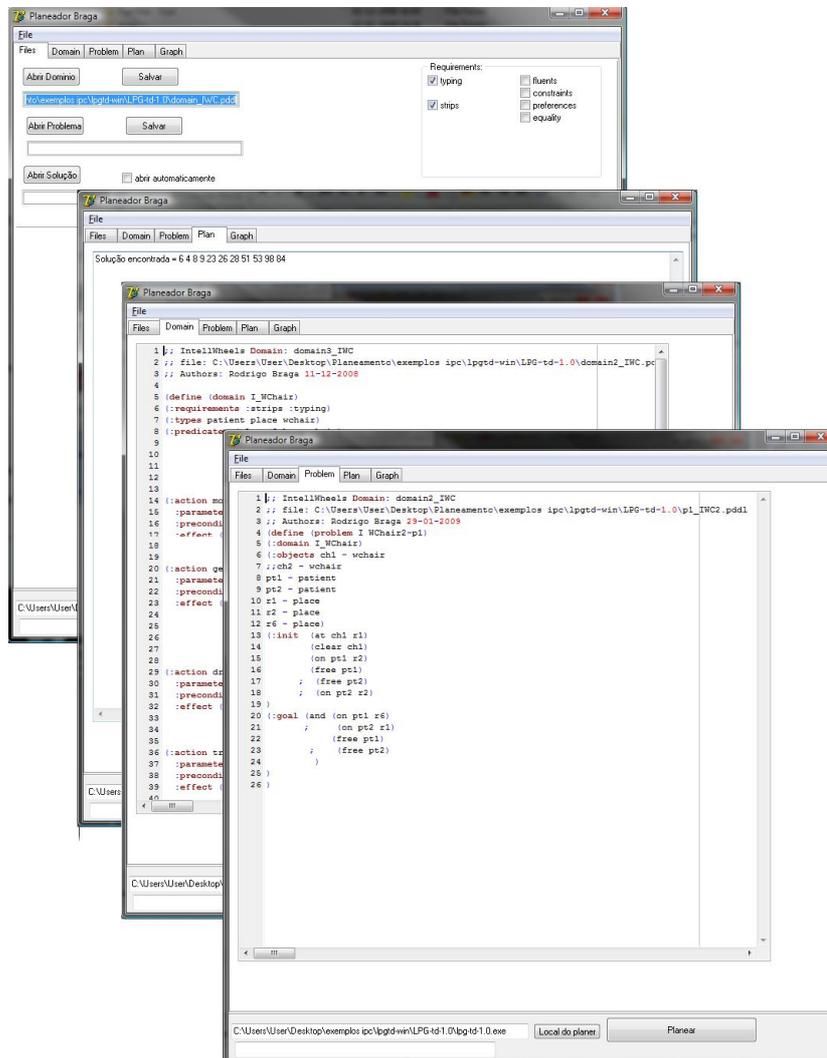


Figura 58: Janelas de configurações do Planeador

A Figura 59 apresenta a descrição do domínio das CRI em PDDL. É possível observar os cinco predicados (linhas de 8 a 12) pertencentes ao domínio das CRI (cadeira 'ch' livre, paciente 'pt' sobre a cadeira 'ch', cadeira 'ch' no sítio 'p', paciente 'pt' no sítio 'p' e paciente 'pt' sem cadeira), assim como, a definição de quatro acções (pegar paciente, largar paciente, transportar paciente e mover cadeira vazia de um sítio para outro).

```

1 ;; IntellWheels Domain: domain3_IWC
2 ;; file: C:\Users\User\Desktop\Planeamento\exemplos ipc\lpqtd-win\LPG-td-1.0\domain2_IWC.pddl
3 ;; Authors: Rodrigo Braga 11-12-2008
4
5 (define (domain I_WChair)
6 (:requirements :strips :typing)
7 (:types patient place wchair)
8 (:predicates (clear ?ch - wchair)
9              (on-chair ?pt - patient ?ch - wchair)
10             (at ?c - wchair ?p - place)
11             (on ?pt - patient ?p - place)
12             (free ?pt - patient) )
13
14 (:action move
15  :parameters (?ch - wchair ?p1 - place ?p2 - place)
16  :precondition (and (clear ?ch ) (at ?ch ?p1))
17  :effect (and (not (at ?ch ?p1))
18             (at ?ch ?p2)))
19
20 (:action get
21  :parameters (?ch - wchair ?pt - patient ?p1 - place)
22  :precondition (and (clear ?ch) (at ?ch ?p1) (on ?pt ?p1) (free ?pt))
23  :effect (and (not (clear ?ch))
24             (not (free ?pt))
25             (on-chair ?pt ?ch)
26             (at ?ch ?p1)
27             (on ?pt ?p1)))
28
29 (:action drop
30  :parameters (?ch - wchair ?pt - patient)
31  :precondition (and (on-chair ?pt ?ch))
32  :effect (and (not (on-chair ?pt ?ch ))
33            (clear ?ch )
34            (free ?pt)))
35
36 (:action transport
37  :parameters (?ch - wchair ?pt - patient ?p1 ?p2 - place)
38  :precondition (and (on-chair ?pt ?ch ) (at ?ch ?p1 ) (on ?pt ?p1))
39  :effect (and (not (at ?ch ?p1))
40            (not (on ?pt ?p1))
41            (at ?ch ?p2)
42            (on ?pt ?p2)
43            )
44 )
45 )

```

Figura 59: Descrição do domínio da CRI em PDDL

A descrição do problema da CRI em PDDL (Figura 60) informa o planeador que existe duas CRI, dois pacientes, e três salas (ambientes). Inicialmente, a ‘CRI 1’ está livre na ‘sala 1’, os dois pacientes estão na ‘sala 2’, sem cadeira (linha 13 à 16). Neste momento o objectivo é levar o ‘paciente 1’ para a ‘sala 6’ e ‘paciente 2’ para a ‘sala 1’ (linha 20 à 24). Esta descrição deve ser alterada sempre que o ambiente muda e/ou um novo objectivo é determinado.

```

1 ;; IntellWheels Domain: domain2_IWC
2 ;; file: C:\Users\User\Desktop\Planeamento\exemplos ipc\lpgtd-win\LPG-td-1.0\p1_IWC2.pddl
3 ;; Authors: Rodrigo Braga 29-01-2009
4 (define (problem I_WChair2-p1)
5 (:domain I_WChair)
6 (:objects ch1 - wchair
7          ch2 - wchair
8          pt1 - patient
9          pt2 - patient
10         r1 - place
11         r2 - place
12         r6 - place)
13 (:init (at ch1 r1)
14        (clear ch1)
15        (on pt1 r2)
16        (free pt1)
17
18
19 )
20 (:goal (and (on pt1 r6)
21            (free pt1)
22
23
24 )
25 )
26 )

```

Figura 60: Descrição do problema da CRI em PDDL

Outra função que está presente neste módulo é o planeador de trajetórias (Figura 61), este gera o caminho (trajetória) para atingir os objectivos, propostos pelo planeador de alto nível, tendo em conta as informações do modelo de mundo (mapa, localização e etc.). Para encontrar um caminho a partir de um determinado ponto inicial até um ponto objectivo determinado, o sistema utiliza o algoritmo A\* (Braga et al., 2009), (Martens et al., 2001).

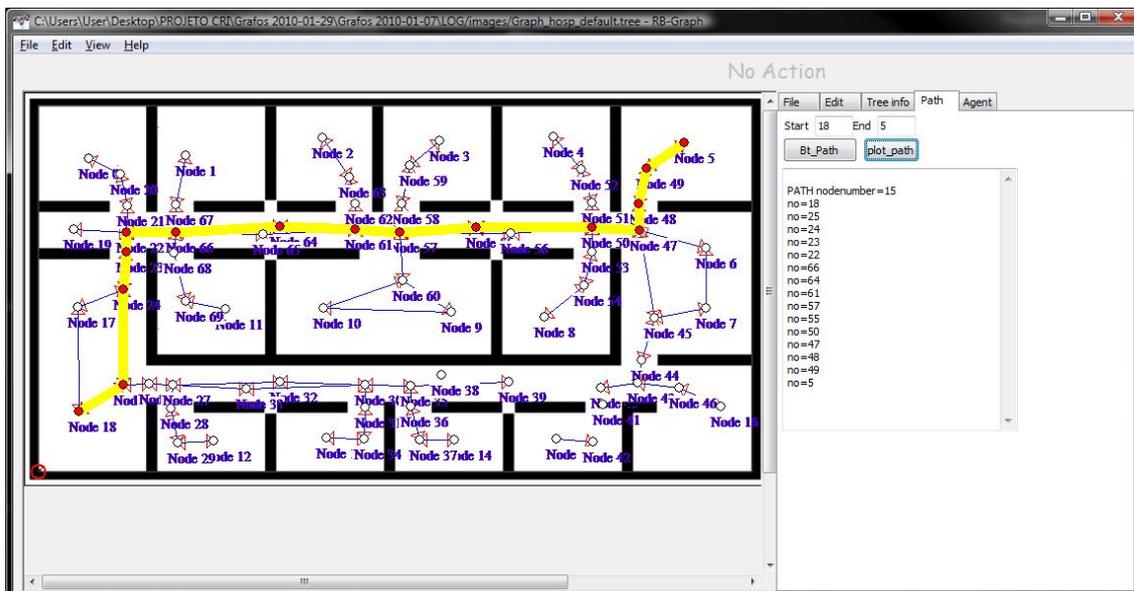


Figura 61: Planeador de trajetórias

Esta função foi implementada na forma de uma ferramenta que une dois conceitos, mapa métrico e topológico. Os passos de utilização desta ferramenta são:

1. Inicialmente, utiliza-se uma representação métrica do ambiente em formato Bitmap (bmp);
2. Criar a o sistema referencial no mapa do ambiente (indica o ponto zero do mapa);
3. Calibrar as dimensões no mapa (encontra a escala do mapa) para o mesmo representar as medidas reais do ambiente;
4. Criar o mapa topológico do ambiente. Esta ferramenta utiliza o conceito de grafos com as informações das coordenadas x,y dos nós obtidas do mapa métrico. A tarefa de criar o mapa topológico resume-se em colocar os nós nos sítios adequados e gerar os arcos entre os nós.

Desta forma o algoritmo de pesquisa A\* utiliza este grafo para encontrar o melhor caminho entre dois pontos. O objectivo de se utilizar um planeamento de rotas com base num mapa topológico, neste desenvolvimento, foi facilitar a cooperação entre as CRI para evitar conflito de rotas. Desta forma, os caminhos estão restritos aos nós e arcos e para avaliar conflitos de rotas basta comparar os nós e arcos de cada plano com uma estimativa temporal da ocupação dos mesmos. Um outro recurso possível é armazenar conhecimento nos nós e arcos, com o objectivo de auxiliar a navegação. A Figura 62 ilustra as janelas, deste planeador de trajetórias, que permitem abrir, salvar, editar, criar mapas, visualizar as informações dos nós e arcos, informações da rota e configuração do agente assim como as informações das tarefas deste agente.

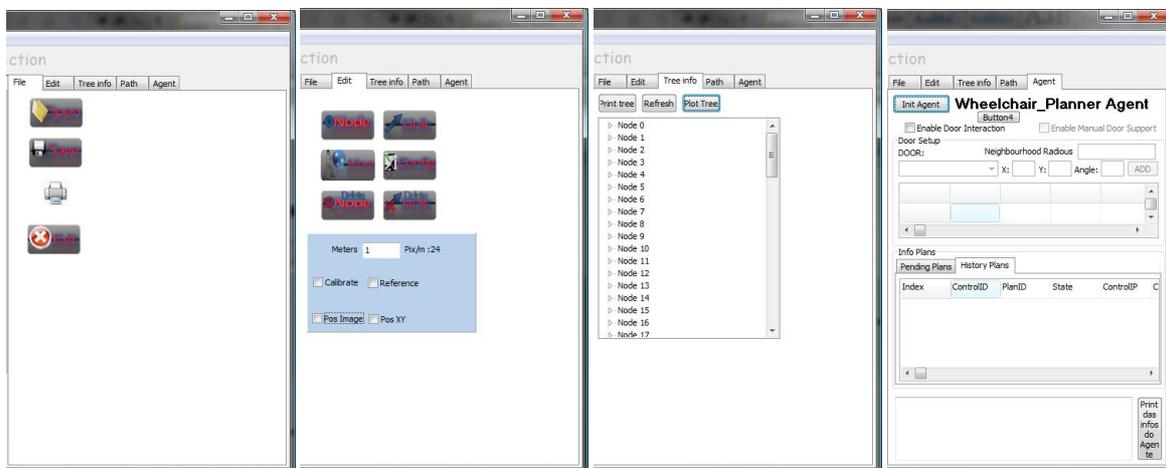


Figura 62: Janelas de configuração do planeador de trajetórias

A função de planear rotas geralmente é realizada pelo agente Planeador mas, em caso de falta deste agente no sistema, o agente Controlo tem capacidades para assumir este papel e gerar sua própria trajetória utilizando A\* num mapa métrico pré-informado ou num mapa criado de forma dinâmica pelo agente Percepção (Figura 63).

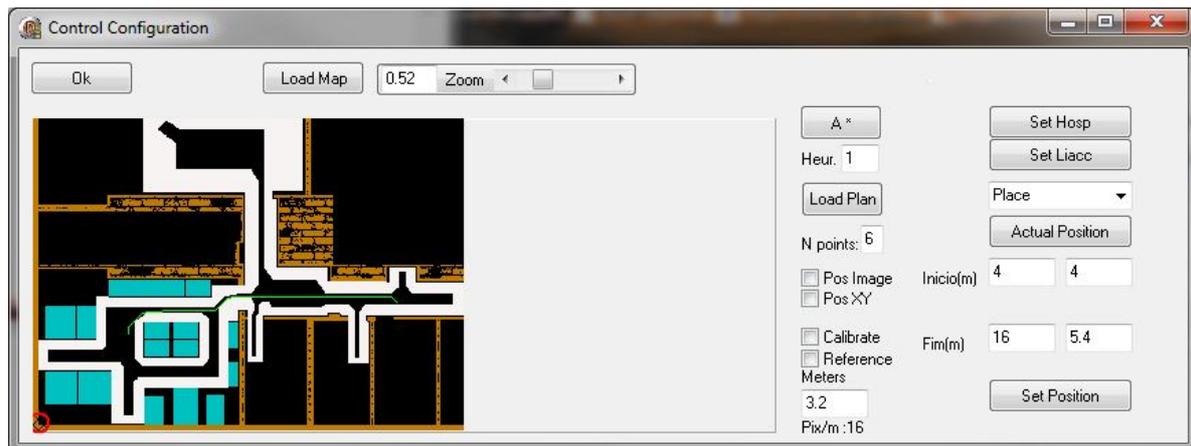


Figura 63: Planeador de trajetórias com mapa métrico

## 4.5 Navegação e Controlo

Navegação é a arte, ciência, prática ou tecnologia para um robô movimentar-se (executar e/ou planejar uma viagem) de um ponto de partida até seu ponto de destino (Borenstein et al., 1996) e envolve, principalmente: percepção, acção, planeamento, representação, controlo e hardware. Os sistemas de navegação buscam responder a quatro questões essenciais: Onde estou? Para onde ir? Por onde ir? Onde estive? Uma actividade importante para permitir a de navegação é a de se determinar a posição actual, para possível comparação com posições previstas ou desejadas. Alguns equipamentos (como por exemplo, bússola, o radar, o rádio, *encoder* e GPS) auxiliam nesta tarefa. Outro factor de extrema importância na navegação é a existência de mapas ou modelos que permitam descrever a representação do ambiente.

O módulo de Navegação da PIW inclui um conjunto de algoritmos responsável por realizar o tratamento dos sensores da cadeira de rodas, sua localização e mapeamento do ambiente (Braga et al., 2009). O conjunto de funções relacionadas com este módulo é implementado de forma distribuída entre os agentes Percepção, Controlo e Inteligência.

A PIW apresenta uma arquitectura de controlo multi-nível, subdividida em três camadas: camada estratégica, camada táctica e camada de controlo básico (descritas na Figura 64) (Braga et al., 2008).



Figura 64: Arquitectura de controlo multi-nível da PIW

No topo das camadas do controlo multi-nível tem-se a camada estratégica que é implementada pelo agente Inteligência, responsável pelo planeamento. Os algoritmos que compõem este agente já foram descritos na subsecção anterior – Planeamento Inteligente. Utilizou-se planeamento baseado em grafos para os planos de acções de alto nível. No quarto nível, foi utilizado um algoritmo baseado em A\* e grafos para calcular o menor caminho de um ponto a outro.

A camada táctica é responsável por subdividir o caminho calculado pelo módulo planeamento em formas básicas (linhas, círculos, pontos) e em seguida calcular as referências de velocidades para as rodas da cadeira (linear e angular) a fim de movimentar a cadeira de rodas.

O nível mais baixo do controlo, que pertence a camada de controlo básico, consiste basicamente em controlar a velocidade de cada roda. Nos últimos protótipos da CRI IntellWheels esta função é realizada pelo módulo de potência. As referências de velocidades lineares e angulares são transferidas por comunicação série para a placa de interface que por sua vez as envia ao módulo de potência da cadeira. Foi utilizado um módulo comercial produzido pela PG Drives Technology modelo VR2<sup>14</sup>. Este módulo possui altos padrões de desempenho e são dedicados ao controlo de cadeira de rodas motorizadas. O módulo VR2 inclui um módulo de alimentação compacta e um módulo de joystick e permite configurar um elevado número de parâmetros, tais como acelerações, velocidades máximas, protecções de sobre corrente, etc.

<sup>14</sup> PG Drives Tecnologia, anteriormente conhecido como Penny & Giles Unidades de Tecnologia, é uma das principais fabricantes mundial de controladores para cadeiras de rodas e Scooters de mobilidade, VR2 é um módulo controlador de cadeira de rodas convencional. <http://www.pgdt.com/>.

A Localização da CRI é realizada pelo agente Percepção que utiliza as informações provenientes do conjunto de sensores do RM. Para tal, foi desenvolvido um módulo de odometria que estima a posição do robô. No caso da odometria virtual, este módulo fica responsável por intermediar a troca de dados com o simulador e actualizar o estado do mundo interno que é enviado aos outros agentes. Estas informações, de posição, vão permitir que o módulo de controlo efectivamente controle os movimentos da cadeira. Na odometria real, as informações de velocidade e posição são calculadas a partir das informações obtidas pelos *encoders*.

Utilizou-se ainda a informação proveniente do sistema de visão para auxiliar na navegação. O reconhecimento de marcos artificiais (*landmarks*) torna o sistema de localização mais robusto. O módulo de visão consiste no processamento de imagens fornecidas por uma *webcam* direccionada para o piso do ambiente, onde actualmente a imagem capturada é utilizada para o reconhecimento de marcos rectangulares vermelhos de 11x2cm colados no chão. Tais marcos são reconhecidos e comparadas com as informações armazenadas na memória do agente. O agente tem armazenado na sua memória informações referentes à posição dos marcos artificiais, para um determinado ambiente. O software de reconhecimento é activado quando a CRI está aproximadamente a um raio de um metro do marco. Desta forma, quando este é reconhecido o sistema calcula a posição em relação a este marco conhecido e actualiza a sua posição, o que ocasiona uma correcção automática do erro de posição acumulado pela odometria. A Figura 65 apresenta a janela de configuração da aplicação de reconhecimento de marcos artificiais. Nesta aplicação é possível configurar a posição da webcam e os critérios de reconhecimento do marco artificial.

O algoritmo de reconhecimento é muito simples e segue os seguintes passos: segmentar as cores (neste caso foi determinado a cor vermelha pela baixa probabilidade de encontrar objectos e/ou pisos desta cor); identificar as bordas pelo lado direito e esquerdo; validar as bordas encontradas a partir de parâmetros de validação que indicam a distância e a tolerância; cálculo do ponto médio entre os pontos das bordas; determinação da equação que representa a recta entre os pontos médios do marco (utiliza-se o método dos mínimos quadrados); e, por fim, cálculo da posição global com base nas informações da imagem e da informação da posição do marco no ambiente.

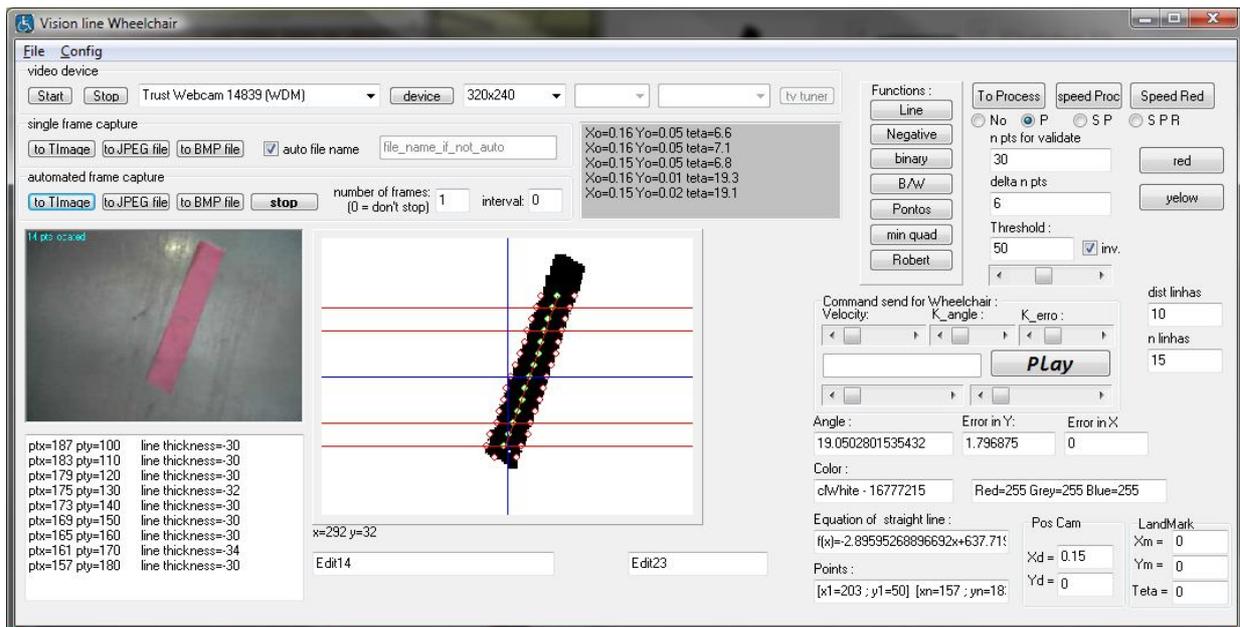


Figura 65: Aplicação de reconhecimento de marcos artificiais

Outro método de localização global utilizado na PIW, foi o sistema de localização em tempo real (Real Time Location Systems - RTLS) fabricado pela Ubisense<sup>15</sup> (Figura 66). O sistema Ubisense foi projectado para detectar e responder a eventos, com latências inferiores a um segundo, para o controle independentemente do número de pessoas e objectos ou do tamanho do espaço. Ubisense usa uma arquitectura celular escalonável, podendo ser aplicado desde pequenas instalações até áreas muito grandes. Em áreas grandes existe a necessidade de várias "células" com um pequeno número de sensores a trabalhar em conjunto. Milhares desses sensores podem ser integrados numa única empresa para monitorar uma área ilimitada e gerir milhares de *tags*. Os sensores podem ser ligados entre si numa variedade de configurações, o que permite otimizar os custos da infra-estrutura juntamente com a precisão de localização requerida pela aplicação. O sistema é composto por sensores de precisão ultra-wideband (UWB) organizados em forma de uma matriz de antenas e receptores de rádio de UWB. Deste modo, o sensor detecta pulsos das *tags* UWB Ubisense, o que permite a localizar as posições das *tags* com erro de 15cm em 3D. Para calcular a orientação da cadeira foram necessárias a utilização de duas *tags* por cadeira.

<sup>15</sup> Ubisense é um dos líderes mundiais em sistemas precisos de localização em tempo real (*Precise Real-Time Location Systems – RTLS*), <http://www.ubisense.net/en/>.



Figura 66: Sistema Ubisense: Antena, tag fina e tag compacta

As funções de Controlo, cuja função principal é o controlo da CRI, também permitem estudar e avaliar novos algoritmos de controlo assim como os algoritmos base já implementados. Os parâmetros dos algoritmos base podem ser alterados com a finalidade de possibilitar o seu estudo e permitir o ajuste dos parâmetros com a finalidade de melhorar o desempenho do controlo em modo automático (Figura 67). Actualmente é possível utilizar oito tipos de acções básicas: seguir recta, ir para o ponto (x,y), girar para ( $\theta$ ), seguir parede à esquerda, seguir parede à direita, esperar, parar, assim como a função de desvio de obstáculo.

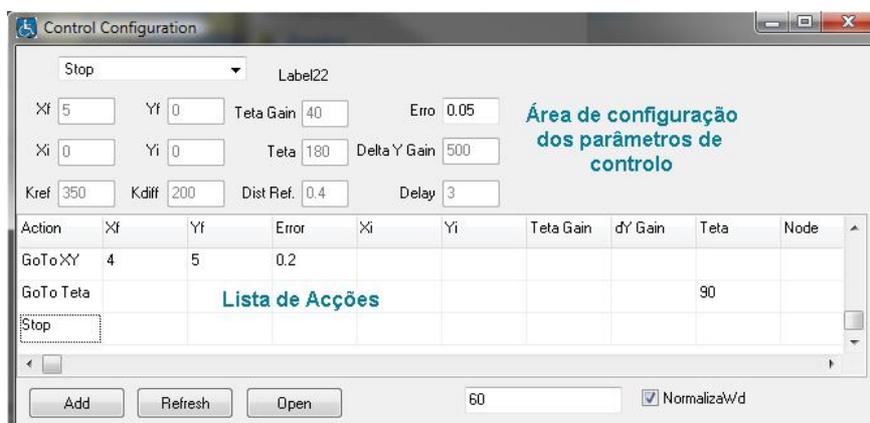


Figura 67: Janela de configuração dos parâmetros de controlo

#### 4.5.1 Controlo de Acções Básicas

Foram implementados algoritmos simples para o controlo de acções básicas, como por exemplo: seguir recta, ir para o ponto, girar para o ângulo. Estes algoritmos são baseados na posição cartesiana da cadeira de rodas. Outros algoritmos baseados nas informações de sensores de distância (sensor de ultra-som e infravermelho) também foram implementados, por exemplo: seguir paredes e desvio de obstáculos.

Uma trajectória, mesmo que complexa, pode ser entendida como a segmentação de inúmeros segmentos de rectas. A utilização desta abordagem visa reduzir a carga computacional necessária para a movimentação da CRI, uma vez que para determinar uma recta é necessário apenas a definição de dois pontos no espaço. Além disso, o uso de segmentos de recta torna possível definir e ajustar as velocidades linear e angular em cada ponto do segmento, levando em consideração os valores correspondentes aos pontos inicial e final.

Neste desenvolvimento, foram utilizados controladores que possuem feedback parcial dos estados  $(x_c, y_c, \theta)$  e permitem corrigir a posição  $(x_c, y_c)$  que representa as coordenadas cartesianas do ponto central da cadeira e a sua orientação  $(\theta)$  no plano cartesiano. Todas as velocidades calculadas, para os controladores a seguir, são saturadas no valor limite determinado pelo parâmetro de velocidade máxima. Os controladores implementados foram: seguir recta, ir para o ponto  $(x, y)$ , girar para  $(\theta)$  e seguir parede.

### Seguir Recta

O controlo *Seguir recta* é utilizado para manter o robô sobre uma determinada recta de referência. Para isso, considera-se a trajectória desejada, descrita pelo segmento de recta definido pelos pontos A  $(x_a, y_a)$  e B  $(x_b, y_b)$ , na direcção de A para B. A Figura 68 ilustra as representações da posição, orientação da cadeira de rodas e a trajectória definida pelo segmento de recta.

O ângulo  $\varphi$  é o erro de orientação da cadeira de rodas, e consiste na diferença entre o ângulo  $\alpha$  do segmento [AB] e o ângulo  $\theta$  do robô no Sistema Global de Coordenadas (SGC) e é calculado através da equação:

$$\varphi = \theta - \alpha \quad (\text{Eq. 15})$$

Para o cálculo da velocidade angular de referência  $(\omega)$  leva-se em consideração a componente de erro de orientação e o erro  $e_y$  (que representa a distância entre o ponto C e o segmento de recta [AB], sendo que a recta é definida pela função  $ax_r + by_r + c = 0$ ). O valor do erro  $e_y$  é calculado através da equação:

$$e_y = \frac{|ax_c + by_c + c|}{\sqrt{a^2 + b^2}} \quad (\text{Eq. 16})$$

Desta forma, a velocidade angular é calculada através da seguinte equação:

$$\omega = -k_1 \varphi - k_2 e_y \quad (\text{Eq. 17})$$

Onde  $k_1$  e  $k_2$  representam os ganhos proporcionais de ajuste do controlo.

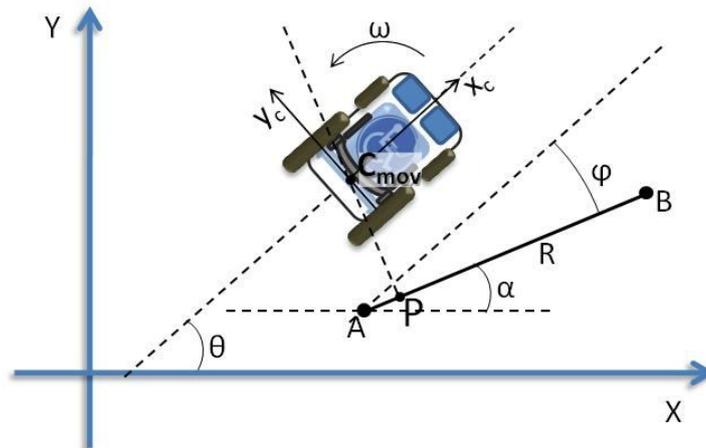


Figura 68: Representação esquemática do controlador para seguimento de recta

A equação que representa a velocidade linear ( $v$ ) do robô é:

$$v = v_n + k_2 e_p \quad (\text{Eq. 18})$$

O erro  $e_p$  consiste da distância entre a projecção do ponto C sobre a recta AB e o ponto B e é calculado pela equação:

$$e_p = \sqrt{(x_b - x_p)^2 + (y_b - y_p)^2} \quad (\text{Eq. 19})$$

Este erro é responsável por incrementar a velocidade linear ( $v$ ) até a velocidade máxima determinada, quando a cadeira se encontra distante de seu do ponto B, e reduzi-la proporcionalmente à medida que se aproxima.

### Ir para o Ponto (x,y)

O objectivo do controlo *Ir para o ponto (x,y)* é permitir que a cadeira de rodas se mova desde a sua posição actual C ( $x_c, y_c$ ) até determinado ponto objectivo B ( $x_b, y_b$ ). Para isso, é traçado uma trajectória R definida a partir do segmento de recta [CB] com a orientação  $\alpha$  definida no Sistema Global de Coordenadas. Este controlador é uma simplificação do controlo *Seguir recta*, uma vez que a cadeira de rodas sempre se mantém sobre a trajectória R que é recalculada a cada ciclo com os dados de posição actualizados. A Figura 69 apresenta a representação esquemática deste controlador.

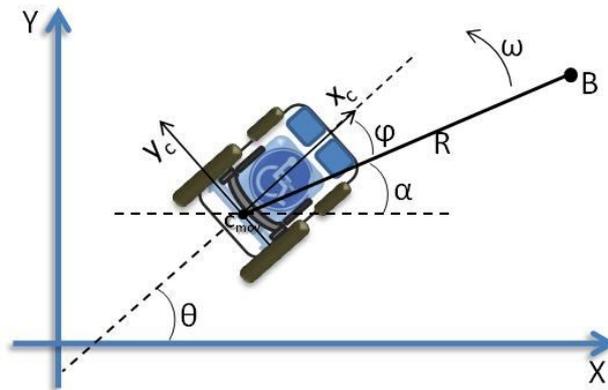


Figura 69: Representação esquemática do controlador para *Ir para o ponto*  $(x,y)$

O ângulo  $\varphi$  é o erro de orientação da cadeira de rodas, e consiste na diferença entre o ângulo  $\alpha$  do segmento [CB] e o ângulo  $\theta$  do robô no sistema de global coordenadas SGC. O valor de  $\varphi$  é determinado pela equação:

$$\varphi = \theta - \alpha \quad (\text{Eq. 20})$$

Para este controlador, o erro de orientação  $\varphi$  da cadeira de rodas mantém sua relação entre  $\alpha$  e  $\theta$ . Entretanto, a definição de  $\alpha$  sofre uma pequena alteração, sendo neste caso o ângulo formado entre segmento [CB] e o eixo da abcissas do SGC. Nesta modelação, o erro  $e_y$  é nulo, uma vez que o ponto C faz parte da trajectória R. Sendo assim, a velocidade angular ( $\omega$ ) do robô é definida utilizando-se apenas o erro de orientação e calculado através da equação:

$$\omega = -k_1 \varphi \quad (\text{Eq. 21})$$

O termo  $k_1$  representa o ajuste de ganho proporcional para este controlador. Neste caso, a CRI roda sobre si própria até estar quase a apontar para o ponto destino e depois chama-se o controlador de *Seguir recta* recalculando a recta em cada iteração.

### Girar para ( $\theta$ )

O controlo para posicionar a cadeira num determinado ângulo é um algoritmo muito simples, mas não menos importante que os demais. Este é utilizado em situações em que é necessário apenas um movimento angular da cadeira de rodas, como na orientação inicial de trajectórias e em certas correcções de rotas. Este controlo é uma simplificação do controlo *Ir para o ponto*  $(x,y)$ , onde a cadeira já se encontra no ponto objectivo e falta apenas ajustar sua orientação. O ângulo  $\varphi$  necessário para a correcção da orientação da cadeira de rodas é calculado pela equação:

$$\varphi = \theta - \alpha \quad (\text{Eq. 22})$$

Onde  $\theta$  é o ângulo da posição da cadeira de rodas no SGC e  $\alpha$  o ângulo de posição de referência (ver Figura 70).

