



UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS

MARCKSON MONTEIRO DA SILVA

**CONFIGURAÇÃO DE CENÁRIOS EXPERIMENTAIS EM LABORATÓRIO À
DISTÂNCIA USANDO SISTEMA MULTIAGENTE**

Manaus

2013

MARCKSON MONTEIRO DA SILVA

**CONFIGURAÇÃO DE CENÁRIOS EXPERIMENTAIS EM LABORATÓRIO À
DISTÂNCIA USANDO SISTEMA MULTIAGENTE**

Dissertação apresentada ao Curso de Mestrado em
Informática da Universidade Federal do Amazonas,
como requisito parcial à obtenção do título de Mestre
em Informática.

Orientador: Prof. Dr. José Francisco de Magalhães Netto

Manaus

2013

MARCKSON MONTEIRO DA SILVA

**CONFIGURAÇÃO DE CENÁRIOS EXPERIMENTAIS EM LABORATÓRIO À
DISTÂNCIA USANDO SISTEMA MULTIAGENTES**

Esta dissertação foi julgada adequada à obtenção do título de Mestre em Informática e aprovada em sua forma final pelo Curso de Mestrado em Informática da Universidade Federal do Amazonas.

Manaus, ____ de _____ de 2013.

Prof. José Francisco de Magalhães Netto, Dr.
Orientador
Universidade Federal do Amazonas.

Prof. Davidson Cury, Dr.
Universidade Federal do Espírito Santo.

Prof. José Luiz de Souza Pio, Dr.
Universidade Federal do Amazonas.

Prof. Raimundo da Silva Barreto, Dr.
Universidade Federal do Amazonas.

Dedico este trabalho aos meus pais, irmãos e amigos por todo suporte necessário para realização desse projeto.

AGRADECIMENTOS

Agradeço inicialmente a DEUS, a quem devo tudo o que sou.

Queria agradecer aos meus pais que sempre me deram as condições ideais para que eu me dedicasse aos estudos, além de terem me dado o exemplo de caráter a ser seguido. Obrigado Valdéa Monteiro da silva e Lazaro Souza da Silva!

Obrigado a meus irmãos por todo o carinho e confiança em mim. São eles Valéria e Marcelo.

Ao meu orientador, Professor José Francisco de Magalhães Netto, Dr., pela paciência, pelas sugestões, por ter acreditado na realização desta pesquisa e confiado em meus ideais.

Aos professores, colegas e todos os integrantes do curso de pós-graduação, que direta ou indiretamente contribuíram para a conclusão desse trabalho.

Agradeço também à CAPES pela bolsa de estudos, que possibilitou uma dedicação integral a este curso e também ao CNPq, por apoiar o projeto LaboREAm possibilitou auxílio financeiro em atividades no decorrer do mestrado.

Obrigado a todos!

“Exalta a sabedoria, e ela te exaltará, e abraçando-a ti, ela te honrará”.

Provérbios 4:8.

RESUMO

Laboratório de experimentação Hands-On é limitado temporalmente e geograficamente. O usuário tem que se deslocar até ao ambiente físico em seu horário de funcionamento para utilizá-lo, em seguida esperar que seja montado o experimento por um técnico e somente depois disso realizar os procedimentos experimentais. Na busca por amenizar essas limitações surgiram os Laboratórios de Experimento Remoto, também chamado de Laboratório de Acesso Remoto (ou LAR). Uma característica em Laboratórios de Acesso Remoto é a configuração de cenários experimentais encontra-se pronto para ser utilizado, tirando a oportunidade do usuário de escolher um domínio experimental a ser executado. Em geral, são limitados a executarem sempre os mesmos experimentos, se tornando um problema. Esta característica, primeiramente, foi identificada no trabalho de Mestrado de um integrante do Laboratório de Robótica Educativa do Amazonas (ou LaboREAM) com o projeto Laboratório de Acesso Remoto de Robótica (ou LARRE). Então, neste projeto de Mestrado realizou-se inicialmente o levantamento bibliográfico sobre o estado da arte dos Laboratórios de Acesso Remoto e foi identificada a mesma característica. O problema abordado nesse trabalho é como realizar a configuração e montagem autônoma de ambientes experimentais de ciências? A estratégia adotada a fim de solucionar o problema foi a combinação de um Sistema Multiagente (ou SMA) com o XML para assim definir os cenários experimentais a serem montados. Para montar o cenário experimental de forma autônoma, o Sistema Multiagente é usado em conjunto com um Sistema Robótico Móvel. E na modelagem do sistema foi usada a metodologia O-MaSE. Foi desenvolvida uma Rede de Petri (ou RdP) que representa as interações entre os agentes robóticos móveis montando o cenário e simulando a correta ordem de sincronismo ou paralelismo das tarefas. A partir dessa concepção, foi implementado o Sistema Multiagente usando framework JADE e firmware LeJOS. Os resultados obtidos com os testes qualitativos e quantitativos aplicados no sistema comprovam a validade e a viabilidade da solução. Em testes foram verificados a qualidade em montar os cenários, ou seja, a precisão em posicionar os objetos e também quantos cenários seguidos conseguem ser montados.

Palavras-chave: Tele-experimentação, experimentação remota, laboratório de experimentação, WEB laboratório, Laboratório de Acesso Remoto, laboratório de ciências, configuração autônoma.

ABSTRACT

Hands-on laboratory is limited temporally and geographically. To use the laboratory the user has to scroll to the laboratory environment for their opening hours, then wait for the experiment to be mounted by a laboratory technician and only then perform experimental procedures. In seeking to minimize these limitations arose laboratories remote experiment, also called remote access laboratory. One problem in laboratories remote access is configuration experimental scenarios, these laboratories the experiment was already ready for experimentation, taking out the opportunity of the user to choose a domain experimental, then it is always limited to performing the same experiment. This problem, was identified in work master's thesis of a member of the project LaboREAM, called LARRE (SOUZA, 2011). in literature for the state of the art of Remote Access Laboratories (or LARs), in section 2.2, was identified this same problem. The definition of the problem this work is how to perform configuration and mounting autonomous the environments experimental? The strategy for solving this problem is to use Multi Agent System (or SMA) combined with XML to define the scenarios to be assembled. To assembly the experimental scenario, SMA is used in conjunction with a Mobile Robotic System (or MRS). To model the system was used to modeling methodology O-Mase, also developed a Petri net that represents the interactions between mobile robotic agents setting the stage, simulating the correct order or timing of task parallelism. Based on this concept, the SMA was implemented using JADE framework and lejos firmware. The results obtained from the tests applied to the system prove validity and feasibility of the solution.

Keywords: Tele-experimentation, remote experimentation, laboratory experimentation, WEB laboratory, remote access laboratory, science laboratory, autonomous configuration.

LISTA DE SIGLAS

AD	Aprendizado à Distância
ACL	Agent Communication Language
AID	Agent Identifier
AVA	Ambiente Virtual de Aprendizagem
CAI	Instrução Assistida pelo Computador
CLRC	Construir Laboratório Remoto Colaborativo
CMS	Content Management System
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CSAC	Computador com Suporte a Aprendizagem Colaborativa
DTD	Document Type Definition
FIPA	Foundation for Intelligent Physical Agents (1997 / 2000)
IVI	Interchangeable Virtual Instrument
IEEE	Institute of Electrical and Electronics Engineers
IEEE LOM	IEEE 1484 Learning Objects Metadata
JADE	Java Agent Development Framework
LaboREAM	Laboratório de Robótica Educativa do Amazonas
LAR	Laboratório de Acesso Remoto
LARRE	Laboratório de Acesso Remoto de Robótica
LeJOS	Lego Java Operating System
LR	Laboratório Remoto
LRC	Laboratório Remoto Colaborativo
MaSE	Multiagent Systems Engineering
MDA	Model Driven Architecture
Moodle	Modular Object-Oriented Dynamic Learning Environment
MRTA	Multi-Robot Task Allocation
OMACS	Organization Model for Adaptive Computational Systems
O-MaSE	Engineering Organization-Based Multi Agent Systems
PHP	PHP: Hypertext Preprocessor
PIM	Platform Independent Model
PSM	Platform Specific Model
RdP	Rede de Petri
SCORM	Shareable-Content Object Reference Model
SMA	Sistema Multiagente
SR	Sistema Robótico
SRM	Sistema Robótico Móvel
TIC	Tecnologia da Informação e Comunicação
VISA	Virtual Instrument Software Architecture
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WAMP	Apache, PHP, MySQL on Windows
WWW	World Wide Web
XML	Extensible Markup Language

LISTA DE ILUSTRAÇÕES

Figura 1. 1 Subsistema do LaboREAM.....	16
Figura 2. 1 Caracterização de experimentos usando mapas conceituais	21
Figura 2. 2 Ano por número de publicações atualizado de Gravier (GRAVIER <i>et al</i> , 2008) para trabalhos de 2008 a 2011	26
Figura 2. 3 Número de artigos por área científica atualizado de Gravier (GRAVIER <i>et al</i> , 2008) para trabalhos de 2008 a 2011	26
Figura 2. 4 Tecnologias usadas na construção dos laboratórios remotos atualizado de Gravier [Gravier <i>et al</i> , 2008] para trabalhos de 2008 a 2011	27
Figura 2. 5 Interseção de Laboratório Remoto Colaborativo modificado de (GRAVIER <i>et al</i> , 2008).....	29
Figura 2. 6 Multidisciplinaridade dos LRC modificado de (GRAVIER <i>et al</i> , 2008).....	29
Figura 2. 7 Ferramentas de comunicação usadas nos Laboratórios Remotos atualizado de Gravier (GRAVIER <i>et al</i> , 2008) para trabalhos de 2008 a 2011	30
Figura 2. 8 Etapas, Fases e Diagramas da metodologia MaSE (WOOD <i>et al</i> , 2001)	38
Figura 2. 9 Modelo de RdP da camada ambiente, movendo a bola (COSTELHA & LIMA, 2010).....	42
Figura 2. 10 Modelo de RdP da camada executor de ações, pegando a bola (COSTELHA & LIMA, 2010).....	42
Figura 2. 11 Modelo de RdP da camada coordenador de ações, marcando gol (COSTELHA & LIMA, 2010).....	43
Figura 3. 1 Controle mestre do laboratório (MASÁR <i>et al</i> , 2011).....	45
Figura 3. 2 GUI laboratório controle de guindaste (MASÁR <i>et al</i> , 2011).....	46
Figura 3. 3 WebLab-Box (GARCÍA-ZUBIA <i>et al</i> , 2010)	47
Figura 3. 4 GUI inicial do LARRE (SOUZA, 2011)	49
Figura 4. 1 Arquitetura geral do sistema	54
Figura 4. 2 Arquivo XML cenário.....	55
Figura 4. 3 Arquivo DTD cenário	56
Figura 4. 4 Arquivo XML Schema cenário	57
Figura 4. 5 Arquivo de tarefas	57

Figura 4. 6 Fluxograma dos passos da tarefa posicionar objeto.....	60
Figura 4. 7 Representação em mapa a) métrico por grade de ocupação e b) topológico	61
Figura 4. 8 Representação em mapa a) antes da montagem e b) depois da montagem.....	62
Figura 4. 9 Modificação do mapa antes da execução do Dijkstra a) para robô 1 e b) para robô 2	63
Figura 4. 10 Objeto levado até ponto objetivo a) do robô 1 e b) do robô 2	63
Figura 4. 11 Voltar para ponto de partida com a) robô 1 e b) robô 2.....	63
Figura 4. 12 Modelos de robôs móveis	65
Figura 4. 13 Objetos de captura.....	65
Figura 4. 14 Processo O-MaSE	67
Figura 4. 15 Modelo de Objetivo	68
Figura 4. 16 GMoDS	68
Figura 4. 17 Modelo de Domínio	69
Figura 4. 18 Modelo de Organização	69
Figura 4. 19 Modelo de Papéis	70
Figura 4. 20 Modelo de Classes de Agentes.....	71
Figura 4. 21 Modelo dos Protocolos de Agendamento	71
Figura 4. 22 Modelo dos Protocolos de Montagem.....	72
Figura 4. 23 Modelo de Capacidades	72
Figura 4. 24 Modelo de Modificar Cenário do Agente Professor	73
Figura 4. 25 Modelo de Plano Solicitar Cenário do Agente Estudante.....	73
Figura 4. 26 Modelo de Agendar Cenário do Agente Administrador	73
Figura 4. 27 Modelo de Plano Controlar SRM do Agente SRM.....	74
Figura 4. 28 Modelo de Plano Coordenar Montagem e Extrair Tarefas do Agente Coordenador	74
Figura 4. 29 Passos da RdP na tarefa posicionar objetos	75
Figura 4. 30 RdP para montagem do cenário com agentes robóticos concorrentes	76
Figura 4. 31 RdP para montagem do cenário com agentes robóticos sincronizados.....	77
Figura 5. 1 Ciclos de desenvolvimento do robô garra simples.....	79
Figura 5. 2 Ciclos de desenvolvimento do robô garra dupla.....	80
Figura 5. 3 Qualidade do cenário montado pelo robô garra simples.....	81
Figura 5. 4 Qualidade do cenário montado pelo robô garra dupla	81
Figura 5. 5 Montagem do cenário 1.....	83
Figura 5. 6 Montagem do cenário 2 depois de ter montado o cenário1	83

Figura 5. 7 Montagem do cenário 3 depois de ter montado o cenário 2	83
Figura 5. 8 Estados finais do a) cenário inicial, b) cenário 1, c) cenário 2, d) cenário 3	84
Figura 5. 9 Primeiro grupo de mensagens para montagem do cenário.....	85
Figura 5. 10 Segundo grupo de mensagens para montagem do cenário.....	86
Figura 5. 11 Terceiro grupo de mensagens para montagem do cenário	87

LISTA DE TABELAS

Tabela 2. 1 Modelagem da caracterização de experimentos usando tabela	21
Tabela 2. 2 Comparação dos tipos de laboratórios (DENIZ <i>et al</i> , 2003)	24
Tabela 2. 3 Detalhamento dos Kits educacionais robóticos (SASAHARA, 2007).....	35
Tabela 2. 4 Detalhamento das linguagens de programação para Lego Mindstorms	35
Tabela 2. 5 Fragmentos do método O-MaSE (DELOACH & GARCIA-OJEDA, 2010).....	40
Tabela 4. 1 Tarefas e robôs.....	58
Tabela 4. 2 Tarefa (T4) fictícia adicionada	58
Tabela 4. 3 Atribuição ótima de 4 tarefas.....	59
Tabela 4. 4 Passos para tarefa posicionar objeto	59
Tabela 4. 5 Grupo de robôs (G1) adicionada.....	64
Tabela 4. 6 Modelagem da caracterização de experimentos usando tabela	70
Tabela 5. 1 Desempenhos dos robôs para cada um dos passos em cada objeto.....	81

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA.....	15
1.2	OBJETIVOS	16
1.3	ESTRATÉGIA DE SOLUÇÃO DO PROBLEMA E JUSTIFICATIVA.....	17
1.4	ORGANIZAÇÃO DO TRABALHO.....	19
2	REVISÃO BIBLIOGRÁFICA E ESTADO DA ARTE	20
2.1	CLASSIFICAÇÃO DOS LABORATÓRIOS	20
2.2	ESTADO DA ARTE DE LABORATÓRIO DE ACESSO REMOTO	25
2.2.1	Aplicações científicas.....	25
2.2.2	Arquitetura de hardware e software	27
2.2.3	Questões e desafios	28
2.3	TENDÊNCIAS ATUAIS EM LABORATÓRIO DE ACESSO REMOTO.....	31
2.3.1	Padronização no desenvolvimento	31
2.3.2	Acessibilidade	32
2.3.3	Liberdade de configuração	32
2.4	SISTEMA MULTIAGENTE.....	33
2.5	FERRAMENTAS	34
2.5.1	Java Agent Development Framework	34
2.5.2	Legu MindStorms.....	34
2.5.3	Legu Java Operating System.....	36
2.6	METODOLOGIAS MASE E O-MASE.....	36
2.7	MODELAGENS DE TAREFAS ROBÓTICAS USANDO REDES DE PETRI	41
2.8	FÁCIL RECONFIGURAÇÃO DE LABORATÓRIO REMOTO	44
2.9	CONFIGURAÇÕES DISTRIBUÍDAS DE AMBIENTES DE EXPERIMENTAÇÃO	47
2.10	RESOLVENDO PROBLEMA DE ALOCAÇÃO DE TAREFAS	48
2.11	AMBIENTE ROBÓTICO EDUCACIONAL BASEADO EM SMA	49
2.12	CONCLUSÕES DO CAPÍTULO.....	50
3	LAR MULTI CONFIGURÁVEL E MONTAGEM AUTÔNOMA.....	53
3.1	ARQUITETURA DE SOFTWARE DO SISTEMA	53
3.1.1	Funcionamento da extração de tarefas	55