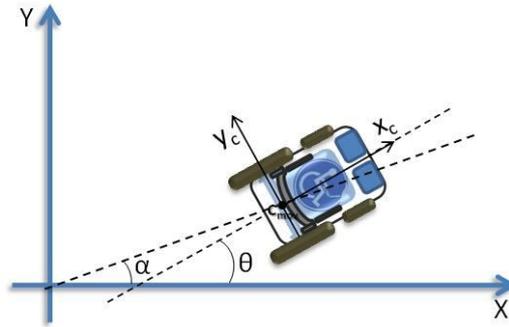


Figura 70: Representação esquemática do controlo girar para ( $\theta$ )

Neste controlador a velocidade angular  $\omega$  é calculada pela equação:

$$\omega = -k_1 \varphi \quad (\text{Eq. 23})$$

Onde  $k_1$  é o ganho de ajuste. Neste caso, a velocidade linear ( $v$ ) do robô móvel é nula.

### **Seguir Parede**

O controlo *Seguir parede*, ao contrário dos controlos anteriores, não depende da localização global da CRI. O controlo é realizado com base nas medidas obtidas dos sensores localizados na parte lateral da CRI. Desta forma é possível obter a distância e o ângulo da CRI em relação à parede.

A diferença entre a distância de referência ( $d_R$ ), distância desejada entre a cadeira e a parede, e a distância medida pelo sensor frontal lateral resulta no erro de distância frontal da CRI ( $e_F$ ) e é calculada através da equação:

$$e_F = d_R - \text{Sonar}_F \quad (\text{Eq. 24})$$

Da mesma forma, o termo  $e_d$  representa o erro da diferença entre as medidas do sensor frontal lateral ( $\text{Sensor}_{FL}$ ) e o traseiro lateral ( $\text{Sensor}_{TL}$ ) e é calculado através da equação:

$$e_d = \text{Sonar}_{FL} - \text{Sonar}_{TL} \quad (\text{Eq. 25})$$

Para calcular a velocidade angular ( $\omega$ ) da CRI utiliza-se a equação:

$$\omega = k_1 e_F - k_2 e_d \quad (\text{Eq. 26})$$

Para ajustar a dinâmica do controlo são utilizados dois ganhos de ajuste,  $k_1$  e  $k_2$ , onde o primeiro parâmetro tem influência no erro da distância da CRI em relação à parede e o segundo parâmetro está relacionado com a perpendicularidade da CRI em relação à parede e ajuda a suavizar a reacção do controlador. A velocidade linear ( $v$ ) é um parâmetro independente e constante durante o deslocamento da CRI, este pode ser determinado pelo utilizador, como representado na equação a seguir.

$$v = v_n \quad (\text{Eq. 27})$$

Onde  $v_n$  é a velocidade nominal determinada pelo utilizador.

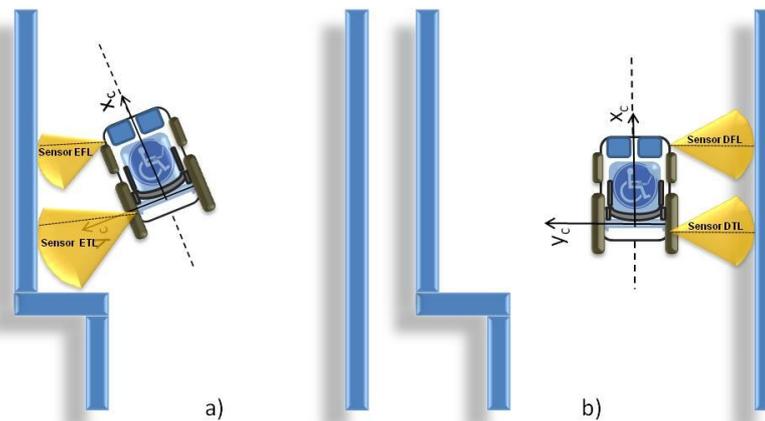


Figura 71: Esquema da utilização dos sensores laterais para o controlo de *Seguir paredes*

A Figura 71 ilustra duas situações diferentes na utilização dos sensores para seguir paredes. Em (a) a cadeira está a aproximar-se da parede esquerda e, neste caso, utiliza-se apenas a informação dos sensores laterais esquerdos. Em (b), a CRI encontra-se a seguir a parede pela direita e neste caso as informações dos sensores a ser utilizada são apenas dos sensores do lado direito. Entretanto, em qualquer dos casos, deve-se utilizar o desvio automático de obstáculos.

### ***Desvio de obstáculos***

O algoritmo de desvio de obstáculos é baseado na arquitectura de subsunção. Este algoritmo possui camadas de comportamentos como inversão de marcha, desvia para esquerda, desvia para direita entre outros. Estes comandos estão organizados de forma hierárquica sendo que um comportamento quando activado pode inibir os comportamentos de níveis mais baixos. Neste caso, o nível mais baixo da hierarquia é o comando do utilizador. Este algoritmo foi utilizado também para o modo autónomo, neste caso, o comando oriundo do utilizador foi substituído pelo comando do controlo autónomo. Por exemplo, quando o utilizador, por descuido, envia um comando que pode causar uma colisão este é suprimido e o comando de desvio de obstáculo é enviado ao motor, como representado na Figura 72.



Figura 72: Arquitectura de subsunção utilizada no controlo de desvio de obstáculo

## 4.6 Interface Multimodal

Uma *interface* é um elemento que estabelece limites entre duas entidades. Actualmente, a maioria das interfaces homem-máquina tradicionais tem por base a correlação de uma entrada com uma saída (uma relação de um para um e não personalizável). A evolução deste paradigma, e uma maneira mais natural de criar uma interacção com o utilizador, é o estabelecimento de uma interacção multimodal, que contempla uma ampla gama de formas e canais de comunicação, como vídeo, voz, caneta, etc. Segundo Oviatt (Oviatt, 2000) uma Interface multimodal (*MultiModal Interface* – MMI) "Processa dois ou mais modos de entrada do utilizador (como a fala, caneta, toque, gestos manuais, posição dos olhos, movimentos da cabeça e do corpo) de forma coordenada com a saída do sistema multimédia. As interfaces multimodais são uma nova classe de interfaces que visam reconhecer formas naturais da linguagem humana ou comportamentos, e que incorporam uma ou mais tecnologias com base no reconhecimento".

O módulo MMI da PIW foi projectado para permitir a conexão de diversos dispositivos de entrada simultaneamente (voz, expressões faciais, gestos da cabeça, teclado touch-screen e joystick) (Reis et al., 2009). Desta forma é possível comandar a cadeira de rodas através de uma sequência de entrada do mesmo canal de comunicação ou mesmo a partir da combinação de canais distintos (dispositivos de entrada). Através deste módulo, os utilizadores podem criar uma sequência de entrada mais adequada com as suas limitações (o que pode estar associado a um ou mais comandos de saída). Além disso, esta aplicação pode proporcionar uma interacção entre o ambiente e os métodos de entrada, de modo que em qualquer caso a informação de entrada pode ser analisada e verificada se é confiável, para garantir a segurança do utilizador.

A interacção entre o módulo MMI e o driver de dispositivo de entrada é baseado numa arquitectura cliente / servidor, onde o módulo MMI actua como um servidor e o driver de dispositivo de entrada como cliente. Durante a conexão, a MMI requisita informações do driver de dispositivo de entrada quanto as suas características (por exemplo: nome, tipo e número de entradas, etc.). Uma vez que a conexão é estabelecida, quando este dispositivo de entrada é utilizado o driver envia o novo estado do dispositivo para o módulo MMI. A conexão entre o driver do dispositivo de entrada e o módulo MMI está representado na Figura 73.



Figura 73: Arquitectura do módulo MMI

O módulo MMI tem por base 3 listas dinâmicas:

- **Acções:** Lista de acções de alto-nível disponibilizada pelo agente Controlo. Esta lista é sempre actualizada quando as habilidades do sistema são incrementadas ou reduzidas (Figura 74);
- **Dispositivos:** Lista de dispositivos de entrada regista todas as entradas activas no agente interface (Figura 74);
- **Associações:** Lista que associa as entradas (por exemplo, sequência de botões) com as saídas (acções). O sistema permite criar novas associações e também definir uma relação entre uma sequência de entradas e uma dada acção de alto-nível (Figura 75).

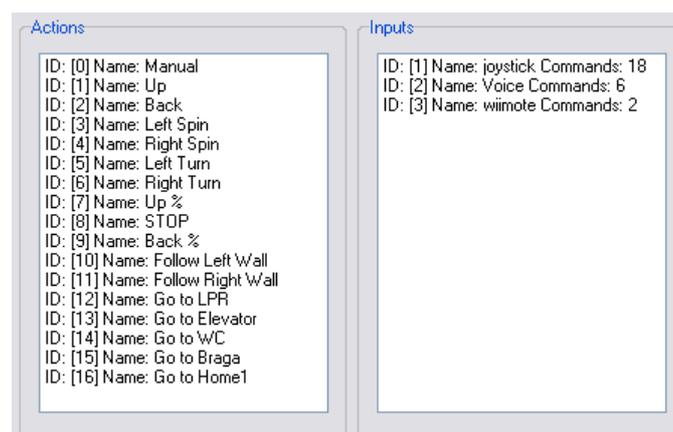


Figura 74: Listas de acções e dispositivos de entrada disponíveis

Sequence List		
	Sequence	Action
1	#1.1T#1.1F	1
2	#1.1T#1.1F#1.14%	7
3	#1.2T#1.2F	2
4	#1.2T#1.2F#1.14%	9
5	#1.3T#1.3F	3
6	#1.3T#1.3F#1.5T#1.5F#1.14%	5
7	#1.4T#1.4F	4

Figura 75: Lista que armazena as associações entre as seqüências de entradas e saídas

Sempre que uma entrada é alterada esta informação é enviada à MMI que faz a busca na lista das associações. Se existir alguma acção associada com a seqüência de sinais de entrada então o agente interface envia ao agente controlo o pedido de execução de tal acção. A Figura 76 apresenta seis exemplos de seqüências, em a) uma seqüência simples de entrada (carrega e solta o botão 3 do dispositivo de entrada 1) que está associada com a acção girar no próprio eixo no sentido anti-horário, em b) uma seqüência errada e as demais imagens apresentam outros exemplos.

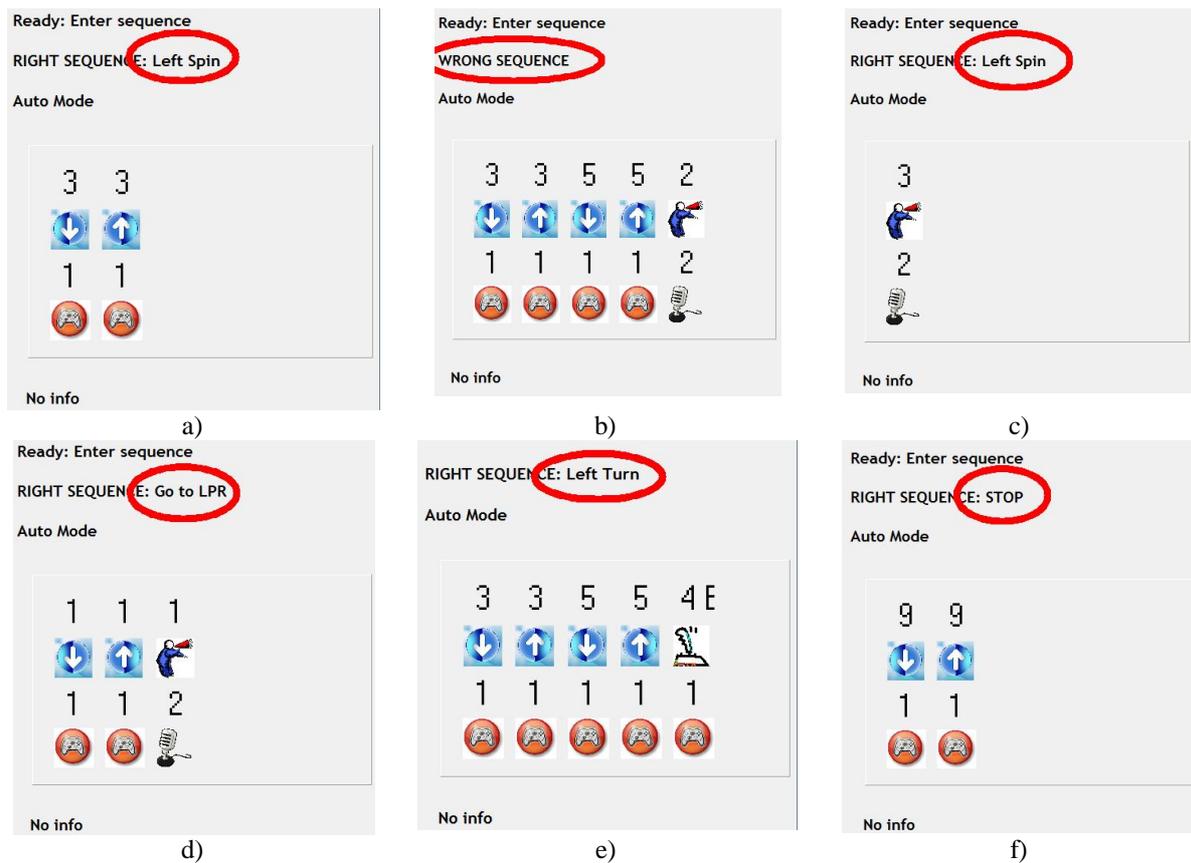


Figura 76: Exemplos de uma seqüência de entrada

Os itens da Lista de associações são habilitados ou desabilitados de acordo com a disponibilidade dos dispositivos de entrada, com as acções disponíveis pelo controlo. Em

trabalho futuro, poderá ser desenvolvido algoritmo que avaliará as condições do ambiente, habilidades do utilizador, desta forma será possível habilitar ou desabilitar o gatilho das acções de acordo com o resultado desta avaliação.

## 4.7 Sistema Multi-agente

Uma cadeira de rodas verdadeiramente inteligente deve ser capaz de comunicar e interagir com outros dispositivos inteligentes, como por exemplo, outras cadeiras de rodas inteligentes, sistemas de accionamento de portas, elevadores, entre outros. Para atender este requisito foi estudada a metodologia dos Sistemas Multi-Agente (SMA). Esta metodologia permitiu implementar o conjunto de funções descrito a seguir:

- **Comportamento cooperativo entre um grupo de CRI:** Nesta funcionalidade as CRI podem negociar as tarefas de deslocamento de um determinado número de pacientes. Neste caso, o resultado seria um escalonamento dinâmico em tempo real. Outra possibilidade é a de cooperar no planeamento automático das acções para realizar uma determinada tarefa, assim como, comparar planos para que uma cadeira não tenha acções que possam entrar em conflito com o plano de uma outra cadeira. Um exemplo de conflito, seria a necessidade das duas cadeiras passarem pelo mesmo corredor apertado ao mesmo tempo.
- **Comportamento colaborativo entre utilizador e outros sistemas:** Esta funcionalidade é mencionada na literatura sobre SMA (Wooldridge, 2001)(Gonzalez et al., 2006). Neste caso, o utilizador e a CRI poderão colaborar em comandos compartilhados, como no caso da navegação em modo seguro, modo semi-automático e até mesmo, na colaboração da construção de planos para a navegação automática. Outros dispositivos, como por exemplo portas automáticas, pode colaborar numa tarefa onde a cadeira necessita trocar de sala. Neste caso a porta irá realizar as acções de abrir e fechar de forma colaborativa e automática.

Uma outra característica desejável que os SMA apresentam é a facilidade de integrar novos recursos, serviços, funcionalidades ao sistema, além de facilitar a criação de redundância no sistema, o que resultará em maior robustez (uma característica muito interessante em sistemas críticos).

Por estes motivos, no desenvolvimento da PIW foi utilizado o paradigma de SMA. Pode-se considerar o SMA desenvolvido como um agrupamento de vários módulos distribuídos em agentes autónomos (micro agentes) que, por sua vez, numa visão global, formam um único agente do SMA (macro agentes). Assim sendo, um conjunto base de agentes (Controlo, Inteligência, Interface e Percepção) denominados micros agentes formará os agentes embarcados em cada CRI e a sua coligação formará um macro agente. Este conjunto de

agentes, através da interacção com o utilizador e a comunicação entre si próprios, agentes de outras cadeiras e outros dispositivos “inteligentes” poderão colaborar, cooperar, trocar conhecimento, planos, desejos, etc.

Existem diversos projectos de investigação voltados para as CRI e protótipos de CRI desenvolvidos para testar e validar teorias. No entanto, poucos trabalhos publicados utilizam metodologias de planeamento inteligente sob o paradigma de SMA. Este trabalho, pelo contrário, aborda simultaneamente o uso de SMA, de forma macro e micro, e ao mesmo tempo apresenta a modelização formal do SMA.

O SMA desenvolvido permite controlar e coordenar CRI tanto no mundo real como num mundo virtual. Este sistema de controlo para agentes autónomos e semi-autónomos opera de forma colaborativa e cooperativa em qualquer tipo de ambiente onde possa ser interessante o uso de múltiplas cadeiras de rodas, tais com ambientes hospitalares ou centros de paralisia cerebral. Os outros objectivos em utilizar o paradigma de SMA são: facilitar a integração de novos agentes com novas habilidades; permitir que emergja uma auto-organização e comportamentos complexos do sistema através de estratégias individuais de todos os seus agentes simples.

A arquitectura do SMA proposta para a plataforma de desenvolvimento de CRI foi projectada para implementar e facilitar a utilização de vários módulos independentes. Os principais agentes projectados foram:

- 1) **Agente Inteligência:** Este agente foi projectado para implementar o módulo de planeamento que é responsável pela camada de estratégia. É nesta camada que as decisões de alto nível são tomadas, assim como um possível planeador contínuo, sistema para monitorar em tempo real a execução do plano e a cooperação com outros agentes. O agente Inteligência possui um planeador automático responsável por criar uma sequência de acções de alto nível necessárias para atingir a meta global (com base num algoritmo de planeamento inteligente). Além disso, esse agente também é responsável por gerar planos de acção com sequências de acções básicas (algoritmo de planeamento do caminho).
- 2) **Agente Controlo:** O agente Controlo implementa a camada táctica que inclui um controle das acções básicas (por exemplo, seguir recta, girar, seguir parede, ir para o ponto (x,y)) e gerar as referências (calcular as velocidades linear e angular da CRI). Este agente também implementa a camada de controlo básico, responsável pelo controlo de velocidade em baixo nível.
- 3) **Agente Interface:** O agente Interface é responsável por colectar as informações das entradas com utilizador (por meio do módulo de interface Multimodal) e por exhibir as informações mais relevantes (por exemplo, distância dos obstáculos, velocidade, posição) através de uma interface gráfica com utilizador (GUI, do inglês *Graphical*

*User Interface*). Além disso, é responsável por fazer a interação entre o utilizador e os outros agentes do sistema, encaminhando os pedidos do utilizador para o agente mais adequado.

- 4) **Agente Percepção:** Este agente representa o sistema de percepção do robô móvel. Os seus objectivos incluem ler o sensor adequado e actualizar a representação do mundo interno, mapeamento e localização.

Outros agentes que completam o conjunto de agentes da PIW são designados por agentes de serviços. Estes agentes foram criados com o objectivo principal de auxiliar o sistema de CRI a atingir os seus objectivos globais.

Os agentes de serviço podem cooperar e colaborar com os agentes que estão embarcados no robô móvel (CRI). O **Agente Porta** é responsável por controlar as portas e portões no ambiente CRI, abrindo e fechando as portas para permitir ou impedir acesso em áreas restritas. O **Agente Registo** é responsável por criar os arquivos de registo permanente sobre as mensagens trocadas entre agentes, seus estados, com a finalidade de auxiliar no processo de depuração e análise do sistema. O **Agente Controlador de Acções das CRI** é responsável por centralizar o controlo de todo o tráfego no ambiente das CRI, desta forma, ele busca evitar os conflitos de tráfego. O papel deste agente é monitorizar todas as actividades e agir sempre que necessário para evitar eventuais conflitos e situações sem saída. O **Agente Assistente** é responsável pela interação humano/sistema, recebendo e tratando os objectivos globais. Este agente é a interface entre enfermeiros, médicos, terapeutas e assistentes com o sistema de CRI.

### 4.7.1 Plataforma de Comunicação

Algumas das restrições aplicáveis aos robôs móveis são as comunicações seguras em sistemas de transmissão abertos, navegação segura e evitar obstáculos. A forma de comunicação dos sistemas está em constante evolução devido à proliferação dos dispositivos e das tecnologias sem fios, enquanto estas novas tecnologias trazem vantagens elas também apresentam desvantagens, especificamente no domínio dos sistemas de segurança ou de segurança crítica, onde, na eventualidade de uma falha, possa causar danos às pessoas, nas suas propriedades ou para o ambiente (Malm et al., 2007), (Fowler, 2004).

Se um robô móvel é um sistema crítico de segurança ou parte de um, o sistema de comunicação deve evitar falhas e mostrar-se seguro para acesso não autorizado, mantendo o nível desejado de compatibilidade com as camadas de transmissão do meio de comunicação físico disponível. Para enfrentar e resolver esses problemas, determinados standards (por exemplo a *CENELEC EN 50159-2*) (CENELEC EN 50159-2, 2001) devem ser seguidos. Estes standards descrevem as ameaças conhecidas para comunicações e os métodos de defesa aplicáveis para sistemas críticos de segurança que usam camadas de transmissão em meios de comunicação abertos.

Normalmente, uma plataforma multi-agente como o *Java Agent DEvelopment Framework* (JADE) (Bellifemine et al., 2007) poderia ser utilizada para permitir as comunicações e organizar os diferentes agentes. No entanto, com as plataformas multi-agente comuns não é possível personalizar e aprimorar algumas funcionalidades do sistema, como, por exemplo, adaptar o sistema para os problemas de um sistema crítico de segurança. A solução, usada na PIW, para este problema foi desenvolver novos métodos de comunicação no âmbito de uma nova plataforma multi-agente.

Na implementação do sistema de comunicação da PIW seguiram-se as directrizes FIPA (*Foundation for Intelligent Physical Agents*), assim como a linguagem de comunicação de agente (ACL, do inglês *Agent Communication Language*) e um conjunto de serviços tal como um sistema de gestão de agente, um sistema de transporte de mensagens e um directório facilitador (Cunha et al., 2010). A arquitectura de comunicação, ilustrada na Figura 77, foi projectada com cinco camadas, sendo possível escolher quais as camadas que devem ser utilizadas na aplicação pelo utilizador sem comprometer a funcionalidade do agente. Esta arquitectura seguiu o modelo de Referência OSI (*Open Systems Interconnection*) e implementação de métodos de tolerância a falhas descritos em (CENELEC EN 50159-2, 2001) e (IEC 60812, n.d.). Além disso, o sistema aceita a adição de novas camadas na arquitectura, o que permite personalizar a arquitectura do sistema, e desta forma é possível adaptá-la a um domínio específico ou para melhorar os serviços prestados.

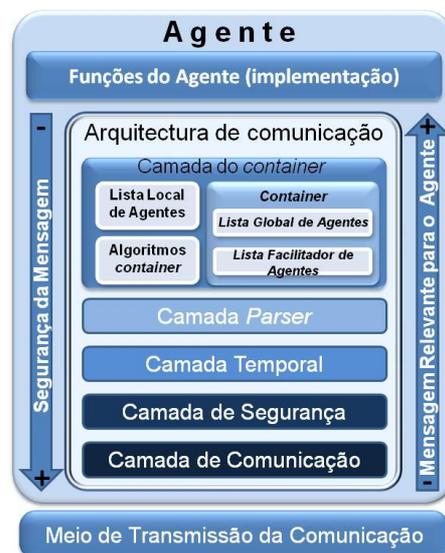


Figura 77: Arquitectura multi-camadas da plataforma de comunicação

As principais características desta arquitectura são a eleição de uma entidade designada por *container*, semelhante ao que existe na plataforma JADE, e a distribuição de uma lista de agentes local (LAL), bem como uma lista de agentes global (LAG), assim como o uso do paradigma orientado à mensagem. Estas listas contêm configurações dos aplicativos que permitem a comunicação e a distribuição da chave pública de criptografia entre os agentes. O *container* foi projectado para ser responsável pelas operações de manutenção das listas, que

incluem criação, actualização e exclusão. No entanto, e ao contrário de outros sistemas, o *container* não foi projectado como uma entidade separada ou como a base para a criação de agentes e das suas actividades. A principal motivação para isso é evitar que a admissível e provável situação na qual uma cadeira de rodas possa perder a conectividade com a rede, ou simplesmente alterar sua configuração de rede, possa causar o mau funcionamento do sistema. Ao adicionar as funções directamente aos agentes, permite-se a auto reconfiguração do sistema (caso o agente responsável pelo *container* pare de funcionar por motivo qualquer) e a eleição de um novo *container* entre os agentes remanescentes é realizada.

Para permitir este comportamento de reconfiguração, os agentes devem interagir entre si e reconhecer qual dos agentes presentes é mais adequado para desempenhar o papel do *container*. O agente eleito para tal papel fica responsável em reconfigurar o sistema e gerir os serviços de páginas brancas e amarelas.

A Camada de Comunicação é responsável por receber e enviar mensagens de/para a camada de transporte de mensagem. Esta camada permite que o utilizador escolha entre TCP/IP, UDP ou até mesmo mensagens http. Essa camada também impede a interpretação de mensagens repetidas presentes no meio físico de comunicação e permite a retransmissão de mensagens evitando perda de pacote no nível da rede. Outra função importante é a de impedir que o aplicativo receba mensagens com um tamanho maior do que o especificado pelo utilizador durante a implementação do agente.

A Camada de Segurança é responsável pela segurança da mensagem, impedindo a interceptação e modificação das mensagens. O método de criptografia é escolhido em conformidade com o destino da mensagem e conhecimento das plataformas no ambiente. Os métodos de criptografia envolvem o uso de uma chave privada e par de chaves públicas ou uma chave AES (*Advanced Encryption Standard*) pré-compartilhada. Cabe ainda a esta camada a verificação da integridade da mensagem.

A Camada Temporal é responsável pela inserção de restrições temporais para as mensagens. Estas restrições podem ser vistas como uma medida defensiva. Ao adicionar um selo temporal aos dados de uma mensagem é possível filtrar mensagens desactualizadas. Este método pode também servir como controle da sequência das mensagens. Para facilitar um processo de construção do conhecimento correcto, a camada temporal do *container* força o sistema operacional a sincronizar o seu relógio com um relógio da rede. Os sistemas operacionais actuais já utilizam algoritmos de tempo padronizado de sincronização como o *Network Time Protocol* (NTP) e *Precision Time Protocol* (PTP). Logo, através destes recursos os agentes podem trabalhar de forma sincronizada em *container* diferentes.

Finalmente, a camada *Parser* é responsável pela construção de acordo com o padrão FIPA-ACL (FIPA, 2009) e escrita utilizando a norma FIPA-SL (FIPA, 2009) da mensagem. Ela também selecciona as mensagens que deverão ser aceites de acordo com a estrutura

correcta e com a presença do remetente na plataforma, impedindo assim qualquer comunicação de um aplicativo não autenticado.

Com o objectivo de cumprir o padrão FIPA, um sistema deve fornecer o serviço de páginas brancas e amarelas. O serviço de páginas brancas facilita o acesso dos agentes aos demais agentes da plataforma, enquanto as páginas amarelas permitem a consulta de serviços fornecidos pelos agentes presentes na plataforma. Como descrito anteriormente, o agente com o papel de *container* fica responsável por criar, manter e distribuir estas listas entre os agentes presentes no sistema.

Na implementação, a Lista de Agentes é composta por diferentes listas (Cunha et al., 2010):

- Lista de agentes locais (LAL) – responsável pela manutenção configuração dos agentes locais. Esta lista é criada após a eleição do *container* local;
- Lista de agentes global (LAG) – o seu objectivo é fornecer aos agentes locais a configuração dos agentes remotos, ou seja, agentes que estão noutra *container* (computador). O seu conteúdo não deve incluir as informações fornecidas pela LAL;
- Lista de agentes facilitadores (LAF) – é responsável por armazenar e apresentar a configuração do facilitador do sistema.

Por simplicidade, na implementação dos sistemas propostos, as cinco camadas descritas anteriormente foram implementadas como bibliotecas (componentes) individuais, como mostrado na Figura 78, isto permite que sejam adicionadas ao formulário (*form*) de projecto para criar a base de comunicação do agente.

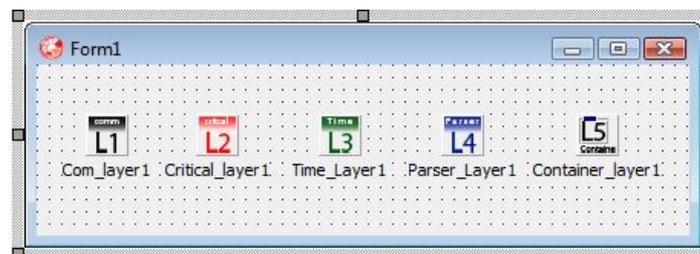


Figura 78: Bibliotecas que compõem a plataforma de comunicação da PIW

Ao definir um modelo padrão de camada torna-se possível estender o número de camadas. Deste modo pode-se criar e executar novas camadas, o que vem facilitar a sua integração com as camadas já desenvolvidas. O objectivo principal deste modelo padrão foi abstrair o emissor e receptor das mensagens. Com isso, a configuração das camadas necessárias para serem utilizadas numa certa aplicação resume-se apenas a configurar a sua ordem, bem como se será utilizado processamento paralelo da mensagem, ou não, nesta camada. Este modelo padrão serve de base funcional. As funções da arquitectura proposta (Figura 77) foram implementadas e a estrutura detalhada é mostrada na Figura 79.

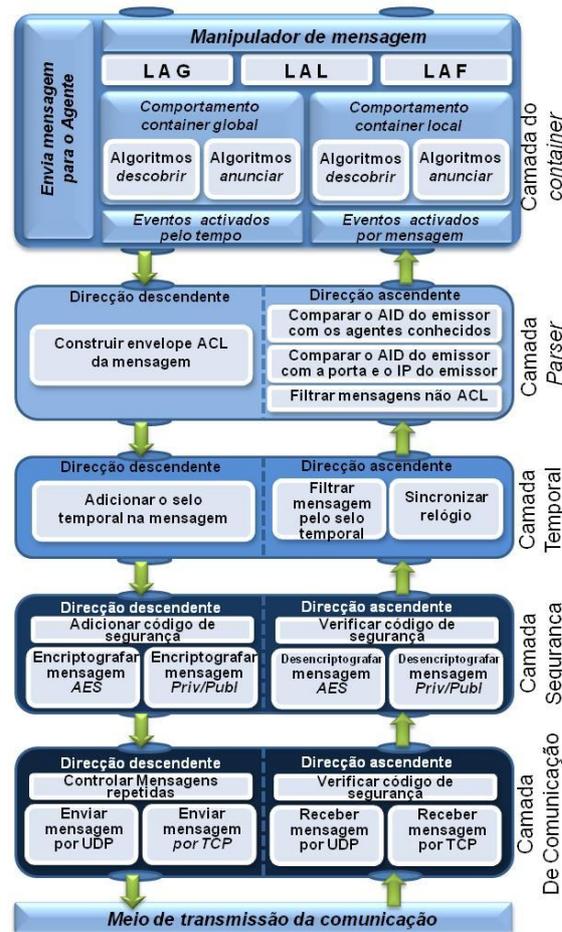


Figura 79: Camadas de comunicação e suas funções

Outro aspecto desses componentes é que a mensagem é transmitida no formato XML, representado pelas setas verdes na Figura 79.

## 4.7.2 Modelização do SMA

Nesta subsecção são apresentados os requisitos e a modelização do SMA que compõe a PIW. Em seguida, são apresentados também a análise detalhada e o projecto do modelo do SMA seguindo a metodologia GAIA (Wooldridge et al., 2000). Inicialmente, é introduzido um metamodelo para um sistema de agente robótico genérico utilizando o SPEM (*Software Process Engineering Metamodel*) (Group, 2005) (Silva et al., 2010). O objectivo de desenvolver este metamodelo foi evitar ou reduzir tarefas repetitivas comuns para sistemas que possuem as mesmas características estruturais. Esta redução na tarefa de desenvolvimento, para estes tipos de sistemas, pode ser alcançada por meio de uma metodologia que suporta a fase de projecto da arquitectura do sistema como a metodologia GAIA. A metodologia GAIA pode ser descrita como uma metodologia genérica que exclui o levantamento de requisitos e implementação, concentrando-se na análise e concepção do sistema (Wooldridge et al., 2000). Embora a metodologia GAIA tenha sido originalmente

destinada à concepção de sistemas organizacionais, esta também pode ser utilizada na concepção de sistemas abertos.

Também foi introduzido um metamodelo para uma versão estendida da metodologia GAIA, usando uma linguagem para modelação de processos de Software SPEM. Este metamodelo pode ser considerado como um contributo original que proporciona uma maior formalização e também vem facilitar o entendimento da metodologia GAIA.

### 4.7.2.1 Metamodelo com GAIA-SPEM

Neste trabalho, foi utilizado o SPEM versão 2.0 (Group, 2005) e a versão modelada de GAIA foi baseada na GAIA v.2 (conforme descrito na secção 2.4.2). Foram ainda utilizadas funções e diagrama de interacção, como proposto em (Castro & Oliveira, 2008).

A Figura 40 mostra uma lista de alguns dos estereótipos conforme definido pelo SPEM 2.0. Embora a “UML/ modelo formal” não seja definido pela especificação SPEM 2.0, este foi incluído como um estereótipo. Considerou-se útil ter esse estereótipo para ajudar a identificar produtos de trabalho com uma apresentação formal (diagramas UML ou outros modelos estruturados formalmente).

Representação	Estereótipo	Representação	Estereótipo
	<i>WorkProduct</i> (Artefacto ou produto do trabalho)		<i>Activity</i> (Atividade)
	<i>UML/Formal Model</i> (Modelo Formal em UML)		<i>Phase</i> (Fase)
	<i>RoleDefinition</i> (Definição do Papel)		<i>ProcessPackage</i> (Pacote de processos)
	<i>TaskDefinition</i> (Definição da Tarefa)		

Figura 80: Estereótipos mais utilizados

A metodologia GAIA foi aplicada em quatro estágios: colecta de dados, análise, projecto de arquitectura e projecto detalhado, como mostrado na Figura 81.

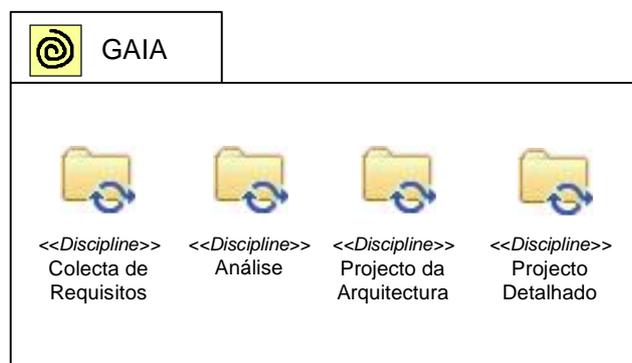


Figura 81: Estágios da metodologia GAIA

Embora o primeiro estágio (Colecta de Requisitos) não faça realmente parte da metodologia GAIA, representa-se este estágio para apresentar formalmente o documento de

declaração de requisitos, ilustrado pela Figura 82a, que é considerada a base para as etapas restantes, ou seja, a entrada de dados da metodologia GAIA.

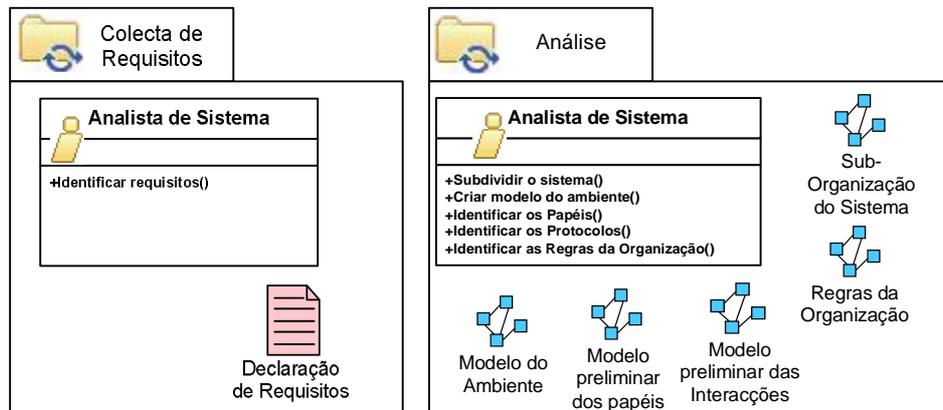


Figura 82: a) Pacote de Requisitos; b) Pacote de Análise

O segundo estágio (Análise) tem o objectivo de desenvolver uma visão geral (entendimento) do sistema e sua estrutura, o que resulta na produção de um total de cinco produtos de trabalho (modelos que podem ser expressos com o uso de modelos de bancos de dados ou descrições textuais informais), tais produtos são representados na Figura 82b.

O estágio de Projecto da arquitectura destina-se a transformar os modelos de análise num nível de abstracção suficientemente baixa para que técnicas tradicionais de desenvolvimento de projecto possam ser aplicadas a fim de implementar os agentes. O projecto da arquitectura envolve quatro produtos de trabalho (incluindo as funções e o diagrama de interacção, conforme proposto em (Castro & Oliveira, 2008), este pacote é ilustrado pela Figura 83a.

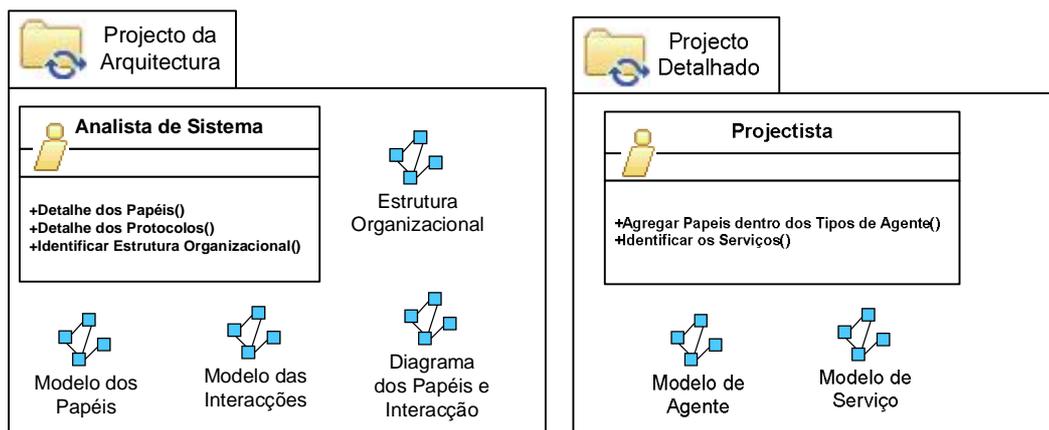


Figura 83: a) Pacote projecto da arquitectura; b) Pacote projecto detalhado

Finalmente, o estágio Projecto Detalhado envolve dois produtos de trabalho (Figura 83b): modelo de agente e modelo de serviço.

A Figura 84 apresenta a inter-relação entre os estágios do processo GAIA, e o grupo de documentos produzidos em cada estágio.

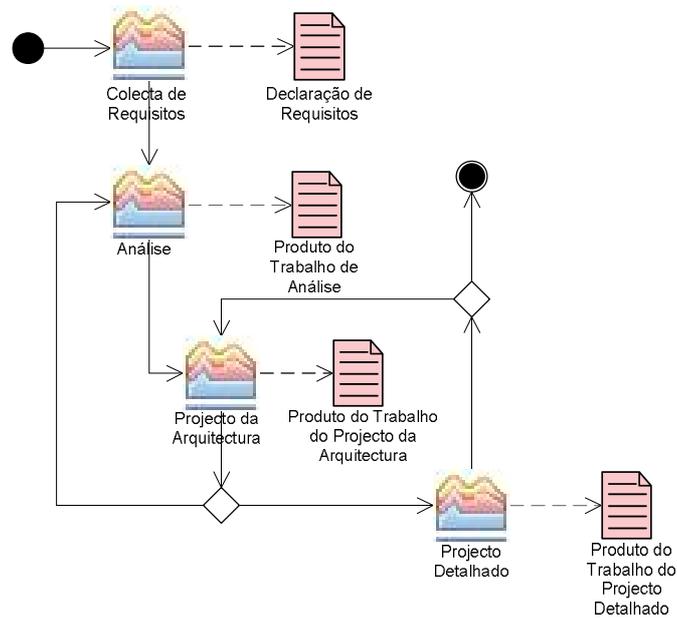


Figura 84: Processo GAIA

O estágio Análise identifica as sub-organizações presentes no sistema, produz um modelo do ambiente, uma versão preliminar dos papéis e modelo preliminar de interacção (padrões de interacção entre diferentes papéis) e as regras de organização (Figura 85).

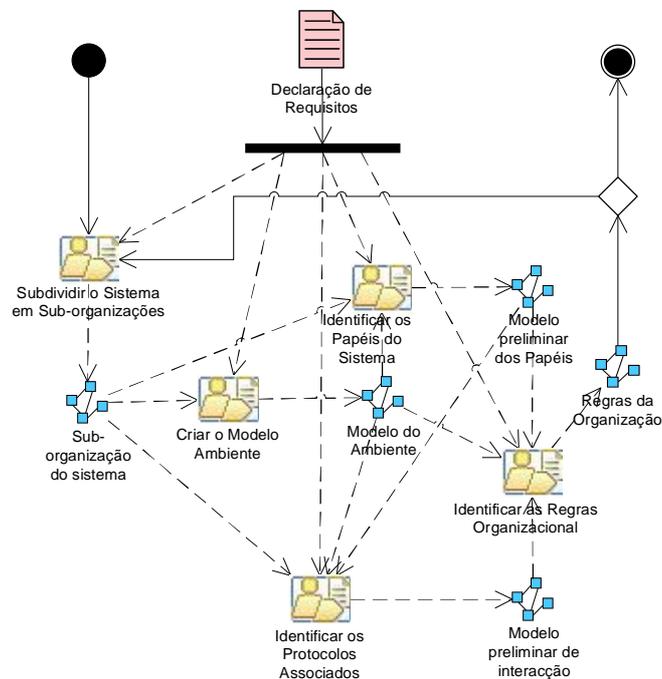


Figura 85: Fases da análise

A fase do Projecto da Arquitectura identifica a estrutura organizacional, detalha os modelos dos papéis, os modelos de interacção e cria o diagrama dos papéis e interacção (ver Figura 86).

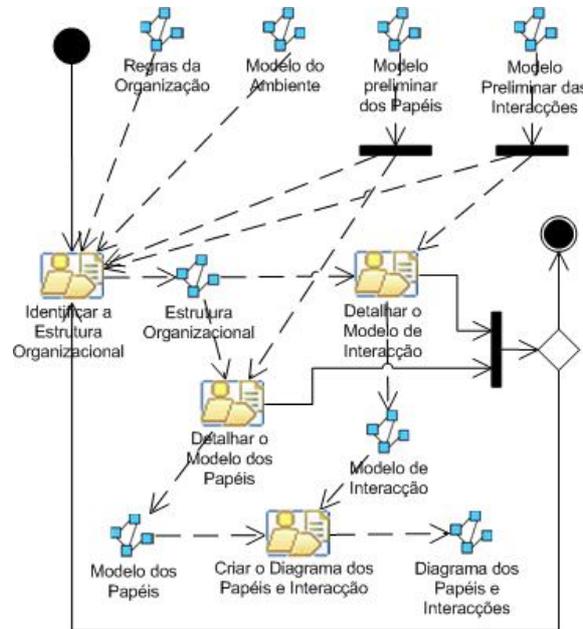


Figura 86: Estágio Projecto da Arquitectura

A fase que detalha o projecto resulta em dois modelos: o modelo de agente e o modelo de serviços. O modelo de agente identifica os tipos de agente que compõem o sistema e as instâncias desses tipos agente no sistema. O modelo de serviços identifica os principais serviços que são necessários para cada papel (Figura 87).

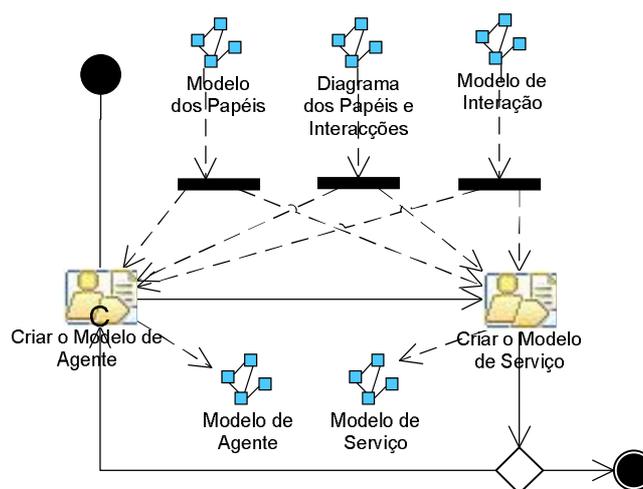


Figura 87: Estágio Projecto Detalhado

### 4.7.2.2 Captura de Requisitos

Existem muitos projectos de sistema multi-agente compostos por robôs móveis, onde a maioria destes sistemas possui requisitos (de alto nível) semelhantes, o que permite o desenvolvimento de um metamodelo comum que agregue todos esses requisitos e possa ser usado como base para o desenvolvimento mais rápido de um sistema específico. Desta forma, os próximos dois parágrafos fazem uma descrição mais genérica sobre agentes robóticos móveis com a finalidade de fornecer os requisitos mais genéricos.

A presença de agentes robóticos móveis é um dos requisitos mais genéricos nesses sistemas. Estes veículos robóticos normalmente têm vários sensores e actuadores, bem como capacidades de comunicação (por exemplo, robôs enfermeiros, AGVs (Automatic Guided Vehicle), cadeiras de rodas inteligentes, robôs futebolísticos (RoboCup), aviões, barcos, submarinos e muitos outros). Esses robôs são geralmente autónomos, mas um grande grupo destes sistemas exige a possibilidade que um humano possa assumir ou compartilhar o controlo. Estes veículos geralmente são usados para executar tarefas ou missões, que podem ser delegadas a um único robô ou a um grupo de robôs ou até mesmo ser realizada de forma distribuída. Para isso, os robôs móveis devem ser capazes de cooperar, otimizar os recursos e melhorar o desempenho da missão.

Como exemplo comum a outras aplicações de sistemas multi-agente, a comunicação entre os agentes deve seguir as orientações do FIPA, e um conjunto de serviços deve ser disponibilizado, ou seja, um sistema de gestão de agente, um sistema de transporte de mensagens e um directório facilitador (FIPA, 2009). Estes veículos robóticos costumam operar num ambiente dinâmico, que normalmente é apenas parcialmente acessível, e podem ser variados em termos de estrutura, presença de outros agentes (robóticos ou humanos) ou nível de inteligência de dispositivos – por exemplo: portas, janelas, luzes, ar condicionado, sistema de carregamento/descarregamento de carga (braço robótico) inteligentes, entre outros. Além dos agentes robóticos no ambiente, vários agentes utilitários são normalmente necessários, tais como um mecanismo de registo, um sistema (redundante e centralizado) para detecção e resolução de eventuais conflitos entre dois ou mais agentes e uma interface com seres humanos para permitir especificar as tarefas e missões.

Como resultado da primeira fase da metodologia GAIA, os requisitos do sistema são recolhidos e estruturados num documento que irá agir como a entrada para as fases seguintes. A Figura 88 ilustra a arquitectura geral do sistema (ainda sem formalismo), na qual se pode ver, de forma gráfica, a especificação dos requisitos do sistema. Esta arquitectura especifica um conjunto de quatro agentes básicos para compor as habilidades do robô móvel, um outro grupo de agentes mais genérico que podem colaborar com os diversos robôs móveis no sistema e ainda conta com a necessidade de uma plataforma para gerir o SMA com serviços básicos de sistema seguindo as especificações FIPA.

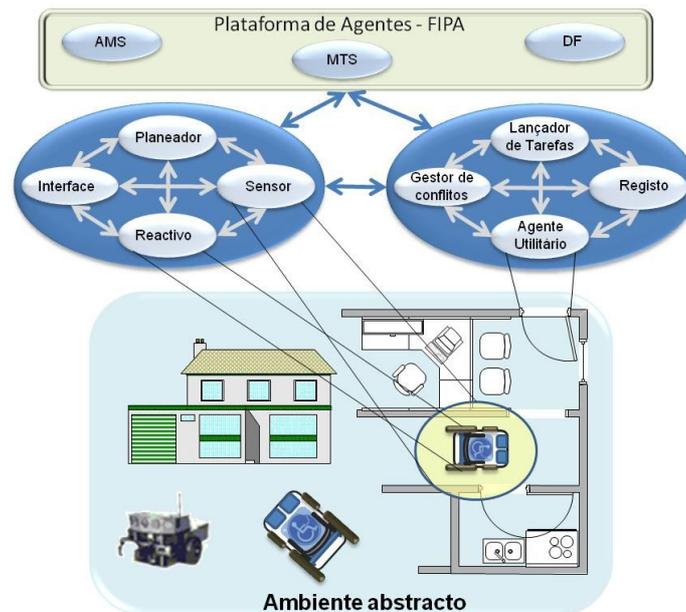


Figura 88: Arquitectura informal do sistema

### 4.7.2.3 Análise

#### *Sub-Organização do Sistema*

O primeiro resultado da fase de análise é a identificação de sub-organizações que constituem o sistema. Dada a natureza do sistema a ser implementado, a identificação das sub-organizações não é possível na perspectiva GAIA, uma vez que não há hierarquia estabelecida ou estrutura organizacional. Por outro lado, estes tipos de sistemas podem ser organizados em grupos funcionais, por semelhanças lógicas ou físicas, de acordo com a divisão representada na Figura 89.

Neste sistema, podemos identificar dois grupos diferentes de agentes. O primeiro, denominado neste trabalho como Robô Móvel (RM), que inclui quatro agentes que compõem e representam a plataforma robótica móvel. O segundo, chamado de Outros Serviços, que inclui agentes que executam várias tarefas no sistema, em nível global. Os agentes no primeiro grupo estão fortemente ligados (geralmente executando a bordo da plataforma robótica e formam um macro agente – como definido no início deste capítulo), enquanto os agentes do segundo grupo estão fracamente ligados. Uma instância do primeiro grupo (um RM) interage com outras instâncias do mesmo grupo (outros RMs) e com os agentes do segundo grupo (agentes que prestam outros serviços). Estes conceitos são representados graficamente pela Figura 89.

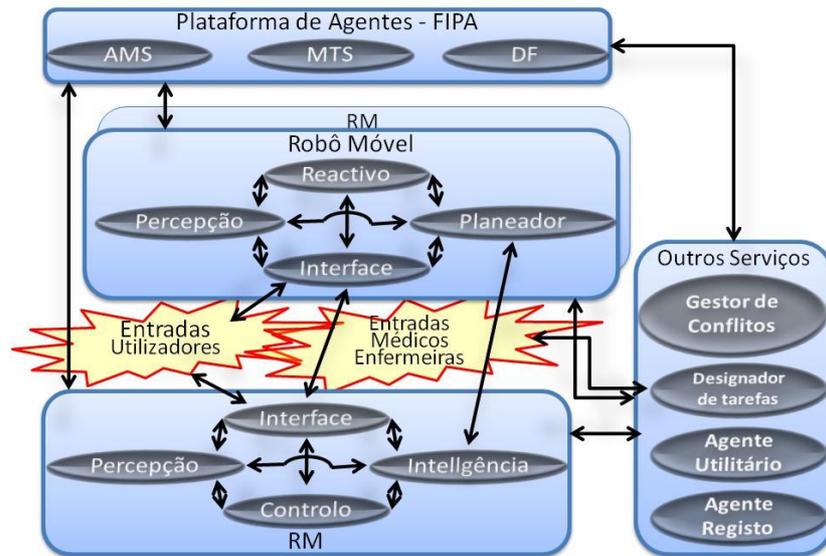


Figura 89: Organização do SMA e suas Sub-organizações

**Modelo do Ambiente**

Outro produto desta fase é o modelo do ambiente. Uma vez que o ambiente em que esses sistemas destinam-se a operar é o mundo real (e não um ambiente controlado, fechado), com todas as suas variáveis e incertezas, criar um metamodelo que represente um modelo de ambiente completo não é uma tarefa fácil. A Figura 90 apresenta, de forma genérica, um diagrama de recursos do ambiente, mostrando uma versão preliminar de funções existentes e recursos genéricos comuns neste tipo de ambiente. Cada sistema, implementando a partir deste metamodelo, deverá descrever melhor o ambiente em que ele irá operar e as particularidades possíveis de cada ambiente.

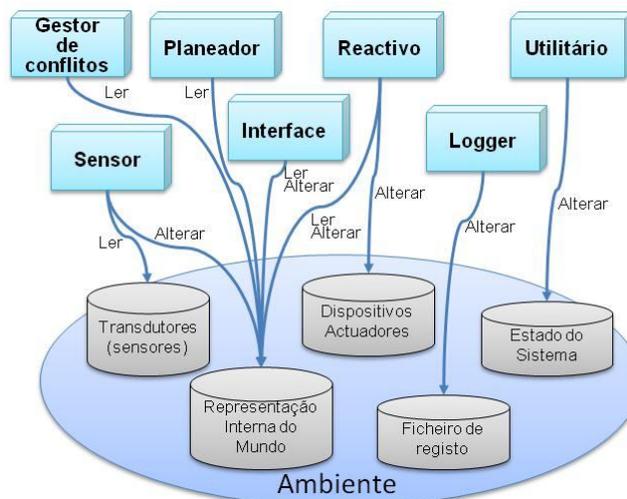


Figura 90: Diagrama de recursos do ambiente

### ***Papéis Preliminares e Modelo de Interacção***

Um total de nove papéis e seis protocolos foram identificados e introduzidos no metamodelo. Cinco desses papéis são incluídos no grupo do RM e os quatro restantes estão agrupados no grupo dos outros serviços. Os modelos de interacção são apresentados na próxima subsecção.

### ***Regras de Organização***

Outro resultado da fase de análise é as regras organizacionais. Dada a natureza desses sistemas e o carácter genérico do metamodelo pretendido, as regras organizacionais que podem ser aplicados nestes domínios são bastante raras e, portanto, devem ser determinadas separadamente para cada sistema que derive deste metamodelo.

## **4.7.2.4 Projecto da Arquitectura**

### ***Estrutura da Organização***

A estrutura organizacional, de forma similar ao modelo de sub-organizações, também não se aplica a esses sistemas, dado que não existe nenhuma estrutura hierárquica comum fixa, e, por consequência, esta estrutura não é desenvolvida no metamodelo. No caso de sistemas que venham a utilizar este metamodelo e que desejam incluir algum grau de estrutura de hierarquia ou controle entre agentes ou agentes de serviços globais, será necessário que o projectista inclua essa informação no modelo.

### ***Modelo dos Papéis***

Como mencionado anteriormente, cinco dos nove papéis identificados pertencem aos agentes da plataforma do RM, nomeadamente os papéis: Sensor, Reactivo, Planeador, Interface e Broadcaster.

O papel Sensor é um dos papéis mais básicos. A sua função é reunir todas as informações sobre o ambiente, usando informações recolhidas pelos sensores, processá-las e actualizar a representação interna do mundo, para que outros agentes/funções possam utilizá-las (Figura 91).

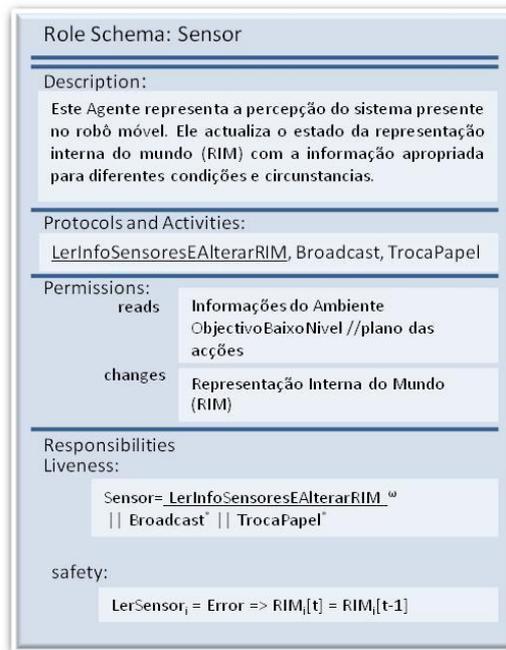


Figura 91: Papel Sensor

O papel Reactivo (Figura 92) é também um papel básico e é responsável por todos os controlos de baixo nível. Através da representação interna do ambiente, juntamente com objectivo de baixo nível (referência de controlo), o agente de controlo calcula a acção de controlo mais adequada.

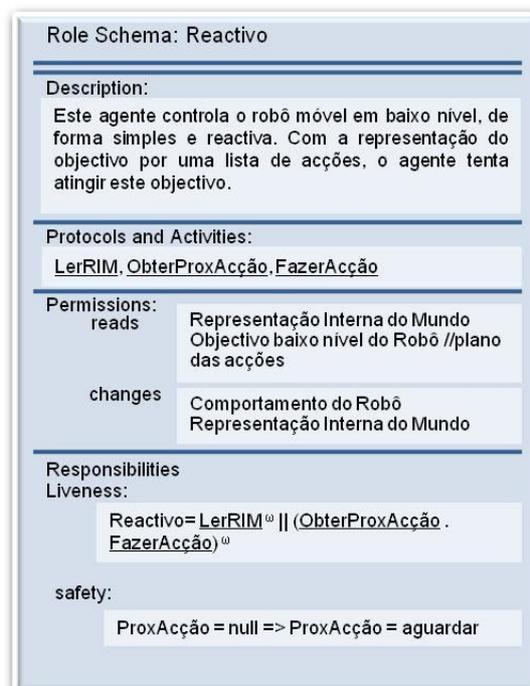


Figura 92: Papel Reactivo

O papel Planeador é responsável pelo controlo de alto nível, e cabe ao mesmo criar a sequência de acções de alto nível necessária para atingir a meta global. Ele também integra as características de colaboração e cooperação entre a plataforma robótica que ela representa e outras plataformas robóticas ( Figura 93).

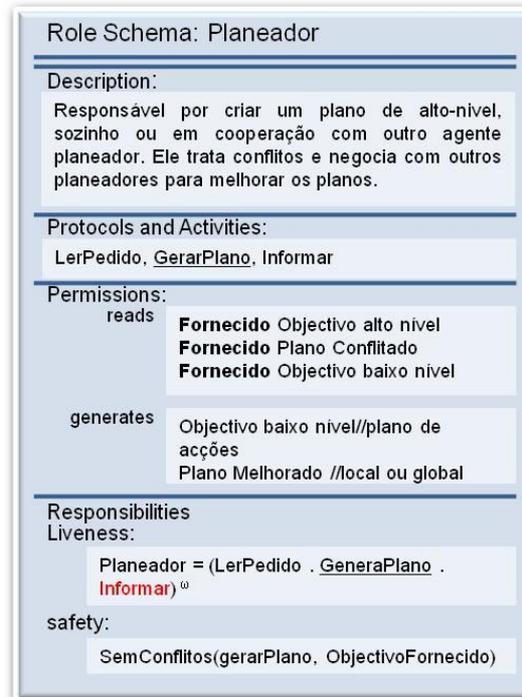


Figura 93: Papel Planeador

O papel Interface (ver Figura 94) estabelece a interacção entre o utilizador e o RM; este papel tem as seguintes funções: recolher todas as informações e seleccionar as informações relevantes ao utilizador e exibi-las na GUI; receber ordens de utilizador e encaminhá-las para o agente apropriado; permitir que o utilizador assuma o controlo manual do RM.

O agente que implementa o papel Broadcaster é responsável pela divulgação, em intervalos regulares, da representação de mundo interno e o estado do RM para que os demais agentes possam actualizar a sua representação interna do mundo – RIM (Figura 95).

Os agentes que implementam os papéis Logger, Utilitário e Gestor de conflitos (detalhados abaixo) podem inscrever-se na lista de informação (protocolo Monitor de Ambiente, apresentado a seguir). Desta forma, passam a receber a informação da RIM, actualizar os seus conhecimentos sobre as várias plataformas robóticas existentes no ambiente e ajustar suas acções de acordo com as informações actualizadas.

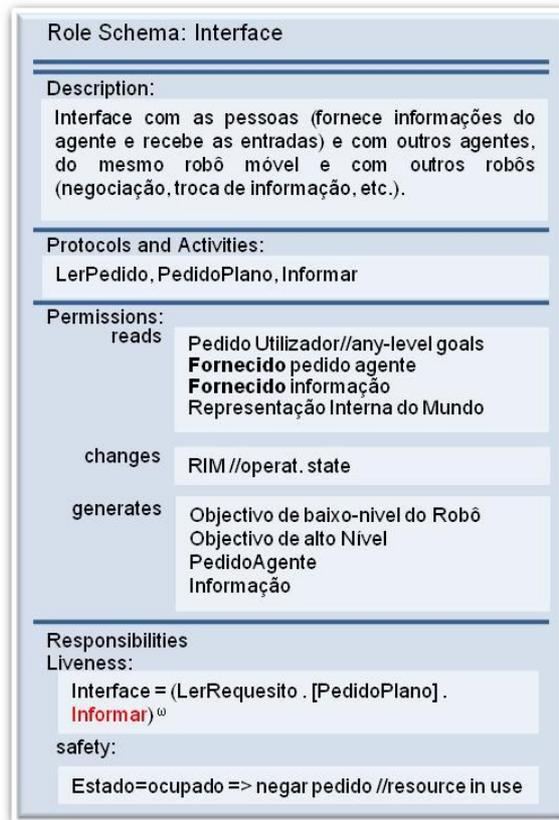


Figura 94: Papel Interface

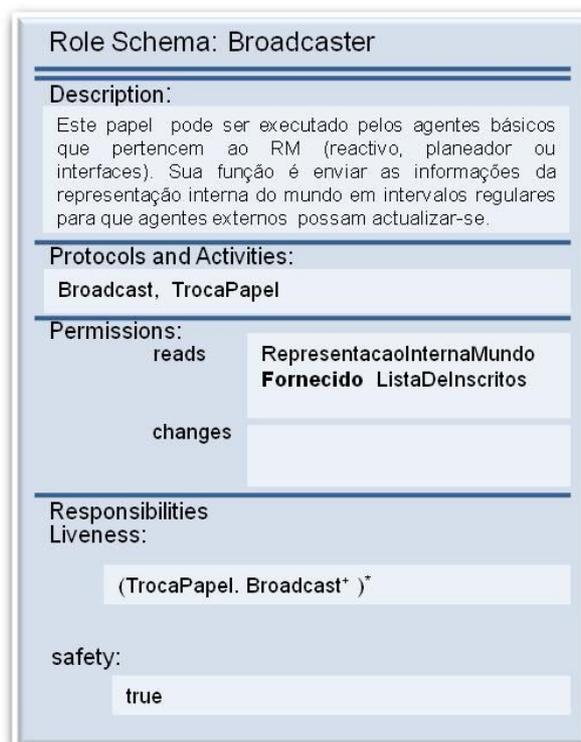


Figura 95: Papel Broadcaster

Os quatro papéis que fazem parte dos agentes que não estão embarcados no RM são: Logger, Utilitário, Gestor de conflitos e Designador de tarefas. Os dois primeiros, pela sua simplicidade, aqui não são graficamente representados. O papel Logger é responsável pela criação de um conjunto de arquivos contendo informações relevantes sobre os agentes e o ambiente. O papel Utilitário pode ser instanciado num grande número de agentes a representar portas, janelas ou outros elementos do ambiente, para que estes elementos possam interagir com as plataformas robóticas móveis e desta forma facilitar a navegação através do ambiente.

O Gestor de Conflitos é responsável por monitorar o ambiente e os agentes móveis, procurando possíveis conflitos ou situações nas quais os robôs possam ficar sem saída (Figura 96). Quando um conflito for encontrado, o agente que implementa este papel fica responsável por resolver esse conflito, e impor uma solução para os RM, ou pedir que eles cooperem para encontrar uma solução adequada com a finalidade de resolver antecipadamente o conflito futuro.

**Role Schema: Gestor de Conflitos**

---

**Description:**  
 Este papel é monitorar os movimentos de cada robô móvel, tenta detectar conflitos inesperados (colisões) e executa ações apropriadas para resolver o conflito. Ele também avalia novos planos e verifica se conflita com os planos existentes, executa ações apropriadas para solucionar o conflito.

---

**Protocols and Activities:**  
 MonitorarAmbiente, ResolverConflitos

---

**Permissions:**

**reads** **Fornecido** Objectivo baixo nivel robô //plano de acções  
**Fornecido** Informação odometry robô

**generates** **Pedir Solução Conflito**

---

**Responsibilities**  
**Liveness:**  
 Gestor Conflito = MonitorarAmbiente<sup>o</sup> ||  
 ResolverConflito<sup>o</sup>

**safety:**  
 SemConflito(PlanoBaixoNivelGerado)

Figura 96: Papel Gestor de Conflito

O papel Designador de Tarefas (Figura 97) é responsável por fornecer aos humanos um meio de interacção com o sistema como um todo. Isso permite especificar as missões que devem ser efectuadas pelo sistema (quer por um único RM, ou por vários RMs).

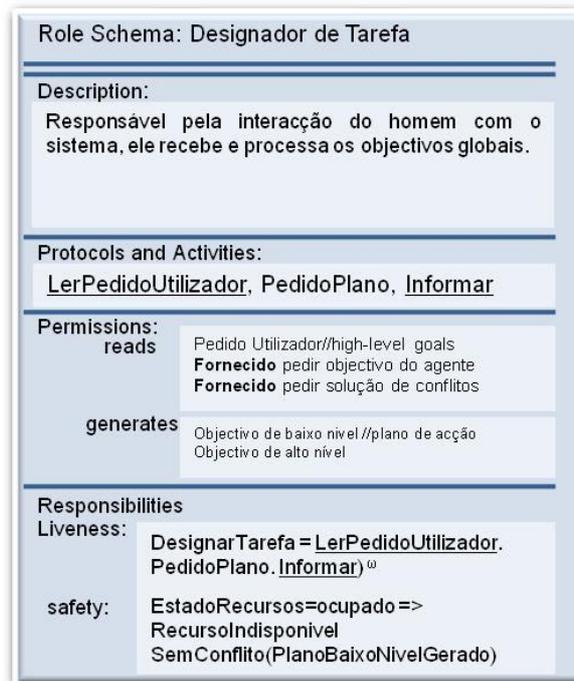


Figura 97: Papel Designador de tarefas

### Modelo de Protocolos

Neste metamodelo são apresentados seis protocolos no total, embora mais protocolos possam ser incorporados para uma versão mais completa do metamodelo.

O protocolo Troca Papéis pode ser usado por qualquer um dos agentes percepção, controlo, planeador e interface. Este protocolo é usado para solicitar uma transferência do papel Broadcaster para outro agente e geralmente é disparado por um aumento da carga de trabalho das tarefas de núcleo do agente (Figura 98).