3.1.2	O Papel do Multi Robot Task Allocation na montagem do cenário	57
3.1.3	Estratégia para evitar colisões	59
3.1.4	Planejando a trajetória.....	60
3.1.5	Cooperação entre agentes robóticos	63
3.2	AMBIENTE OPERACIONAL.....	65
3.3	MODELAGEM DO SISTEMA USANDO METODOLOGIA O-MASE	66
3.4	MODELAGEM DO SISTEMA USANDO REDE DE PETRI	74
3.5	CONCLUSÕES DO CAPÍTULO.....	77
4	TESTES E AVALIAÇÃO DO SISTEMA.....	78
4.1	DESCRIÇÃO DOS TESTES EMPREGADOS.....	78
4.1.1	Testando os agentes robóticos separadamente	78
4.1.2	Testando os agentes robóticos colaborativos	82
4.2	CONCLUSÕES E DISCUSSÕES SOBRE OS TESTES.....	87
5	CONCLUSÕES E TRABALHOS FUTUROS.....	89
	REFERÊNCIAS	92
	ANEXOS	101
	ANEXO A – EXEMPLOS DE LAR	102
	ANEXO B – HIERARQUIA DE PAPÉIS EM VISÃO GERAL	103
	ANEXO C – PUBLICAÇÕES REALIZADAS.....	104

1 INTRODUÇÃO

Em experimentação pedagógica são usados laboratórios experimentais de ciências que fazem a integração entre teoria e prática. Para experimentos, o estudante desloca-se até uma sala laboratorial onde manipula equipamentos para o procedimento experimental, em laboratórios que são conhecidos como *Hands-On*. A construção e manutenção de laboratórios experimentais *Hands-On* com equipamentos de primeira qualidade têm um custo elevado. O custo elevado faz com que escolas com recursos financeiros limitados não tenham acesso a esse amparo pedagógico, tornando-se sem dúvida um sonho longe de ser alcançado. Como soluções para limitações financeiras de tais escolas, surgem os laboratórios alternativos, nos quais os equipamentos acadêmicos precisos são substituídos por outros de baixo custo, entretanto a baixa precisão na aquisição de dados pode gerar interpretações erradas dos experimentos. A alternativa pedagógica são os laboratórios simulados por software, chamados de laboratórios virtuais, que podem ser usados localmente e remotamente, mas ainda apresentam uma baixa sensação de realidade, causando também baixa qualidade experimental. Sendo assim, os laboratórios *Hands-On* são transformados em laboratórios remotos, também chamados de Laboratórios de Acesso Remoto (ou LAR), com objetivo de democratizar laboratórios de experimentação na modalidade chamada educação automatizada. Os tipos de laboratórios são discutidos no capítulo 3 - Revisão bibliográfica e estado da arte.

O tema e objeto de pesquisa desse trabalho de Mestrado são os Laboratórios de Acesso Remoto. No capítulo 3 são apresentadas várias áreas de aplicação desses laboratórios, entretanto pode-se generalizar as aplicações e afirmar que esses tipos de laboratórios podem ser aplicados em universidades quando equipamentos de um laboratório são caros, como por exemplo no projeto *Quantum Random Bit Generator Service* (QRBG, 2011). Outra aplicação é em escolas quando não possuem disponíveis laboratórios de experimentação, criando oportunidade de utilização, por exemplo, o *E-Lab* (E-LAB, 2011), permitindo estudantes realizarem experimentos de física *online*. Ainda, é aplicado em indústria quando se quer avaliar e testar novos instrumentos e dispositivos para finalidade comercial, exemplo do Project REAL (REAL, 2011), onde são testado padrões de resposta de antena em um laboratório remoto (ou LR).

Os Laboratórios de Acesso Remoto apresentam muitos desafios como à falta de reutilização de software em sua construção, falta de interoperabilidade entre laboratórios com

mesmo campo científico de experimentação e falta do uso de ferramentas de aprendizado colaborativo apoiado por computador, estes assuntos são discutidos nas seções do capítulo 3. Diversas questões de pesquisa têm surgido com a difusão do uso dos laboratórios remotos pelo Brasil e pelo Mundo, sendo que uma delas é o problema descrito nesta dissertação, detalhado na Seção a seguir.

O problema abordado neste trabalho de Mestrado é como configurar e montar um cenário de forma autônoma em um Laboratório de Acesso Remoto. Na comunidade científica temos pesquisas no mesmo campo de interesse deste Mestrado. Por isso a importância de conhecer essas comunidades é descobrir quem são os principais pesquisadores e qual o estado da arte do campo de pesquisa.

1.1 MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA

Este trabalho de Mestrado foi apoiado pelo Projeto 575894/2008-3, intitulado Ampliação de Acessibilidade a Laboratório de Ciências Usando Ambientes Virtuais e Tele robótica, mantido pelo CNPq que propõe implementar, desenvolver e tornar disponível um Laboratório Virtual para o Ensino de Ciências, projeto que foi conduzido pelos pesquisadores do Grupo de Sistemas Inteligentes do LaboREAM. O elemento motivador inicial foi a identificação de uma característica no projeto LARRE (SOUZA & NETTO, 2010), (SOUZA, 2011) e (SOUZA *et al*, 2011), que também faz parte do LaboREAM. Este trabalho é um subsistema do LaboREAM, que soluciona essa característica dos Laboratórios de Acesso Remoto (ou LARs) tornando-os mais acessíveis, apresentado na Figura 1.1. A modelagem e implementação do LARRE não contempla o uso de agentes para configurar e fazer montagem autônoma de cenários experimentais. O motivo é que o Sistema MultiAgente (ou SMA) foi elaborado com foco na teleoperação robótica, então, é o administrador do LARRE quem monta o cenário tele operado. Foi feita uma investigação para verificar se esta característica aparecia em outros laboratórios e foi confirmado, resultados que são detalhados na Seção 2.

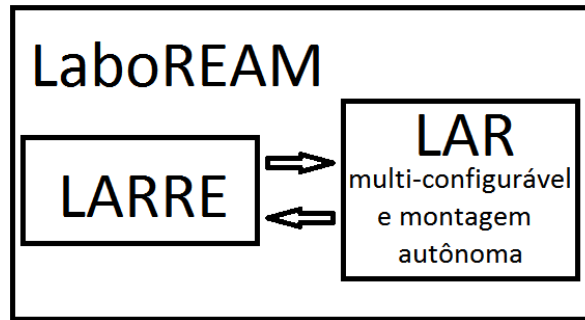


Figura 1. 1 Subsistema do LaboREAM

Para um melhor entendimento do problema abordado nessa dissertação de Mestrado vamos ilustrar com a seguinte situação: um aluno de ensino médio depois de assistir suas aulas teóricas de física clássica quer acessar um laboratório de experimentação para verificar na prática o que foi aprendido na teoria, mas infelizmente sua escola não possui laboratórios de experimentação de física clássica. Então o aluno recorre aos laboratórios remotos via Internet, também chamados de Web laboratórios ou LAR. Depois de muitas buscas na Web o estudante encontra um LAR em funcionamento. No site o aluno consegue realizar o único experimento de física clássica, exemplo de experimento de “Conservação de Energia”, e depois da experimentação não havia outra opção de novos experimentos no campo de experimentação em física clássica. O estudante, então, busca novos LARs em funcionamento. O problema acontece porque os experimentos em LAR são fixos e únicos, não permitindo a configuração do ambiente de experimentação. O problema de configurações e montagens de ambientes de experimentação é abordado neste trabalho de Mestrado. Definindo o problema: dado um ambiente real com objetos e instrumentos e uma lista de cenários, como configurar objetos e instrumentos para uma das opções de cenário solicitado e montá-lo deixando o ambiente pronto para experimentação. Para montar o cenário da configuração escolhida pelo usuário são usados robôs móveis.

1.2 OBJETIVOS

O objetivo geral é prover um mecanismo que possibilite a configuração e montagem de cenários experimentais em LAR, pois, essa característica não está presente nos LARs atuais.

A montagem deve ter qualidade superior a 75% de sucesso na montagem e de menos de 5 cm na precisão do posicionamento dos objetos.

Para alcançar o objetivo geral são realizados objetivos secundários. São os objetivos específicos:

- 1) Elaborar e definir parâmetros para configuração e montagem de cenários em LAR.

O objetivo identificar a *baseline* em liberdade de configuração experimental automatizada no levantamento do estado da arte. Assim, é gerado um *survey* do estado da arte e tendências atuais em LAR, focando em trabalhos relacionados à parametrização da liberdade de configuração experimental.

- 2) Criar uma camada baseada em Sistema MultiAgente (ou SMA) e Sistema Robótico Móvel (ou SRM) que permita a configuração e montagem de cenários experimentais.

O objetivo é elaborar uma arquitetura SMA combinada com SRM. Também é elaborada uma metodologia de construção de LAR de ciências multi-configuráveis e montagem autônoma. Ainda, para esse objetivo é estudado formalismo de representação de tarefas em sistemas robóticos autônomos.

- 3) Oferecer ao usuário de LAR a possibilidade de escolher o experimento a ser montado pelo sistema autônomo.

O objetivo é concretizado depois de codificado e implementado um LAR com os objetivos abordados nos itens 1 e 2.

- 4) Avaliar a efetividade do sistema proposto por meio de experimentos.

Para avaliar a qualidade do LAR multi-configurável e montagem autônoma serão realizados testes e avaliações para esse tipo de sistema, principalmente testes quantitativos referentes à precisão da montagem dos cenários.

1.3 ESTRATÉGIA DE SOLUÇÃO DO PROBLEMA E JUSTIFICATIVA

Como estratégia de solução, propõe-se neste trabalho uma camada de agentes inteligentes em sociedade, um SMA, para configuração de ambientes. Para atuar no ambiente real são usados agentes robóticos móveis que montam uma configuração de cenário escolhida pelo usuário do

sistema. Então, também é abordado neste trabalho: como robôs autônomos podem configurar cenários para experimentação?

Os estudos do SMA e robótica móvel começam antes do início do mestrado, comprovada por publicações no Anexo C, a intenção preliminar em resolver problemas por esta abordagem é assim justificada. A justificativa principal para usar SMA para resolver o problema deve-se ao fato de que a configuração e montagem de cenários de experimentação podem ser resolvidas de forma distribuída. A solução distribuída de problemas é abordada em diversos trabalhos, como, por exemplo, (O'HARE & JENNINGS, 1996) e (RUSSELL & NORVIG, 2004). Outra justificativa é que o uso de SMA oferece uma maior rentabilidade de recursos para problemas onde as atividades são delegadas (REIS, 2003). Pode ainda ser como justificada, o problema tem as atividades distribuídas aos agentes robóticos autônomos. Ainda, com paralelismo de atividades, podem ser atribuídas diferentes tarefas a diferentes agentes robóticos de forma que a execução da configuração do cenário seja mais rápida, ou seja, tem-se a montagem do cenário em menor tempo. Como a solução do problema é distribuída, podemos ter um sistema mais robusto, flexível e escalável.

O SMA é chamado de flexível quando os agentes adaptam-se para resolver problemas que passam por mudanças. E também, antecipam ações para futuros objetivos do SMA. Exemplo do SMA LAR que foi construído para ter capacidade de montar uma grande categoria de cenários que precisam de movimentação de objetos. Suas ações para montagem de cenários acontecem antes da atuação no ambiente, com trocas de mensagens para alcançar o objetivo do sistema. O SMA é robusto e escalável ao participar de ambientes dinâmicos e operáveis em estruturas organizacionais flexíveis com aumento substancial de carga sem que seu desempenho piore ao ponto de torna-lo inútil. Para explicar melhor é exemplificado com SMA LAR que aloca os robôs de acordo com o cenário a ser montado. Caso agentes robóticos fiquem *off-line* ou com capacidade de operação reduzida, o SMA LAR procura adaptar-se para montar o cenário eficientemente, alocando apenas os agentes robóticos *online* e com capacidade de operação que maximize a eficiência do SMA LAR em atingir seus objetivos.

1.4 ORGANIZAÇÃO DO TRABALHO

Além do capítulo de introdução, a presente dissertação está organizada com mais seis capítulos, resultando na seguinte estrutura: o 2 capítulo aborda a revisão bibliográfica relacionada com o projeto desenvolvido, mostrando a classificação dos laboratórios, o estado da arte e as tendências atuais em Laboratório Acesso Remoto, além de modelagens de Sistema Multiagente usando *Multiagent Systems Engineering* (ou MaSE) e *Engineering Organization-Based Multi Agent Systems* (ou O-MaSE) como metodologia e a Rede de Petri na modelagens de tarefas robóticas. Continuando, o 3 capítulo mostra os trabalhos correlatos ao problema apresentado. Já o 4 capítulo apresenta os métodos utilizados no desenvolvimento do Laboratório de Acesso Remoto, mostrando como funciona a configuração e montagem de cenários pelos agentes e o uso do Sistema Robótico Móvel com alocação de tarefas robóticas, a modelagem das tarefas do Sistema Robótico Móvel em Rede de Petri e a modelagem do Sistema Multiagente em O-MaSE. Em seguida, o 5 capítulo apresenta os experimentos realizados, a avaliação e a discussão dos resultados obtidos. E por último, o 6 capítulo relata as conclusões da estratégia para solução do problema e os possíveis trabalhos futuros para as pesquisas relacionadas aos Laboratórios de Acesso Remoto.

2 REVISÃO BIBLIOGRÁFICA E ESTADO DA ARTE

Este capítulo tem por objetivo contextualizar o projeto de Mestrado apresentando a fundamentação teórica dos conteúdos que embasam seu desenvolvimento. A seguir é apresentada a classificação dos tipos de laboratórios, o estado da arte e tendências atuais em LAR, exposição de algumas abordagens atuais e principais que existem na literatura envolvendo os agentes de software e SMAs, ferramentas utilizadas para o desenvolvimento do projeto, modelagem de SMAs usando metodologia O-MaSE e modelo de tarefas robóticas em Redes de Petri.

2.1 CLASSIFICAÇÃO DOS LABORATÓRIOS

Laboratório é um ambiente virtual ou real que proporciona condições controladas para realizar pesquisa científica, experimentos e medições. Outras instalações também são chamadas assim quando têm processos e equipamentos semelhantes ao dos ambientes científicos, exemplos dos laboratórios de informática – *Dicionário Michelin*.

Há vários tipos de laboratórios e JingMa (JINGMA & NICKERSON, 2006) discute que o tópico de estudar cada tipo de ambiente de experimentação é muito importante, por causa da similaridade tecnológica. Tomando por exemplo, o controle remoto dos dispositivos em automação dos robôs usados nas manufaturas, observa-se que há similaridades com os usados nos diversos tipos de ambientes de experimentação, como os dispositivos de acesso remoto. Com isso, pode ser reaproveitada a mesma tecnologia para diversos fins.

Seguindo o mapa conceitual, apresentado na Figura 2.1, os laboratórios experimentais podem ser caracterizados em dois eixos: “Experimento Usa” e “Tipos de Localização”. É entendido que o “Experimento Usa” “Modelo Físico” quando o procedimento experimental é simulado por software, e “Experimento Usa” “Dispositivo Físico” quando o procedimento experimental é executado em maior parte por dispositivo de hardware. São laboratórios de experimentação Hands-On os experimentos que usam dispositivos físicos e os usuários estão no mesmo lugar dos experimentos. São laboratórios de experimentação remota os experimentos que usam dispositivos físicos e os usuários não estão no mesmo lugar dos

experimentos. São laboratórios de simulação remota os experimentos que usam modelos físicos e os usuários não estão no mesmo lugar dos experimentos. São laboratórios de simulação local os experimentos que usam modelos físicos e os usuários estão no mesmo lugar dos experimentos. Foram conceituados quatro tipos de laboratórios de experimentação: Hands-On, experimentação remota, simulação remota e simulação local.