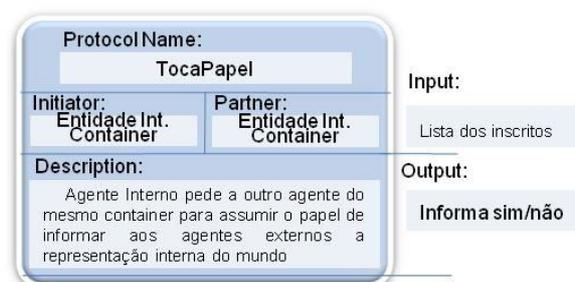


Figura 98: Protocolo Troca papéis

O protocolo Broadcaster (Figura 99) é usado para transmitir informações (sobre o RM e o ambiente) para os agentes inscritos na lista de agentes que requerem a informação, desta forma estes agentes receberão as informações actualizadas do sistema.

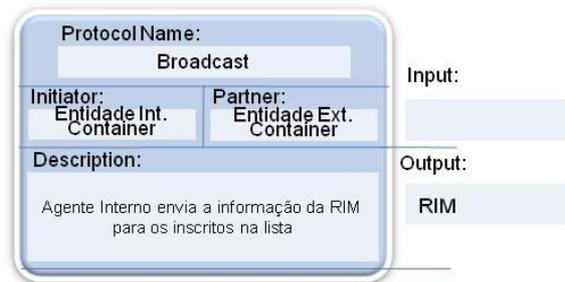


Figura 99: Protocolo Broadcast

O protocolo Monitor de ambiente (Figura 100) é usado pelos agentes fora do RM para pedir informações sobre o estado e informações da plataforma robótica (fornecido pelo agente que executa o papel Broadcaster).

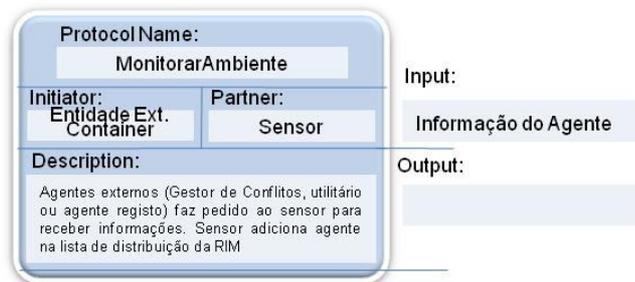


Figura 100: Protocolo Monitor de Ambiente

O protocolo Resolução de Conflitos é iniciado, pelo agente que executa o papel de gestor de conflito, quando um conflito entre as acções das plataformas robóticas é detectado, para que os agentes envolvidos supostamente cooperem na resolução do conflito. Este protocolo é usado para se comunicar com o agente Planeador de cada plataforma robótica envolvida no conflito e visa encontrar uma solução através da cooperação dos agentes (Figura 101).



Figura 101: Protocolo Resolução de Conflitos

O protocolo Solicitação de Plano (Figura 102) é iniciada pelo agente de Interface ou agente Controlo e permite solicitar ao agente Planeador num plano de acções de alto nível ou planos de trajectória para o RM atingir o objectivo da missão.



Figura 102: Protocolo Solicitação de Plano

O protocolo Atender Objectivo geralmente é iniciado pelo agente Designador de Tarefas e permite requisitar ao agente Interface de uma plataforma específica, para que gere e execute um plano para alcançar um determinado objectivo de alto nível (Figura 103).

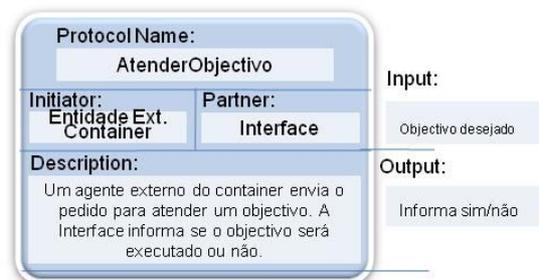


Figura 103: Protocolo Atender Objectivo

### ***Diagrama das Funções e Interação***

Para uma melhor compreensão de todos papéis e protocolos de interação presentes no sistema, um diagrama de papéis e interação, como proposto por Castro (Castro & Oliveira, 2008) e também incluído no modelo apresentado acima, é ilustrado pela Figura 104. Os papéis identificadas no sistema são apresentadas como classes e os protocolos entre eles são apresentados como associações, incluindo a direcção em que o protocolo é activado.

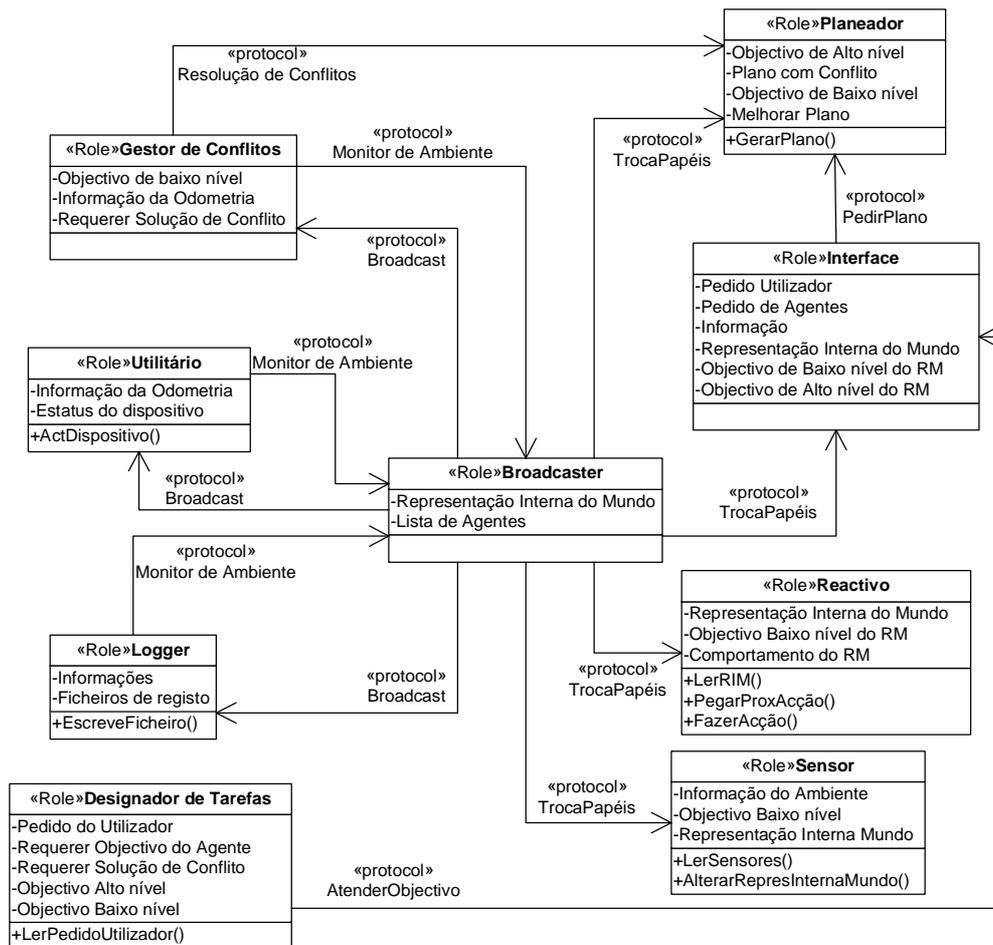


Figura 104: Diagrama dos papéis e interacção

### 4.7.2.5 Projecto Detalhado

#### Modelo de Agente

O modelo de agente é um dos dois produtos desta fase e pode ser visto como um mapeamento entre agentes e papéis, indicando quantas ocorrências de cada agente vai existir no sistema e os papéis que cada agente irá implementar (ver a Tabela 21). Neste caso específico, os quatro agentes que representam a plataforma robótica (percepção, controlo, planeador e interface) terá  $N$  instâncias, correspondente ao número de plataformas robóticas no sistema. O agente utilitário pode ter  $U$  instâncias e o Gestor de conflito pode ter  $C$  instâncias. É importante salientar que embora o papel Broadcaster esteja presente e implementado pelos quatro agentes embarcados no RM, apenas um desses agentes irá executar este papel em cada momento.

Tabela 21: Modelo de Agente

Sensor	$1 \dots N$	$\xrightarrow{play}$	Sensor, Broadcaster
Reactivo	$1 \dots N$	$\xrightarrow{play}$	Reactivo, Broadcaster
Planeador	$1 \dots N$	$\xrightarrow{play}$	Planeador, Broadcaster
Interface	$1 \dots N$	$\xrightarrow{play}$	Interface, Broadcaster
Agente Utilitário	$0 \dots U$	$\xrightarrow{play}$	Utilitário
Gestor de Conflito	$0 \dots C$	$\xrightarrow{play}$	Gestor de Conflito
Designador de Tarefas	$0 \dots 1$	$\xrightarrow{play}$	Designador de Tarefas
Logger	$0 \dots 1$	$\xrightarrow{play}$	Logger

### Modelo de Serviço

O modelo de serviço destina-se a identificar os serviços associados a cada classe de agente ou papel. Como proposto em (Castro & Oliveira, 2008), a tabela de modelo de serviço foi substituída por um diagrama de classes UML – as informações em falta (saída) também foram incluídas como notas para os serviços no diagrama. A Figura 105 mostra alguns serviços prestados pelo sistema.

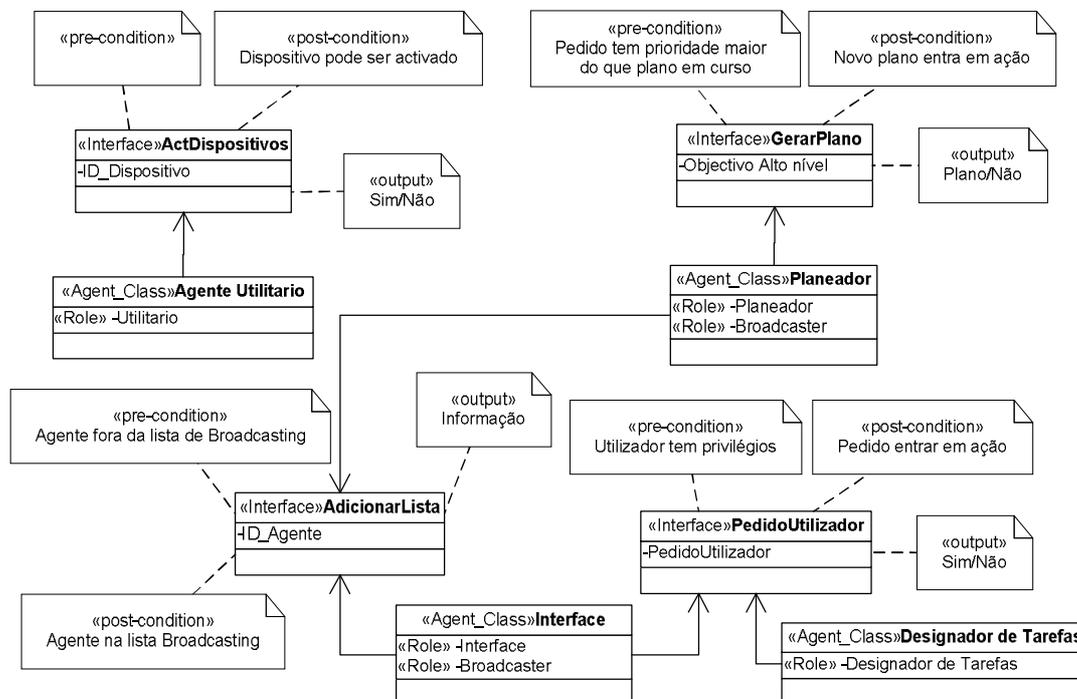


Figura 105: Modelo de Serviço

### 4.7.2.6 Particularização do Metamodelo

Esta subsecção apresenta a particularização do metamodelo aplicada à plataforma IntellWheels, o que resultou na modelização do SMA da PIW. A Figura 106 ilustra a arquitectura completa da PIW.

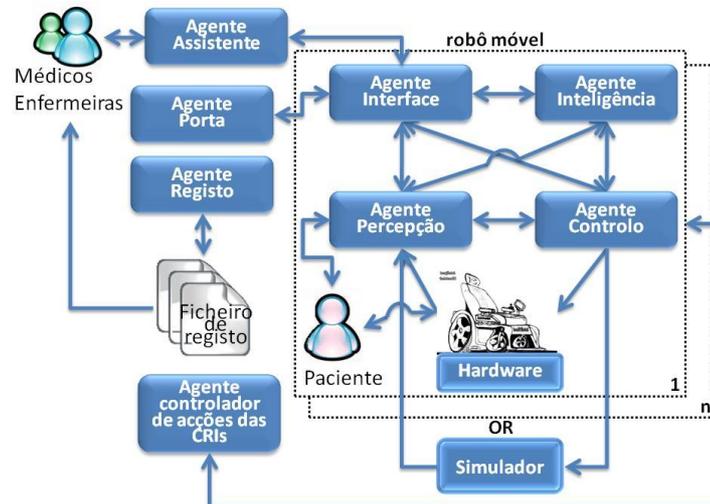


Figura 106: Arquitectura completa da PIW

A correlação entre os agentes do metamodelo com os agentes da PIW é apresentada através da Tabela 22, junto também é apresentada uma pequena descrição das funções destes agentes disponíveis nesta aplicação mais específica.

Tabela 22: Correlação entre o metamodelo e a PIW

Agentes		Descrição
Metamodelo	PIW	
Agente Planeador	Agente Inteligência	Este agente é a parte mais cognitiva do sistema, desta forma ele é responsável por gerar os planos de alto nível e calcular a melhor trajectórias.
Agente Controlo	Agente Controlo	Este agente é o controlo de baixo nível do sistema, possui um comportamento quase reactivo.
Agente Sensor	Agente Percepção	Responsável em obter os dados dos sensores e processa-los, criar mapas, e fazer a localização da CRI.
Agente Interface	Agente Interface	Ele é responsável em tratar as informações e disponibilizar as informações mais importantes através da GUI.
Agente Utilitário	Agente Porta	Vários agentes foram criados para ajudar o sistema das CRI com os objectivos globais. Esses agentes podem cooperar com o RM. Este agente é responsável por controlar as portas e portões no ambiente da CRI. Ele pode permitir ou inibir o acesso a algumas áreas restritas
Agente Logger	Agente Registo	Responsável em registar os estados, mensagens e

		informações gerais das CRI.
<b>Agente Gestor de conflitos</b>	Controlador de Acções das CRI	Este agente é responsável por um controlo central de todo o tráfego no ambiente CRI, evitando conflitos.
<b>Designador de tarefa</b>	Agente Assistente	Responsável por todo o sistema interacção humana externa (médico, enfermeira, etc). Ele recebe e processa os objectivos globais.

## 4.8 Sumário

Neste capítulo foram apresentados o conceito e projecto de uma plataforma de desenvolvimento de cadeira de rodas inteligente com os seus módulos básicos, assim como os aspectos da implementação desta plataforma. Esta plataforma tem como base um sistema multi-agente com o foco de facilitar a integração de diferentes sensores, actuadores, dispositivo de interface mais flexível, metodologias de navegação, técnicas de planeamento, etc.

Neste projecto foi seleccionado um conjunto de dispositivos para compor o Kit de Hardware da plataforma com o foco de garantir uma fácil instalação em cadeiras de rodas comerciais. Foram levados em conta aspectos como: um número alargado de dispositivos de entrada para permitir uma flexibilidade maior na interacção do utilizador com a cadeira de rodas inteligente (CRI); conjunto dispositivos para compor a percepção; baixo custo; manter o conforto e a ergonomia da cadeira de rodas original.

O simulador IntellWheels, apresentado neste capítulo, foi desenvolvido com base no simulador ciber-rato. O seu objectivo foi facilitar realização de um grande número de testes dos algoritmos presentes na plataforma IntellWheels (PIW). Devido à tecnologia de realidade aumentada desenvolvida, tornou-se possível que as CRI reais possam interagir com objectos e CRI virtuais. Desta forma, resultou a possibilidade de realizar testes mais abrangentes com uma interacção com o ambiente e um número de agentes e objectos muito maior. A execução de testes mais abrangentes, utilizando exclusivamente cadeiras de rodas reais, necessitaria de muito tempo e resultaria num custo muito mais elevado. Este simulador também possibilita desenvolver um laboratório de treino de pacientes. Neste laboratório, os pacientes poderão adaptar-se e treinar a condução da CRI de forma segura e menos stressante, antes de experimentarem o equipamento real.

O módulo de Planeamento inteligente é responsável por gerar planos de acções de alto nível e trajectórias para a CRI. Este módulo é composto por dois tipos de planeadores: um planeador inteligente com base em GraphPlan utilizando a linguagem PDDL para descrever o domínio e o problema; um outro planeador mais simples com base em A\* e grafos foi utilizado para gerar as trajectórias da CRI.

Na secção Navegação e Controlo foram apresentadas diversas tecnologias para permitir a movimentação da cadeira. Nesta secção foram descritas as funções básicas de controlo, a arquitectura de controlo, métodos de localização utilizados. Um dos aspectos discutidos nesta secção foi a arquitectura de controlo multi-nível e distribuída da PIW.

Para atender a alguns requisitos como a modularidade, flexibilidade, extensibilidade, robustez e reduzir a complexidade do problema, eficiência computacional, a PIW foi projectada e desenvolvida utilizando o paradigma dos SMA. Neste contexto, desejava-se que o sistema apresentasse as seguintes características: capacidade de comunicação em alto-nível com outras cadeiras e equipamentos; facilidade e robustez na comunicação entre os agentes; comportamentos colaborativos e cooperativos; possibilidade de formar alianças; entre outras características. Devidas às necessidades deste projecto, foi desenvolvida uma nova plataforma de comunicação para sistemas multi-agente a fim de garantir segurança, flexibilidade e robustez no sistema de comunicação. Este SMA desenvolvido pode ser considerado um SMA dentro de outro SMA. Por outras palavras, foi aplicado o conceito de macro e micro agentes.

Muitos projectos de CRI preocupam-se em melhorar, adaptar e desenvolver novos métodos de interacção com o utilizador. Podem-se citar diversos projectos que procuram desenvolver interfaces que permitam ao utilizador comandar a CRI de forma alternativa, como com gestos da cabeça, expressões faciais, comando de voz, entre outros. Como apresentado neste capítulo, a PIW trata esse problema de forma mais abrangente, com uso do conceito de interface multimodal aplicada as CRI. A ideia foi criar uma interface muito mais flexível e adaptável ao utilizador. Dentro deste contexto a interface multimodal processa as entradas dos diversos dispositivos de entrada e associa-as com as respectivas saídas. O utilizador pode criar diferentes sequências de entrada, adequadas às suas limitações e associar essas sequências às saídas (acções) disponíveis.

Para formalizar o desenvolvimento do SMA, utilizou-se a metodologia GAIA para modelizar o sistema. Este modelo partiu de um metamodelo descrito com uma linguagem de modelação de processos de software – SPEM. A implementação deste sistema a partir deste metamodelo e de outros trabalhos relacionados (Silva et al., 2010) mostram que este metamodelo pode ser usado na concepção de sistemas robóticos móveis distintos, reduzindo significativamente a maioria das tarefas comuns no projecto. O metamodelo reduz também as dificuldades de implementação, fornecendo, ao mesmo tempo, uma visão geral do sistema.

A união dos módulos apresentados forma a plataforma de desenvolvimento de cadeiras de rodas inteligente, que resultou numa ferramenta para auxiliar os investigadores na geração de novas tecnologias e na disseminação dos conceitos envolvidos nesta área científica.

## Capítulo 5

# Experimentação e Resultados

Este capítulo é constituído, inicialmente, por uma secção que tem por objectivo apresentar a metodologia utilizada na realização dos testes práticos, relacionados com a plataforma de desenvolvimento de cadeiras de rodas inteligentes. Na segunda secção apresentam-se os protótipos reais desenvolvidos, assim como os seus detalhes de desenvolvimento e construção. Durante o processo de desenvolvimento da plataforma foram implementados três protótipos que serviram como ferramentas de auxílio no aperfeiçoamento da plataforma de hardware. As secções seguintes apresentam os resultados dos testes práticos, sendo que os primeiros testes são destinados à avaliação individual de alguns módulos básicos e, finalmente, alguns testes para avaliar o funcionamento do sistema como um todo. De forma resumida este capítulo apresenta os protótipos desenvolvidos, análise do controle compartilhado e dos algoritmos de planeamento, navegação autónoma, testes do agente de interface, realidade mista e interacção dos agentes.

### 5.1 Introdução e Metodologia de Teste

Uma descrição detalhada do ambiente de teste será fornecida permitindo a reprodução do mesmo. Além disso, os cenários de teste serão caracterizados sendo a sua descrição complementada com os objectivos dos testes e o diagrama dos protocolos implementados, quando aplicável. Para finalizar, os resultados alcançados serão apresentados em conjunto com uma breve interpretação sobre seu significado e significância, para a validação do sistema.

Para todos os testes, onde foi utilizado o Simulador IntellWheels, os parâmetros destinados a configurar a cadeira de rodas, estão descritos na Tabela 23. Assim, para uniformizar os testes, foram utilizados os mesmos parâmetros em todos os testes (tamanho, distribuição dos sensores e dinâmica da CRI), sendo estes determinados com objectivo de

representar as características do protótipo real de maneira mais realista possível. Um teste muito simples foi realizado com a finalidade de estimar os parâmetros que representam o comportamento da cadeira real. Aplicou-se um degrau de velocidade de referência (0 a 100%) e adquiriram-se os dados de evolução da velocidade real.

Tabela 23: Parâmetros de configurações da Cadeira de Rodas

Corpo da CRI				
Comprimento	Largura	Centro de Mov.	Velocidade Max	Aceleração
1.04 m	0.66 m	0.81	0.49 m/s	0.83
Configuração dos Sensores				
Sensor	Posição x (m)	Posição (m)	Direcção (graus)	Cone (graus)
S1	0,72	0,24	0,0	60
S2	0,72	-0,24	0,0	60
S3	0,54	0,24	90	60
S4	0,54	-0,24	-90	60
S5	0,05	0,28	90	60
S6	0,05	-0,28	-90	60
S7	-0,03	0,28	180	60
S8	-0,03	-0,28	180	60

O gráfico da Figura 107 apresenta as curvas do ensaio com a cadeira real e virtual. Com base nestes resultados foi possível determinar os valores dos parâmetros do modelo da cadeira de rodas a ser simulada. Este modelo permite simular o comportamento dinâmico e velocidade máxima da cadeira.

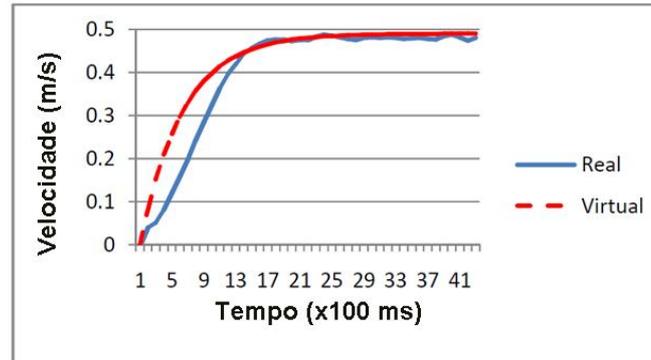


Figura 107: Curva de aceleração da cadeira de rodas real e virtual

Para o teste com a cadeira real obteve-se uma velocidade máxima de 0,49 m/s e um tempo de subida (de 10% a 90% da velocidade máxima), com valor de 1,4 segundos. Os parâmetros da CRI virtual foram configurados para VelMax = 0,49 m/s e o factor da curva de aceleração = 0,83. Os resultados (Figura 107) comprovaram que a velocidade máxima foi a mesma de 0,49 m/s e o tempo de subida de 1,6 segundos muito perto do real e o tempo de estabelecimento foi ajustado de modo a ser mesmo. O comportamento dinâmico foi semelhante, apesar do formato da curva não ser exactamente o mesmo. Isso deu-se devido às características dos controladores das cadeiras de rodas terem rampas de aceleração e desaceleração, resultando nesta diferença de 0,2 segundos no tempo de subida. Entretanto,

esta diferença não compromete em nada as simulações, pois neste momento estamos interessados apenas nos resultados de alto nível.

Os agentes desenvolvidos para avaliar a arquitectura proposta foram desenvolvidos utilizando Delphi 2010 e as mensagens utilizaram o protocolo UDP. Para os agentes do JADE foram utilizados o JADE versão 3.7 e a IDE Eclipse.

Além disso, para permitir que sejam feitas conclusões válidas a partir dos resultados, foram adoptadas algumas metodologias que consistem na repetição dos testes um número apropriado de vezes. Para análises quantitativas, todos os testes foram repetidos 20 vezes, mantendo-se as condições do ambiente. Os dados recolhidos são então analisados e, após isso, a média e o intervalo de confiança de 95% são calculados. Para análises qualitativas, foram repetidas 5 vezes as experiências (para testes em que não se efectuam análises estatísticas). Já nos testes mais complexos que envolvem voluntários, os dados foram submetidos a um teste não paramétrico para amostras emparelhadas (teste de *Wilcoxon Signed Ranks 1-tailed* (Corder & Foreman, 2009)), pois devido a complexidade do teste tem-se um pequeno número de dados, que não permite conhecer o tipo de distribuição da variável.

## 5.2 Protótipos

No período de desenvolvimento desta tese de doutoramento, compreendido entre os anos de 2006 a 2010, foram desenvolvidos e implementados três protótipos reais, sendo que um deles ainda sofreu uma actualização mais recente. O propósito dos protótipos reais, desenvolvidos neste período, foi criar uma plataforma de hardware que apresentasse simplicidade, baixo custo, facilidade de adaptação em cadeiras de rodas comerciais e não alterasse o conforto e a ergonomia original da cadeira. A construção de um protótipo também tinha como objectivo permitir e auxiliar no desenvolvimento, testes e aperfeiçoamento do conjunto *hardware/firmware/software* da plataforma de desenvolvimento da cadeira de rodas, assim como a sua disposição e instalação em cadeiras de rodas comerciais.

A construção dum protótipo de CRI teve início na adaptação de uma cadeira de rodas motorizada comercial. A adaptação visa permitir que software do sistema possa controlar a cadeira. O software é executado num computador portátil embarcado na cadeira de rodas. As tarefas de desenvolvimento dum protótipo incluem:

- Montagem do módulo de controlo de cadeira de rodas e adaptação para recepção dos sinais de referência para os motores;
- Adaptação do sistema de odometria na cadeira de rodas (montagem de *encoders*);
- Montagem e adaptação dos sensores de proximidade (sonares, infravermelhos), e periféricos (dispositivos de entrada e saída para a interacção com o utilizador);

- Testes simples de funcionamento do protótipo no laboratório.

No início da criação da plataforma foram realizadas investigações sobre projectos de CRI, estudo e avaliação das adaptações necessárias numa cadeira de rodas comercial para o desenvolvimento dum protótipo.

### 5.2.1 Protótipo IntellWheelsChair-I

A partir de uma cadeira de rodas motorizada comercial, os primeiros testes foram focados na integração, através das placas com microprocessadores, dos componentes: módulo electrónico de origem da cadeira, sensores e computador portátil embarcado na cadeira. O sinal de comando gerado pelo computador é enviado para a placa de interface, que emula o sinal gerado pelo joystick original, desta forma é possível controlar a velocidade das rodas da cadeira através dos módulos de potência originais da mesma. Foram instalados, ainda:

- Dois *encoders*: montados independentes das rodas da cadeira e instalados pelo lado interno das mesmas. Estes sensores fornecem as informações necessárias para a estimação da posição e velocidade da cadeira de rodas, sendo estas essenciais para o sistema de navegação (Figura 108-a). O preço e a simplicidade de funcionamento foram factores determinantes na escolha dos *encoders*, assim como a possibilidade de utilização em ambientes internos e sua boa precisão quando combinados com métodos automáticos de correcção;
- Dez sensores de proximidade: estes sensores são utilizados para possibilitar o desvio de obstáculos, seguimento de paredes e percepção de desníveis no chão (Figura 108-b). A forma de instalação na cadeira de rodas, visa criar um “pára-choques virtual” ao redor dela, protegendo o paciente de possíveis colisões.

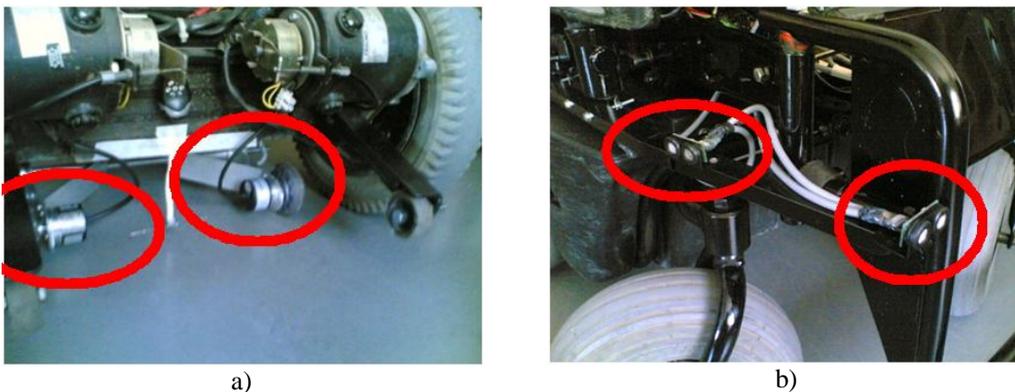


Figura 108: Montagem dos sensores, a) posicionamento dos *encoders* e b) instalação dos sensores de ultra-som

Após testes neste protótipo, verificou-se que as rodas propulsoras da cadeira apresentam deslizamento no início do movimento, o que inviabiliza o acoplamento dos *encoders* directamente ao eixo dos motores. A solução encontrada para este problema foi a instalação de duas rodas auxiliares posicionadas internamente às rodas de tracção da cadeira de rodas,

seguindo a linha do eixo destas rodas (Figura 108-a). Esta configuração garante a medição dos deslocamentos reais da cadeira, sem o efeito de escorregamento no sentido do movimento.

O posicionamento dos sensores de ultra-som objectivou criar uma área de percepção ao redor da cadeira, utilizada para criar uma barreira de protecção virtual. A Figura 109 ilustra a área de percepção de cada sensor. Assim, usou-se como referência um ângulo de 50° na abertura do cone de detecção do sensor, este valor foi retirado dos resultados obtidos a partir de testes práticos realizados.

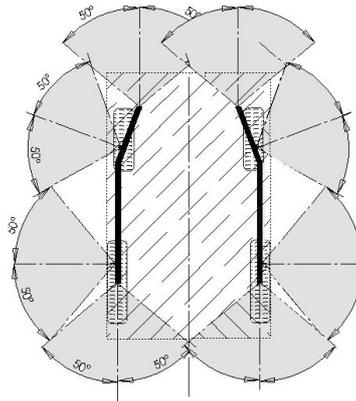


Figura 109: Posicionamento dos sensores de ultra-som instalados no protótipoI

Além dos dispositivos de percepção citados anteriormente, este protótipo dispõem de outro grupo de equipamentos:

- Um computador portátil responsável por proporcionar “inteligência” à cadeira de rodas. Este item pode ser considerado o cérebro da CRI, sendo que o computador portátil utilizado é um HP Pavilion tx1270EP, AMD Turion 64 X2 TI60. O equipamento possui uma tela sensível ao toque, sendo um mais um possível canal de interação com o utilizador;
- Duas placas electrónicas para leitura dos sensores de proximidade, as quais recebem dados dos sensores de ultra-sons, enviando-os de seguida ao computador (Figura 110);
- Duas placas electrónicas de PWM/Encoder, sendo estas utilizadas para enviar ao computador a informação de deslocamentos da cadeira (utilizada no cálculo da odometria). São também utilizadas para receber do computador o sinal de referência de velocidade para as rodas. Desta forma, o controlo de velocidade gera o sinal analógico através de PWM<sup>16</sup>. Estas funções são implementadas no *firmware*.

<sup>16</sup> PWM – *Pulse Width Modulation* ou Modulação por largura de pulso: neste contexto, é utilizado para controlar o valor de alimentação entregue à carga através da modulação de sua razão cíclica (*duty cycle*).

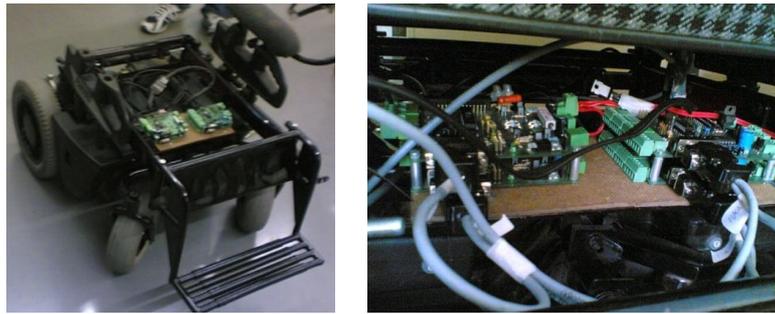


Figura 110: Montagem das placas electrónicas para a interface dos dispositivos e o computador

O projecto das placas de interface utilizadas neste projecto, bem como o desenvolvimento de seu *firmware* e comunicação foi completamente desenvolvido por alunos e docentes da Faculdade de Engenharia da Universidade do Porto (FEUP), sendo originalmente utilizadas na equipa de futebol robótico (Moreira et al., 2002). A Figura 111 ilustra a evolução deste protótipo, sendo que a terceira imagem representa resultado final da montagem do primeiro protótipo, com a instalação dos sensores, as placas de interface e as conexões necessárias.



Figura 111: Evolução do protótipo IntellWheelsChair-I

## 5.2.2 Protótipo IntellWheelsChair-II

Após a experiência adquirida com o primeiro protótipo, foi desenvolvida uma arquitectura modular para os dispositivos electrónicos, construindo-se um *kit* de hardware (ou plataforma de hardware). Este *kit* é facilmente instalado em cadeiras de rodas motorizadas comerciais.