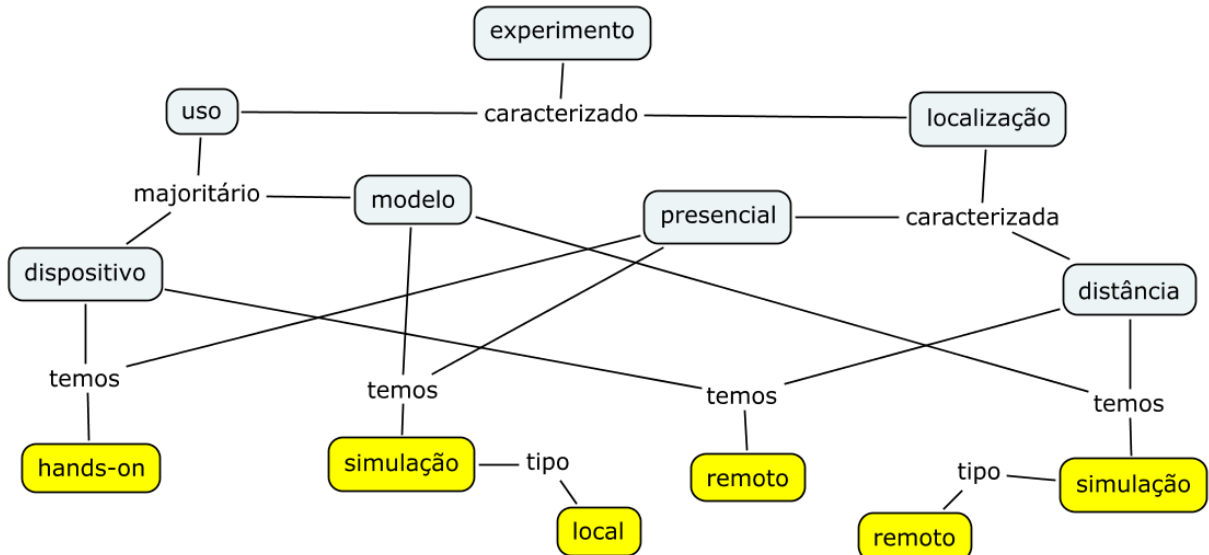


Figura 2. 1 Caracterização de experimentos usando mapas conceituais

Para melhor entender os limites laboratoriais são trocados os conceitos “Modelos Físicos” por “Software”, “Dispositivos Físicos” por “Hardware”, “usuários estão no mesmo lugar dos experimentos” por “Presencial” e “usuários não estão no mesmo lugar dos experimentos” por “Não presencial”. Essa mudança não altera a semântica dos conceitos. Então é refeito o mapa conceitual da Figura 2.1 usando tabela, como mostrado na Tabela 2.1.

Tabela 2. 1 Modelagem da caracterização de experimentos usando tabela

Experimento usa Localização	Software	Hardware
Presencial	Simulação local	Experimentação <i>Hands-On</i>
Não presencial	Simulação remota	Experimentação remota

Na prática, os limites entre os quatro tipos de laboratórios de experimentação não são definidas rigorosamente. O autor afirma que as definições de laboratório *Hands-On*, simulado e remoto são inconsistentes e confusas (*e.g* laboratório remoto chamado de *web labs*, *virtual labs* e laboratório de aprendizado distribuído). A maioria dos laboratórios é mediada por computadores e acontecem combinações com outros tipos de laboratórios de experimentação. Esta classificação ajuda que seja discutido os valores pedagógicos em cada

um dos tipos, ainda que o autor declare não há critérios padronizados para avaliar eficiência dos trabalhos desenvolvidos em cada ambiente, pois são realizados em várias áreas tornando difícil ter um critério padrão de avaliação.

Naef (NAEF, 2006) coloca um experimento de química nesses tipos de laboratórios de experimentação para descobrir o laboratório experimental mais adequado. O autor conclui que os laboratórios de experimentação *Hands-On* (nomeado por ele como laboratório real) e laboratório remoto fazem com que os usuários tenham uma reflexão maior sobre os experimentos, diferentemente dos experimentos em laboratórios de simulação remota e simulação local (nomeado por ele como laboratório virtual). Um motivo dessa reflexão menor em laboratórios simulados seria que o tempo de execução dos experimentos em laboratórios simulados é mais rápido do que em laboratórios reais, e com isso, os alunos em fases iniciais não entendiam o motivo de experimento ter sido executado rapidamente. Naef ainda observa a grande dificuldade de aproximar laboratórios virtuais de um experimento real devido à complexidade computacional do modelo físico simulado. Os laboratórios de experimentação *Hands-On* têm o custo de investimento e manutenção elevados, exemplo disso é que um sistema de instrumento ter no máximo dupla de estudantes por experimento e necessitar de pessoas técnicas para manter as instalações. Ainda são limitados geograficamente e temporalmente, ou seja, o estudante tem que estar presente ao laboratório em horário agendado para realizar os experimentos.

Da Tabela 2.2 (DENIZ *et al*, 2003), trabalho de JingMa (JINGMA & NICKERSON, 2006) e da dissertação de Mestrado (LOPES, 2007) podemos destacar as vantagens e desvantagem desses laboratórios. Podemos agrupar os laboratórios de simulação remota e simulação local como laboratórios virtuais tendo as seguintes vantagens e desvantagens:

- Estende a capacidade dos laboratórios em tempo e espaço, ou seja, não são limitados temporalmente e geograficamente. A experiência pode ser realizada em qualquer lugar (casa, escola ou escritório) e em qualquer momento (noite, manhã ou tarde);
- É necessário ter computador ligado à Internet para experiências virtuais remotas;
- O custo da utilização dos laboratórios virtuais é baixo porque a simulação pode ser copiada e distribuída;

- O custo da construção de laboratórios virtuais não é sempre mais baixo que os demais tipos. A construção de um modelo físico simulado por computador exige mão de obra especializada, tempo e grande investimento financeiro;

- O conhecimento adquirido pelo estudante depende da qualidade do software de simulação e da habilidade/competência do estudante. Um aluno inexperiente não entende e não sabe explicar porque simulação foi executada rapidamente;

- Não apresentam falhas experimentais retirando a possibilidade do aluno de questionar um erro do procedimento experimental, ou seja, o estudante não tem oportunidade de aprender por tentativa e erro;

- Apresenta fator de ficção que é indesejável em trabalho experimental;

- Possibilidade da adaptação do ambiente do laboratório para portadores de deficiências.

Para os laboratórios remotos temos as vantagens e desvantagens:

- Conta com possibilidade de partilha de recurso por várias instituições. Para ter acesso a um equipamento raro, basta ter acesso remoto ao equipamento para ser compartilhado, porém apenas um por vez pode acessar equipamento experimental;

- Estende a capacidade dos laboratórios em tempo e espaço, ou seja, não são limitados temporalmente e geograficamente. A experiência pode ser realizada em qualquer lugar (casa, escola ou escritório) e em qualquer momento (noite, manhã ou tarde);

- Não podem ser distribuídos, pois não são baseados somente em software igual os virtuais;

- O fator de ficção é reduzido com o uso de vídeo em tempo real junto com o controle supervisionado do equipamento, isso faz dos estudantes serem aprendizes ativos da experiência e não meros observadores;

- Possibilidade da adaptação do ambiente do laboratório para portadores de deficiências;

- Preparação e treinamento do estudante para melhorar destreza (habilidade motora) para o teste prático em laboratório de experimentação *Hands-On*;

- A comunicação presencial para apoio técnico e tutorial ao estudante é substituída por comunicação à distância como *chat*, fórum de discussão, *e-mail* e vídeo

conferência. Os questionamentos dos estudantes são mais facilmente respondidos em uma comunicação presencial do que em uma comunicação à distância;

- A liberdade de configuração experimental é limitada por opções de pré-configuração. Exemplo da conversão de um laboratório de química para laboratório de experimentação remota exige muito do sistema automatizado, pois na troca de experimentos é necessária a constante troca de material;
- Sendo uma Aplicação Web com ligação à Internet pode deixar o sistema instável. Exemplo do *delay* da recepção por vídeo depois da alteração em um cenário de sistema robótico quando estudante envia um comando de movimento;
- Ainda existe mentalidade tradicionalista de educadores que não acreditam na experimentação remota como mais uma forma de democratizar educação.

Tabela 2. 2 Comparação dos tipos de laboratórios (DENIZ *et al*, 2003)

Laboratório de Experimentação	Hands-On	Remota	Simulação Local e Remota
Características			
Experiência Hands-On	Não há nada como uma experiência real.	Muito perto da realidade.	Completamente virtual.
Sentimento de realidade	Alto.	Razoavelmente alta (se utilizar câmera).	Baixo.
Sensação de controle real	Alto.	Razoavelmente alto.	Razoavelmente alta.
Liberdade de configuração e experimentação	Possíveis experimentos adaptáveis. Limitado às instalações do laboratório.	Limitado por opção de pré-configuração.	Limitado por possibilidades de programação.
Apoio a Tutor	Normalmente presente durante os laboratórios ou horário de trabalho.	Estudantes fazem perguntas por <i>e-mail</i> . Assistentes pedagógicos ou técnicos em informática respondem.	
Apoio técnico	Técnico disponível.	Estudantes solicitam <i>e-mail</i> .	
Tempo de acesso	Limitado ao que foi agendado.	Limitado por configuração de <i>Set-Up</i> .	
Limite de acesso	Os alunos completam experimentos em um período de laboratório.	Sem limites, quando é acesso em fila.	
Supervisão	Assistente de laboratório fica presente para perguntas.	Questões por <i>e-mail</i> ou Chat de discussão.	
Verificar progresso	Relatórios enviados podem ser verificados em todo semestre.	Docente usa software para monitorar o acesso e o progresso do estudante. São enviados aos docentes relatórios de acesso e desempenho do estudante.	
Melhoramento pedagógico	Tradicional.	Adequado para aprendizagem local/remoto. Melhoramento pedagógico com uso de <i>Links Web</i> , animação e software de realidade virtual.	

2.2 ESTADO DA ARTE DE LABORATÓRIO DE ACESSO REMOTO

Gravier (GRAVIER *et al*, 2008) faz um levantamento bibliográfico para descobrir o estado atual dos laboratórios remotos e a evolução da próxima geração desses laboratórios. Este autor elaborou uma revisão bibliográfica dos últimos 10 anos, totalizando o levantamento de 42 publicações. Os gráficos da Figura 2.2, Figura 2.3, Figura 2.4 e Figura 2.7 representam o mapeamento modificado dessa revisão, onde foi feito uma extensão do ano de 2008 até 2011. A modificação foi feita da seguinte forma: realizada no site “*periodicos.capes.gov.br*” (CAPES, 2011) e buscamos pela palavra-chave *remote laboratory* na área de conhecimento Ciências Exatas e da Terra – Acesso Livre. A mesma palavra chave foi usada para buscar artigos no *IEEE Xplore digital library*. Foram examinados 29 trabalhos, que estão listados no Anexo A e seus trabalhos relacionados são citados na Seção de referências.

2.2.1 Aplicações científicas

No gráfico da Figura 2.2 de barras azuis, de 1997 a 2007, é observado pelo autor um pico de oito publicações a respeito de laboratórios remotos em 2002, depois foi diminuindo até novo pico em 2006. Esse último pico em 2006 uma grande parte é de revisão sobre laboratórios remotos, o que infere estar passando por uma revisão tecnológica para saber quais as próximas evoluções desses laboratórios. Depois de atualizar o gráfico de 2008 a 2011 é observado no gráfico da Figura 2.2 em barras vermelhas que o número de publicações cresce por causa principalmente das inovações tecnológicas em software e hardware.

São observadas pelo autor no gráfico da Figura 2.3 barras azuis que a maior quantidade das publicações tem concentração no domínio científico de eletrônica, seguido de robótica, física e automação. O motivo dessa concentração é a aproximação tecnológica desses domínios científicos com as tecnologias usadas na construção de laboratórios remotos.

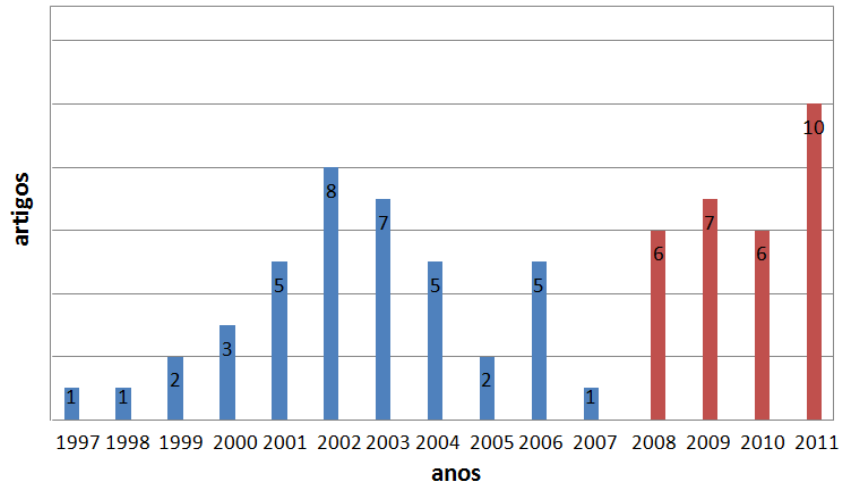


Figura 2. 2 Ano por número de publicações atualizado de Gravier (GRAVIER *et al*, 2008) para trabalhos de 2008 a 2011

Depois da atualização com outros trabalhos de 2008 á 2011 observamos no gráfico da Figura 2.3 barras vermelhas empilhadas nas barras azuis que a ordem muda um pouco mais mantém eletrônica em primeiro lugar, seguido de robótica, área multidisciplinar, automação e física. Um motivo de o campo científico multidisciplinar entrar para os três primeiros é que pesquisadores buscam laboratórios universalizados com objetivo de reduzir custos de construção e operação. Por exemplo, os benefícios educacionais em relação aos custos de construção operação têm valor maior em laboratórios remotos de um módulo de física geral do que de um laboratório específico, como de ondas.

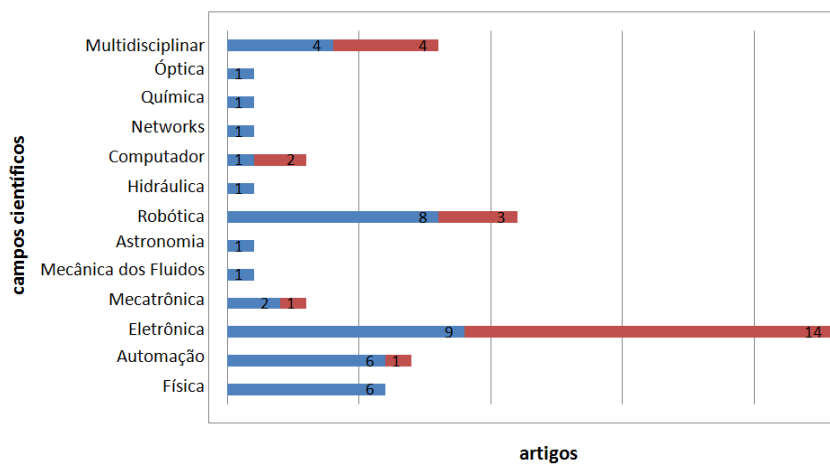


Figura 2. 3 Número de artigos por área científica atualizado de Gravier (GRAVIER *et al*, 2008) para trabalhos de 2008 a 2011

No trabalho de JingMa (JINGMA & NICKERSON, 2006) a maioria dos laboratórios é do domínio da engenharia. Na divisão feita (engenharia, ciências naturais e

outros) a engenharia apresentou 39 artigos. Um das principais justificativas é que engenharia é uma ciência aplicada.

2.2.2 Arquitetura de hardware e software

A arquitetura de software dos laboratórios remotos geralmente é a mesma em todas as publicações levantadas: comunicação entre cliente (ou usuário), servidor e dispositivo remoto. Essa arquitetura de software continua sendo majoritário para levantamento de 2008 a 2011 apresentados na Tabela do Anexo A. Uma das soluções de comunicação usada entre usuário e laboratório experimental é a *Virtual Network Computing*, mas é considerado inseguro por dar controle total do hardware para usuário e ineficiente por exigir muita largura de banda na comunicação do usuário com laboratório remoto.

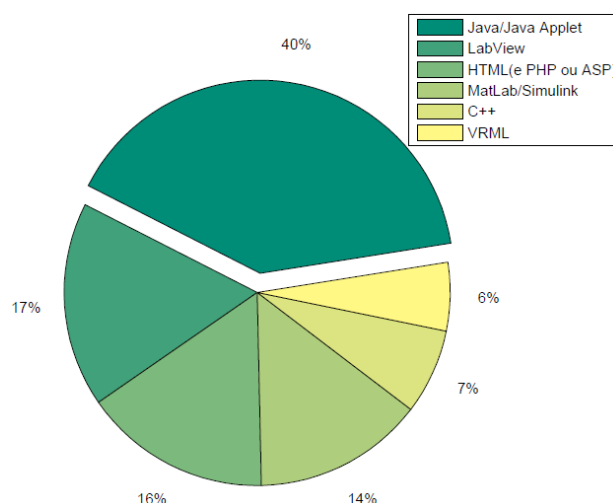


Figura 2. 4 Tecnologias usadas na construção dos laboratórios remotos atualizado de Gravier [Gravier et al, 2008] para trabalhos de 2008 a 2011

Existem várias tecnologias de software para desenvolver laboratórios remotos e virtuais (CHEN *et al*, 2010), exemplo de Java, Flash, VPN, XML, MatLab/Simulink, LabView e AJAX. Nas publicações pesquisadas na comunicação entre cliente e servidor foi usada em maior quantidade a tecnologia *Java* e *Java Applet* seguido de HTML (e PHP ou ASP). O uso majoritário da tecnologia *Java* é mantido mesmo depois da atualização de 2008 a 2011, como pode ser visto na Figura 2.4. Observamos também que a tecnologia LabVIEW está entre os três primeiros mais utilizados. A justificativa dessa tendência é que LabVIEW passou por amadurecimento tecnológico possibilitando interligar diversos hardwares. As

tecnologias usadas variam muito impedindo reusabilidade de software na construção dos laboratórios e interoperabilidade com outros laboratórios com a mesma abordagem laboratorial. A tecnologia de software usada para fazer a comunicação entre servidor e dispositivo remoto depende da arquitetura de laboratório a ser colocada na Internet e os dispositivos de hardware usados. Os dispositivos remotos geralmente são heterogêneos impedindo o reuso da mesma implementação em outros laboratórios. Para tentar resolver esse problema surgiram padrões *Virtual Instrument Software Architecture* (ou VISA), *Interchangeable Virtual Instrument* (ou IVI), *Shareable-Content Object Reference Model* (ou SCORM) e *IEEE 1484 Learning Objects Metadata* (ou IEEE LOM) apontam para soluções com característica de reusabilidade.

2.2.3 Questões e desafios

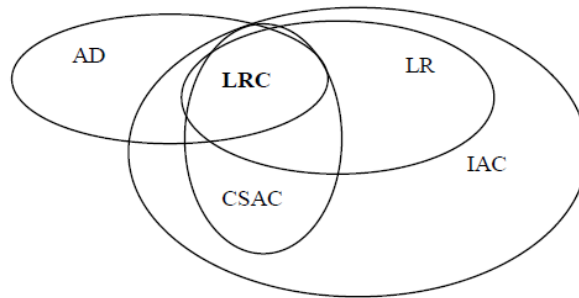
Algumas questões desafiadoras em laboratórios remotos são levantadas por Gravier (GRAVIER *et al*, 2008): reusabilidade (ou reutilização), interoperabilidade, colaboratividade e convergência com sistema de gerenciamento de conteúdo (SGC).

- *Reusabilidade (ou reutilização)*

O software desenvolvido tende a ser exclusivos, não podendo ser utilizados por outros laboratórios com mesma abordagem e domínio. Existe essa falta de reutilização dos laboratórios, e com a reutilização a construção de laboratórios remotos ficaria mais barato. Uma solução parcial é a formalização sobre a arquitetura de software, baseado em menor tempo de integração e habilidade para construção do laboratório remoto. Este tópico é uma tendência discutida na Seção 2.3.1.

- *Interoperabilidade*

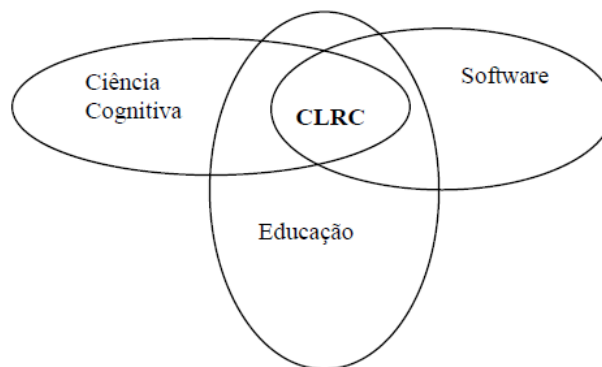
Geralmente os laboratórios remotos operam somente um serviço experimental, mas o ideal seria interligar sistemas de laboratórios remotos existentes (*workbench distributed*). Nesse ponto que temos a falta de interoperabilidade impossibilitando interligar sistemas laboratórios remotos. A interoperabilidade pode ser alcançada com uso dos serviços na *Web*.



AD: Aprendizado à Distância.
 LR: Laboratório Remoto.
 LRC: Laboratório Remoto Colaborativo.
 IAC: Instrução Assistida por Computador.
 CSAC: Computador com Suporte a Aprendizagem Colaborativa.

Figura 2. 5 Interseção de Laboratório Remoto Colaborativo modificado de (GRAVIER *et al*, 2008)

A localização dos serviços experimentais (*localization transparency of devices*) possibilitará a troca de dispositivo que apresente falha por outro semelhante em outro laboratório. A descoberta de serviços de laboratórios remotos evitaria investimento desnecessário com construção de novas *workbench* e evitaria também reinventar a roda com criação de laboratórios já existentes. A distribuição geográfica de laboratórios remotos requer gerenciamento de recursos, assim se um aparelho do laboratório falhar tem que trocar o aparelho por outro sem usuário perceber e sem perda de serviço, uma solução é o uso de *Logs*.



CLRC: Construir Laboratório Remoto Colaborativo.

Figura 2. 6 Multidisciplinaridade dos LRC modificado de (GRAVIER *et al*, 2008)

- *Colaboratividade e convergência com SGC*

Os laboratórios remotos devem ser apoiados nas teorias de aprendizagem. O ideal é casar CSAC e laboratórios remotos com vários domínios acadêmicos, gerando interseção

LRC, como mostrado na Figura 2.5 Construir Laboratório Remote Colaborativo (ou LRC) requer conhecimento de ciência cognitiva, educação e engenharia de software, como mostrado na 2.2.3.2. Em ciência cognitiva devemos ter conhecimentos como psicologia cognitiva, comunicação e teoria da atividade. Em educação temos que ter conhecimentos como cenarização pedagógica, teorias de aprendizagem e avaliação. Em engenharia de software temos que ter conhecimento tais como arquitetura de software, GUI e *Middleware*.

Nos trabalhos pesquisados pelo autor apenas 12 são LRC em diferentes níveis. Na atualização de sua pesquisa de 2008 a 2011 foi observado que parte majoritária dos laboratórios foi construída usando algum sistema de gestão de conteúdo (SGC) ou ambiente virtual de aprendizagem (AVA), como Joomla e *Modular Object-Oriented Dynamic Learning Environment* (ou Moodle). E constatou que a mídia de interação com usuário mais usada é voz sobre IP (VOIP), seguido de Chat e Vídeo. Uma justificativa para voz sobre IP ser a mais usada é a intenção de aproximar os laboratórios remotos dos *Hands-On*. Era de esperar que a mais utilizada fosse Mensagem Instantânea, por causa da simplicidade da implementação computacional em relação à VOIP. Depois da atualização de 2008 a 2011 somente a ordem mudou, temos Chat, seguido de VOIP e Vídeo, como pode ser visto na Figura 2.7 Isso porque a maioria dos trabalhos concentravam seus esforços em comprovar sua contribuição de inovação em laboratórios remotos e acabavam deixando essas ferramentas de interação de lado.

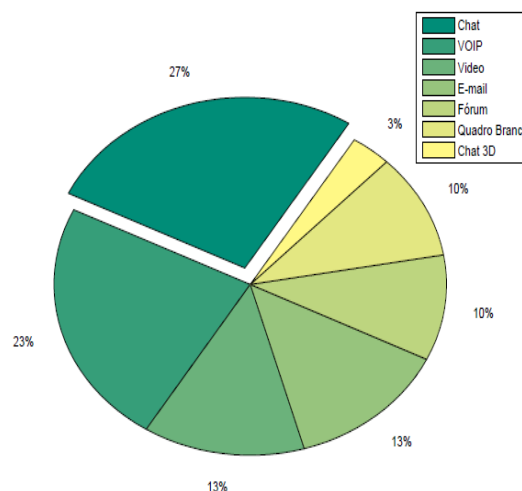


Figura 2. 7 Ferramentas de comunicação usadas nos Laboratórios Remotos atualizado de Gravier (GRAVIER *et al*, 2008) para trabalhos de 2008 a 2011

Sistema de Gestão de Conteúdo de Aprendizagem – SGCA (do inglês, *Learning Content Management Systems*) são estudados há muito tempo a fim de fornecer melhor exposição de conteúdo para aluno, melhor reusabilidade de conteúdo e facilidades de produção pelo professor. Segundo autor, existe uma distância entre SGCA e laboratórios remotos que impede que o professor acompanhe a evolução do aluno, impedindo uma ajuda adequada a cada aluno a fim de tirar melhor rendimento das atividades laboratoriais. Gravier conclui falando que afastando os problemas de reusabilidade, interoperabilidade, prática de colaborativas e convergência com SGCA vamos poder proporcionar uma experiência mais rica de aprendizado para alunos e professores em laboratórios remotos.

2.3 TENDÊNCIAS ATUAIS EM LABORATÓRIO DE ACESSO REMOTO

Essa Seção mostra as tendências atuais em LAR baseado no trabalho de Gomes (GOMES, 2009) que apresenta uma visão geral do estado da arte de LAR em eletrônica, depois identifica e questiona as tendências atuais e desafios em LAR.

2.3.1 Padronização no desenvolvimento

A primeira observação de Gomes (GOMES, 2009) é que a evolução de LAR é incrementada pela evolução tecnológica. Com essa evolução pesquisadores da área deixam de lado assuntos importantes como oferecer solução de padronização no desenvolvimento de novos laboratórios. A falta de padronização impede a modularidade, portabilidade, escalabilidade e interoperabilidade. O autor aponta soluções para esse problema como modelos de arquiteturas orientadas a serviços SCORM e IEEE LOM, *open frameworks* VISA e IVI, e desenvolvimento baseado em *Model Driven Architecture* (ou MDA).

SCORM usa normas técnica que não se preocupa com fatores pedagógicos, mas se preocupa com fatores de interoperabilidade, acessibilidade e reusabilidade de conteúdo *e-learning*. SCORM 1.2 é suportado no Moodle 1.9.3 e no Moodle 1.8.7 (SCORM, 2011). O IEEE LOM tem mesmos objetivos de interoperabilidade do SCORM.

IVI define uma arquitetura de *driver* aberta para instrumentação virtual, esses padrões fornecem elementos críticos que garantem a conexão dos instrumentos (intercambiabilidade). IVI é usado com LabView, MatLab, C++ e C# (IVI, 2011). VISA atualmente é mantida pela *IVI foundation*.

MDA é uma forma de desenvolver aplicações e escrever especificações baseado em *Platform Independent Model* (ou PIM) criado pelo *Object Management Group*, depois de criado PIM, ele é transformado em *Platform Specific Model* (ou PSM), e por último, PSM é transformado em código. Essa metodologia faz com que a modelagem tenha forte importância no processo de desenvolvimento. O processo MDA oferece os benefícios: produtividade, portabilidade e interoperabilidade (MDA, 2011).

2.3.2 Acessibilidade

Outra preocupação destacada por Gomes (GOMES, 2009) é a acessibilidades em LAR em lugares onde a Internet tem largura de banda de rede e número de computadores limitados. Exemplo do trabalho de Mittal (MITTAL *et al*, 2009) que apresenta solução prática para transmissão de vídeo em tempo real de um experimento em LAR com melhor qualidade dependendo da largura de banda da rede. Ainda nesse mesmo ponto da acessibilidade é questionado o poder de uso desses laboratórios como facilitadores para pessoas com necessidades especiais, iniciativa apoiada pela WAI do grupo W3C. Na forma mais ampla da acessibilidade Gomes afirma que não é garantida a disponibilidade de operação 24 por 7 em LAR. Chen (CHEN *et al*, 2010) e Mittal afirmam que isso acontece porque os pesquisadores estão mais preocupados em provar novos conceitos educacionais em LAR e não se preocupam com a questão de garantir operação contínua.

2.3.3 Liberdade de configuração

Outra tendência importante destacada por Gomes (GOMES, 2009) são as pesquisas para liberdade de configuração dos experimentos. Em simulação local e remota a configuração é limitada a parâmetros predefinidos da configuração do software, nesse ponto acabam

afastando-se dos laboratórios *Hands-On* que é limitado às instalações do laboratório. No caso dos laboratórios remotos essa tendência depende dos avanços tecnológicos na área de Engenharia de Controle e Automação, pois os procedimentos experimentais dependem dessa capacidade de controle automático.

2.4 SISTEMA MULTIAGENTE

Antes de discutir SMA é necessário definir Agente. É uma entidade de software onde em busca de seus objetivos apresenta as seguintes características (WOOLDRIDGE, 2002) (WOOLDRIDGE & JENNINGS, 1995):

- **Autonomia.** Um agente tem sua própria *thread* de execução interna, tipicamente orientada para a realização de uma tarefa específica, e decide por si quais ações ele deve executar em que momento.
- **Contextualização.** Agentes executam suas ações enquanto situado em um ambiente particular. O ambiente pode ser computacional (por exemplo, *Web Site*) ou físico (por exemplo, fabricação de gasoduto) e o agente pode sentir esse ambiente.
- **Pró-atividade.** A fim de realizar seus objetivos em um ambiente dinâmico e imprevisível o agente pode agir a sua necessidade para garantir que seus objetivos sejam alcançados e definir novas metas.

Muitas definições de Agente podem ser encontradas na literatura, mas elas não chegam a um acordo comum. Temos os atributos de Agentes mais comuns nas literaturas, como ser autônomo, social (tem linguagem de comunicação), reativo e proativo (SOWA, 1999).

Então, SMA é um sistema computacional em que dois ou mais agentes interagem ou trabalham em conjunto de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objetivos. A investigação científica e a implementação prática de SMA estão focalizada na construção de *standards*, princípios e modelos que permitam a criação de pequenas e grandes sociedades de agentes semiautônomos, capazes de interagir convenientemente de forma a atingirem os seus objetivos (REIS, 2003). O SMA é usado quando o problema é complexo e usamos a técnica de divisão e conquista para resolvê-lo. Ainda quando o problema é intrinsecamente distribuído e exige rápido tempo de resposta.

2.5 FERRAMENTAS

No desenvolvimento computacional do SMA foi usado o *Java Agent DEvelopment Framework* (ou JADE). Para o controle físico do ambiente foi usada a plataforma de robótica educacional *Lego MindStorms* sendo programado em linguagem orientada a objetos chamada *Lego Java Operating System* (ou LeJOS). Essas ferramentas são detalhadas nas seções seguintes.

2.5.1 Java Agent Development Framework

É um middleware de agentes que implementa uma plataforma distribuída e um framework de desenvolvimento para SMA. O JADE foi desenvolvido pelo *CSELT - Centro Studi E Laboratori Telecomunicazioni e Univ. Parma - Università degli Studi di Parma*. O JADE é completamente implementado em Java e obedece às especificações da *Foundation for Intelligent Physical Agent* (ou FIPA) (FIPA, 2011) e utiliza classes já definidas em Java.

2.5.2 Lego MindStorms

Com a proposta de ter um kit mais versátil, com montagem e programação mais rápida deixando de lado a parte eletrônica, Lego e outras empresas criaram Kits educacionais robóticos. Os kits educacionais existentes no mercado são o Super Robby, Robótica Fácil, Robo Fácil, SIROS, Hajiime, GoGoBoard e Lego Mindstorms. Mais detalhes são mostrados na tabela 2.3.

Neste projeto é usado o LEGO Mindstorms NXT e LEGO Mindstorms NXT 2.0. O primeiro kit educacional da Lego foi o RCX 1.0, depois foi lançado o RCX 2.0, em seguida o Mindstorms NXT, e atualmente foi lançado o Mindstorms NXT 2.0.

Três diferentes ambientes de programação são oferecidos pela LEGO para o NXT: NXT-G, ROBO LAB e ROBOTC. Os ambientes de programação tornam a programação para os robôs Mindstorms relativamente fácil.