O segundo protótipo, nomeado de IntellWheelsChair-II, seguiu a arquitectura modular ilustrada na Figura 112.

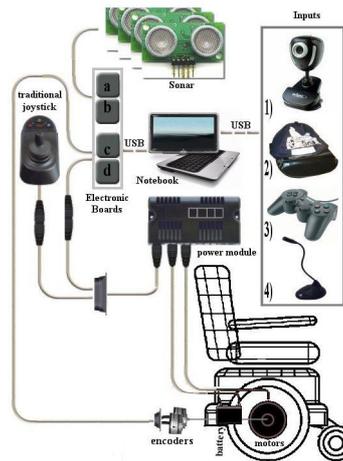


Figura 112: Arquitectura preliminar de hardware

IntellWheelsChair-II é composto por diversos dispositivos comerciais (Figura 113), tendo por base uma cadeira de rodas motorizada marca Vassilli, modelo Evolution Electronic<sup>17</sup>, um módulo electrónico VR2, da PG Drives Technology. O drive VR2 substitui o drive original da cadeira de rodas motorizada, portanto, o conjunto electrónico é totalmente independente da marca da cadeira.



Figura 113: Hardware utilizado na construção do protótipo 2

Para compor o sistema sensorial do IntellWheelsChair-II, foram instalados os seguintes dispositivos:

- Sensores de proximidade: detecção de obstáculos e auxílio à navegação. Estes sensores foram distribuídos em duas barras instaladas nas laterais da cadeira de rodas.
  - 8 sensores de infravermelhos: desempenham funções de seguir paredes, mapeamento do ambiente e localização;

<sup>17</sup> Maiores informações encontram-se disponível em <http://www.vassilli.it/pages/ing/evolut/evola.html>.

- 10 sensores de ultra-sons: para o desvio de obstáculos, detecção de escadas, degraus, buracos e desníveis no pavimento.
- Sensor de velocidade e deslocamento: mede o movimento das rodas e, com base nesta informação, é realizada a odometria no robô.
  - 2 *encoders*: instalados directamente no eixo dos motores.
- Webcam para auxílio à navegação: reconhecimento de marcos artificiais.

Dispositivos de interface com o utilizador utilizados neste protótipo foram: microfone, câmara do utilizador (ambos para uso de comandos através de som e expressões faciais), sensor de movimentos da cabeça, e ecrã sensível ao toque.

Para completar a plataforma de hardware foram ainda utilizados:

- 2 Placas electrónicas para a leitura dos sensores de proximidade.
- 2 Placas electrónicas de PWM/Encoder (as mesmas utilizadas no primeiro protótipo).
- Computador portátil.

O resultado da construção do segundo protótipo pode ser observado na Figura 114.



Figura 114: Protótipo IntellWheelsChair-II

De modo distinto da maioria dos trabalhos relacionados, os cuidados com o impacto visual e alteração do conforto da cadeira estiveram presentes em todo o desenvolvimento deste projecto. A alteração no visual da cadeira e no conforto causado por suportes, câmaras, sensores, computador, fios e scanners infra-vermelhos acabam por afastar os utilizadores e criar barreiras psicológicas quanto a utilização das CRI. A consequência disso é a restrição na quantidade e disposição dos sensores, assim como a escolha de certos tipos de sensores que possuem um tamanho demasiado grande, aparência e necessidades especiais de montagem.

### 5.2.3 Protótipo IntellWheelsChair-III

O último protótipo desenvolvido do ponto de vista de hardware (ver Figura 115), teve como meta a redução no número de placas electrónicas e melhorar a sua aparência (redução de fios visíveis).



Figura 115: Protótipo IntellWheelsChair-III

No desenvolvimento deste novo protótipo utilizou-se uma placa Arduíno<sup>18</sup>, em substituição das 4 placas electrónicas utilizadas no protótipo anterior (Figura 116). Esta é uma placa comercial desenvolvida com base num microcontrolador, com suporte de diversas entradas e saídas. Assim, surgiu o novo desafio de desenvolver um *firmware* dedicado a cadeira de rodas. Optou-se, ainda, por adicionar ao *firmware*, um algoritmo que forneça à CRI a habilidade de desvio de obstáculos. Outra inovação neste protótipo é a captura e envio ao Arduíno do sinal do joystick original. Logo é possível conduzir a CRI através desse joystick juntamente com a habilidade de desvio de obstáculos, isto sem necessidade do computador portátil.



Figura 116: Placa Arduíno

---

<sup>18</sup> Arduíno Mega é uma plataforma de hardware livre, projectada com um microcontrolador baseado no Atmega1280 com suporte a várias entradas/saídas. Ver <http://arduino.cc/>.

A arquitectura e demais equipamentos desse último protótipo foram formalizados e descritos no Capítulo 4, secção 4.2.

O novo algoritmo de desvio de obstáculo implementado é baseado no conceito de campo de potencial (*Potential Field Methods – PF*) (Andrews & Hogan, 1983) (Khatib, 1986), devido à sua simplicidade e, especialmente, a possibilidade de adaptar-se facilmente às necessidades específicas dos paradigmas de controlo compartilhado, e também a possibilidade de executá-lo sobre a limitada capacidade computacional, sendo assim possível a sua implementação em *firmware* e em software. No entanto, este trabalho difere do PF original, porque não tenta construir internamente um mapa do meio ambiente. Ao contrário, a abordagem é mais próxima da implementação descrita por Estela Bicho (Bicho et al., 1998), onde cada leitura do sensor de ultra-som é tratada como uma força de repulsão (Petry et al., 2010).

Quando um objecto é detectado pelo sensor  $S_i$ , uma força repulsiva virtual  $F_i$  é calculada. A direcção de cada força repulsiva é determinada pela direcção  $\sigma_i$ , do objecto  $O_i$  para o centro do robô  $C$  (Figura 117). Como os sensores de ultra-som retornam medidas radiais do ambiente, não é possível determinar com precisão a localização angular do objecto. No entanto, é muito mais provável a detecção do objecto mais próximo do eixo acústico do sensor de ultra-sons, do que na periferia de cone do campo de visão (Khatib, 1986). Assim, a posição do obstáculo  $O_i$  é calculada como a distância medida  $D_i$  sob o eixo acústico do sensor.

$$\sigma_i = \tan^{-1} \frac{O_{iy}}{O_{ix}} \quad (\text{Eq. 28})$$

Onde

$O_{ix}, O_{iy}$  Posição relativa do obstáculo detectado pelo sensor  $S_i$ .

$\sigma_i$  Direcção do objecto  $O_i$  até o centro da cadeira de rodas  $C$ .

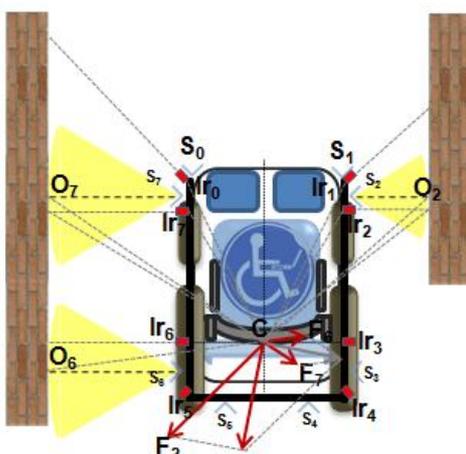


Figura 117: Forças repulsivas

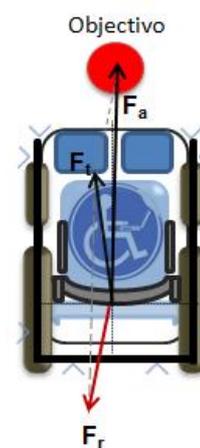


Figura 118: Forças resultantes que agem sobre a cadeira

A fim de manter ao máximo a autonomia do utilizador, os sinais de controlo apenas são adaptados em situações em que o utilizador enfrenta risco eminente de colisão. Portanto, as forças de repulsão começam a influenciar no controlo da CRI apenas quando uma distância de segurança é atingida. Devido à inércia, a distância necessária para parar completamente a cadeira de rodas aumenta com a sua velocidade  $s$ . Assim, o risco de colisão é considerado como uma variável bidimensional, dependente da distância e da velocidade (Figura 119 e Figura 120).

A distância de segurança destina-se não apenas a evitar obstáculos nas proximidades da cadeira de rodas, mas também visa evitar oscilações que objectos longe, não críticos, podem causar no comportamento do controlo. A magnitude das forças repulsivas cresce exponencialmente, em consequência, do par  $(D_i, S)$ :

$$|F_i| = \alpha * \exp(-\beta * D_i + \omega * S) * |F_a| \tag{Eq. 29}$$

Onde

$\alpha, \beta, \omega$  Constantes positivas deduzidas da distância de segura desejada

$F_a$  Força atractiva

$D_i$  Distancia medida pelo sensor  $S_i$

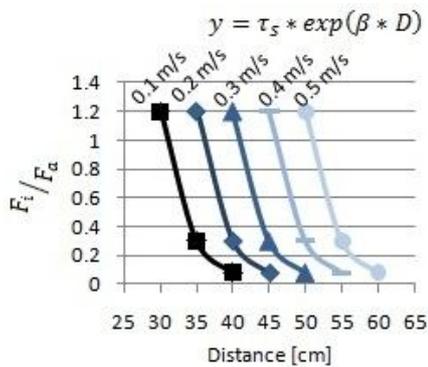


Figura 119: Distância segura em função da velocidade e distância da CRI

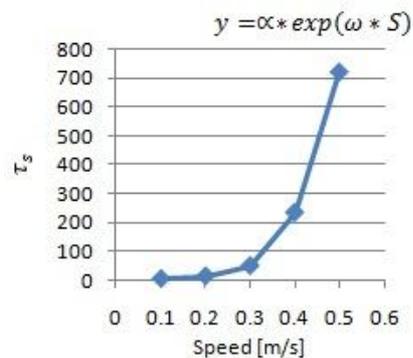


Figura 120: Componente da distância segura

Uma vez calculadas todas as forças repulsivas, elas são somadas para produzir uma força repulsiva resultante  $F_r$

$$F_r = \sum_{i=0}^n F_i \tag{Eq. 30}$$

Em seguida, a força de atracção virtual  $F_a$  gerada pelo alvo é actualizada. A força  $F_a$  produzida é directamente proporcional à entrada do utilizador, a qual pode ser o joystick da cadeira ou uma interface especial do utilizado, como, por exemplo, a que se baseia na posição de cabeça do utilizador. Assim, a soma da força repulsiva resultante  $F_r$  com a força de atracção  $F_a$  resulta na força final  $F_t$  que age sobre a cadeira de rodas (Figura 118).

$$F_t = F_a + F_r \tag{Eq. 31}$$

## 5.2.4 Dispositivos de Interação com o Utilizador

Nas primeiras etapas de desenvolvimento, iniciou-se a criação de *drivers* para capturar as informações provindas dos dispositivos de entrada (*inputs* do utilizador) tais como: *driver* para joystick USB e *driver* para controlo com os movimentos da cabeça. Os *inputs* do utilizador implementados para a interacção com a CRI foram os seguintes:

- **Joystick USB:** o objectivo de usar este dispositivo foi possibilitar o uso da plataforma em modo de simulação sem a necessidade do hardware da CRI. Outro motivo é o facto de possuírem diversos botões configuráveis para auxiliar na navegação. Neste sentido, foi desenvolvido um *driver* (Figura 121) para obter os dados, fazer o pré-processamento dos dados e enviá-los para o agente interface.



Figura 121: *Driver* para o Joystick

- **Detecção de expressões faciais:** O módulo de interacção por expressões faciais tem por objectivo gerar comandos para CRI sem requerer o uso das mãos. Essa interface foi proposta para solucionar a dificuldade de pacientes sem capacidade de utilizar as mãos de forma a controlar a cadeira de rodas, de modo a tornar confortável o uso de expressões faciais. As expressões faciais são capturadas por meio de uma câmara digital e interpretadas por uma aplicação que é executada num computador portátil na CRI. O software inclui algoritmos de processamento de imagem digital para a detecção de certas características, tal como a segmentação da cor e a detecção da borda, seguidas pela aplicação de uma rede neuronal que usa estas características para detectar as expressões faciais desejadas. Assim sendo, as expressões faciais são capturadas previamente para treinar o sistema. As expressões reconhecidas servem tanto para tratar de comandos básicos, como: andar para frente, à esquerda e parar; até comandos de alto nível: ir à casa de banho, ir à enfermaria ou ir ao quarto (Figura 122). Este módulo foi desenvolvido em trabalhos prévios (Faria et al., 2007),(Faria et

al., 2007) e, mais tarde, integrado ao projecto da CRI. Foram realizados alguns testes práticos que revelaram uma sensibilidade à variação de luz e como este trabalho específico não está no escopo da tese, não serão apresentados resultados práticos da sua utilização/testes.



Figura 122: Interface do módulo de controlo por expressão facial

- **Comando por Voz:** inicialmente, o módulo de comando de voz foi implementado com base no software IBM ViaVoice (IBM, 2009), um sistema comercial de reconhecimento de fala. Antes da sua primeira utilização é necessário que o utilizador treine o software para que este reconheça um conjunto de palavras específicas. Após o treino, o utilizador deve pronunciar o comando desejado num microfone e em seguida a fala será interpretada. Para a prova de conceito, foi desenvolvido um driver, o qual recebe a acção gerada pelo software da IBM e envia-a ao agente Interface. Os comandos por voz foram associados a um evento de carregar num botão. Actualmente, o sistema de reconhecimento de voz faz parte integrante do agente interface através do uso da API do Windows, nomeadamente da SAPI.
- **Detecção dos movimentos da cabeça:** a detecção de movimentos da cabeça é um método alternativo proposto para a condução da cadeira, indicado para utilizadores com dificuldades motoras nos braços. Utilizou-se um boné como suporte para os dispositivos que capturam os movimentos dos três eixos de coordenadas cartesianas através do uso de acelerómetros (Figura 123).



Figura 123: Sensor para detectar os movimentos da cabeça

O *driver* para este dispositivo de entrada recebe os dados via *Bluetooth*, faz o pré-processamento e envia ao agente Interface os dados.

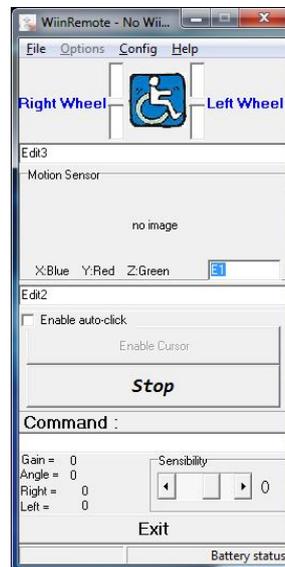


Figura 124: *Driver* do sistema de detecção dos movimentos da cabeça

- **Interface do utilizador:** No estado actual do projecto, o agente de interface foi criado para auxiliar nas tarefas de desenvolvimento e avaliação do comportamento da CRI durante os testes. Sendo assim, a interface gráfica mostrada na Figura 125, composta por vários grupos de informações, não está focada no utilizador final. Na parte superior esquerda um painel contém uma visualização da câmara e a localização resultante do reconhecimento de marcos. No canto inferior esquerdo, a figura esquemática de uma cadeira de rodas mostra a distância para objectos próximos medida por cada sonar e sensor infra-vermelho. No centro, um painel mostra as informações fornecidas pela odometria, como a velocidade de cada roda e os botões para escolher o modo de operação. Finalmente, o lado direito da janela contém as informações sobre o módulo de interface homem/máquina (por exemplo, lista de acção, lista de entradas, lista de sequências).

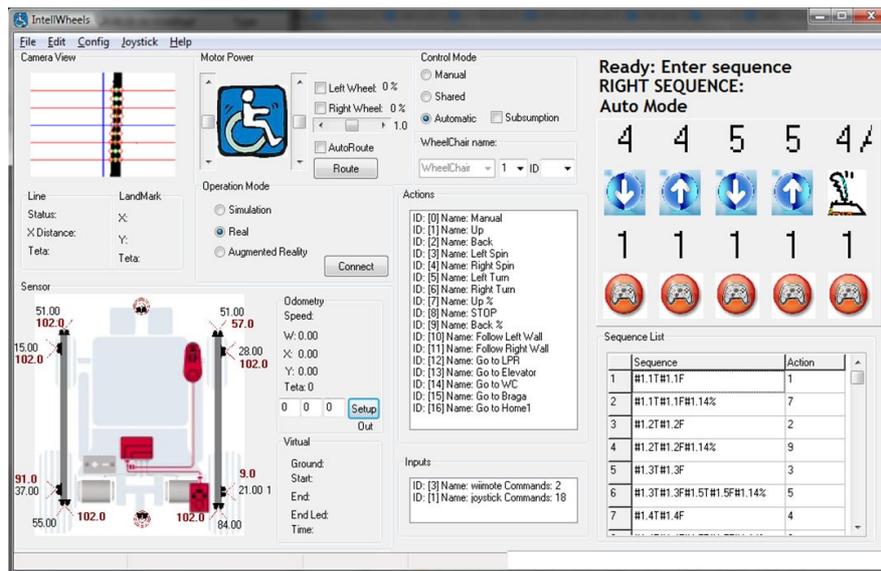


Figura 125: Agente Interface

### 5.3 Cenários de Teste

Para provar os conceitos apresentados anteriormente foram criados alguns cenários de teste. Desta forma procurou-se realizar desde experiências simples que comprovem o funcionamento de módulos individuais, até algumas mais complexas para se avaliar o sistema como um todo. É importante lembrar que o foco deste trabalho não está no desempenho individual dos módulos ou dos algoritmos, mas sim no comportamento global do sistema. Logo, espera-se que o sistema sirva como ferramenta para desenvolvimento de novos algoritmos e conceitos.

Os cenários de teste implementados estão apresentados de acordo com sua relevância e complexidade, para que se possa avaliar e validar o conceito do sistema proposto e seus módulos. Cada teste representa um nível funcional a ser avaliado, e os níveis mais altos passam a avaliar a integração dos módulos, sendo divididos em 6 grupos nomeados a seguir:

- Teste 1 – Teste da Interface Multimodal;
- Teste 2 – Navegação Autónoma Simulada;
- Teste 3 – Realidade Mista;
- Teste 4 – Plataforma de Comunicação;
- Teste 5 – Controlo Compartilhado;
- Teste 6 – Interação da CRI com Macro Agentes.

Para facilitar o entendimento de cada teste, estes serão descritos em detalhes nas suas respectivas subsecções.

### 5.3.1 Teste da MMI em Ambiente Real

Este teste tem por objectivo avaliar o funcionamento da Interface Multimodal. A experiência consistiu em conduzir a cadeira de rodas, em modo manual com desvio de obstáculos, ao longo de um corredor estreito com um obstáculo no percurso (Figura 126). Foram realizados basicamente dois testes nesta experiência: o primeiro teste foi com a condução através de comandos do joystick, juntamente com comandos de voz, e no segundo teste utilizou-se comandos com gestos da cabeça e com comando de voz.



Figura 126: Representação do percurso para o teste com a MMI

Neste teste, será considerado um êxito se o utilizador cumprir a tarefa estipulada utilizando o recurso da MMI sem ajudas extras e livre de colisões.

### 5.3.2 Navegação Autónoma

O objectivo desta experiência é realizar um teste simples da integração da tarefa de planeamento autónomo de acções de alto nível com a navegação. Para realizar este teste foi utilizado um agente CRI, o módulo de planeamento para tarefas de alto nível em STRIPS e o Simulador IntelliWheels para simular o movimento da CRI num ambiente hospitalar. Este teste consistiu no planeamento e transporte do paciente entre duas salas diferentes. A sequência de acções necessárias para atingir a meta supostamente seria: pegar o paciente 1 no quarto 1, levar este paciente para o quarto 2 e finalmente ir para o corredor.

O sucesso desta experiência ocorre caso a CRI consiga completar a tarefa sem a intervenção humana. Após a definição do objectivo a CRI deverá planear e executar o plano de forma automática.

### 5.3.3 Realidade Mista

#### Desvio de obstáculos

A experiência de Realidade Mista foi planeada para avaliar a interação entre o protótipo real da CRI com um ambiente virtual. Como mencionado anteriormente, o modo realidade mista torna possível a realização de muitos testes com uma CRI real em vários cenários (por exemplo, corredores estreitos, lugares lotados, movimentação de objectos), de forma muito económica (redução do tempo exigido para criar cenários e mínimo custo em infra-estrutura) e segura (livre de colisões com objectos reais, risco reduzindo a probabilidade de danificar o equipamento).

Nesta experiência a cadeira de rodas é utilizada em modo autónomo com desvio de obstáculos. O simulador foi configurado para simular um ambiente igual ao real. Desta forma o mapa modelado para representar o ambiente de teste real foi carregado no simulador. O primeiro passo foi colocar a CRI no modo realidade mista (software conecta-se ao hardware e ao simulador ao mesmo tempo), e posicioná-la no meio do corredor. Os sensores de proximidade real foram desconectados e os seus dados de percepção substituídos pelo seu homólogo simulado. Em seguida, foi dado um comando simples para a CRI seguir a parede do corredor.

O requisito mínimo para se considerar que ocorreu um sucesso nesta experiência será a CRI conseguir percorrer o corredor sem colisões utilizando apenas os dados de odometria real e o restante da percepção provinda do simulador.

#### Laboratório de reabilitação

O segundo teste com o sistema em realidade mista foi para avaliar a possibilidade de criar um ambiente de reabilitação para que o paciente possa treinar a utilização da CRI no simulador com a CRI real ao mesmo tempo.

### 5.3.4 Plataforma de Comunicação

Para avaliar e validar o funcionamento da plataforma de comunicação, realizaram-se três tipos de testes básicos.

#### 5.3.4.1 Efectividade e a eficiência da plataforma

O objectivo deste teste foi obter dados sobre a eficácia e a eficiência dos algoritmos em si, sem o uso de um cenário de agente. Para esse efeito um agente simples foi criado e lançado várias vezes. Neste cenário, foram medidos os seguintes dados:

- Fase 1 – O tempo necessário para que os aplicativos consigam eleger um *container* local;
- Fase 2 – O tempo necessário, gasto pelo *container*, para reunir todas as informações de aplicativos e criar uma LAL;

- Fase 3 – O tempo necessário, gasto pelo *container*, para informar todos os aplicativos de uma nova LAL e distribuí-los;
- Fase 4 – O tempo necessário para eleger um novo *container*, quando o *container* original é encerrado.

### 5.3.4.2 Teste comprador/vendedor de livros

Neste teste o objectivo foi testar a eficácia dos algoritmos num cenário bem conhecido em simulação de agentes e comparar os resultados obtidos nesta plataforma com os obtidos numa simulação equivalente em JADE. O cenário implementado foi o conhecido vendedor e comprador de livros (Bellifemine et al., 2007)(Wooldridge, 2001). A Figura 127 representa a interacção desejada entre agente vendedor com o agente comprador.

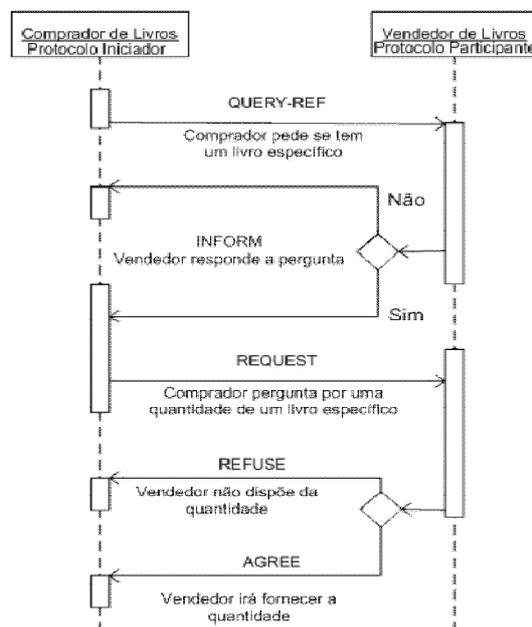


Figura 127: Modelo de interacção esperada entre os agentes comprador/vendedor de livros

Depois de enviar as mensagens QUERY-REF para o pedido do livro, o comprador activa um temporizador de 1s para determinar o tempo limite para receber novas mensagens, após esse tempo o aplicativo ignora as outras respostas.

### 5.3.4.3 Escalabilidade da plataforma

O teste de escalabilidade da plataforma tem por objectivo avaliar o desempenho e escalabilidade da plataforma no mesmo cenário do comprador/vendedor de livros. Os resultados obtidos na plataforma IntellWheels foram comparados com os resultados obtidos, no mesmo cenário e nas mesmas condições, usando a plataforma JADE. O protocolo implementado segue o modelo de interacção do comprador e vendedor de livros, visto na Figura 127.

Os tempos necessários que o agente comprador leva para realizar com sucesso o pedido e a aquisição do livro são registados, para os testes com a plataforma JADE e para os testes com a plataforma proposta. Para complementar o protocolo, o agente Comprador implementado foi configurado com um cronómetro para impor um tempo limite. Após passar o tempo limite, o comprador poderá prosseguir com a interacção, mesmo se um dos vendedores não responder. Nestes casos o agente Comprador descarta a interacção e os dados colectados e, em seguida, inicia uma nova interacção. Para atingir objectivos do teste, apenas um agente Comprador foi instanciado. O número de agentes Vendedor é incrementado em cada iteração de teste até ao máximo de 15 agentes.

### 5.3.5 Controlo Compartilhado em *Firmware*

Com o objectivo de avaliar a eficiência do algoritmo de controlo compartilhado implementado em *Firmware*, seis voluntários foram convidados a realizar um conjunto de quatro testes, sendo estes compostos por quatro voltas (Figura 128): duas voltas em um ambiente simulado (com e sem a assistência do algoritmo), e duas voltas em um ambiente real (com e sem a assistência do controle compartilhado). Os voluntários, cuja idade é compreendida entre 26 e 39 anos, despenderam cerca de 40 minutos para realizar os testes e responder a um questionário pós-teste.

Com base no trabalho proposto por Parikh (Parikh et al., 2007), um protocolo foi desenvolvido para realizar o teste e garantir que os dados fossem recolhidos com precisão. Este protocolo de teste foi definido para padronizar os resultados dos dois ensaios, e consiste basicamente em sete passos:

1. Voluntários foram instruídos sobre o procedimento de teste e os seus objectivos durante o teste;
2. Foram dados a cada participante 10 minutos para treinar no ambiente simulado. Assim, o utilizador poderia experimentar a cadeira de rodas e fazer os ajustes necessários da interface homem-máquina.
3. O participante foi convidado a conduzir a cadeira de rodas (1 volta) através do circuito no ambiente simulado utilizando o sistema manual tradicional.
4. Pediu-se ao voluntário para conduzir a cadeira de rodas (1 volta) através do mesmo circuito simulado, mas com a ajuda do controle compartilhado.
5. O utilizador conduziu a cadeira de rodas (1 volta) no ambiente real com o controle manual.
6. O utilizador conduziu a cadeira de rodas (1 volta) no ambiente real com o controlo compartilhado.
7. Para avaliar o paradigma de controlo compartilhado, a percepção de segurança do utilizador, foi aplicado um questionário após os testes.

Os participantes foram instruídos do objectivo principal de conduzir em segurança e como segundo objectivo realizar a volta no menor tempo possível. O tempo só foi mencionado como um objectivo secundário para impedir que voluntários navegassem muito lentamente, não sendo, no entanto, utilizado no processo de avaliação. Os testes no mundo real foram realizados com o protótipo 3. Na simulação, configurou-se o sistema para que apresentasse as mesmas características do protótipo real e do ambiente real.

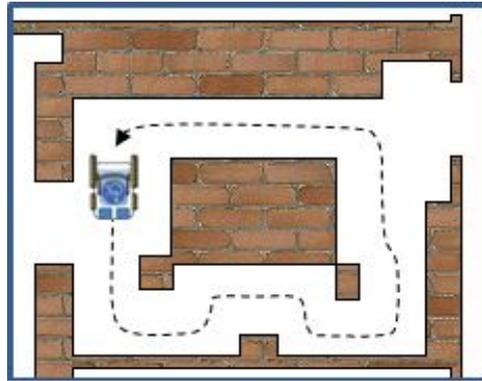


Figura 128: Representação do ambiente com o circuito fechado onde ocorreram os testes

Durante a execução do teste, os voluntários foram convidados a controlar a cadeira através dos movimentos da cabeça. Como nenhum dos voluntários tinha já utilizado este método de interação antes, isso simulou a dificuldade que normalmente os deficientes apresentam na condução de uma cadeira de rodas.

### 5.3.6 Interação da CRI com Macro Agentes

O objectivo deste teste foi avaliar o novo sistema de comunicação dos agentes e seus comportamentos (planeamento, navegação) num ambiente simulado.

O cenário é composto de um Agente Médico que solicita uma CRI para transportar um determinado paciente a um lugar específico. Assim, para este propósito um mapa hipotético de um hospital foi modelado e carregado no simulador IntellWheels. O mesmo mapa adicionado de um grafo de caminhos foi carregado no Agente Inteligência, a negociação entre o agente médico e as CRI para realizar a tarefa utiliza o protocolo FIPA Contract Net (FIPA, 2009), e a utilidade da CRI para realizar a tarefa é dada pela(Eq. 32).

$$Utilidade = \frac{100 * (3 + PT)}{(3 * Custo) + [PT * (Trajecto + Descarga)]}, PT \in \{0; 1,5; 3\} \quad (Eq. 32)$$

A Figura 129 e a Figura 130 ilustram os protocolos de interacção entre os agentes: Médico-Controlo e Controlo-Inteligência, respectivamente.

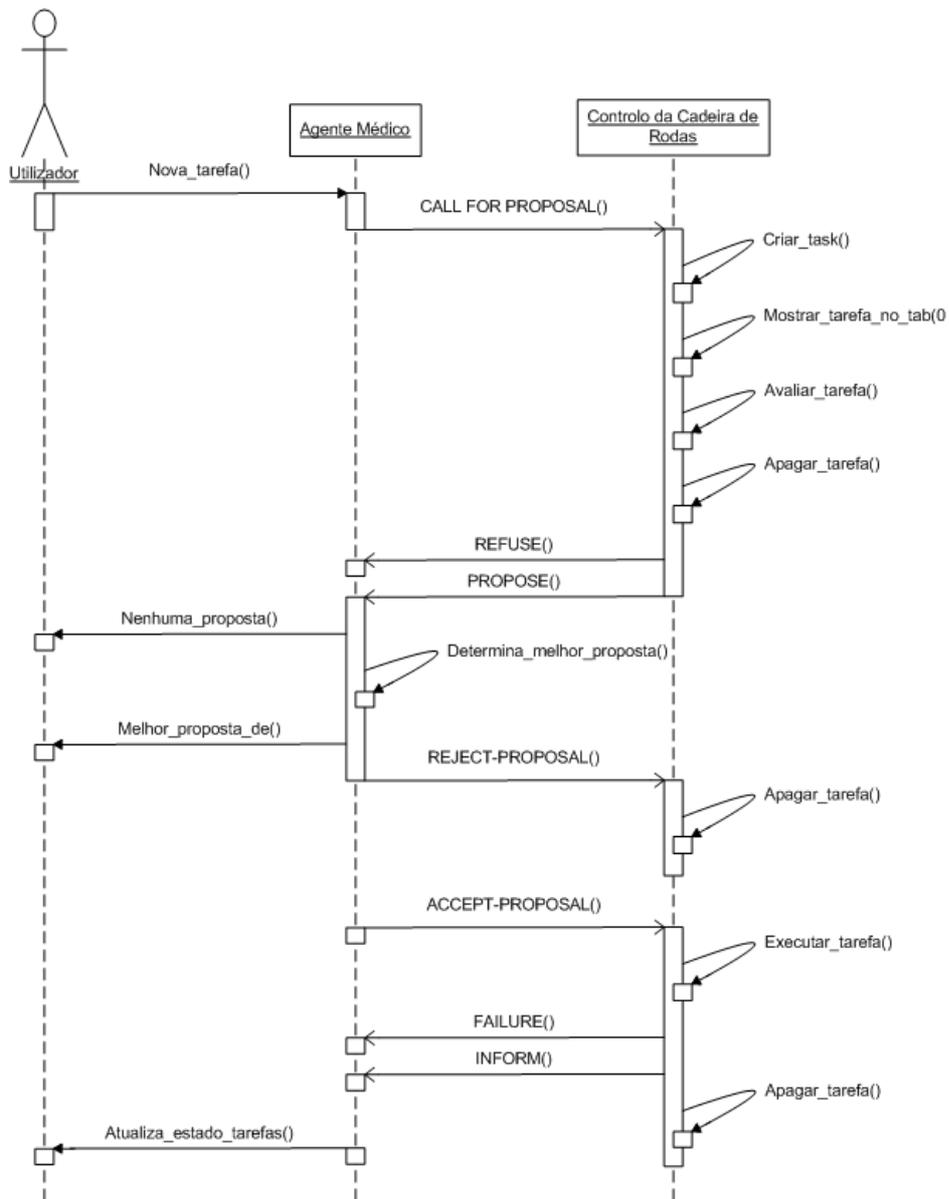


Figura 129: Diagrama de interação entre os agentes Médico x Controlo

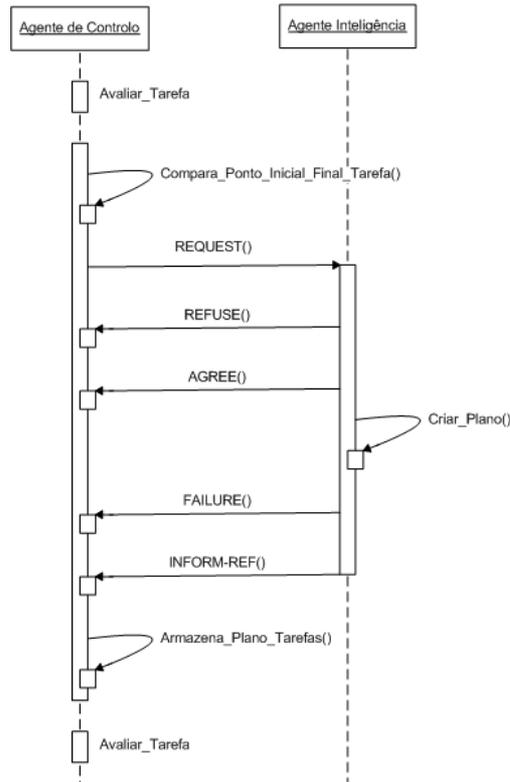


Figura 130: Diagrama de interação entre os agentes Controlo x Inteligência

A tarefa solicitada foi definida como pegar paciente presente no nó 3 e transportá-lo para o nó 9, localizado em outra divisão hospital. No entanto, o teste foi configurado para que a CRI, no momento do pedido, estivesse no nó 19. O mapa, o grafo de caminhos com destaque aos nós 3, 9 e 19 podem ser vistos na Figura 131.