Tabela 2. 3 Detalhamento dos Kits educacionais robóticos (SASAHARA, 2007)

Kit	Nacionais					Importados	
	Super Robby	Robótica Fácil	RoboFácil	SIROS	Hajime	GoGo Board	Legu Mindstorms
Tipo Arquitetura	Fechada	Fechada	Pública	Fechada	Pública	Pública	Fechada
Open Source	Não	Não	Sim	Não	Sim	Sim	Não
Software Livre	Não	Não	Não	Não	Sim	Não	Não
Tipo de Interface	Externa	Externa	Externa	Externa	Interna	Externa	Externa
Usa Microcontrolador	Sim	Sim	Sim	Sim	Não	Sim	Sim
Programação Icônica	Indisponível	Disponível	Disponível	Indisponível	Disponível	Disponível	Disponível
Autônomo	Não	Não	Não	Não	Não	Sim	Sim
Utiliza sucata	Não	Não	Não	Não	Sim	Não	Não
Uso de sensores	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Suporte à reprogramação	Não	Não	Sim	?	Sim	?	Sim
Acesso via Web	Não	Não	Não	Sim	Não	Não	?

Além disso, a comunidade *open source* tem desenvolvido soluções alternativas em programação para o NXT como NXC, pbLUA e NXJ. Mais detalhes apresentados na tabela 2.4.

Tabela 2. 4 Detalhamento das linguagens de programação para Lego Mindstorms

Linguagens para Lego NXT								
Recursos	NXT-G Retail	NXT-G Educational	RoboLab 2.9	NBC	NXC	RobotC	NI LabVIEW Toolkit	leJOS NXJ
Tipo de Linguagem	Gráfica	Gráfica	Gráfica	Assembly	C-like	C	Gráfica	Java
Firmware	Padrão	Padrão	Padrão	Padrão	Padrão	Padrão	Padrão	leJOS
IDE (incluído?)	Sim	Sim	Sim	Sim	Sim	Sim	Não	Plugins para Eclipse e NetBeans.
Windows	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Mac OSX	Sim	Sim	Sim	Sim	Sim	Ainda não	Sim	Sim
Linux	Não	Não	Não	Sim	Sim	Não	Não	Sim
Eventos	Não	Não	Sim	Não	Não	Sim	Não	Standard Java events
Multithreading	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Bluetooth módulo para PC	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim
Bluetooth módulo a módulo	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim
Bluetooth módulo para outro dispositivo	Não	Não	Não	Não	Não	Sim	Não	Sim
Sistema de arquivo	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Ponto Flutuante	Não	Não	Sim	Não	Não	Sim	Não?	Sim
Como obtê-lo	Incluído como versão NXT Retail no site: mindstorms.lego.com	Disponível com versão educacional da NXT no site: legoeducation.com	Disponível com versão educacional da NXT no site: legoeducation.com	Web Site da BricxCC, bricxcc.sourceforge.net	BricxCC Web Site bricxcc.web site	RobotC.net ou LEGO Education	LabVIEW toolkit Site	Web Site lejos: lejos.sourceforge.net

Linguagens para Lego NXT								
O que você quer fazer...	NXT-G Retail	NXT-G Educational 2.9	RoboLab 2.9	NBC	NXC	RobotC	NI LabVIEW Toolkit	leJOS NXJ
Fazer robôs sem ter conhecimento de programação	Sim	Sim	Sim					
Aprender a usar o Programa NXT				Não				
Escrever programas "rápidos"				Sim	Sim	Sim		Talvez?
Escrever rapidamente programas	Sim	Sim	Sim					
Saber conceitos avançados de programação				?	Sim	Sim	Sim	Sim
Adaptado para...	Pessoas começar a programar simples tarefas no módulo	Escolas com início no NXT	Escolas que querem atualizar o módulo RCX para o NXT	Programador avançado	Programador de C	Aplicações que exijam velocidade máxima	Utilizadores LabVIEW, ou as pessoas que desejam melhorar NXT-G	Programadores de Java

2.5.3 Lego Java Operating System

NXJ LeJOS é um sistema operacional *Open Source* baseado em Java para programar *Lego Mindstorms* em linguagem orientada a objeto. LeJOS consiste em três partes principais:

1. Uma máquina virtual para execução de *Java bytecode*.
2. Um API para programar o tijolo inteligente do Lego.
3. Ferramentas adicionais de software.

LeJOS possui implementação própria dos pacotes *java.lang.java.utile java.ioe* ainda têm as características:

- Linguagens Orientada a Objetos, igual ao Java.
- *Threads*, permitindo multitarefas.
- Vetores multidimensionais.
- Inclui tipos como *float*, *longestring*, igual o Java.
- Um API bem documentada.

2.6 METODOLOGIAS MASE E O-MASE

Para modelar sistemas complexos usando o paradigma de SMA inúmeras metodologias surgiram para prover mecanismos para sua construção (HENDERSON-SELLERS & GIORGINI, 2005) (NETTO, 2006). Nessa subseção são citadas algumas dessas metodologias

como GAIA (ZAMBONELLI *et al*, 2003), TROPOS, MAS-ML, ANOTE (LM-SMA) (CHOREN & LUCENA, 2003) e MAS – CommonKADS, mas neste trabalho são apresentadas e exploradas o MaSE e O-MaSE.

O objetivo principal da MaSE é prover ao desenvolvedor de um SMA uma metodologia para construir passo-a-passo seu sistema partindo de um conjunto inicial de requisitos até chegar às fases de análise, *Design* e implementação (SOUZA & NETTO, 2010), (WOOD *et al*, 2001). Ainda segundo Souza e Netto (SOUZA & NETTO, 2010), MaSE considera os agentes uma especialização dos objetos, tornando possível aplicar as técnicas utilizadas de Orientação a Objetos nos modelos de especificação e de *Design* de SMA. Esta metodologia reconhece que os agentes possuem características próprias que os tornam capazes de coordenar outros agentes, comunicando-se por meio de troca de mensagens e de ações pró-ativas, voltadas para atingir os objetivos individuais de cada agente e de todo o SMA. Essa metodologia é independente de qualquer arquitetura de agente, linguagem de programação, ou *framework* de comunicação (WERNECK *et al*, 2011).

A metodologia consiste em três etapas na Análise (*Analysis*), em quatro etapas no *Design* e em seus nove respectivos diagramas. São as três etapas na fase de Análise: Capturando os Objetivos (*Capture Goals*), Estabelecendo os Casos de Uso (*Applying Use Case*) e Definindo os Papéis (*Refining Roles*). E as quatro etapas na fase do *Design*: Criando as Classes de Agentes (*Creating Agent Classes*), Desenvolvendo os Diálogos (*Constructing Conversations*), Agrupando as Classes de Agentes (*Assembling Agent Classes*) e *Design* do Sistema (*System Design*). As fases, etapas e diagramas da metodologia são apresentados na Figura 2.8 Cada seta indica como um diagrama pode afetar o outro, não é a ordem de processo de desenvolvimento.

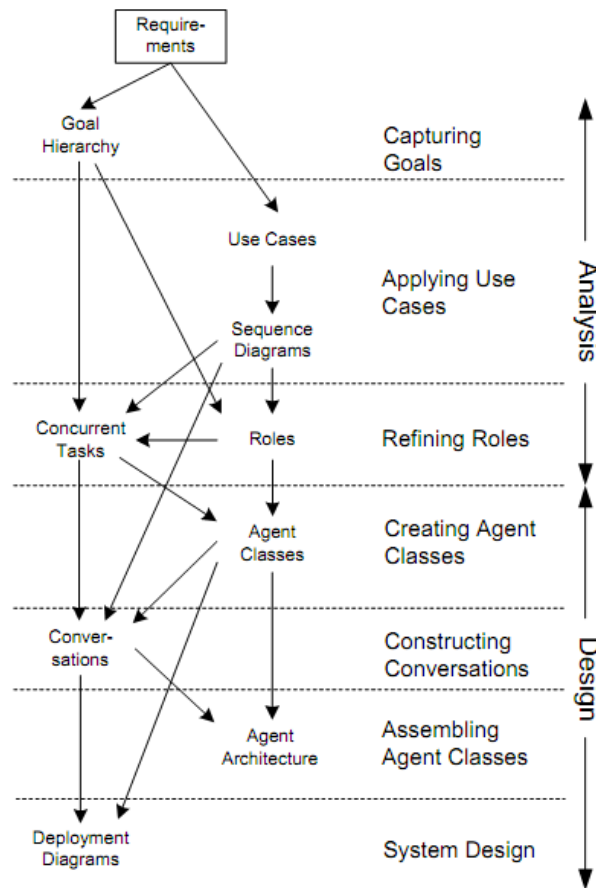


Figura 2. 8 Etapas, Fases e Diagramas da metodologia MaSE (WOOD *et al*, 2001)

A primeira etapa na fase de Análise é Captura de Objetivos onde são expressos de forma hierárquica os objetivos a serem alcançados pelo sistema, para isso é usado diagrama de hierarquia de objetivos. Na etapa Estabelecendo os Casos de Uso os requisitos funcionais identificados na etapa anterior são representados em diagramas de casos de uso. A última etapa da fase de Análise é Definindo os Papéis onde são os comportamentos esperados dos agentes, para isso é usado diagrama de hierarquia de papéis. Esses papéis são detalhados por um conjunto de tarefas independentemente executadas de forma simultânea e descritas por modelo de estado finito em um diagrama de tarefas simultâneas.

A primeira etapa na fase de *Design* é Criando as Classes de Agentes onde os papéis são agrupados em classes sendo estabelecido o diálogo entre elas, para isso é usado diagrama de classes de agentes. Na etapa Desenvolvendo os Diálogos é detalhado por dois agentes e definidos por um protocolo de coordenação usando dois diagramas de classe de comunicação. A etapa de Agrupando as Classes de Agentes onde primeiro é definido arquitetura do agente e depois seus componentes. O *Designer* pode escolher entre as

arquiteturas de agentes: Interação de crença e desejo (*Belief-Desire-Intention*), Reativa (*Reactive*) ou Base de conhecimento (*Knowledge Base*). Para essa etapa é usado o diagrama de arquitetura de agente. E a última etapa é *Design* do Sistema que é definido pelo diagrama de desenvolvimento para mostrar os números, os tipos e as locações de instâncias de agentes em um sistema, semelhante ao do UML.

Exemplos de modelagens são realizadas por Souza (SOUZA & NETTO, 2010) para o LARRE e Gago (GAGO *et al*, 2009) para um sistema educacional. Os autores usam uma ferramenta específica chamada de *agentTool* (aT) (DELOACH & WOOD, 2001) para modelar o SMA apoiada na metodologia MaSE e O-MaSE.

Foram identificados problemas com MaSE:

- Não modela interações do SMA com ambiente.
- Não permite que na modelagem o SMA projete sua própria organização em tempo de execução.
- Não modelar e integrar subequipes em um SMA, possibilitando maiores níveis de abstrações, como em SMA de times de futebol.
- Não modelava na fase de análise conversas com mais de dois agentes, fazendo com que a conversa se encaixa com mais dificuldade.
- Não permite o desenvolvedor do SMA pular etapas da metodologia, o *AgentTools* tem estruturas rígidas obrigando o MaSE também seguir essa rigidez.

O-MaSE é um framework derivado do MaSE que é constituída por metamodelo, um conjunto de fragmentos de métodos e diretrizes. O metamodelo é um conjunto de conceitos para design e implementação do SMA. Os fragmentos de método são tarefas executadas para produzir produtos como modelos, documentação ou códigos, Tabela 2.5. E as diretrizes orientam como usar os produtos dos fragmentos de métodos para criar uma instância do O-MaSE. Fragmentos O-MaSE são instâncias de elementos SPEM como tarefas, produtos de trabalho e papéis, essas instâncias são definidas em termos de conceitos para O-MaSE. Por exemplo, o modelo de papéis O-MaSE é uma instância de produto de trabalho SPEM e é definida em termos de papéis, objetivos e capacidades, e cada uma dessas são definidas em metamodelo O-MaSE.

Tabela 2. 5 Fragmentos do método O-MaSE (DELOACH & GARCIA-OJEDA, 2010)

<i>Activities</i>	<i>Tasks</i>	<i>Work products created/modified</i>	<i>Responsible method-roles</i>
Requirements gathering	Requirements specification	Requirements spec	Requirements engineer
Problem analysis	Model goals	Goal model	Goal modeller
	Refine goals		
Solution analysis	Model domain	Domain model	Domain modeller
	Model organisation interfaces	Organisation model	Organisation modeller
	Model roles	Role model	Role modeller
	Define roles	Role description document	
Architecture design	Define role goals	Role goal model	
	Model agent classes	Agent class model	Agent class modeller
	Model protocols	Protocol model	Protocol modeller
	Model policies	Policy model	Policy modeller
Low level design	Model plans	Agent plan model	Plan modeller
	Model capabilities	Capabilities model	Capabilities modeller
	Model actions	Action model	Action modeller
Code generation	Generate code	Source code	Programmer

O framework O-MaSE é baseado em dois meta-modelos, o SPEM 2.0 para definir conceitos da metodologia relacionada e O-MaSE para definir conceitos de produtos relacionados. O metamodelo O-MaSE é baseado em *Organization Model for Adaptive Computational Systems* (ou OMACS), abordagem organizacional que permite organizações de decomposição baseado em time, hierarquia e *holonic*, e ainda permite reorganização em tempo de execução. Os principais atributos da O-MaSE são objetivo, papéis, agente, capacidade, modelo de domínio, modelo de protocolos e organização.

O-MaSE define atividades e tarefas e não fases, porque existem várias maneiras de organizar as atividades e a metodologia O-MaSE não predefine uma. O Engenheiro de SMA que adota/cria sua organização em fases, pode seguir uma abordagem, ver exemplo modelo em cascata usando o-MaSE. Existem dependências entre atividades que são definidas em diretrizes de construção do método.

2.7 MODELAGENS DE TAREFAS ROBÓTICAS USANDO REDES DE PETRI

Na literatura encontramos trabalhos que representam comportamentos dos agentes robóticos fazendo uso de sistema de eventos discretos baseado em autômato de estado finito. Como exemplo, temos o trabalho de Carvalho (CARVALHO *et al*, 2006) que apresenta uma arquitetura hierárquica de comportamentos para dois desafios de competições de robótica. Os planos de tarefas são atributos a serem representados por agentes em um Sistema Robótico (ou SR). Os planos de tarefas robóticas são geralmente modelados como sistemas distribuídos concorrentes, e para esse fim o uso de autômato de estados finitos não é tão robusto quando as RdP (MARRANGHELLO, 2005). O trabalho de Costelha (COSTELHA & LIMA, 2010) representa o plano de tarefa robótica usando RdP.

O *framework* apresentado por Costelha permite uma abordagem sistemática para modelagem por módulo, análise e execução de tarefas robóticas. O *framework* é dividido nas camadas: modelos de plano das tarefas, de ações e de ambiente. A RdP usada é do tipo ordinária (ou clássica) marcada e estocásticas generalizada, onde os lugares representam tarefas, ações primitivas e conjunto de predicados lógicos definidos por leitura de sensores. Costelha faz uso do *framework* proposto para modelar um cenário de futebol robótico, onde são representados os diferentes níveis de granularidade em cada uma das camadas do modelo *Bottom-Up*. O *framework* contém respectivamente: camadas de ambiente, de executor de ações, de coordenador da ação e de organização. Onde temos a camada de ambiente como a primeira *Bottom* e a de organização a primeira *Up*. Todas as quatro camadas fazem parte do processo de análise, mas somente as duas últimas e parte da segunda fazem parte do processo de execução. Na camada de ambiente é onde acontecem às mudanças realizada por agentes ou pelo meio físico, exemplo da bola em livre movimento é representado na RdP pela Figura 2.9

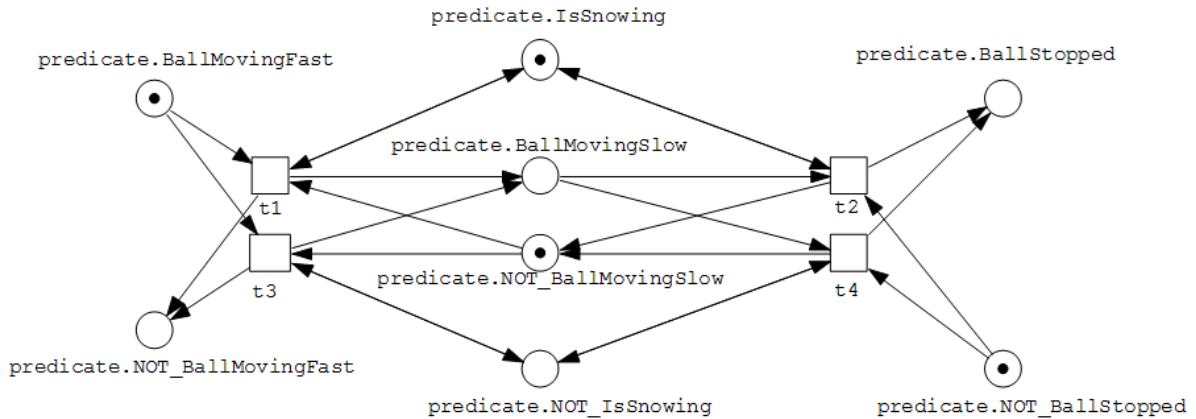


Figura 2. 9 Modelo de RdP da camada ambiente, movendo a bola (COSTELHA & LIMA, 2010)

A camada executor de ações é onde mudanças no ambiente acontecem quando ocorrem ações em certas condições, exemplo do agente pegando a bola é representado na RdP pela Figura 2.10 Observe que os lugares *predicate.r.SeeBall* e *predicate.r.CloseToBall* representam as condições sendo executadas (justificando o uso do *.r.*) e os lugares *predicate.NOT_HasBall* e *predicate.e.HasBall* são os efeitos (justificando o uso do *.e.*).

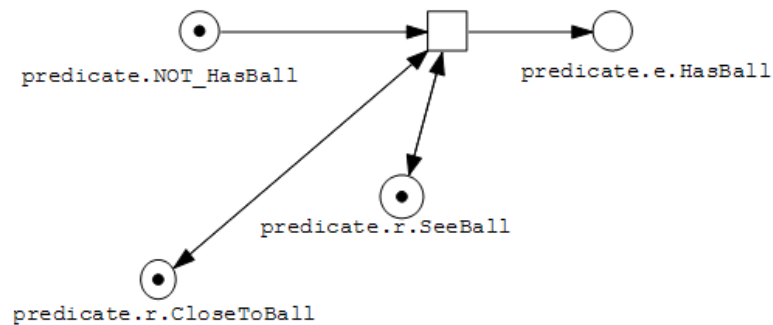


Figura 2. 10 Modelo de RdP da camada executor de ações, pegando a bola (COSTELHA & LIMA, 2010)

A camada coordenador de ações é onde acontece a composição das ações para fazer os planos de tarefas, exemplo do modelo em RdP que representa a tarefa marcando gol, como pode ser visto pela Figura 2.11 Observe que no lugar *action.o.Kick2Goal* o prefixo de *Kick2Goal* e sufixo de *action* representa que esse local deve ser marcado no final da execução da RdP.

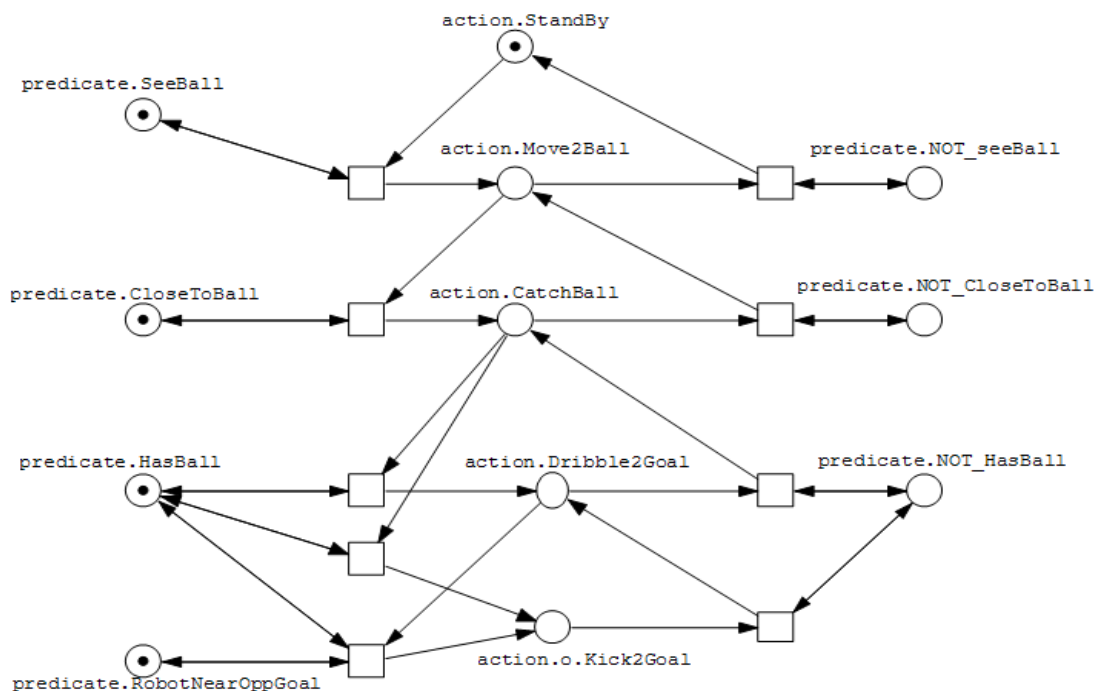


Figura 2. 11 Modelo de RdP da camada coordenador de ações, marcando gol (COSTELHA & LIMA, 2010)

É importante salientar que a camada de organização não é implementada por Costelha. Com todas as camadas modeladas em RdP. O autor apresenta um algoritmo do processo de expansão que permite reunir os modelos de plano de tarefas, de ações e de ambientes em uma única RdP, assim é feita a análise nessa única RdP. A análise pode ser lógica e/ou propriedade de desempenho probabilístico, exemplo *deadlocks* e/ou probabilidade de atingir um estado de marcação.

Este capítulo também é apresentado os principais trabalhos correlatos que exploram de alguma forma o aspecto da liberdade de configuração em LAR. Foram tabelados vários laboratórios remotos, entretanto, poucos deles apresentam algum grau na liberdade de configuração. Para seção a seguir foram selecionados os trabalhos que apresentam maior grau. Ainda são apresentados trabalhos que ofereceram sustentação para a ideia de integrar agentes de software, formando uma comunidade de um SMA, para configuração de LAR no apoio às atividades colaborativas de montagens do ambiente de experimentação, objeto de estudo desta pesquisa.

2.8 FÁCIL RECONFIGURAÇÃO DE LABORATÓRIO REMOTO

Masár (MASÁR *et al*, 2011) desenvolve um framework para facilitar design de novos laboratórios remotos ou reconfiguração de laboratórios remotos existentes. A principal justificativa para o trabalho é que os laboratórios remotos são construídos de maneira *ad hoc*, tornando difícil desenvolver tarefas experimentais fora do escopo desses laboratórios específicos, pois eles não foram concebidos para esse fim. Para desenvolver o framework, são analisados vários laboratórios remotos e destacados alguns fatores comuns nos experimentos:

- O controle é feito em tempo real.
- O usuário pode mudar vários parâmetros do controle experimental.
- Os dados produzidos são apresentados para o usuário poder fazer avaliação experimental.
- O usuário deve observar o experimento realizado para garantir a segurança experimental.

Ainda são destacadas características comuns no processo de execução experimental:

- Segurança. Esta característica envolve condições de operações que podem levar a condições perigosas, tais estados são reconhecidos para serem evitados e caso ocorra devem existir controle automático do sistema para sair destes estados.
- Fácil operação. A operação experimental deve ser fácil o suficiente para cumprir o objetivo do experimento.
- GUI atrativa. Os objetos de controle bem como parâmetros experimentais devem ser claros e acessíveis para usuário, exemplo, uso de vídeos e gráficos dos dados coletados devem ser sincronizados para não prejudicar a conclusão do aluno sobre o experimento.

E também identifica as fases de operação comuns em experimentos remotos:

- Inicialização. Para começar um experimento remoto os dispositivos devem estar inicializados, pois podem estar em estados indefinidos.
- Set-Up. O experimento pode requerer que tenha dispositivos pré-definidos para começar experimentação.

- Experimentação. Essa fase é onde acontece a experimentação, caso usuário queira repetir experimento o sistema já tem Set-Up da fase anterior.
- Controle seguro. Caso sistema entre em um estado indesejado ele será automaticamente comutado para modo de controle de segurança.

Depois de identificar as fases de operação foi desenhada a unidade de controle mestre na Figura 3.1.

No primeiro uso do laboratório o sistema pode não estar inicializado, então o usuário deve aguardar a inicialização do sistema, esse comando de inicialização pode ser enviado pelo próprio usuário. Depois de inicializado o laboratório fica pronto para ser controlado e caso seja necessário o usuário pode Set-Up procedimento desejado para um determinado experimento já pré-definido. Se o laboratório ficar fora de controle é automaticamente ativado o controle de segurança.

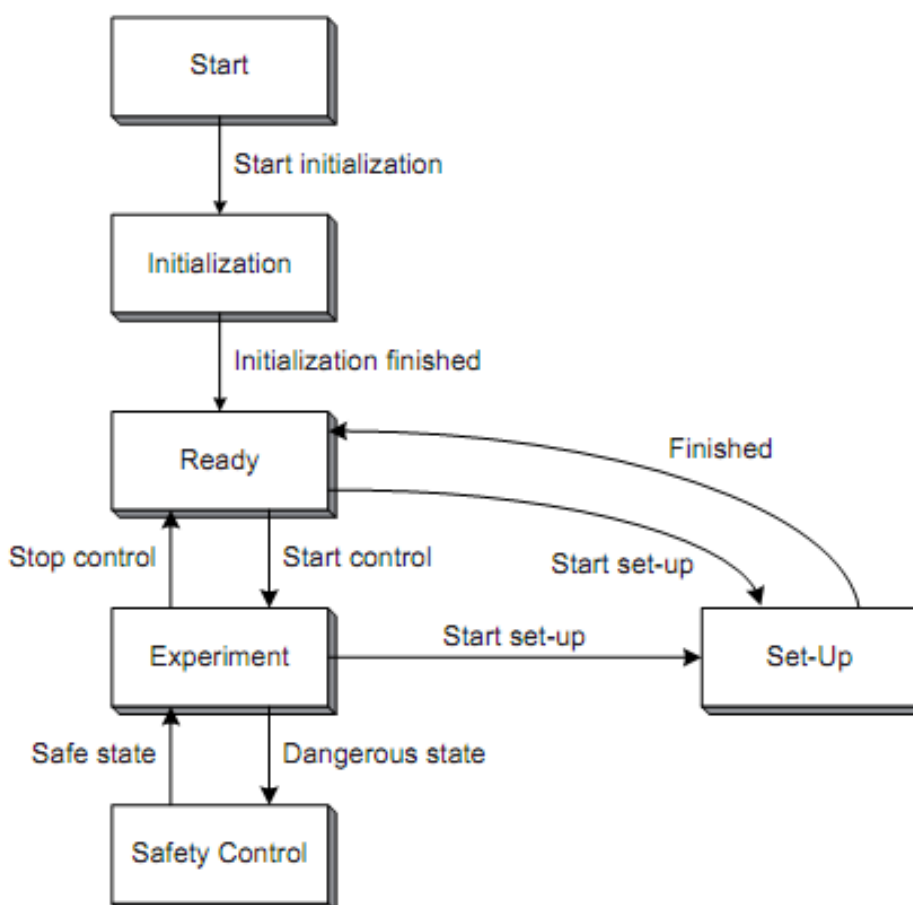


Figura 3. 1 Controle mestre do laboratório (MASÁR *et al*, 2011)

Essas fases do controle mestre podem ser incorporadas em qualquer experiência em particular.

O autor divide ferramentas e componentes de software usados em laboratórios em três grupos:

- Set-up da experimentação. Nesse grupo inclui design do algoritmo de controle em Matlab/Simulink/xPC Target.
- Componentes do servidor. Grupo que interconecta o cliente ao experimento.
- Componentes do cliente. Grupo tem componentes para operar e visualizar experimento.

A modificação de um laboratório existente envolve as seguintes etapas:

- Design o modelo experimental no Simulink.
- Preencher estrutura de inicialização do servidor com parâmetros do novo experimento.
- Modificar o GUI para acesso do laboratório remoto.

Usando esse conceito em um framework foi desenvolvido um caso de estudo de três laboratórios remotos: controle de guindaste, DC motor e termo-optical, como pode ser visto na Figura 3.2.

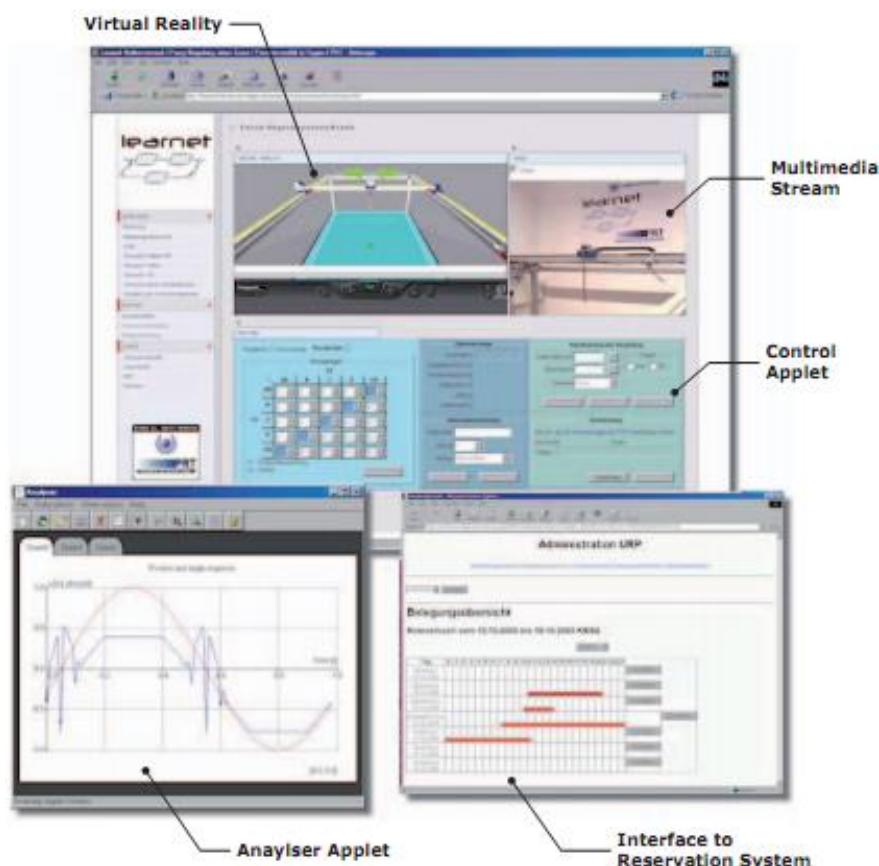


Figura 3. 2 GUI laboratório controle de guindaste (MASÁR *et al*, 2011)

Com essa abordagem ficou mais fácil configurar novas plantas experimentais, como foi apresentado nesse caso de uso.

2.9 CONFIGURAÇÕES DISTRIBUÍDAS DE AMBIENTES DE EXPERIMENTAÇÃO

O grupo de pesquisa WebLab-Deusto (WEBLAB-DEUSTO, 2012) da Universidade de Deusto desenvolveu um projeto chamado WebLab-Box de autoria de García-Zubia (GARCÍA-ZUBIA *et al*, 2009a), (GARCÍA-ZUBIA *et al*, 2009b) e (GARCÍA-ZUBIA *et al*, 2010), onde o principal objetivo é facilitar implantação de experiências remotas de eletrônica através de uma plataforma que fornece instrumentos necessários como web câmera, computador digital/analogico embutido e experiências sugeridas. A plataforma de hardware pode ser facilmente adaptada para experimentos de micro controladores em qualquer instituição de ensino. O WebLab-Box, mostrado na Figura 3.3, foi premiado como melhor ferramenta educacional na conferência ICELIE 2009 (AWARD ICELIE, 2009).