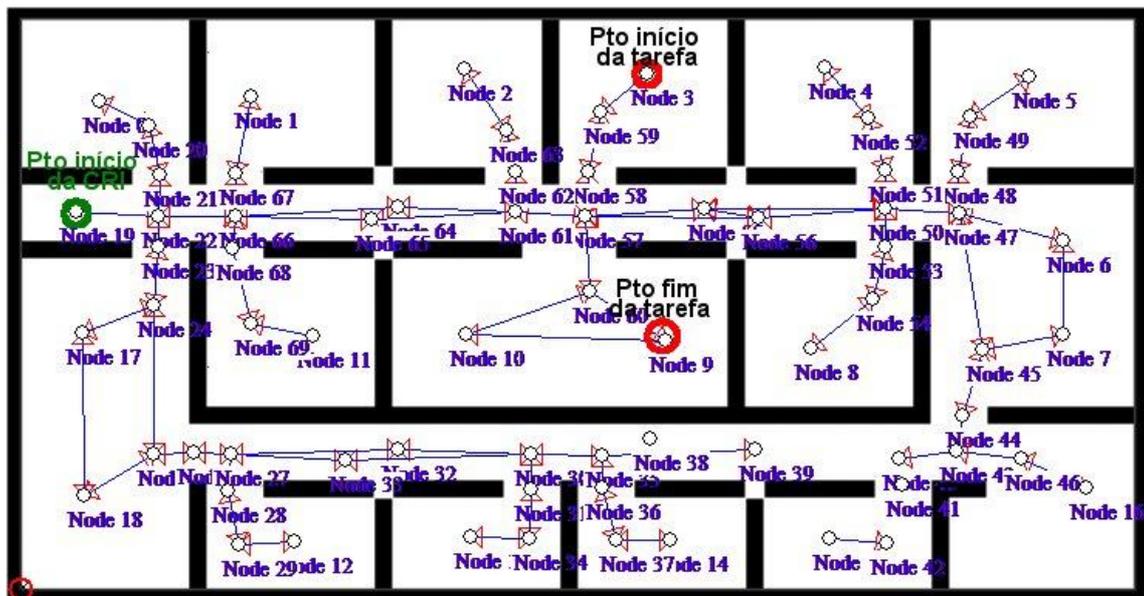


Figura 131: Mapa hipotético do hospital com o grafo de trajetórias

Para este ensaio os agentes Médico, controlo, inteligência e percepção foram utilizados (Figura 132).

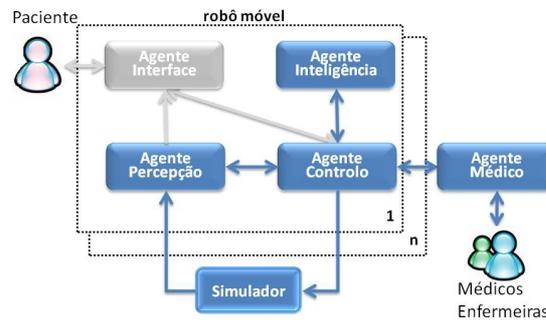


Figura 132: Diagrama dos agentes envolvidos neste ensaio

Onde PT = prioridade da tarefa, Custo = custo para realizar a tarefa, Trajecto = custo relacionado para chegar no local da tarefa, Descarga = custo relacionado com o tempo de subir e descer da CRI.

O teste será considerado bem sucedido se o agente reactivo for capaz de oferecer uma proposta para a tarefa e, em seguida, cumprir a tarefa do plano, deslocando a partir do nó 19 ao nó 3 e em seguida ao nó 9.

## 5.4 Resultados dos Testes

Ao longo desta secção os diferentes resultados dos testes serão apresentados, divididos nas subsecções descritas anteriormente

### 5.4.1 Teste da MMI em Ambiente Real

Na primeira etapa deste teste utilizou-se o joystick e comando de voz como canais de interacção com a CRI. Para a CRI passar a movimentar-se o sistema necessita de passar para o estado manual. Inicialmente ele encontrava-se no estado de espera, e neste estado qualquer comando com o bastão (*stick*) do joystick não gera acção nenhuma. Neste caso, foi utilizado um comando de voz (*“Manual”*) para activar o controlo manual. Em seguida a cadeira foi conduzida pelo utilizador até o fim do corredor onde parou após o comando de voz (*“Stop”*) o que fez o sistema sair do modo manual. De seguida, um novo comando de voz foi dado (*“Right Spin”*) o que fez a cadeira girar no próprio eixo para a direita e novamente um comando de *“Stop”* e outro de *“Manual”* foram dados à cadeira que prosseguiu, sendo depois conduzida até o final do percurso. Em seguida, o mesmo teste foi realizado trocando os comandos do joystick pelos comandos dados por gestos com a cabeça. A Figura 133 ilustra estes dois testes com a MMI.



Figura 133: Controlo da CRI através MMI em um corredor

A experiência adquirida nestes testes comprovou que é possível conduzir a cadeira de rodas apenas usando movimentos da cabeça com bom desempenho. No entanto, esses testes foram feitos sem qualquer fonte de distração. Noutras experiências foi confirmado que controlar a cadeira de rodas usando apenas com os movimentos da cabeça, configurado de forma muito sensível e com diversas fontes de distração é uma tarefa muito complexa.

### 5.4.2 Navegação Autónoma

Neste teste foram seleccionadas as seguintes opções do planeador: objectivo = "Ir para o quarto", lugar = "Quarto 2" e paciente = "Paciente 1". Esta selecção de parâmetros resultou na seguinte representação de estados para o objectivo:

$\text{On}(P1,R2) \wedge \text{WithoutC}(P1) \wedge \text{On}(C1,H)$ , ou seja, o paciente 1 está no quarto 2 e paciente 1 sem cadeira e cadeira 1 está no corredor.

A Figura 134 apresenta a janela do planeador com a representação do "estado do mundo" antes de gerar o plano e depois da execução do planeador.

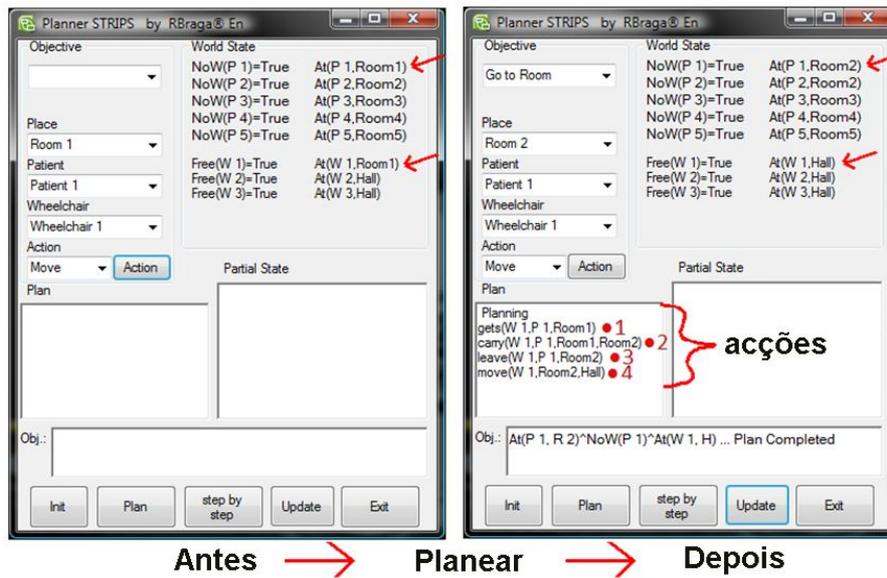


Figura 134: Geração de plano de alto nível

O plano gerado pelo planeador foi a seguinte sequência de acções: gets(W1, P1, Room1), carry(W1, P1, Room1, Room2), leave(W1, P1, Room2), move(W1, P1, Hall), ou seja, pegar o paciente 1 no quarto 1, levar este paciente para o quarto 2, e finalmente ir para o corredor conforme o era suposto.

Após a geração do plano, este é enviado ao agente controlo que executa de forma sequencial as acções descritas no plano. Para cada acção de movimento da CRI (move) é gerado um plano de trajectória utilizando o algoritmo A\*. A rota total planeada e o caminho percorrido pela CRI na simulação estão representadas na Figura 135.

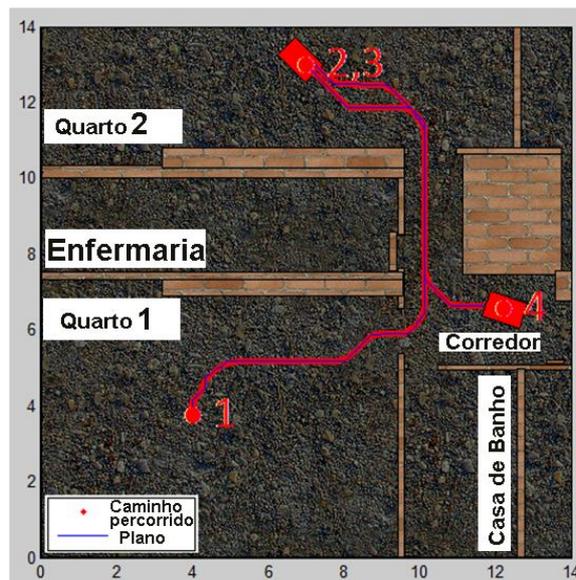


Figura 135: Rotas planeadas e percorridas em modo autónomo.

### 5.4.3 Realidade Mista

#### Desvio de obstáculos

Os resultados deste teste estão ilustrados na Figura 136 por uma sequência de quatro imagens. Nessa sequência de imagem é possível observar a CRI real, a evitar uma mesa e paredes virtuais através das informações dos sensores virtuais. Salienta-se que objectos reais, como a caixa e as paredes do ambiente real, não são percebidos pela CRI pois seus sensores de proximidade estão desligados durante esta experiência.

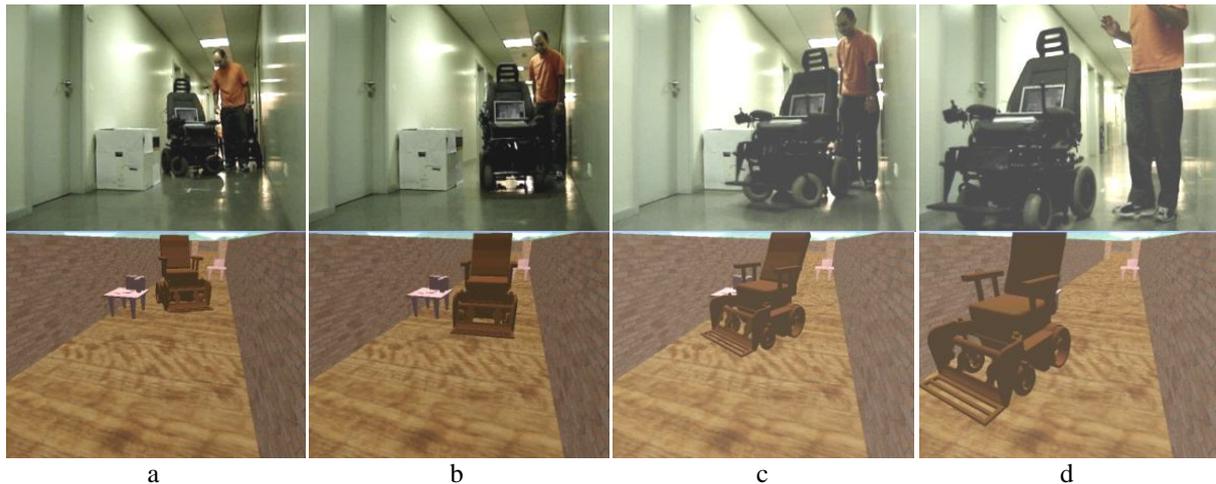


Figura 136: Teste de Realidade Mista: CRI real (imagens superiores) interagindo com objectos virtuais (imagens)

#### Laboratório de reabilitação

A Figura 137 ilustra o sistema configurado para auxiliar os pacientes em treinos com a cadeira real num ambiente virtual. Nesta configuração a CRI é colocada no modo de RM mas com os motores desconectados das rodas (sistema original que liberta as rodas para que a cadeira possa ser empurrada), deste modo a cadeira não se move mas os *encoders* detectam a rotação dos motores e é calculada uma “nova posição” da cadeira que é enviada ao simulador. Este comportamento permite que o utilizador conduza a cadeira virtual fornecendo comandos na cadeira real.



Figura 137: Laboratório de reabilitação em realidade mista

Outra possibilidade, com o auxílio de óculos de realidade virtual, será, numa sala vazia, utilizar a cadeira rodas em modo RM com as rodas conectadas aos motores normalmente. Deste modo pode-se criar um ambiente com os obstáculos todos virtuais e o utilizador navegar a cadeira real neste ambiente. Um ambiente simples de RM permite auxiliar os pacientes a melhorar sua capacidade de condução da cadeira de rodas motorizada. Exercícios com os pacientes a utilizar cadeiras de rodas reais em cenários virtuais podem ser executados com algum realismo, eliminando o risco de ferimentos e reduzindo o stress da condução da cadeira de rodas em ambiente real.

## 5.4.4 Plataforma de Comunicação

Esta subsecção apresenta os resultados dos diferentes testes focados em avaliar a eficácia e eficiência da plataforma de comunicação.

### 5.4.4.1 Efectividade e a eficiência da plataforma

Os dados de tempo obtidos nos testes estão em milissegundos e foram utilizados para estabelecer uma relação com o número de aplicativos em execução. A Figura 138 representa a relação entre o tempo necessário para as aplicações para descobrirem e elegerem um *container* entre eles e o número de agentes/aplicações no sistema.

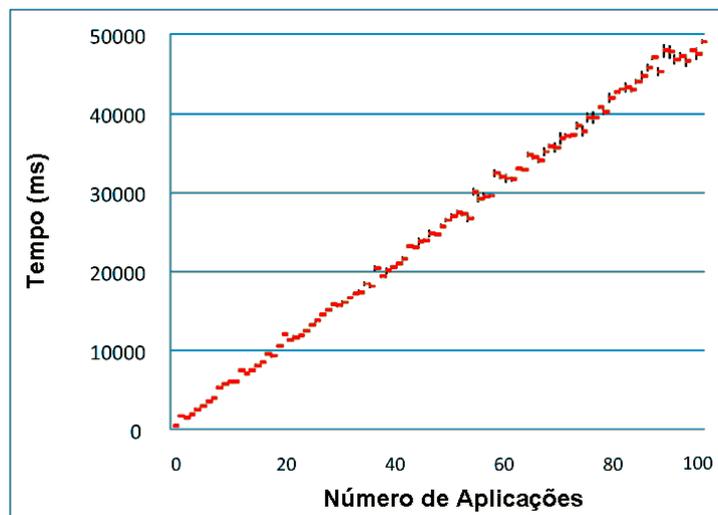


Figura 138: Tempo necessário para que os aplicativos consigam eleger um *container* local

Os dados colectados mostram que o tempo necessário para a eleição do *container* cresce linearmente com a quantidade de aplicativos presentes. A Figura 139 representa os resultados obtidos durante o ensaio que adquire o tempo necessário para o *container* eleito colectar as informações dos aplicativos e construir uma nova LAL.

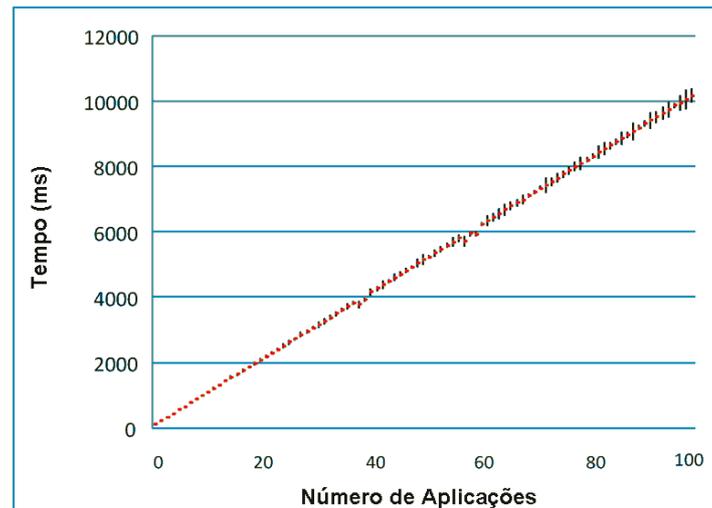


Figura 139: Tempo necessário para reunir todas as informações dos aplicativos e criar uma nova LAL

Os resultados obtidos demonstram que o tempo necessário para colectar informações de todos os aplicativos cresce também linearmente com quantidade de aplicativos presentes no ambiente. A Figura 140 representa os resultados obtidos ao medir o tempo necessário pelo *container* eleito para distribuir a LAL entre os aplicativos.

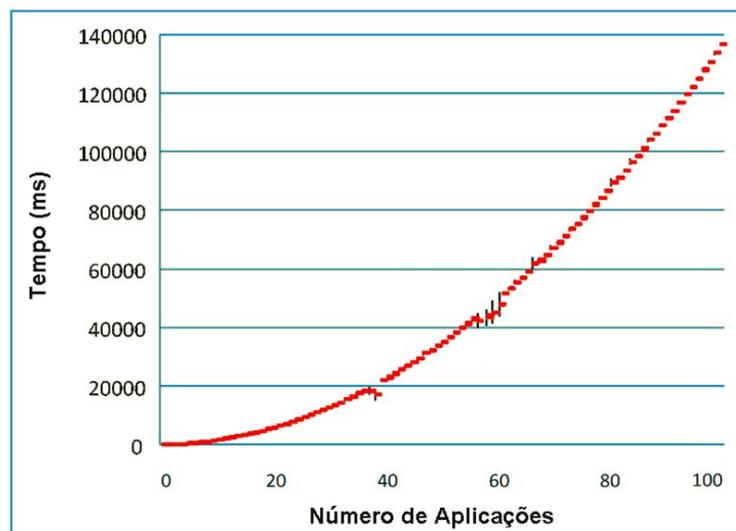


Figura 140: Tempo necessário para informar todos os aplicativos de uma nova LAL e distribuí-la

Os resultados obtidos mostram que o tempo necessário para distribuir um LAL evolui exponencialmente com a quantidade de aplicativos presentes. O mau desempenho é devido ao método de enviar uma linha LAL cada vez. Estes resultados motivaram uma mudança no algoritmo da distribuição. O novo algoritmo constrói uma mensagem com a quantidade máxima de linhas da LAL até que seja atingido o limite de tamanho da mensagem. Desta forma o número de mensagens enviadas foi reduzido para seu mínimo.

A Figura 141 mostra o tempo necessário para eleger um novo *container* quando o original, por algum motivo, não conseguiu comunicar dentro do prazo estabelecido. Este tempo cresce linearmente na proporção que aumenta o número de aplicações. Os dados colectados mostram a eficácia dos algoritmos e baixa sobrecarga do sistema.

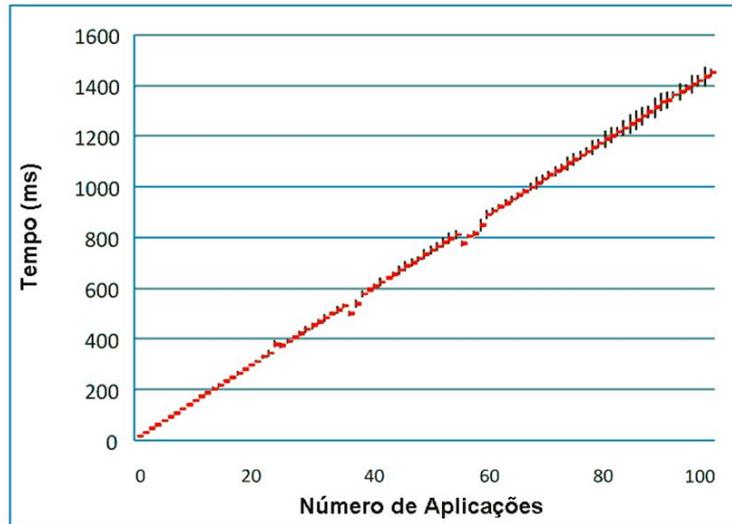


Figura 141: Tempo necessário para eleger um novo *container*

### 5.4.4.2 Teste comprador/vendedor de livros

Este teste tem por objectivo verificar apenas o comportamento de uma interacção simples e fazer a comparação entre as duas plataformas. A Figura 142 descreve a interacção entre os agentes comprador e vendedor de livros implementado em JADE e a Figura 143 descreve a interacção na plataforma IntellWheels. Os dados foram registados e comparados entre as plataformas.

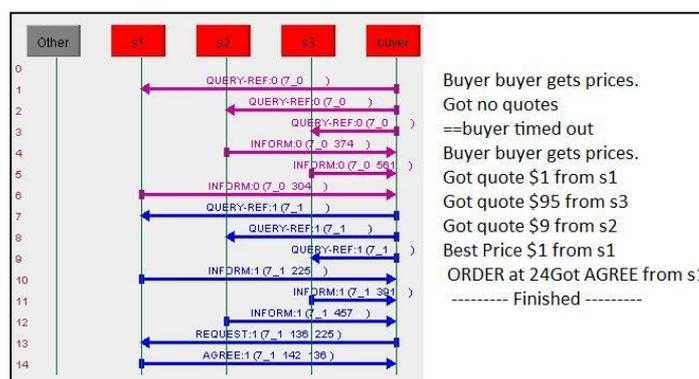


Figura 142: Resultados da interacção comprador/vendedor em JADE obtidos através do *Sniffer* e *command line*.

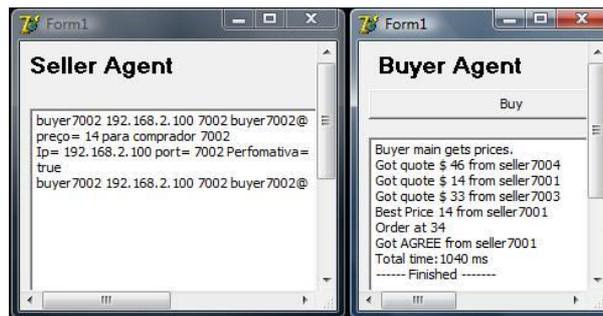


Figura 143: Mensagens das interações comprador/vendedor na plataforma de comunicação IntelWheels

Comparando os dados registados, obtidos do teste com a plataforma proposta e obtidos dos testes JADE, nota-se que os resultados são similares, a única diferença fica no tempo necessário para mensagens chegarem e sua ordem de chegada.

### 5.4.4.3 Escalabilidade da plataforma

Objectivo deste teste foi avaliar o desempenho do sistema e compará-lo com resultados de um mesmo cenário implementado na plataforma JADE, que é comumente utilizada. A fim de obter dados de desempenho significativos, o teste foi implementado usando um único agente Comprador e vários agentes Vendedores. Este teste foi concebido desta forma porque o agente mais exigente neste cenário é o comprador. É ele que inicia e decide o curso da interação.

Em ambos os casos, o tempo necessário para concluir com sucesso o pedido dum livro e sua aquisição foi medido em cinco repetições do teste e calculado o valor médio. Os resultados obtidos podem ser vistos na Figura 144 para o JADE e na Figura 145 para o sistema proposto.

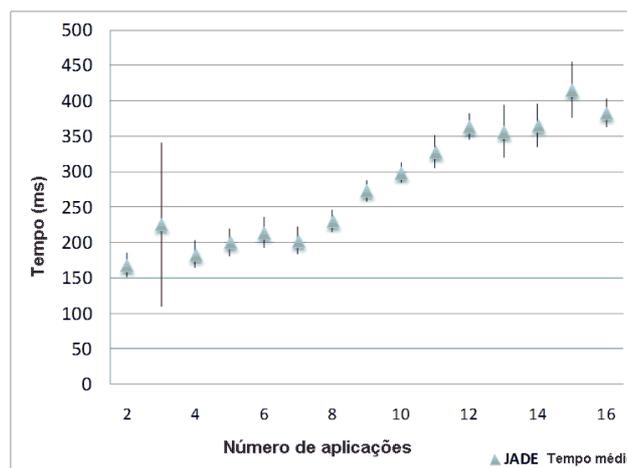


Figura 144: Resultados obtidos para a compra de livros utilizando JADE

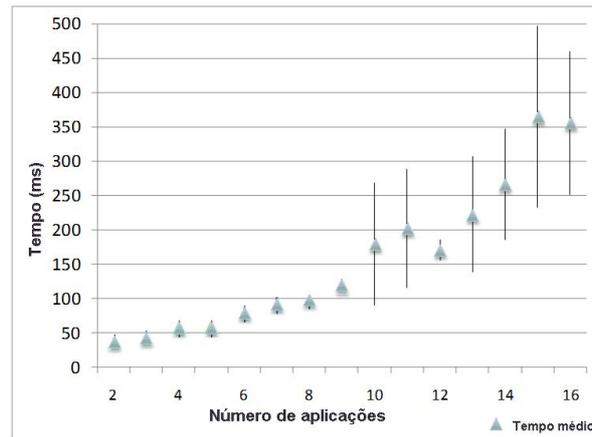


Figura 145: Resultados obtidos para a compra de livros utilizando a plataforma IntellWheels

É possível ver nos resultados deste teste que a plataforma JADE exige mais tempo do que a plataforma proposta para executar a interação. Esse comportamento é mais evidente para um número reduzido de agentes. Nestes casos a nova plataforma foi capaz de executar acima de 5 vezes mais rápido do que o JADE. No entanto, quando o número de agentes aumenta, a diferença entre as duas plataformas diminui, mas mesmo assim o sistema proposto é ainda capaz de superar o JADE.

### 5.4.5 Controle Compartilhado em *Firmware*

O conjunto de experiências, descritas na proposta deste cenário de teste, gerou dados quantitativos e qualitativos. Todos os dados obtidos foram utilizados para avaliar se cada voluntário foi realmente auxiliado na condução da CRI, e não apenas avaliar o seu desempenho individual.

Com base nos números de colisões de cada tentativa (sem e com controle compartilhado), o desempenho do algoritmo de controle compartilhado será avaliado tanto no ambiente de simulação (Figura 146), como no ambiente real (Figura 147).

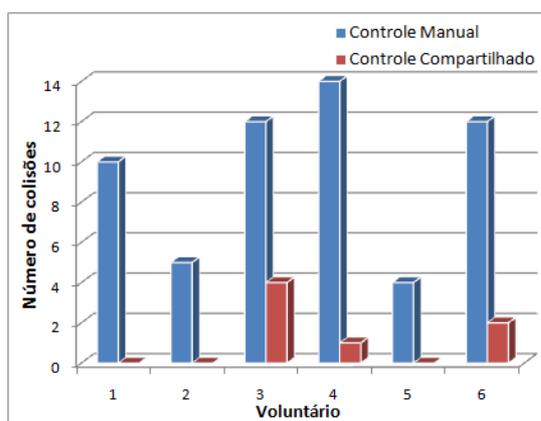


Figura 146: Número de colisões por voluntário em ambiente simulado

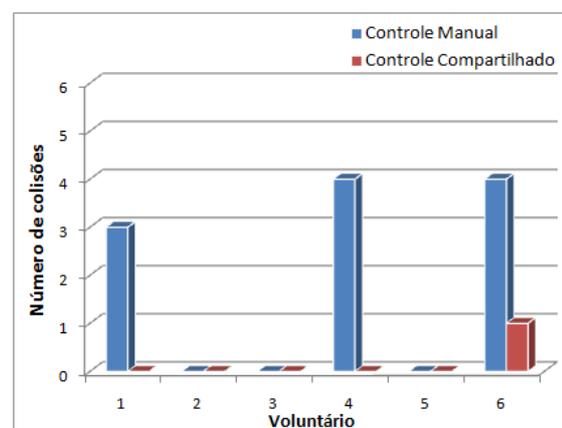


Figura 147: Número de colisões por voluntário em ambiente real

Por outro lado, é importante avaliar o algoritmo tendo em conta o ponto de vista do utilizador. Para medir a percepção do utilizador, foi aplicado um questionário composto por cinco afirmações referente ao paradigma de controlo, sendo que os entrevistados foram convidados a especificar em que nível eles concordam com a afirmação numa escala Likert<sup>19</sup> de um a cinco (1 = discordo totalmente, 2 = discordo parcialmente, 3 = indiferente, 4 = concordo parcialmente, 5 = concordo totalmente):

1. Eu sinto-me confortável quando dirigindo a cadeira de rodas.
2. Eu sinto o comportamento do controle sobre a cadeira de rodas.
3. É fácil de conduzir a cadeira de rodas em espaços desordenados.
4. A cadeira de rodas requer pouca atenção na condução.
5. A cadeira de rodas tem o mesmo comportamento no ambiente simulado e no ambiente real.
6. Eu acredito que o controle compartilhado me ajudou durante a tarefa de navegação.

Para efeitos de análise, a percepção de segurança do utilizador foi tratada como uma variável indirecta medida através da soma dos pontos das quatro primeiras afirmações (Figura 148).

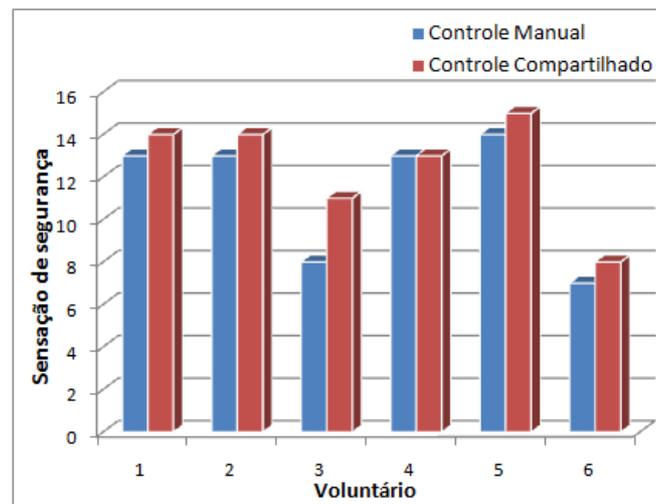


Figura 148: Sensação de segurança com e sem a ajuda do controle compartilhado

Outra inferência pode ser feita sobre o comportamento da cadeira de rodas no simulador. Por meio da quinta afirmação, tentou-se medir o quão realista é o comportamento simulado da cadeira de rodas. Um valor de limite de 3 foi usado para comparar os resultados (Figura 149). Lembre-se que na escala de Likert um valor de 3 significa que os entrevistados não

<sup>19</sup> A Escala Likert é um tipo de escala de resposta psicométrica utilizada comumente em questionários, e é muito utilizada em pesquisas de opinião.

concordam e nem discordam com a declaração, assim, um valor maior que 3 significa que a cadeiras de rodas simulada reage da mesma forma que as cadeiras de rodas reais na perspectiva do utilizador.

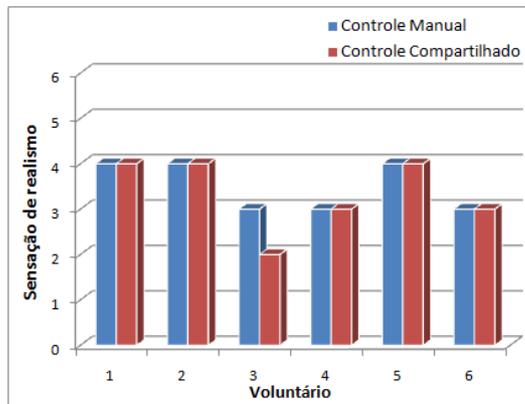


Figura 149: Quão realista os voluntários acharam o simulador

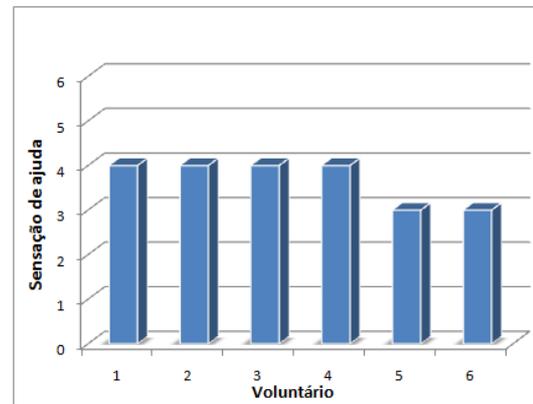


Figura 150: Sensação de ajuda com o controlo compartilhado

Finalmente, um último resultado do questionário é a percepção do utilizador referente a ajuda fornecida pela cadeira de rodas. Neste caso foi avaliada através da sexta afirmação, apenas presentes na secção controlo compartilhado no questionário (Figura 150). Além disso, de uma forma semelhante ao que foi mencionado anteriormente, um valor maior que 3 significa que o utilizador sentiu-se ajudado pelo algoritmo.

A partir da análise dos dados com teste Wilcoxon, em ambiente real, os resultados indicam uma diferença significativa entre o número de colisões com e sem o paradigma de controlo compartilhado ( $T = 0$ ,  $n = 8$ , valor prova = 0.0135). No ambiente simulado, os resultados também indicam que há uma diferença significativa entre o número de colisões com e sem o paradigma de controlo compartilhado ( $T = 0$ ,  $n = 8$ , valor prova = 0,009). Portanto, a nossa análise fornece evidências de que o paradigma de controlo compartilhado está a fornecer assistência para a redução do número de colisões (Petry et al., 2010).