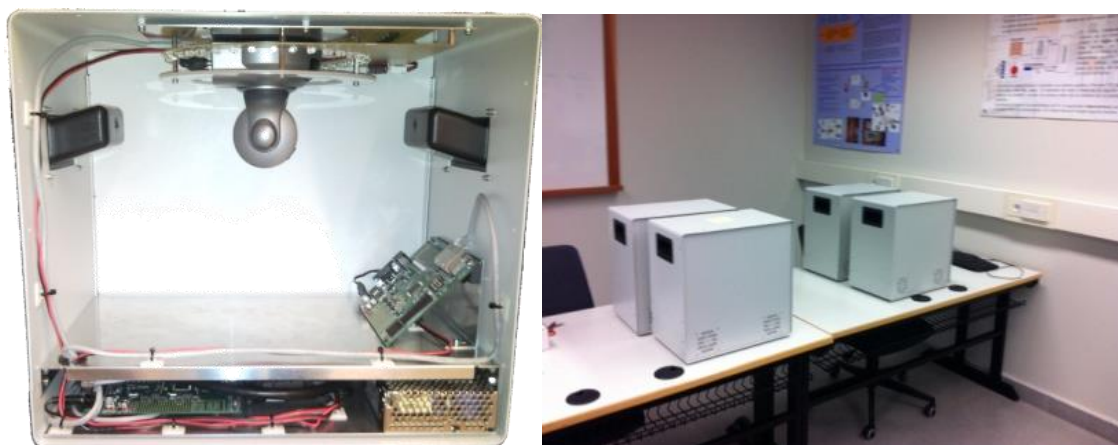


Figura 3. 3 WebLab-Box (GARCÍA-ZUBIA *et al*, 2010)

A WebLab-Box tem aplicativo de operação próprio, o WebLab-Deusto PIC, e também é integrado a plataforma WebLab Deusto, o WebLab-Deusto PLD&FPGA.

As principais contribuições desse projeto para laboratórios remotos foram melhorias nas características de adaptabilidade, estabilidade e extensibilidade.

2.10 RESOLVENDO PROBLEMA DE ALOCAÇÃO DE TAREFAS

O problema de alocação de tarefas para sistemas multi-robóticos, do inglês *Multi-robot task allocation* (ou MRTA), tem recebido, nos últimas décadas, um interesse significativo e crescente na comunidade de pesquisa. Muitos pesquisadores tem se feito a seguinte pergunta: "qual robô deve executar qual tarefa?". E esta questão deve ser respondida de forma convincente, mesmo que para sistemas multi-robóticos relativamente simples. A importância da atribuição de tarefas cresce à medida em que a complexidade do sistema em estudo cresce, em tamanho e capacidade.

O problema de atribuição de tarefas robóticas têm várias configurações. Gerkey (GERKEY 2003a) (GERKEY & MATARIC, 2003) define uma taxonomia onde são adotados três campos que serão utilizados na descrição de problemas de alocação de tarefas:

- *Single-taskrobots (ST) X Multi-taskrobots (MT)*: robôs de tarefa única vs. robôs multi-tarefa, ou seja, um robô é capaz de executar uma única tarefa ou mais de uma tarefa ao mesmo tempo;
- *Single-robottasks (SR) X Multi-robottasks (MR)*: tarefas de robô único vs. tarefas multi-robô, ou seja, cada tarefa requer exatamente um robô para ser executada ou algumas tarefas requerem mais de um robô;
- *Instantaneous assignment (IA) X Time-extendedassignment (TE)*: atribuição instantânea vs. atribuição estendida no tempo, ou seja, se instantaneamente alocarmos todos os robôs para as tarefas ou se teremos mais tarefas que robôs, por exemplo.

A variação mais simples da taxonomia de Gerkey (GERKEY, 2003), ST-SR-IA, pode ser representada como um problema de emparelhamento de grafo bipartido, que pode ser resolvido como um problema de atribuição ótima – *Assignment Problem*, sendo resolvido pelo Método Húngaro. Outras variações podem ser modeladas como problema de escalonamento, exemplo ST-SR-TE que pode ser descrito, pela taxonomia Brucker (BRUCKER, 2006), na classe de problema de escalonamento dado por:

$$R || \sum w_j C_j$$

A classe do problema é um superconjunto do caso mais simples de máquinas paralelas idênticas $P || \sum w_j C_j$. Problemas nesta classe são conhecidos por serem fortemente

NP-difícil. Podemos reduzir esse problema fazendo relaxamento de variáveis, chegando a $R \parallel \sum C_j$ que pode ser resolvido em $O(mn^3)$, (BRUNO, 1974), ou em $O(mn^2)$, (KUHN, 1955).

A principal contribuição do trabalho do autor foi definir configurações de atribuições de tarefas. Contribuição que ajudou a definir a configuração do SRM do LAR multi-configurável e montagem autônoma.

2.11 AMBIENTE ROBÓTICO EDUCACIONAL BASEADO EM SMA

Trabalho de Mestrado de Souza (SOUZA, 2011)(SOUZA & NETTO, 2010) que apresenta um arcabouço de um ambiente telerrobótico educacional baseado em SMA. O ambiente foi modelado usando a metodologia MaSE e implementado usando ambiente virtual de aprendizagem Moodle, firmware LeJOS para plataforma robótica Lego Mindstorms NXT, e por último, framework JADE para programar os agentes do SMA. O nome do ambiente é LARRE – Laboratório de Acesso Remoto de Robótica Educativa, interface inicial do LARRE conforme pode ser visto na Figura 3.4.



Figura 3. 4 GUI inicial do LARRE (SOUZA, 2011)

Com o ambiente LARRE concluído o autor respondeu às perguntas fundamentais de sua pesquisa: como disponibilizá-lo sobre Internet? Como definir comportamentos dos agentes apoiando as atividades do ambiente? Como auxiliar usuário através de troca de mensagens? Os serviços oferecidos pelo LARRE são cadastramento no sistema, agendamento para experimento e manipulação remota do robô móvel. O SMA LARRE é formado pelos agentes *Administrator*, *Scheduler*, *Assistant*, *User* e *Execute*. São oferecidos dez desafios de experimentação que são valorados em tempo por pontos que servem para qualificar e identificar perfil do usuário. O controle dos robôs acontece por eventos de clicar em botões virtuais que aparecem para usuário que acessa experimento em seu tempo agendado.

A principal contribuição da pesquisa foi criar um laboratório de acesso remoto de robótica usando uma camada SMA, viabilidade confirmada com implementação e testes.

2.12 CONCLUSÕES DO CAPÍTULO

Neste capítulo, apresenta-se a revisão bibliográfica, na qual se fez uma discussão de conceitos, definições e ferramentas que serviram de base para o desenvolvimento desta dissertação, sendo fundamentais, principalmente, para fazer um levantamento do estado da arte do objeto de estudo e identificar o problema definido no capítulo 1, caracterizado preliminarmente elementos para a solução para o problema. Além disso, foram elaborados resumos das ferramentas utilizadas na fase de codificação do Laboratório Acesso Remoto baseado em Sistema Multiagente e com multi-configuração e montagem autônoma de cenários.

Neste capítulo fez-se também um estudo dos trabalhos correlatos que servem de base teórica para uma abordagem que utiliza Sistema Multiagente em Laboratório de Acesso Remoto multi-configurável e montagem autônoma de seus cenários. Desta forma, estruturou-se este capítulo nas seguintes seções:

Na Seção 3.1 foi apresentado o framework que auxiliou na organização de construção ou modificação de laboratórios remotos, valendo salientar que esse trabalho de desenvolvimento é custoso, visto que para modificar um laboratório devem-se modificar todas as fases de operação do laboratório. Para fazer a devida configuração é importante que todo laboratório seja paralisado, e assim, realizar as alterações nas fases de operação, pois essas

modificações não são automatizadas, e sim realizadas manualmente pelo administrador. Por fim, observou-se a possibilidade de execução de um laboratório por vez, mesmo com essa proposta de fácil configuração.

Na Seção 3.2 foi apresentado uma plataforma chamada WebLab-Box que tem como principal objetivo facilitar a implantação de experiências remotas da área de eletrônica em outras universidades. A plataforma permite configurar vários experimentos da área de eletrônica sobre microcontrolador, ou seja, tem fácil adaptação dos experimentos, ainda, a plataforma permite criar extensões do experimento. Pela integração da plataforma com WebLab-Deusto PLD&FPGA os experimentos têm localização transparente, esse problema de localização foi explicado no capítulo 2. A abordagem do WebLab-Box permite uma configuração distribuída do ambiente de experimentação da área de eletrônica, ou seja, pode-se construir experiências laboratoriais em cada uma das WebLab-Box, em seguida, pode ser realizado uma combinação de WebLab-Box's para outro experimento. A limitação do WebLab-Box é a mesma da apresentada para o framework anterior, a adaptação e a implantação do cenário experimental é manual, além do mais, o administrador deve projetar as adaptações e implementar em hardware.

Na Seção 3.3 foi explicada a taxonomia que define uma organização para classificar um grupo de robôs de acordo com alocação de tarefa a eles ou robô alocado a tarefas, outro critério usado nessa classificação é o espaço temporal da quantidade de tarefas por robô. A classificação é importante para limitar o problema de configuração de ambiente utilizando tarefas distribuídas. Para a implementação que ilustra o uso de Sistema Multiagente na montagem de cenários é encaixado na configuração ST-SR-IA, que significa: um robô é capaz de executar uma única tarefa, cada tarefa requer exatamente um robô para ser executada e instantaneamente alocaremos todos os robôs para as tarefas, respectivamente para cada campo da taxonomia.

Na Seção 3.4 foi apresentado Laboratório de Acesso Remoto de robótica educacional, chamado LARRE. O processo desenvolvido começa com modelagem, passa pelas implementações e termina em as devidas conclusões dos testes. Modelagem usa uma metodologia MaSE, com implementação em JADE e LeJOS e os testes são realizados nos casos de uso, onde as funcionalidades modeladas são verificadas no sistema implementado. A modelagem e implementação do LARRE não contempla o uso de agentes para configurar e montar cenários de experimentação, o motivo é que o Sistema Multiagente foi elaborado com

foco na teleoperação robótica, então é o administrador do LARRE que monta o cenário teleoperado.

Os trabalhos correlacionados apresentam o estado da arte de Laboratórios de Acesso Remoto em configuração de cenários e indícios de como melhorar os Laboratórios de Acesso Remoto usando Sistema Multiagente. O único trabalho encontrado que usa abordagem Sistema Multiagente para Laboratório de Acesso Remoto foi o trabalho de Souza com o LARRE (SOUZA, 2011) (SOUZA & NETTO, 2010). Na pesquisa bibliográfica realizada não foi identificado Laboratório de Acesso Remoto que usa agentes robóticos autônomos para montagens de cenários.

3 LAR MULTI CONFIGURÁVEL E MONTAGEM AUTÔNOMA

Neste capítulo são apresentados os métodos utilizados no desenvolvimento do sistema inteligente com configuração e montagem de cenários para LAR. Primeiro é explicado em detalhes os agentes na arquitetura de software. Em seguida, é apresentada a arquitetura de hardware do sistema, que é o ambiente de controle robótico móvel. A seguir, é mostrada a modelagem em Rede de Petri e metodologia O-MaSE com o processo de troca de mensagens pelos agentes para configurar e montar cenários. Nesse capítulo é mostrado também a implementação de prova de conceito

3.1 ARQUITETURA DE SOFTWARE DO SISTEMA

A arquitetura geral do sistema é apresentada na Figura 4.1 A organização de agentes é dividida em três categorias: atores externos e agentes de interface em (1), agentes de gerenciamento do LAR em (2) e os agentes modificadores do ambiente laboratorial real em (3). As setas indicam o fluxo de informações por mensagens entre os agentes. Para atingir o objetivo da organização de agentes cada uma das metas dos agentes deve ser cumprida. São explicadas a seguir as metas dos agentes e atores externos.

A categoria de atores externos e agentes de interface (1) é formada por:

- *Usuário Estudante*: escolhe um cenário para ser experimentado em determinado horário. Depois aguarda informe de montagem do cenário.

- *Usuário Professor*: pode modificar ou criar um cenário de experimentação.

Depois aguarda informe da modificação do cenário.

- *Agente Estudante*: serve de interface principal para funcionalidades do Usuário Estudante. Nessa interface o estudante escolhe e solicita um cenário em um horário. Essa solicitação é enviada para Agente Administrador, que também aguarda informe de montagem do cenário.

- *Agente Professor*: serve de interface principal para funcionalidades do Usuário Professor. Nessa interface o professor cria ou modifica cenários experimentais. A interface

acompanha e verifica essa funcionalidade informando o Usuário Professor da validação das modificações.

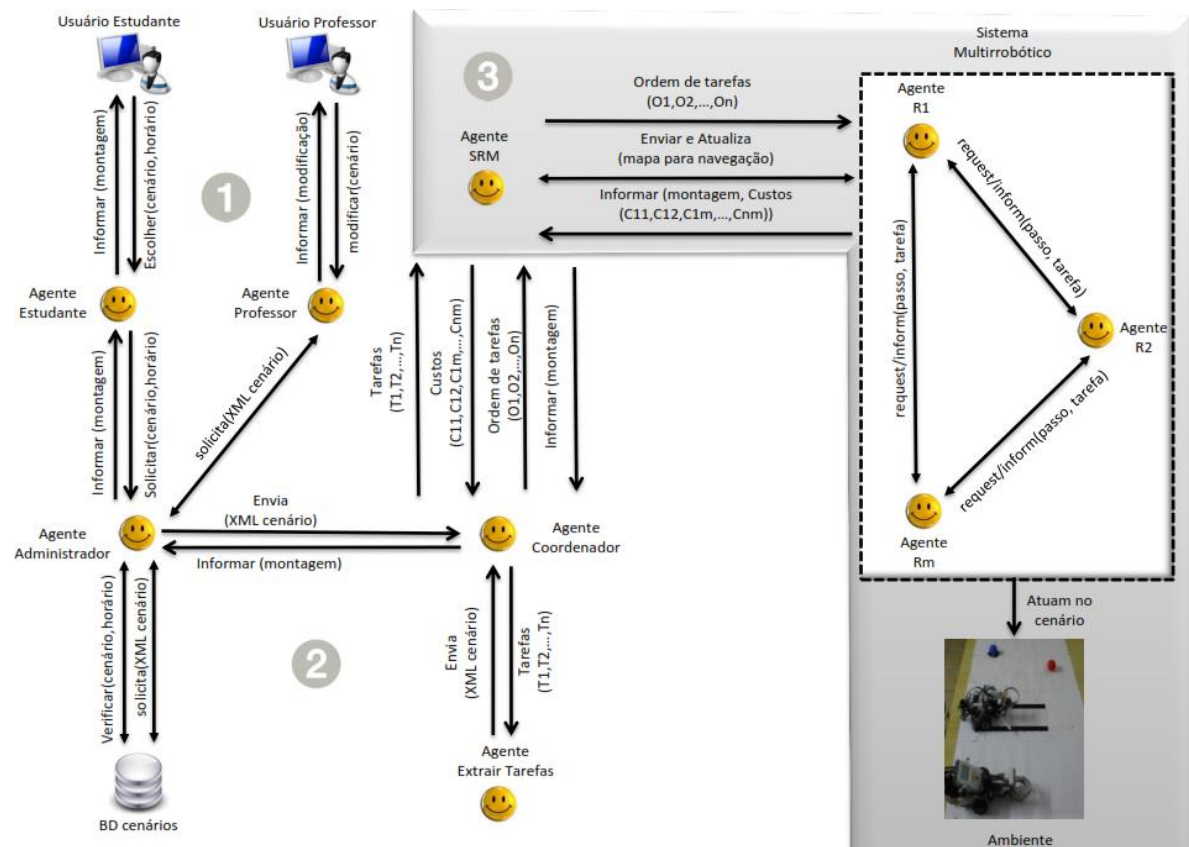


Figura 4. 1 Arquitetura geral do sistema

A categoria de agentes de gerenciamento de LAR (2) é formada pelos:

- *Agente Administrador*: verifica com DB cenário possibilidade de agendamento de cenário para horário escolhido pelo estudante, em seguida, caso agendamento seja realizado com sucesso, o administrador solicita o código XML do cenário escolhido, então BD Cenário responde com XML do cenário. Por último, informa Agente estudante sobre a montagem.
- *Agente Coordenador*: é responsável por solicitar extração de tarefas do código XML cenário para o *Agente Extrair Tarefas* e executar algoritmo de alocação de tarefas extraídas para os robôs do sistema robótico. O Agente Coordenador também aguarda informe de montagem do cenário.

A categoria de agentes modificadores do ambiente laboratorial real (3) é formada por:

- *Agente SRM (Sistema Robótico Móvel)*: encaminha as ordens de tarefas para os robôs. Ainda, compartilha um mapa de navegação com Sistema Multirrobótico. O cenário é montado quando todas as tarefas são realizadas pelos robôs, informando novos custos para aquelas tarefas que já foram realizadas. O Agente SRM informa para o Agente Coordenador os custos dos robôs para cada tarefa. A tabela de custo robô-tarefa do Agente SRM é atualizada periodicamente.