#### 5.4.6 Interação da CRI com Macro Agentes

O objectivo deste teste foi avaliar o sistema de comunicação dos agentes e o seu comportamento num ambiente simulado. Os dados colectados mostram que o teste foi bem sucedido. A plataforma foi capaz de se organizar e anunciar as configurações aos agentes. Cada agente foi ainda capaz de colaborar com a sua habilidade para a realização da tarefa.

O agente Controlo foi capaz de interagir e colaborar com o agente Inteligente formalizando uma proposta referente a tarefa requerida pelo agente Médico. A proposta eleita resultou nos seguintes dados para a tarefa proposta: Custo – 273,47; Trajecto – 134,86; Descarga – 60; Utilidade da Tarefa – 0,43.

A Figura 151 mostra a sequência registada de mensagens recebidas pelos agentes Controlo e Inteligência.

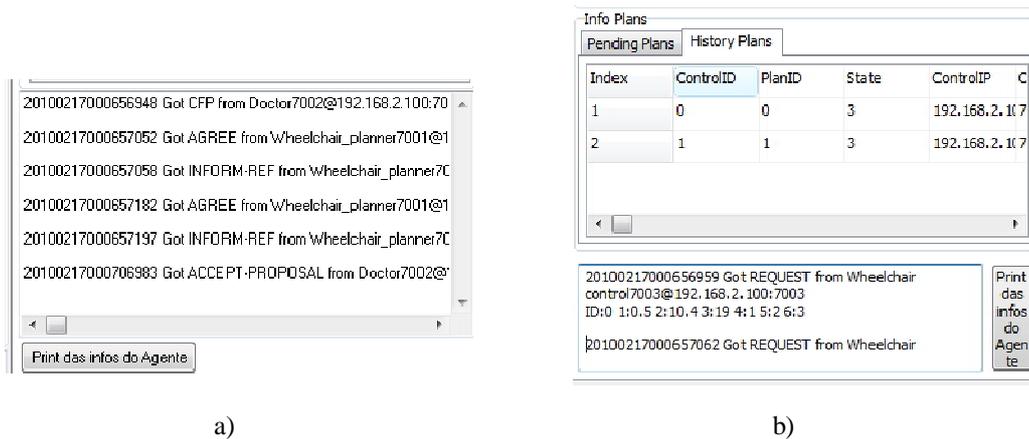


Figura 151: a) Consola de comunicação do agente Controlo. b) Consola de comunicação do agente Inteligência e tabela de planos.

A partir da consola do agente Inteligência é possível verificar os planos para esta tarefa. Esta informação é confirmada pelo agente Controlo que recebeu duas mensagens **AGREE** e duas **INFORM-REF**, como esperado, devido o comportamento implementado e à tarefa solicitada. A primeira interacção refere-se à tarefa de pré-plano necessária para a CRI deslocar-se da posição da sua última acção para a posição inicial da tarefa em questão. A segunda mensagem **AGREE** e **INFORM-REF** refere-se ao plano da tarefa propriamente dita.

Segundo informações contidas na consola do agente Controlo também é possível notar a mensagem **ACCEPT-PROPOSAL**, referindo-se a aceitação do agente Médico. Esta mensagem dispara um evento, o qual traduz um plano pendente numa lista de acções que o agente ira executar para atingir o objectivo. Após definir a lista de acções, o agente passa a executá-las. A lista de acções criadas para essa tarefa pode ser visualizada na Figura 152.

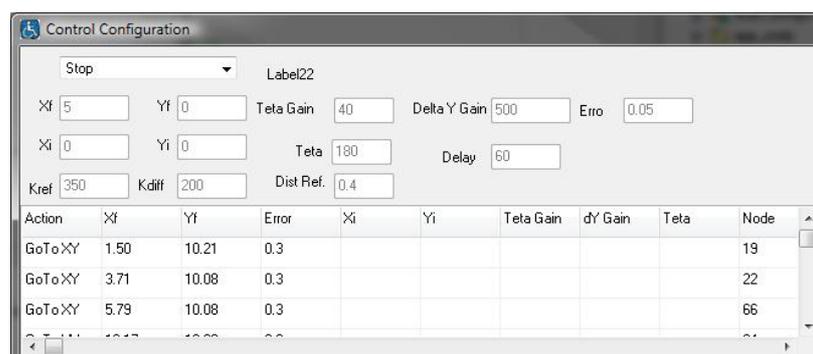


Figura 152: Lista de acções pendentes do agente Controlo

Como esperado, após a lista de acções criada a CRI começa a mover-se no ambiente. Os movimentos resultantes podem ser visualizados nas imagens do simulador 3D (Figura 153).

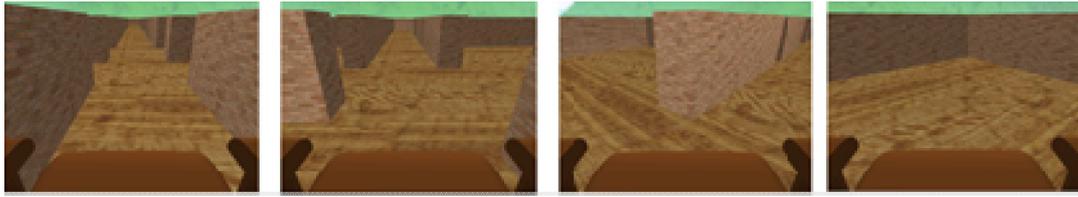


Figura 153: Imagens do visualizador em modo 3D que representam o movimento da CRI do nó 19 para o nó 3.

Este movimento foi registado e pode ser visto na Figura 154, que ilustra o percurso realizado pela CRI. Ela parte do nó inicial 19 para o nó 3, representada em azul. Recolhe o paciente neste local e transporta-o até o local representado pelo nó 9. O que mostra que a tarefa solicitada foi realizada com sucesso.

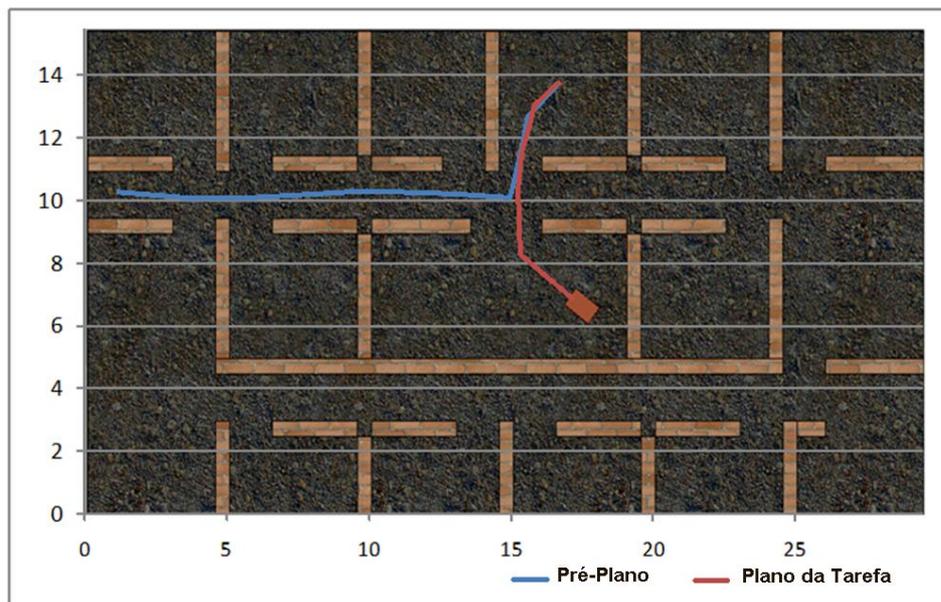


Figura 154: Trajecto percorrido pela CRI

## 5.5 Sumário

O principal objectivo deste capítulo consistiu em apresentar os testes que permitiram avaliar a plataforma desenvolvida. Para este fim, cenários e baterias de testes foram definidos e caracterizados. Além disso, os protocolos implementados também foram descritos para permitir a reprodução dos testes.

Foram apresentados o desenvolvimento e a construção dos protótipos desenvolvidos no âmbito desta tese. Essa etapa teve dois objectivos principais. Em primeiro lugar permitir disponibilizar um equipamento que permitisse testar e avaliar a eficácia dos algoritmos desenvolvidos num ambiente real. O segundo objectivo, foi adquirir experiência para projectar/especificar um conjunto de hardware. Com o resultado do projecto e a

implementação do protótipo 3 pode-se concluir que é possível agregar habilidades à cadeira de rodas motorizada com um hardware de baixo custo sem alterar significativamente a sua aparência e mantendo a ergonomia da cadeira de rodas motorizada original.

Quanto aos ensaios realizados, os resultados obtidos demonstraram a eficácia do sistema numa variedade de situações. Os testes variaram desde experiências simples para prova de conceito até modelos de interação mais complexos, como por exemplo em interações entre os agentes que compõem a plataforma IntellWheels. Sendo assim foram apresentados nesta tese seis grupos de ensaios: teste com a MMI, navegação autónoma simulada, realidade mista, plataforma de comunicação, controlo compartilhado e interação da CRI com agentes macros.

Tendo em consideração a dificuldade que certos pacientes têm em controlar a cadeira de rodas através do comando tradicional, procurou-se adicionar novos meios de interação entre a cadeira e o utilizador, através da criação dum sistema de múltiplas entradas. Uma vez implementado, o sistema faculta ao utilizador a escolha do tipo de comando que melhor se adapta às suas necessidades, o que resulta num aumento do nível de conforto e segurança. Como verificado no cenário de teste da MMI é possível controlar a CRI de forma simples e eficaz utilizando vários canais de entrada.

No decorrer deste projecto gerou-se a necessidade de propor e desenvolver novas soluções de comunicação para agentes, aplicáveis na área dos robôs móveis em ambientes dinâmicos. Os resultados dos testes com a arquitectura proposta, mostram que a nova plataforma de comunicação para SMA pode fornecer comunicações e cooperação entre agentes na organização dos mesmos dentro da plataforma, o que resultou numa nova solução para sistemas baseados em agentes.

Nos cenários de teste *Interação de CRI com macro agentes*, a arquitectura proposta provou ser eficaz. Além disso, a comparação feita entre a plataforma de comunicação implementada e JADE demonstrou que o sistema proposto e os seus algoritmos podem superar o JADE numa variedade de cenários. No entanto, os dados recolhidos mostram que ainda deve ser melhorada a optimização dos algoritmos em referência ao tempo e a eficiência de transmissão de dados, pois, para um elevado número de agentes, os tempos de manutenção da organização da plataforma começam a crescer demasiado.

Com o rápido e crescente número de projectos envolvendo cadeiras de rodas inteligentes, deve ser dada mais atenção aos sistemas de comunicação. Neste campo, a interação com outros dispositivos é absolutamente necessária, de modo a permitir a cooperação/colaboração entre os equipamentos existentes num ambiente dito inteligente.

Foi apresentada uma nova abordagem de melhora da segurança para utilizadores de cadeira de rodas. Para ajudar os pacientes em suas tarefas de navegação, foi adoptada uma metodologia de desvio de obstáculos. Com base numa abordagem dinâmica, nomeadamente o *potential field*, sendo que este trabalho estendeu e complementou esta metodologia do ponto

de vista de controlo compartilhado. Assim, para reduzir o custo computacional e executar o algoritmo em tempo real, cada medida, realizada a partir dos sensores de distância, foi tratada como uma força de repulsão e desta forma não foi necessário criar um mapa do ambiente. É importante notar que esta proposta é muito dependente dos sensores utilizados, sendo a inclusão de filtros nas informações obtidas dos sensores uma importante melhoria para aumentar a robustez e reduzir as oscilações dos valores medidos. A principal razão das colisões durante os testes com o paradigma de controlo compartilhado deve-se a um ponto cego na lateral da cadeira que foi eliminado com a adição de mais dois sensores de ultra-som.

O número de colisões no ambiente simulado é claramente muito maior e pode ser explicado através de comentários dos voluntários e dos seus comportamentos durante os testes. Em primeiro lugar, o ambiente 3D do simulador não fornece uma percepção precisa de profundidade e distância para objectos, causando conflitos no circuito de teste estreito. No entanto, a segunda razão está relacionada com a forma de condução do utilizador, pois no ambiente simulado os voluntários tendem a relaxar-se e a reduzir a atenção ao circuito, principalmente, porque não existe risco de sofrerem danos físicos numa colisão.

No próximo capítulo serão apresentadas as conclusões encontradas nesta tese, inicialmente com uma discussão sobre os resultados e finalmente serão apresentados os trabalhos futuros.



# Capítulo 6

## Conclusões

Este capítulo tem como principal objectivo analisar o trabalho desenvolvido no que respeita ao cumprimento dos objectivos iniciais, bem como analisar, numa perspectiva crítica, os resultados obtidos. Inicialmente, será apresentada uma síntese do trabalho desenvolvido e, em seguida, serão discutidos os resultados que foram obtidos nos ensaios apresentados no capítulo anterior. Além disso, serão discutidas as aplicações do sistema desenvolvido numa visão mais alargada e serão apontados novos caminhos para a evolução deste trabalho.

### 6.1 Síntese do Trabalho

O objectivo geral deste trabalho foi o estudo, projecto e implementação de uma plataforma de desenvolvimento de Cadeiras de Rodas Inteligentes (CRI). As CRI desenvolvidas e baseadas nesta plataforma deverão ser utilizadas em ambientes hospitalares ou similares que possam fornecer mapas e estruturas que facilitem a localização das mesmas. Devem também possuir capacidades de interacção com outros robôs de serviços e dispositivos ditos “inteligentes” (por exemplo, portas e elevadores autónomos) presentes no ambiente. Para atender a estas metas, foram:

- Estudados os conceitos de Robótica Inteligente, Sistemas Multi-Agente, metodologias utilizadas em trabalhos relacionados, assim como os inúmeros projectos de CRI desenvolvidos por outras universidades e laboratórios de investigação;
- Desenvolvidos conceitos e o projecto de uma plataforma de desenvolvimento de cadeiras de rodas inteligente, definindo-se uma plataforma completa de hardware e software;

- Construídos três protótipos reais de cadeira de rodas inteligente, a partir de cadeiras de rodas motorizadas comerciais. Nestes protótipos foram agregados o sistema sensorial e de actuação adequados;
- Desenvolvida uma metodologia de interacção multimodal, flexível e programável para comandar a CRI;
- Estudados metodologias de planeamento e implementado um planeador simples, na forma duma ferramenta gráfica e integrada no sistema. Este planeador baseado em *Graph Plan* é capaz de gerar sequências de acções de alto nível para a CRI;
- Desenvolvido um simulador de CRI que permitiu testar as metodologias desenvolvidas;
- Desenvolvido um metamodelo para robôs móveis e a sua especificação no projecto IntellWheels, o que resultou em uma modelação formal do sistema;
- Desenvolvida uma plataforma de comunicação de SMA e implementadas metodologias de comunicação e cooperação de agentes;
- Implementadas as funções de operação da CRI: manual, semi-automática e automática. Estas funções determinam a forma como as acções da CRI são controladas.
- Implementados diversos módulos, tais como: reconhecimento de comandos de voz, interface com múltiplas entradas e saídas com o utilizador e a CRI, planeamento inteligente, navegação autónoma, desvio de obstáculos em *firmware* e em software. Além disso, os módulos desenvolvidos foram avaliados de forma individual e integrada no conjunto tanto em simulação e como no protótipo real de uma cadeira de rodas inteligente;
- Validada a plataforma, assim como os seus diversos módulos, juntamente com o protótipo real de cadeira de rodas inteligente.

A Plataforma IntellWheels (PIW) foi concebida para ser uma ferramenta de suporte na investigação e desenvolvimento de cadeiras de rodas inteligentes. Com isso, este projecto não possuiu a intenção de apresentar um protótipo para pessoas com danos ou habilidades específicas, mas sim criar uma ferramenta de desenvolvimento genérica. Tendo isso em conta, a plataforma foi desenvolvida como um sistema multi-agente composto de vários módulos. O desempenho individual dos módulos não fez parte dos objectivos principais da tese, sendo que o foco foi a integração de todos os módulos. Da mesma forma, não houve uma elevada preocupação em relação à inteligência individual de cada agente do sistema, estando o interesse voltado para o nível de inteligência que pode emergir com a interacção do conjunto global de agentes.

## 6.2 Resultados

O projecto IntellWheels foi baseado em quatro ideias inovadoras. A primeira, foi o conceito de uma estrutura genérica que permita, de forma facilitada, desenvolver novas cadeiras de rodas inteligentes e algoritmos de controlo. A estrutura é suficientemente flexível para permitir facilmente a transformação de cadeiras de rodas motorizadas comerciais em CRI, com mínimas alterações do hardware. Além disso, isto facilita a introdução de novos módulos e algoritmos no sistema da cadeira de rodas inteligente.

A segunda inovação diz respeito à metodologia de interacção da plataforma IW, a qual se baseia numa interface multimodal flexível. A fim de verificar a eficiência do sistema e o controlo da cadeira de rodas através do módulo MMI, foram conduzidos algumas experiências, nas quais a cadeira de rodas foi controlada a partir de diferentes dispositivos de entrada (joystick, voz e gestos com a cabeça). Os resultados obtidos permitiram confirmar as capacidades resultantes da interface multimodal, entretanto, testes com o módulo de reconhecimento de voz demonstram a falta de robustez deste canal de interacção em situações ruidosas. Foi comprovado que a MMI permite ao utilizador controlar a cadeira de rodas através de sequências e combinações de entradas (por exemplo, botões, comandos de voz, expressões faciais e actuação no joystick) de um mesmo dispositivo de entrada ou de um conjunto de diferentes dispositivos de entrada.

A terceira inovação diz respeito aos cenários de Realidade Mista fornecidos pelo simulador IntellWheels. O simulador demonstrou que não é apenas capaz de simular ambientes e cadeiras de rodas, mas também criar um cenário no qual a interacção entre objectos virtuais e reais é possível (por exemplo, cadeiras de rodas, mesas, paredes e outros obstáculos).

Finalmente, a quarta inovação está relacionada com a plataforma de comunicação de SMA para robôs móveis reais, com base no conceito de cooperação entre os agentes para manter a organização do sistema de comunicação e preocupada com aspectos de segurança crítica no sistema de comunicação. Ao contrário de outros projectos nesta área, o conceito, apresentado neste projecto, refere-se a um SMA que por sua vez é composto por outro SMA. De uma forma geral, um robô móvel é um SMA, composto por micro agentes que interagem com outros robôs e agentes presentes no sistema dando origem a um outro SMA (macro agentes). O sistema possibilita que um dado micro agente de um robô possa colaborar com micro agentes de outro robô. Por exemplo, uma CRI que não tem ou por algum motivo perdeu a habilidade de planeamento, pode, através da comunicação com um planeador de outra CRI, gerar um plano.

O conjunto inicial de objectivos para a plataforma de comunicação foi alcançado. O sistema de comunicação desenvolvido é capaz de: prevenir erros de comunicação; de implementar comunicações seguras no campo da robótica móvel; de facilitar o projecto de

novas interacções entre agentes; de auto-adaptar-se ao ambiente de rede; de auto-reconfigurar-se a fim de prevenir falhas de sistema; e de implementar metodologias de SMA seguindo a padronização FIPA.

Esta tese apresenta uma síntese e análise das perspectivas dos autores mais relevantes nas áreas relacionadas com as CRI. Embora existam muitos trabalhos e diferentes abordagens nesta área, pode-se verificar a inexistência de uma CRI que apresente as reais capacidades que permitam a sua utilização no dia-a-dia. Cada projecto concentra-se em problemas mais específicos e poucos estão preocupados com o todo. Desta forma, é necessário conhecer os principais requisitos de uma CRI e quais as tecnologias que já estão disponíveis. Neste sentido, a tese fornece uma visão geral destes requisitos e apresenta uma visão geral destas tecnologias. Mesmo após uma extensa revisão bibliográfica, não foi possível encontrar nenhum trabalho desenvolvido, nesta área, cujo foco seja o de criar uma ferramenta de desenvolvimento, e que, também, permita gerar um sistema para reabilitação e treino dos pacientes. Com a utilização dos recursos de simulação e realidade mista, apresentados neste trabalho, é possível criar métodos e cenários para auxiliar no treino e na reabilitação dos pacientes que utilizam este tipo de equipamento.

Os conceitos e resultados desta tese acabaram por dar origem a um projecto de investigação financiado pela FCT (Fundação para a Ciência e a Tecnologia) no âmbito da reabilitação e integração das pessoas com deficiência. O projecto RIPD/ADA/109636/2009 foi aprovado e teve seu início em 01/07/2010, com uma duração de dois anos e um orçamento global de \$94 360 Euros.

Como exposto nesta tese, a plataforma desenvolvida apresenta o conceito de modularidade total, sendo este conceito aplicado tanto ao hardware como ao software. Os objectivos desta abordagem são: tornar mais perceptíveis os componentes envolvidos no sistema; permitir a manutenção e substituição dos módulos sem afectar o resto do sistema; simplificar o desenvolvimento; e incrementar a flexibilidade e fiabilidade do sistema. Um ponto importante a salientar é a robustez que resulta da possibilidade de criação módulos redundantes no sistema.

Embora diferentes conceitos tenham sido propostos através da plataforma IntellWheels (IW), até o momento, não foi possível colocar em prática ou testar todos eles. Deste modo, é possível classificar as características propostas em três estágios: implementada e testada; implementada, mas não testada; e planeada mas não implementada.

O primeiro caso inclui aquelas características que estão directamente relacionadas com a ideia geral da plataforma e foram avaliadas na secção Resultados. Assim, com o protótipo IW foi possível verificar a compatibilidade entre o suporte de hardware IntellWheels com cadeiras de rodas motorizadas comuns. Adicionalmente, as características visuais e ergonómicas não foram afectadas pela montagem dos dispositivos de hardware, atingindo assim outra meta proposta do projecto. Além disso, é importante enfatizar que o objectivo de

construir protótipos de baixo custo foram também alcançados, mantendo o custo de hardware abaixo de 3.500,00 € (2000,00 € para a cadeira de rodas motorizada, 500 € para portátil e 1.000,00 € para sensores e outros dispositivos de hardware). Todavia, a plataforma não impede a adição de novos sensores para o suporte de hardware. Pelo contrário, é aberta à adição de outros sensores desde que o impacto do seu custo no custo final do protótipo seja reduzido (por exemplo, *laser range-finders* tendem a alcançar preços muito mais baixos nos próximos anos e, conseqüentemente, dentro da gama permitida para manter o reduzido custo da plataforma).

O segundo caso diz respeito a algumas capacidades (tais como a localização e as entradas através de expressões faciais) que foram implementadas e cujos resultados não foram apresentados. Esse facto deveu-se a que, neste documento procurou-se introduzir a plataforma como um todo e não testar e otimizar especificamente cada capacidade da CRI. Entretanto, muitas dessas capacidades foram já avaliadas e os seus resultados publicados em conferências anteriores. Em (Braga et al., 2008), um modelo de movimento por odometria probabilístico para um sistema de localização activo foi discutido e testado. Uma solução para reduzir os erros de localização foi calcular a incerteza (variação) da odometria e se esta alcançar certo limiar, planejar novamente o caminho de maneira a fazer com que o robô passe através do marco mais próximo para corrigir o erro de localização. A entrada através de expressões faciais utiliza algoritmos de processamento de imagem para detecção de atributos, tais como segmentação de cores e detecção de bordas, seguido pela aplicação de uma rede neuronal para detectar as expressões faciais de interesse. Os resultados mostrados em (Faria et al., 2007) fornecem evidência da possibilidade de se conduzir confortavelmente uma cadeira de rodas inteligente utilizando expressões faciais. Entretanto, tal tipo de entrada ainda apresenta algumas limitações no que diz respeito à segmentação de cores (muito sensível para grandes variações de luz e desconsidera mudanças de cores) e à extracção de formas (sendo necessário melhorar a precisão sem aumentar o tempo de processamento).

A plataforma permitiu estudar e avaliar mecanismos para realizar, de forma manual, semi-autónoma ou autónoma, a execução dos desejos dos utilizadores expressados por linguagem de comando de alto nível.

Os resultados obtidos nos testes evidenciaram a eficácia do comportamento cooperativo entre um grupo de CRI e agente médico desenvolvido, que negociavam a execução de tarefas de transporte de paciente. Neste caso procurou-se otimizar a utilidade do sistema em relação ao menor tempo de espera do paciente. Outros testes demonstraram o comportamento colaborativo entre a CRI e o utilizador para auxiliá-lo no desvio de obstáculos, assim como a colaboração entre a CRI e outros dispositivos inteligentes (portas autónomas).

Finalmente, os conceitos propostos mas que não foram concluídos até a escrita desta tese, bem como ideias para melhorias da plataforma desenvolvida serão discutidos na próxima secção de trabalhos futuros.

## 6.3 Trabalhos Futuros

O projecto IntellWheels teve seu início quase em simultâneo com o início desta tese o que exigiu a formação de um background de estudo e desenvolvimento. Além disso, a alta complexidade do sistema e os diversos domínios científicos envolvidos no tema desta tese, exigiram um grande esforço inicial para o desenvolvimento do projecto. Inevitavelmente, as implementações dos protótipos reais acabaram consumindo o tempo disponível e condicionando o desenvolvimento do projecto. Assim, alguns recursos, funções e ensaios que seria interessante acrescentar a este trabalho, não foram implementadas devido aos prazos do projecto. Diante dessa situação, a seguir são listados importantes recursos, funções e ensaios, que poderão servir de ponto de partida para trabalhos de desenvolvimento e melhoramentos futuros da plataforma implementada no âmbito da investigação científica:

- Finalizar a integração do novo módulo de reconhecimento de voz com o agente Interface utilizando a API do Windows – SAPI (*Speech API*). Esta nova abordagem simplifica o sistema, pois não será necessária a utilização de software externo. Isso também resultará numa redução de custos, uma vez que a SAPI é distribuída gratuitamente;
- Aumentar o realismo do simulador de cadeira de rodas, para que os comportamentos da CRI no mundo real e simulado sejam ainda mais semelhantes;
- Realizar experiências que envolvam um número maior de voluntários em testes de condução e avaliação dos algoritmos, e que os participantes envolvidos não sejam somente indivíduos isentos de deficiências, mas também pessoas deficientes e que apresentem diferentes níveis de dificuldades;
- Desenvolver um controlo de decisão de entradas inteligente e sua integração com o módulo MMI. Tal controlo de decisão de entradas inteligentes deverá ser responsável por estabelecer níveis de confiança e gerir entradas de acordo com a sua percepção, buscando, deste modo, evitar conflitos, comandos com ruídos ou outra situação perigosa. Por exemplo, em situações de pouca ou muita luminosidade, onde o rosto do paciente não é suficientemente reconhecido, o controlo inteligente de entradas deveria diminuir o grau de confiança dos comandos enviados pelo software de reconhecimento de expressões faciais e dar maior importância aos comandos do joystick, voz e/ou movimentos da cabeça. Em certos limites, este sistema de controlo inteligente poderá até desabilitar a entrada que apresente um grau de confiança muito baixo.
- Propor melhorias ao módulo de localização, relacionado com novas tecnologias para reduzir a incerteza sobre a posição e orientação da cadeira de rodas. Novas abordagens devem incluir *map matching*, odometria visual e sistemas de localização global de maneira a reduzir os erros de localização sistemáticos e não sistemáticos.

- Desenvolver uma Interface Gráfica de Utilizador intuitiva e amigável voltada para utilizadores portadores de deficiências com limitações severas ou para pessoas com idades avançadas que já possuem dificuldades motoras. Em conjunto poderiam também ser desenvolvidos um conversor de textos em voz e um assistente virtual para melhorar o processo de integração com o utilizador.
- Avaliar e adaptar todo o sistema levando em consideração os aspectos e normas para sistemas críticos.
- Propor melhorias para aumentar a robustez do reconhecimento de expressões faciais.

No âmbito comercial este projecto também pode ser melhorado. Para já é possível levantar algumas acções neste sentido:

- Recriar o dispositivo de comando através de movimentos da cabeça de forma independente do computador e integrá-lo no *firmware* da cadeira. Deste modo será possível desenvolver um equipamento comercial de custo mais baixo que agregue as cadeiras de rodas motorizadas comuns essa habilidade juntamente com o desvio de obstáculos.
- Generalizar e transformar todos os módulos, funções e bibliotecas desenvolvidos neste trabalho em componentes Delphi e/ou Lazarus, com o objectivo de criar uma plataforma comercial em linguagem Pascal para desenvolvimento de robôs reais e sistemas multi-agente dependentes de hardware simples. A utilização do ambiente de desenvolvimento Lazarus apresenta alguns pontos positivos. Para além de ser gratuito, este ambiente é suportado por uma larga comunidade e, ao mesmo tempo, suporta diversas arquitecturas e sistemas operativos. A portabilidade para outras linguagens de programação não é discutida aqui. No entanto, esta possibilidade certamente permitiria expandir a aplicação e o uso da plataforma implementada.

O desenvolvimento dos trabalhos futuros sugeridos conduziria à criação de uma CRI inovadora com elevadas capacidades no auxílio a pessoas com graves deficiências e com um elevado potencial para ser transformada num produto comercial.



# Bibliografia

- Adachi, Y., Kuno, Y., Shimada, N. & Shirai, Y., 1998. Intelligent wheelchair using visual information on human faces. In *International Conference in Intelligent Robots and Systems.*, 1998.
- Alvares, L.O. & Sichman, J., 1997. Introdução aos Sistemas Multiagentes. *XVI Jornada de Atualização em Informática - JAI'97*, pp.1-38.
- Andrews, J.R. & Hogan, N., 1983. Impedance control as a framework for implementing obstacle avoidance in a manipulator. In *ASME Winter Conference.*, 1983.
- Bacchus, F., 2000. Subset of PDDL for the AIPS2000 planning competition. *The AIPS-00 planning competition comitee.*
- Bacchus, F. & Ady, M., 2001. Planning with Resources and Concurrency: A Forward Chaining Approach. *International Joint Conference on Artificial Intelligence (IJCAI-2001)*, pp.417-24.
- Barr, A., Cohen, P. & Feigenbaum, E., 1986. *Handbook of Artificial Intelligence*. Addison Wesley.
- Barr, A. & Feigenbaum, E.A., 1981. *The Handbook of Artificial Intelligence*. Los Altos, California: William Kaufmann Inc.
- Bauer, B., Müller, J.P. & Odell, J., 2001. Agent UML: A Formalism for Specifying Multiagent Software Systems. *International Journal of Software Engineering and Knowledge Engineering*, June. pp.207-30.
- Bayer, P. & Svantesson, M., 2001. Comparison of Agent-Oriented Methodologies: Analysis and Design - MAS-CommonKADS versus Gaia. In *Student Workshop on Agent Programming.*, 2001. Blekinge Institute of Technology.
- Bell, D.A. et al., 1994. An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance. *IEEE Conf. on Robotics and Automation.*
- Bellifemine, F.L., Caire, G. & Greenwood, D., 2007. *Developing multi-agent systems with JADE*. 1st ed. Chichester, England: Wiley.
- Bergasa, L.M. et al., 1999. Guidance of a Wheelchair for Handicapped People by Face Tracking. *7th IEEE International Conference on Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99*, pp.105 - 111.
- Bergenti, F., Gleizes, M.-P. & F.Zambonelli, 2004. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*. 1st ed.

- Kluwer Academic Publishers. Multiagent Systems, Artificial Societies, and Simulated Organizations: International Book Series.
- Bicho, E., Mallet, P. & Schoner, G., 1998. Using Attractor Dynamics to Control Autonomous Vehicle Motion. In *24th Annual Conference of the IEEE Industrial Electronics Society.*, 1998.
- Bittencourt, G., 1998. Inteligência artificial distribuída. *I Workshop de Computação do ITA, Instituto Tecnológico de Aeronáutica*, 06 Oct.
- Blum, A. & Furst, M., 1995. Fast planning through planning graph analysis. *IJCAI*, pp.1636-42.
- Bolt, R.A., 1980. Put-that-there: voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*. Seattle, Washington, USA, 1980.
- Bonet, B. & Geffner, H., 1999. HSP: heuristic search planner. *Universidad Simon Bolivar*.
- Borenstein, J., Everett, H.R. & Feng, L., 1996. *Where am I? Sensors and Methods for Autonomous Mobile Robots Positioning*. MI: University of Michigan.
- Borenstein, J., Everett, H.R. & Feng, L., 1996. *Where am I? Sensors and methods for mobile robot positioning*. Univ. Michigan.
- Borenstein, J. & Koren, Y., n.d. The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots. *IEEE Journal of Robotics and Automation* , pp.278-88.
- Borgerding, B. et al., 1999. FRIEND: Functional robot arm with user friendly interface for disabled people. *5th European Conf. for the Advancement of Assistive Technology*.
- Borgolte, U. et al., 1998. Architectural Concepts of a Semi-autonomous Wheelchair. *Journal of Intelligent and Robotic Systems*, 22, pp.233-53.
- Bourhis, G. & Agostini, Y., 1998. The Vahm Robotized Wheelchair: System Architecture and Human-Machine Interaction. *Journal of Intelligent and Robotic Systems*, 22, pp.39-50.
- Braga, R.A.M., Malheiro, P. & Reis, L.P., 2009. Development of a realistic simulator for robotic intelligent wheelchairs in a hospital environment. In *Proceedings of the RoboCup 2009 Symposium*. Graz, Austria, 2009. june-july.
- Braga, R.A.M., Petry, M.R., Moreira, A.P. & Reis, L.P., 2008. Platform for intelligent wheelchairs using multi-level control and probabilistic motion model. In *8th Portuguese Conference on Automatic Control, Controlo 2008*. Vila Real, Portugal, July 2008.
- Braga, R.A.M., Petry, M.R., Moreira, A.P. & Reis, L.P., 2009. Concept and design of the Intellwheels development platform for intelligent wheelchairs. *Informatics in Control, Automation and Robotics, Lecture Notes in Electrical Engineering*, April. pp.191-203.

- Braga, R.A.M., Petry, M.R., Reis, L.P. & Moreira, A.P., 2008. IntellWheels: A Development PlatForm for intelligent wheelchairs for disabled people. In *5th International Conference on Informatics in Control, Automation and Robotics*. Funchal, Madeira - Portugal, 2008. ICINCO.
- Braga, R.A.M., Petry, M.R., Reis, L.P. & Oliveira, E.C., 2008. Multi-Level Control of an Intelligent Wheelchair in a Hospital Environment using a Cyber-Mouse Simulation System. In *5th International Conference on Informatics in Control, Automation and Robotics*. Funchal, Madeira, Portugal, 2008. ICINCO.
- Bresciani, P. et al., 2004. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3), pp.203-36.
- Brooks, R., 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Mar. pp.14-23.
- Brooks, R.A., 1991. Intelligence Without Reason. *Proceedings of the 12th International Joint Conference on Artificial Intelligence- IJCAI-91*.
- Buhler, C., Hoelpe, R., Hoyer, H. & Humann, W., 1995. Autonomous robot technology for advanced wheelchair and robotic aids for people with disabilities. *Robotics and Autonomous Systems*, pp.213-22.
- Busso, C. et al., 2004. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th International Conference on Multimodal Interfaces*. State College, PA, USA, 2004.
- Castro, A. & Oliveira, E., 2008. The Rationale Behind the Development of an Airline Operations Control Centre using Gaia-based Methodology. *International Journal of Agent-Oriented Software Engineering*, July. pp.350-77. Inderscience Publishers, Geneva, SWITZERLAND.
- CENELEC EN 50159-2, 2001. ISBN: 0 580 37583 8 *Railway Applications - Communication, Signalling and Processing Systems - Part 2: Safety Related Communication in Open Transmission Systems*. Standard. European Committee for Electrotechnical Standardization.
- Cernuzzi, L., Juan, T., Sterling, L. & Zambonelli, F., 2004. The Gaia Methodology - Basic Concepts and Extensions. In F.B.a.M.-P.G.a.F. Zambonelli, ed. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*. Kluwer Academic Publishing (New York). pp.69-88.
- Cernuzzi, L. & Zambonelli, F., 2004. Experiencing AUML in the Gaia Methodology. In *Proceedings of the Sixth International Conference on Enterprise Information Systems (ICEIS'04)*, 2004. Kluwer Academic Publisher.

- Chai, J.Y., Hong, P. & Zhou, M.X., 2004. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*. Funchal, Madeira, Portugal, 2004.
- Charniak, E. & McDermott, D., 1985. *Introduction to Artificial Intelligence*. Addison-Wesley Publishing Company.
- Corder, G.W. & Foreman, D.I., 2009. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. Hoboken, N.J.: Wiley.
- Costa, E. & Simões, A., 2008. *Inteligência Artificial - Fundamentos e Aplicações*. 2nd ed. Lisboa, Portugal: FCA - Editora de Informática.
- Crisman, J.D.a.C.M.E., 1998. Progress on the Deictically Controlled Wheelchair. In *Assistive Technology and Artificial Intelligence*. Heidelberg, Berlin : Springer. p.137.
- Cunha, F.M., Braga, R.A.M. & Reis, L.P., 2010. A Cooperative Communications Platform for Safety Critical Robotics: An Experimental Evaluation. In *8th International Conference on Practical Applications of Agents and Multiagent Systems, PAAMS 2010*. Salamanca, Spain, 2010.
- Cunha, F.M., Braga, R.A.M. & Reis, L.P., 2010. Evaluation of a Communication Platform for Safety Critical Robotics. In *10th International Conference in Artificial Intelligence and Soft Computing, ICAISC 2010*. Zakopane, Poland, 2010.
- Dam, K.H. & Winikoff, M., 2003. Comparing Agent-Oriented Methodologies. In *Proceedings of the Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2003)*, July., 2003. Springer.
- de Berg, M., Cheong, O., van Kreveld, M. & Overmars, M., 2008. *Computational Geometry: Algorithms and Applications*. 3rd ed. Springer-Verlag.
- DeLoach, S.A. & Wood, M.F., 2001. Developing Multiagent Systems with agent-Tool. In *Proceedings of the 7th International Workshop on Intelligent Agents VII. Agent Theories Architectures and Languages (ATAL'00)*. Boston, MA, 2001.
- DeLoach, S.A., Wood, M.F. & Sparkman, C.H., 2001. Multiagent Systems Engineering. In *International Journal on Software Engineering and Knowledge Engineering.*, 2001.
- Diestel, R., 2010. *Graph Theory*. 4th ed. Heidelberg: Springer-Verlag.
- Doherty, P. & Kvarnstrom, J., 2001. TALPlanner: A Temporal Logic Based Planner. *AI Magazine*.
- Durfee, E.H., Lesser, V.R. & Corkill, D.D., 1989. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, March. pp.63-83.
- Edelkamp, S. & Helmert, M., 2001. The Model Checking Integrated Planning System. *AI-Magazine*, pp.67-71.

- Edelkamp, S. & Hoffmann, J., 2003. PDDL 2.2: The Language for the Classical Part. *4th International Planning Competition Technical Report*.
- Ernst, M.D., Millstein, T.D. & Weld, D.S., 1998. Automatic SAT-Compilation of Planning Problems. *IJCAI-97*, august. pp.1169-76.
- Faria, P.M., Braga, R.A.M., Valgôde, E. & Reis, L.P., 2007. Interface framework to drive an intelligent wheelchair using facial expressions. In *IEEE International Symposium on Industrial Electronics*. Vigo, Spain, 4-7 June 2007. IEEE.
- Faria, P.M., Braga, R.A.M., Valgôde, E. & Reis, L.P., 2007. Interface Framework to Drive an Intelligent Wheelchair Using Facial Expressions. *IEEE International Symposium on Industrial Electronics*, 4-7 June. pp.1791-96.
- Faria, P.M., Braga, R.A., Valgôde, E. & Reis, L.P., 2007. Platform to drive an intelligent wheelchair using facial expressions. In *Proceedings of the 9th International Conference on Enterprise Information Systems - Human-Computer Interaction (ICEIS 2007)*. Funchal, Madeira, Portugal, 2007.
- Fehr, L., Langbein, W.E. & Skaar, S.B., 2000. *Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey*. J Rehabil Res Devel.
- Ferber, J. & Gasser, L., 1991. Intelligence artificielle distribuée. *International WorkShop on Expert Systems & Their Applications*.
- Fikes, R.E. & Nilsson, N.J., 1971. STRIPS: A new approach to the application of the theorem proving to problem solving. *Artificial Intelligence 2*, pp.27-120.
- Fink, E. & Veloso, M., 1994. CMU-CS-94-123 *Prodigy planning algorithm*. Technical Report. Pittsburgh, PA: Carnegie Mellon University Carnegie Mellon University.
- FIPA, 2009. *FIPA, Foundation for Intelligent Physical Agents*. [Online] Available at: <http://www.fipa.org> [Accessed October 2009].
- Fowler, K., 2004. Missio-critical and safety-critical developent. *IEEE Instrumentation & Measurement Magazine*, 7(4), pp.52-59.
- Fox, M. & Long, D., 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*.
- Franklin, S. & Graesser, A., 1996. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*., 1996. Springer-Verlag.
- Galindo, C. et al., 2006. A Multi-Agent Control Architecture for a Robotic Wheelchair. *Applied Bionics and Biomechanics*, January. pp.179-89.

- Galindo, C., Gonzalez, J. & Fernandez-Mandrigal, J.A., 2006. A Control Architecture for Human-Robot Integration: Application to a Robotic Wheelchair. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(1), pp.1053-67.
- García-Ojeda, J.C., Arenas, A.E. & Pérez-Alcázar, J.J., 2005. Paving the Way for Implementing Multiagent Systems: Integrating Gaia with Agent-UML. In Müller, J.P. & Zambonelli, F., eds. *Proceedings of the 6th International Workshop on Agent-Oriented Software Engineering (AOSE 2005), 25-26 July*. Utrecht, The Netherlands, 2005. Springer-Verlag.
- Garro, A. & Turci, P., 2003. *Meta-Model Sources: Gaia*. Technical Report. Foundation for Intelligent Physical Agents, August.
- Gerevini, A., Saetti, A., Serina, I. & Toninelli, P., 2004. LPG-TD: a Fully Automated Planner for PDDL2.2 Domains. *American Association for Artificial Intelligence*.
- Gerevini, A. & Serina, I., 2002. LPG: a Planner based on Planning Graphs with Action Costs. *Proceedings of the Sixth Int. Conference on AI Planning and Scheduling (AIPS'02)*, pp.13-22.
- Ghallab, M., Nau, D. & Traverso, P., 2004. *Automated Planning: Theory & Practice*. São Francisco, EUA: Morgan Kaufmann, Elsevier.
- Giorgini, P., Kolp, M., Mylopoulos, J. & Pistore, M., 2003. The Tropos Methodology: An Overview. In *Methodologies and Software Engineering for Agent Systems*, 2003. Kluwer Academic Press.
- Giunchiglia, F., Mylopoulos, J. & Perini, A., 2002. The Tropos Software Development Methodology: Processes, Models and Diagrams. In *Third International Workshop on Agent-Oriented Software Engineering (AOSE 2002) Revised Papers and Invited Contributions*, 2002. Springer.
- Glover, F., 1989. Tabu Search:1. *ORSA Journal on computing*, 1(3), pp.190-206.
- Gomi, T. & Griffith, A., 1998. Developing Intelligent Wheelchairs for the Handicapped. In V. Mittal, H. Yanco, J. Aronis & R. Simpson, eds. *Assistive Technology and Artificial Intelligence*. Berlin: Springer. pp.150-78.
- Gonzalez, J. et al., 2006. A Description of the SENA Robotic Wheelchair. In *IEEE MELECON*. Benalmádena (Málaga), Spain, 2006.
- Gonzalez-Palacios, J. & Luck, M., 2008. Extending Gaia with Agent Design and Iterative Development. In *8th International Workshop on Agent-Oriented Software Engineering (AOSE 2007), May 14*. Honolulu, HI, USA, 2008.
- Group, O.M., 2005. formal/2005-01-06 *Software Process Engineering Metamodel Specification*. Specification. Object Management Group, January. SPEM1.1.

- Hamagami, T. & Hirata, H., 2004. Development of intelligent wheelchair acquiring autonomous, cooperative, and collaborative behavior. *IEEE International Conference on Systems, Man and Cybernetics*, 10-13 Oct. pp.3525 - 3530.
- Hamagami, T. & Hirata, H., 2005. State Space Partitioning and Clustering with Sensor Alignment for Autonomous Robots. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics 2005.*, 2005.
- Harary, F., 1969. *Graph Theory*. MA: Addison-Wesley.
- Helmert, M., 2002. Complexity results for standard benchmark domains in planning. *Journal of Artificial Intelligence Research - JAIR*.
- Hoyer, H., Borgolte, U. & Hoelper, R., 1997. AN OMNIDIRECTIONAL WHEELCHAIR WITH ENHANCED COMFORT FEATURES. In *5th International Conference on Rehabilitation Robotics - ICORR'97*. Bath, England, 1997.
- Hoyer, H. & Hölpel, R., 1993. Open control architecture for an intelligent omnidirectional wheelchair. in *Proc.1st TIDE Congress, Brussels*, pp.93-97.
- Huhns, M. & Stephens, L.M., 1999. *Multiagent Systems and Societies of Agents*. Cambridge, MA, USA: The MIT Press.
- IBGE, 2000. *IBGE - Censo Demográfico 2000*. [Online] IBGE Available at: [URL: http://www.sidra.ibge.gov.br](http://www.sidra.ibge.gov.br) [Accessed 1 Abril 2009].
- IBM, 2009. *IBM - Embedded ViaVoice - Software*.
- IEC 60812, n.d. Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA). *Brussels*, European Committee for Electrotechnical Standardization.
- Iezzoni, L., McCarthy, E., Davis, R. & Siebens, H., 2001. Mobility difficulties are not only a problem of old age. *Journal of General Internal Medicine*, April. p.235-243.
- Iglesias, C.A., Garijo, M. & Gonzalez, J.C., 1998. A Survey of Agent-Oriented Methodologies. In al., J.P.M.e., ed. *Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL'98)*. Paris, France, 1998.
- Iglesias, C.A., Garijo, M. & Gonzalez, J.C., 1999. A Survey of Agent-Oriented Methodologies. In Muller, J.P. & al., e., eds. *Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL'98), July 1998*. Paris, France, 1999. Springer-Verlag.
- Iturrate, I., Antelis, J., Minguez, J. & Kübler, A., 2009. A Non-Invasive Brain-Actuated Wheelchair based on a P300 Neurophysiological Protocol and Automated Navigation. *IEEE Transactions on Robotics*, 25(3), pp.614-27.

- Jaimes, A. & Sebe, N., 2007. Multimodal Human Computer Interaction: A Survey. *Computer Vision and Image Understanding*, October. pp.116-34.
- Jennings, N.R., 2000. On Agent-Base Software Engineering. *Artificial Intelligence*, 117, pp.277-96.
- Jia, P., Hu, H., Lu, T. & Yuan, K., 2007. Head Gesture Recognition for Hands-free Control of an Intelligent Wheelchair. *Journal of Industrial Robot*, pp.60-68.
- Johnston, M. & Bangalore, S., 2004. MATCHKiosk: a multimodal interactive city guide. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*. Barcelona, Spain, 2004. Article no.: 33.
- Juan, T., Pearce, A. & Sterling, L., 2002. ROADMAP: Extending the Gaia Methodology for Complex Open Systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '02)*. Bologna, Italy, July 2002. ACM.
- Ju, J.S., Shin, Y. & Kim, E.Y., 2009. Intelligent wheelchair (IW) interface using face and mouth recognition. In *Proceedings of the 13th international conference on Intelligent user interfaces*. Sanibel Island, Florida, USA, 2009.  
<http://doi.acm.org/10.1145/1502650.1502693>.
- Kapoor, A., Burleson, W. & Picard, R.W., 2007. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8), pp.725-36.
- Katevas, N.L. et al., 1997. The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs. *IEEE Robotics & Automation Magazine*, pp.60-70.
- Kautz, H., McAllester, D. & Selman, B., 1996. Encoding Plans in Propositional Logic. *International Conference on Principles of Knowledge Representation and reasoning*.
- Kautz, H. & Selman, B., 1992. Planning as satisfiability. *Proceedings of ECAI92*, August.
- Kautz, H. & Selman, B., 1998. Blackbox: A new approach to the application of theorem proving to problem solving. *American Association for Artificial Intelligence - AAAI*, June. pp.58-60.
- Kautz, H. & Selman, B., 1999. Unifying SAT-based and graph-based planning. In *Sixteenth International Joint Conference on Artificial Intelligence*. San Francisco, 1999. Morgan Kaufmann Publishers Inc.
- Khatib, O., 1986. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5, pp.90-98.
- Khosla, R. & Dillon, T., 1997. *Engineering Intelligent Hibrid Multi-Agent*. Norwell, MA, USA: Kluwer Academic Publisher.

- Kuno, Y., Shimada, N. & Shirai, Y., 2003. Look where you're going: a robotic wheelchair based on the integration of human and environmental observations. *IEEE Robotics and Automation*, 10(1), pp.26-34.
- Kuo, C.H., Huang, H.L. & Lee, M.Y., 2003. Development of Agent-Based Autonomous Robotic Wheelchair Control Systems. *Journal of Biomedical Engineering – Applications, Basis and Communication*, 15, pp.12-23.
- Lakany, H., 2005. Steering a wheelchair by thought. In *The IEE International Workshop on Intelligent Environments.*, 2005.
- Lankenau, A., Meyer, O. & Krieg-Brückner, B., 1998. Safety in Robotics: the Bremen Autonomous Wheelchair. In *In Proceedings of AMC'98, 5th Int. Workshop on Advanced Motion Control.*, 1998.
- Lau, N. et al., 2002. Ciber-Rato: Um Ambiente de Simulação de Robots Móveis e Autónomos. *Revista do DETUA*, 3(7).
- Lau, N. et al., 2002. Ciber-Rato: uma competição robótica num ambiente virtual. *DETUA*, 3(7), pp.647-50.
- Lau, N. et al., 2002. O visualizador do ambiente de simulação Ciber-Rato. *DETUA*, 3(7), pp.651-54.
- Leonard, J.J. & Durrant-Whyte, H.F., 1991. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings IEEE/RSJ International Workshop on Intelligent Robots and Systems-IROS '91*. Osaka , Japan, 1991.
- Leonard, J.J. & Durrant-Whyte, H.F., 1992. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers.
- Levine, S.P. et al., 1997. The NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, Dec. pp.443-51.
- Lidoris, G. & Buss, M., 2006. A Multi-Agent System Architecture for Modular Robotic Mobility Aids. *European Robotics Symposium*, pp.15-26.
- Lifschitz, V., 2000. Missionaries and Cannibals in the Causal Calculator. *Principles of Knowledge Representation and Reasoning*, pp.85-96.
- Maçada, D.L. & Tijiboy, A.V., 1998. Aprendizagem Cooperativa em Ambientes Telemáticos. In *IV Congresso Ibero-americano de Informática na Educação - RIBIE 98*. Brasília, Brasil, 1998.
- Madarasz, R.L., Heiny, L.C., Crompt, R.F. & Mazur, N.M., 1986. The design of an autonomous vehicle for the disabled. *IEEE Journal of Robotics and Automation*, September. pp.117 - 126.

- Maes, P., 1996. Intelligent Software: Programs That Can Act Independently Will Ease the Burdens that Computers Put on People. *IEEE Expert Systems*, 11(6), pp.62-63.
- Malm, T., Hérard, J., Boegh, J. & Kivipuro, M., 2007. *Validation of safety-related wireless machine control systems*. NT Tech Report. Norway: Nordic Innovation Centre.
- Martens, C. et al., 2001. A FRIEND for assisting handicapped people. *IEEE Robotics & Automation Magazine*, 8, pp.57-65.
- Mazo, M., 2001. An integral system for assisted mobility. *IEEE Robotics & Automation Magazine*, 8(1), pp.46 - 56.
- McAllester, D. & Rosenblitt, D., 1991. Systematic Nonlinear Planning. *American Association for Artificial Intelligence - AAAI*.
- McCain, N., Remolina, E., Tacchella, A. & Lifschitz, V., 2000. Getting to the airport: the oldest planning problem in AI. *Kluwer International Series In Engineering And Computer Science*, pp.147-65.
- McCarthy, J., 1963. Situations, actions and causal laws. *Technical report*, Jul. pp.410-70. Reprinted in *Semantic Information Processing*, MIT Press, Cambridge, Mass., 1968.
- McCarthy, J. & Hayes, P.J., 1969. *Some philosophical problems from the standpoint of artificial intelligence*. Edinburgh, GB: Edinburgh University Press.
- McCowan, I. et al., 2005. Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), pp.305-17.
- McDermott, D., 1998. PDDL - the planning domain definition language. *Technical report*.
- McDermott, D., 2000. The 1998 AI Planning Systems Competition. *Yale University*, apr.
- Meyer, S. & Rakotonirainy, A., 2003. A Survey of research on context-aware homes. In *Australasian Information Security Workshop Conference on ACSW Frontiers*. Adelaide, Australia, 2003.
- Miller, D., 1998. Assistive Robotics: An Overview. *Assistive Technology and AI*, pp.126-36.
- Miller, D. & Slack, M., 1995. Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, pp.77-88.
- Minton, S., 1988. *Learning effective search control knowledge: an explanation approach*. Carnegie Mellon University.
- Minton, S. et al., 1989. Prodigy 2.0: the manual and tutorial. *Tech. Rep. CMU-CS-89-146*.
- Miyake, N., n.d. Constructive interaction and the iterative process of understanding. *Cognitive Science*, pp.151-77.
- Moraitis, P. & Spanoudakis, N.I., 2006. The Gaia2Jade process for multi-agent systems development. *Applied Artificial Intelligence*, 20(4), pp.251-73.

- Moreira, A.P., Costa, P., Sousa, A. & Reis, L.P., 2002. 5dpo Team Description – Middle Size League. In *Proceedings of International RoboCup Symposium*. Fukuoka, Japan, 2002.
- Nau, D.S., Cao, Y., Lotem, A. & Muhnoz-Avila, H., 1999. H. SHOP: Simple Hierarchical Ordered Planner. *Tech. Rep., UMIACS-TR-9904, University of Maryland (IJCAI-99)*, pp.986-75.
- Ng, P.C. & De Silva, L.C., 2001. Head gestures recognition. *Proceedings International Conference on Image Processing*, Oct. pp.266-69.
- Nielsen, J., 1994. *Usability Engineering*. San Francisco, USA: Academic Press.
- Niniss, H. & Nadif, A., 2000. Simulation of the Behavior of a Powered Wheelchair using Virtual Reality. In *Proceedings of Third International Conference on Disability, Virtual Reality and Associated Technologies*. Alghero, Italy, 2000.
- Nisbet, P.D., 2002. Who's intelligent? Wheelchair, driver or both? In *Proc. IEEE International Conference on Control Applications*. Glasgow, Scotlend, UK, 2002.
- Odell, J., Parunak, H.V.D. & Bauer, B., 2001. Representing Agent Interaction Protocols in UML. In Ciancarini, P. & Wooldridge, M., eds. *First International Workshop on Agent-Oriented Software Engineering (AOSE 2000), June 10*. Limerick, Ireland, 2001. Springer-Verlag New York, Inc.
- Odor, J.P., 1995. ISBN 1 898042 06 3 *The CALL Centre Smart Wheelchair*. Supplementary Booklet. Edinburgh: University of Edinburgh.
- Oviatt, S., 2000. Multimodal system processing in mobile environments. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. San Diego, California, United States, 2000. ACM.
- Padgham, L., Thangarajah, J. & Winikoff, M., 2007. The Prometheus Design Tool - A Conference Management System Case Study. In *Agent-Oriented Software Engineering VIII, 8th International Workshop (AOSE 2007), May 14*. Honolulu, HI, USA, 2007.
- Padgham, L. & Winikoff, M., 2002. Prometheus: a Methodology for Developing Intelligent Agents. In Wei, F.G.a.J.O.a.G., ed. *Third International Workshop on Agent-Oriented Software Engineering (AOSE 2002) Revised Papers and Invited Contributions.*, 2002. Springer.
- Padgham, L. & Winikoff, M., 2004. *Developing Intelligent Agent Systems: a Practical Guide*. 1st ed. John Wiley and Sons. Wiley Series in Agent Technology.
- Pardue, J., 2010. *An Arduino Workshop*. 1st ed. Knoxville, EUA: Smiley Micros.
- Parikh, S.P., Jr., V.G., Kumar, V. & Jr, J.O., 2004. Incorporating User Inputs in Motion Planning for a Smart Wheelchair. In *IEEE International Conference on Robotics & Automation*. New Orleans, LA, 2004.

- Parikh, S.P., Jr., V.G., Kumar, V. & Jr, J.O., 2005. Usability Study of a Control Framework for an Intelligent Wheelchair. In *IEEE International Conference on Robotics and Automation*. Barcelona, Spain, 2005.
- Parikh, S.P., Jr., V.G., Kumar, V. & Jr, J.O., 2007. Integrating Human Inputs with Autonomous Behaviors on an Intelligent Wheelchair Platform. *IEEE Intelligent Systems and Their Applications*, pp.33 - 41.
- Pei Chi, N. & de Silva, L.C., 2001. Head gestures recognition. In *Proceedings International Conference on Image Processing.*, 2001.
- Penberthy, J.S. & Weld, D.S., 1992. Ucpop: A sound, complete, partial order planner for ADL. *Conference on Knowledge representation and reasoning*.
- Perzanowski, D. et al., 2001. Building a multimodal human-robot interface. In *Intelligent Systems, IEEE.*, 2001.
- Petry, M.R., Moreira, A.P., Braga, R.A.M. & Reis, L.P., 2010. Shared control for obstacle avoidance in intelligent wheelchairs. In *IEEE Conference on Robotics, Automation and Mechatronics, RAM 2010*. Singapore, 2010.
- Pires, G. & Nunes, U., 2002. Controller, A Wheelchair Steered through Voice Commands and Assisted by a Reactive Fuzzy-Logic. *Journal of Intelligent and Robotic Systems* , pp.301 - 314.
- Pires, G., Nunes, U. & Almeida, A.T.d., 1998. ROBCHAIR – A Semi-Autonomous Wheelchair for Disabled People. In *Proc. 3rd IFAC Symposium on Intelligent Autonomous Vehicles (IAV'98)*. Madrid, 1998.
- Pope, A. & Tarlov, A., 1991. *Disability in America: Toward a national agenda for prevention*. Washington (DC): National Academies Press.
- Prassler, E., Scholz, J. & Fiorini, P., 2001. A robotic wheelchair for crowded public environments. *IEEE Robotics Autom Mag.*, pp.38-45.
- Prenzel, O., Feuser, J. & Graser, A., 2005. Rehabilitation robot in intelligent home environment - software architecture and implementation of a distributed system. In *ICORR 2005. 9th International Conference on Rehabilitation Robotics, 2005.*. Chicago, Illinois, USA, 2005.
- Pruski, A., Ennaji, M. & Morere, Y., 2002. VAHM: a user adapted intelligent wheelchair. In *Proceedings of the 2002 International Conference on Control Applications, 2002*. Glasgow, Scotland, 2002.
- R.Murphy, R., 2000. *Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)*. Massachusetts, EUA: MIT Press.
- Rabiner, L.R., 1993. *Fundamentals of Speech Recognition, Prentice Hall*. Englewood Cliffs.

- Rebsamen, B. et al., 2007. Controlling a Wheelchair Indoors Using Thought. *IEEE Intelligent Systems and Their Applications*, XXII, pp.18-24.
- Reis, L.P., 2003. 621.3(043)/REII/COO (PISO5) *Coordenação em sistemas multi-Agente : Aplicações na gestão universitária e futebol robótico*. Tese de doutoramento. Porto: Engenharia Electrotécnica e de Computadores. Faculdade de Engenharia. Universidade do Porto.
- Reis, L.P., 2003. *Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico*. Tese de Doutoramento. Porto, Portugal,: Faculdade de Engenharia da Universidade do Porto.
- Reis, L.P., Braga, R.A.M., Sousa,. & Moreira, A.P., 2009. IntellWheels MMI: A Flexible Interface for an Intelligent Wheelchair. In *RoboCup 2009: 13th annual RoboCup International Symposium*. Graz, Austria, 2009. Springer.
- Reis, L.P. & Lau, N., 2001. FC Portugal Team Description: RoboCup 2000 Simulation. *RoboCup-2000: Robot Soccer World Cup IV*, pp.29-40.
- Reis, L.P. & Lau, N., 2002. COACH UNILANG – A Standard Language for Coaching a (Robo) Soccer Team. *RoboCup-2001: Robot Soccer World Cup V*, pp.183-92.
- Rezende, S.O., 2005. *Sistemas inteligentes: fundamentos e aplicações*. SP-Brasil: Manole.
- Röfer, T. & Lankenau, A., 2000. Architecture and Applications of the Bremen Autonomous Wheelchair. *Information Sciences—Informatics and Computer Science: An International Journal*, 126, pp.1-20.
- Rohrer, M.R., 2000. Seeing is believing: the importance of visualization in manufacturing simulation. In *Winter Simulation Conference*., 2000.
- Ronzhin, A. & Karpov, A., 2005. Assistive multimodal system based on speech recognition and head tracking. In *Proceedings of 13th European Signal Processing Conference*. Antalya, 2005.
- Russell, S. & Norvig, P., 2002. *Artificial Intelligence: A Modern Approach*. 2nd ed. Englewood Cliffs, NJ: Prentice Hall.
- S. Kambhampati, E. Parker & E. Lambrecht, 1997. Understanding and extending Graphplan. In *4th European Conference on Planning*., 1997. Steel and Alami, Eds.
- Schilling, K., Roth, H., R., L. & Stutzle, H., 1998. Sensors to improve the safety for wheelchair users. In *3rd Annual TIDE Congress*. Helsinki, Finland, 1998.
- Shapiro, S.C., 2000. *Encyclopedia of Artificial Intelligence*. Wiley-Interscience.
- Shneiderman, B. & Plaisant, C., 2004. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 4th ed. Boston: Pearson/Addison-Wesley.

- Siegwart, R. & Nourbakhsh, I.R., 2004. *Introduction to Autonomous Mobile Robot*. The MIT Press.
- Silva, D.C., Braga, R.A.M., Reis, L.P. & Oliveira, E., 2010. A generic model for a robotic agent system using GAIA methodology: Two distinct implementations. In *IEEE Conference on Robotics, Automation and Mechatronics, RAM 2010*. Singapore, 2010.
- Simpson, R.e.a., 1998. *NavChair: An Assistive Wheelchair Navigation System with Automatic Adaptation*. Springer-Verlag. Lecture Notes in Computer Science.
- Simpson, R.C., 2005. Smart wheelchairs: A literature review. *Journal of Rehabilitation Research & Development*, July/August. pp.423-36.
- Simpson, R.M. & McCluskey, T.L., 1999. An object-graph planning algorithm. In *Proceedings of the 18th Workshop of the UK Planning and Scheduling SIG.*, Sept 1999.
- Smith, S.J.J., Nau, D.S. & Throop, T., 1998. Computer Bridge: A Big Win for AI Planning. *AI Magazine 19*, June. pp.93-105.
- Smith, P., Shah, M. & Lobo, N.D.V., 2003. Determining driver visual attention with one camera. In *IEEE Transactions on Intelligent Transportation Systems.*, 2003.
- Smith, D.E. & Weld, D.S., 1998. Conformant Graphplan. *15th National Conference on AI*.
- Solea, R. & Nunes, U., 2008. Robotic wheelchair control considering user comfort: modeling and experimental evaluation. In *International Conference on Informatics in Control (ICINCO 2008)*. Funchal, 2008.
- Sterling, L. & Taveter, K., 2009. *The Art of Agent-Oriented Modeling, Intelligent Robotics and Autonomous Agents*. Cambridge, Massachusetts: The MIT Press.
- Stone, P. & Veloso, M., 1996. Multi-Agent Systems: A Survey from a Machine Learning Perspective. In *IEEE Transactions on Knowledge and Data Engineering (TKDE).*, 1996.
- Sturm, A. & Shehory, O., 2004. A Comparative Evaluation of Agent-Oriented Methodologies. In Bergenti, F. & Zambonelli, M.-P.G.a.F. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*. Kluwer Academic Publishing (New York). pp.127-49.
- Sussman, G.J., 1975. *A Computer Model of Skill Acquisition*. New York, NY, USA: Elsevier Science Inc.
- Te'eni, D., 2006. Designs that fit: an overview of fit conceptualization in HCI. In P. Zhang & D. Galleta, eds. *Human-Computer Interaction and Management Information Systems: Foundations*. USA: AMIS. Ch. 10. p.205.
- Te'eni, D., Carey, J. & Zhang, P., 2007. *Human Computer Interaction: Developing Effective Organizational Information Systems*. Hoboken, USA: John Wiley & Sons.

- Tefft, D., Guerette, P. & Furumasu, J., 1999. Cognitive predictors of young children's readiness for powered mobility. *Dev Med Child Neurol*, p.665–670.
- Thrun, S. & Bücken, A., 1996. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. In *AAAI/IAAI*. Portland, Oregon, 1996. Proceedings of 13th National Conference on Artificial Intelligence.
- Trefler, E. et al., 2004. Outcomes of wheelchair systems intervention with residents of long-term care facilities. *Assist Technol*, p.18–27.
- Weld, D.S., 1999. Recent advances in AI planning. *AI Magazine*, p.93–123. [citeseer.nj.nec.com/weld99recent.html](http://citeseer.nj.nec.com/weld99recent.html).
- Wellman, P., Krovi, V. & Kumar, V., 1994. An adaptive mobility system for the disabled. *Proc. IEEE Int. Conf. on Robotics and Automation*.
- Winikoff, M. & Padgham, L., 2004. The Prometheus Methodology. In F. Bergenti, M.-P.G.F.Z. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*. New York: Kluwer Academic Publishing. pp.217-34.
- Winston, P.H., 1984. *Artificial Intelligence*. 2nd ed. MA: Addison-Wesley Publishing Company Reading.
- Wood, M.F. & DeLoach, S.A., 2000. An overview of the multiagent systems engineering methodology. In *Proceedings of the First International Workshop on Agent-Oriented Software Engineering (AOSE 2000)*., 2000. Springer-Verlag.
- Wooldridge, M., 2001. *An Introduction to Multi Agents Systems. Intelligent agents (Computer software)*. John Wiley & Sons Ltd.
- Wooldridge, M.J. & Jennings, N.R., 1995. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, pp.115-52.
- Wooldridge, M., Jennings, N.R. & Kinny, D., 2000. The Gaia Methodology For Agent-Oriented Analysis And Design. *Journal of Autonomous Agents and Multi-Agent Systems*, September. pp.285-312.
- Woo, M., Neider, J. & Davis, T., 1999. *OpenGL programming guide : the official guide to learning OpenGL, Version 1.2*. 3rd ed. Addison-Wesley.
- Yanco, A.H., 1998. Wheelesley: a robotic wheelchair system: Indoor navigation and user interface. In V.O. Mittal, H.A. Yanco, J. Aronis & S. R., eds. *Lecture Notes in Artificial Intelligence: Assistive Technology and Artificial Intelligence*. pp.256-68.
- Yang, J. & Kim, J., 1999. Sliding Mode Motion Control of Nonholonomic Mobile Robots. *IEEE Control Systems Magazine*, pp.15-23.

Zambonelli, F., Jennings, N.R. & Wooldridge, M., 2003. Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering Methodologies*, July. pp.317-70.