- *Sistema Multirrobótico*: é responsável por atuar no cenário. Para acontecer isso de forma segura os agentes robóticos trocam mensagem solicitando autorização para realizar passos das tarefas que podem levar a situações indesejadas como colisões.

3.1.1 Funcionamento da extração de tarefas

A extração de tarefas do arquivo XML cenário é realizada pelo Agente Extrair Tarefas. Para entender como funciona a extração dessas tarefas vejamos o exemplo do código XML cenário da Figura 4.2:

```

1      <?xml version="1.0" encoding="UTF-8"?>
2      <!DOCTYPE cenario SYSTEM "C:\Users\marckson\Desktop\CENARIO_EDITOR.dtd">
3      <cenario id="1" quadrante="1">
4          <objetos>
5              <objeto id="0" nome="bola">
6                  <formato geometria="esfera" diametro="48"/>
7                  <ponto x="700" y="0"/>
8              </objeto>
9              <objeto id="1" nome="bola">
10                 <formato geometria="esfera" diametro="50"/>
11                 <ponto x="700" y="350"/>
12             </objeto>
13         </objetos>
14         <entradas>
15         </entradas>
16         <saidas>
17         </saidas>
18     </cenario>

```

Figura 4. 2 Arquivo XML cenário

O código XML cenário é a configuração, disposição de objetos e equipamentos, do cenário. Para Figura 4.2 o cenário são os objetos em pontos (700,0) e (700,350). O uso de XML para descrever experimento é viável tecnologicamente, como foi verificado no projeto

FASE (BEZERRA *et al*, 2004). Para usar o padrão de escrita XML é necessário que se defina uma estrutura de metadados baseada em XML, assim criando uma nova estrutura e utilizando a mesma “filosofia” de *parsers* do XML (BEZERRA *et al*, 2004).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--DTD generated by XMLSpy v2013 sp1 (http://www.altova.com)-->
3 <!ELEMENT saidas (#PCDATA)>
4 <!ELEMENT ponto EMPTY>
5 <!ATTLIST ponto
6   x CDATA #FIXED "700"
7   y (350 | 0) #REQUIRED
8 >
9 <!ELEMENT objetos ((objeto+))>
10 <!ELEMENT objeto ((formato, ponto))>
11 <!ATTLIST objeto
12   id (1 | 0) #REQUIRED
13   nome CDATA #FIXED "bola"
14 >
15 <!ELEMENT formato EMPTY>
16 <!ATTLIST formato
17   diametro (50 | 48) #REQUIRED
18   geometria CDATA #FIXED "esfera"
19 >
20 <!ELEMENT entradas (#PCDATA)>
21 <!ELEMENT cenario ((objetos, entradas, saidas))>
22 <!ATTLIST cenario
23   id CDATA #FIXED "1"
24   quadrante CDATA #FIXED "1"
25 >

```

Figura 4. 3 Arquivo DTD cenário

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--W3C Schema generated by XMLSpy v2013 sp1 (http://www.altova.com)-->
3 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4   <xs:element name="saidas" type="xs:string"/>
5   <xs:element name="ponto">
6     <xs:complexType>
7       <xs:attribute name="y" use="required">
8         <xs:simpleType>
9           <xs:restriction base="xs:short">
10            <xs:enumeration value="0"/>
11            <xs:enumeration value="350"/>
12          </xs:restriction>
13        </xs:simpleType>
14      </xs:attribute>
15      <xs:attribute name="x" use="required">
16        <xs:simpleType>
17          <xs:restriction base="xs:short">
18            <xs:enumeration value="700"/>
19          </xs:restriction>
20        </xs:simpleType>
21      </xs:attribute>
22    </xs:complexType>
23  </xs:element>
24 </xs:schema>

```

```

92 | | | | | </xs:restriction>
93 | | | | | </xs:simpleType>
94 | | | | | </xs:attribute>
95 | | | | | </xs:complexType>
96 | | | | | </xs:element>
97 | | | | | </xs:schema>

```

Figura 4. 4 Arquivo XML Schema cenário

A principal construção do arquivo XML que representa o cenário é o conjunto de *tags* dentro da construção `<objetos></objetos>`, cada uma delas representa um objeto a ser posicionado. Para garantir a conformidade do documento XML é utilizado Definição de Tipos de Documentos (ou DTD), mostrado na Figura 4.3. O DTD é limitado por especificar um documento por gramática *Extended Backus-Naur Form* (ou EBNF) e não por sintaxe XML (GAUTHIR, 2007) (DEITEL & DEITEL, 2003]. Por causa de limitações do DTD, atualmente, a maioria das aplicações usam XML Schema. Neste trabalho foi usado XML Schema, como pode ser visto na Figura 4.4, gerado pelo Altova XML SPY (XML SPY, 2013). O Agente Extrair Tarefas conhece o padrão do XML, então ao encontrar uma determinada *tag* extrai a informação e cria um novo arquivo de tarefas, conforme mostrado na Figura 4.5.

```

1 [b1,350,310]: []
2 [b2,480,985]: []

```

Figura 4. 5 Arquivo de tarefas

Cada linha do arquivo de tarefas representa uma tarefa com a seguinte sintaxe: `[<objeto id>,<pontox>,<pontoy>]:[<requisito tarefa id>]`. Os requisitos para cada tarefa são retirados do arquivo XML cenário, construído pelo professor. O próximo passo é atribuição dessas tarefas para os agentes robóticos, explicado na Seção 4.1.2.

3.1.2 O Papel do Multi Robot Task Allocation na montagem do cenário

Depois de receber todas as tarefas, o Agente Coordenador distribui as tarefas para os robôs do sistema. Como fazer a distribuição de tarefas? Aqui é discutida a importância do papel do Multi Robot Task Allocation (ou MRTA) na montagem do cenário. É utilizada a taxonomia de Gerkey para problemas de alocação de tarefas robóticas. A taxonomia trabalha com várias configurações do MRTA. A configuração robótica “ST-SR-IA” apresentado na Seção 3.3,

onde ST é quando um robô é capaz de executar uma única tarefa por vez, SR é quando cada tarefa requer exatamente um robô para ser executada, IA é quando instantaneamente alocarmos todos os robôs para as tarefas, pode ser representado como um problema de emparelhamento de grafo bipartido, podendo ser resolvido como um problema de atribuição ótima pelo Método Húngaro. Para um entendimento melhor considere o exemplo a seguir.

Existem 4 robôs e 3 tarefas. Robôs com especialidades específicas efetuam determinadas tarefas com melhor desempenho do que outros. Como alocar cada tarefa a um robô de forma que cada tarefa seja efetuada somente por um robô e o desempenho total seja maximizada?

Exemplo 1: A Tabela 4.1 exemplifica um conjunto de tarefas e robôs com respectivos desempenhos em relação a cada tarefa.

Tabela 4. 1 Tarefas e robôs

	Robô Carro	Robô Gancho	Robô Sonar	Robô Humanóide
T1	10	4	7	7
T2	6	10	3	5
T3	6	3	10	8

Sendo:

T1 - Percorrer um caminho seguindo uma tarja preta;

T2 - Mover objetos de A até B;

T3 - Mapear uma região.

Uma maneira de resolver é utilizando o Método Húngaro. Como este método não é aplicado em matriz não quadrada, então, é acrescentada uma linha com zeros correspondente a uma tarefa inexistente, como pode ser visto na Tabela 4.2. A linha fictícia não interfere em atribuir corretamente às tarefas, pois, terá custo zero, por exemplo, T4 - ficar parado.

Tabela 4. 2 Tarefa (T4) fictícia adicionada

	Robô Carro	Robô Gancho	Robô Sonar	Robô Humanóide
T1	10	4	7	7
T2	6	10	3	5
T3	6	3	10	8
T4	0	0	0	0

Aplicando o Método Húngaro, encontra-se a solução ótima, indicada na Tabela 4.3 em negrito:

Tabela 4. 3 Atribuição ótima de 4 tarefas

	Robô Carro	Robô Gancho	Robô Sonar	Robô Humanóide
T1	10	4	7	7
T2	6	10	3	5
T3	6	3	10	8
T4	0	0	0	0

Então 10 é o valor de desempenho máximo que pode ser alcançado. Foi apresentado o caso em que o número de robôs é maior que o número de tarefas, e quando for ao contrário, o número de tarefas maior que o de robôs, como reduzir para uma matriz quadrada? A resposta é apresentada na Seção 4.1.5 Cooperação entre agentes robóticos. Os valores da matriz são calculados conforme apresentado na Seção 4.1.5.

3.1.3 Estratégia para evitar colisões

Na parte da arquitetura geral chamada Sistema Multirrobótico, área em retângulo pontilhado na Figura 4.1, depois do Agente SRM distribuir as tarefas, cada robô possui uma tarefa ou lista de tarefas e o mapa de estados que é compartilhado pelo Agente SRM. As propriedades das tarefas são: robô responsável ou robôs responsáveis, pré-requisito e sequência de ações (ou passos) dessas tarefas. A propriedade robô responsável é quem recebe a tarefa do Agente SRM, a propriedade pré-requisito é um dos parâmetros do arquivo XML cenário que é extraída para o arquivo de tarefas, a propriedade passos da tarefa é codificada pelo agente robótico. A sequência de passos pertencentes a cada tarefa é dividida em passos concorrentes ou não, como mostrado na Tabela 4.4 para tarefa de posicionar objeto. Passos identificados durante construção do modelo RdP, apresentado na Seção 4.4.

Os passos para tarefa posicionar objeto são representados no fluxograma da Figura 4.6. Um passo pode modificar o mapa, por isso entre cada passo podem existir pontos de atualização do mapa. Os pontos são atividades críticas que devem ser realizadas de forma atômica. Somente pode ser executado um passo não concorrente por vez no tempo de execução dos passos das tarefas robóticas, assim evitando as colisões.

Tabela 4. 4 Passos para tarefa posicionar objeto

Passos não concorrente:	Passos concorrente:
Para quando os robôs móveis precisam de movimentação no ambiente para realizar o passo.	Para quando os robôs móveis não precisam de movimentação no ambiente para realizar o

	passo.
- Aproximar do objeto. - Descolar até ao ponto Objetivo. - Voltar ao ponto de partida.	- Capturar objeto. - Soltar objeto.

Os passos não concorrentes podem ser tratados para possibilitar a sua execução concorrente, para isso são usados os protocolos de pedido de autorização para navegação. Essa estratégia é apresentada na Seção 4.1.4 Planejando a trajetória.

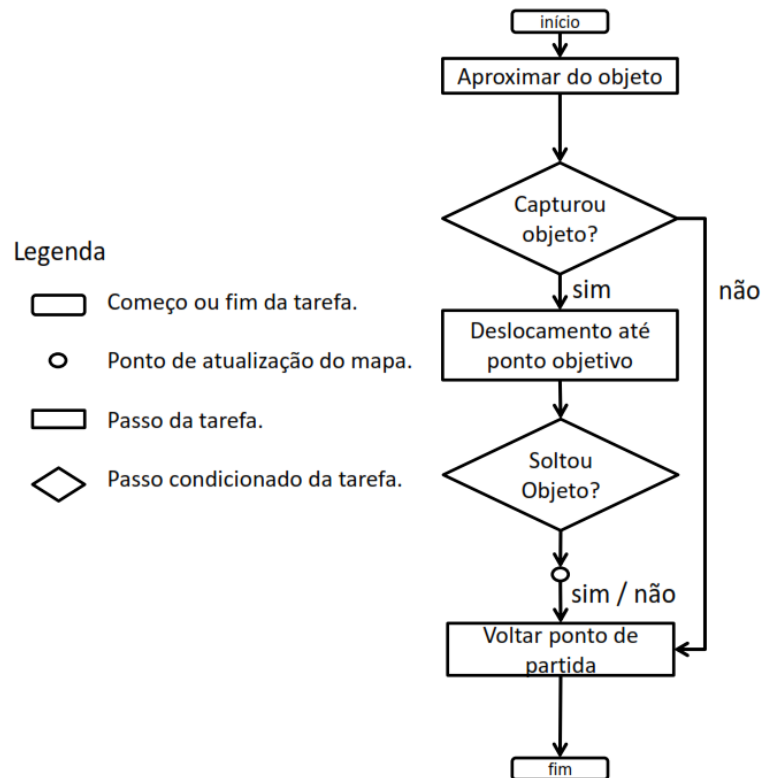


Figura 4. 6 Fluxograma dos passos da tarefa posicionar objeto

Esses passos servem de parâmetros para calcular os desempenhos de cada robô para cada uma das possíveis tarefas atribuídas, explicado na Seção 4.1.6.

3.1.4 Planejando a trajetória

O SRM não precisa construir o mapa, ou seja, os robôs não vão realizar a exploração autônoma, porque o SRM opera em um ambiente essencialmente estático, estruturado e, quase sempre conhecido, assim como robôs industriais que geralmente são operados com um mínimo de informações sensoriais. O mapa é estático até ser alterado pelo agente robótico,

então é necessário ter uma estratégia para modificação do mapa compartilhado por todos os agentes, garantindo fidelidade da representação real. Outra estratégia importante a ser construída é a de planejamento de rotas eficientes evitando colisões.

Em ambientes inteiramente conhecidos, o planejamento de rotas se realiza por meio de *geradores globais de trajetórias*, que usando de um mapa buscam encontrar o melhor caminho com um menor fator de custo. O custo pode ser influenciado pela transitabilidade, prioridade de circulação, densidade de obstáculos e outras características nas trajetórias (SECCHI, 2008). Para o SRM do LAR planejar o objetivo é encontrar um caminho livre de colisões. Quando o ambiente é completamente conhecido, o robô tem um mapa, o cálculo da rota é solucionado por um *algoritmo de busca gráfica* (SECCHI, 2008) (SILVA, 2008). O mapa global é compartilhado por todos os agentes robóticos móveis, por isso há necessidade de uma estratégia para modificação do mapa global.

O mapa baseado em mapa métrico por grade de ocupação e topológico, é mostrado na Figura 4.7. Métrico porque cada célula tem o tamanho máximo que um objeto (incluindo robô) pode chegar a ter no ambiente, que é 35 por 35 cm. Cada célula é representada por uma linha e coluna (Lin,Col). É também baseado em mapa topológico porque o deslocamento do robô deve ser feito em linha ou coluna de cada célula e para facilitar navegação nessa representação de mapa não pode ser feito na diagonal.

Segundo (SECCHI, 2008), mapas geométricos geralmente são usados em navegação baseada em uma planificação com base em coordenadas geométricas de trajetórias pré-fixadas. E mapas topológicos geralmente são usados em navegação baseada em objetivos sensoriais associados a padrões de reação.

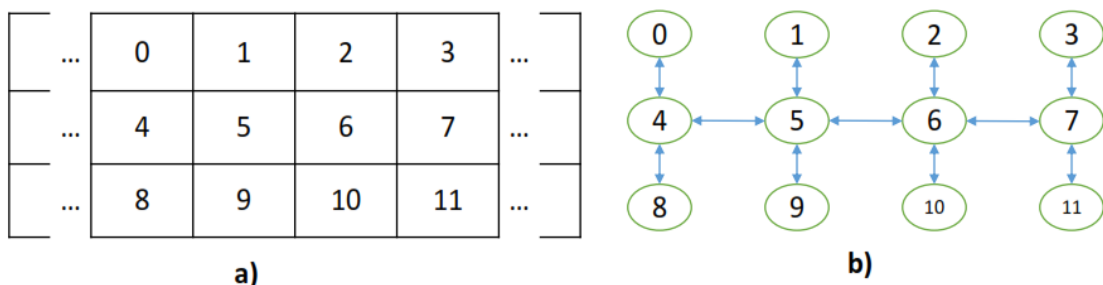


Figura 4. 7 Representação em mapa a) métrico por grade de ocupação e b) topológico

O algoritmo de Dijkstra é usado para realizar a busca gráfica no grafo da Figura 4.7. A estratégia de cálculo de trajetória e mapeamento é explicada no exemplo a seguir, na Figura 4.8.

Suponhamos que o robô 1 (R1) está no vértice 0 e o robô 2 (R2) no vértice 1, e que os objetos 1 e 2 (O1 e O2) em vértices 2 e 3, respectivamente. O cenário a ser montado é posicionar O1 e O2 nos vértices 10 e 11, respectivamente.

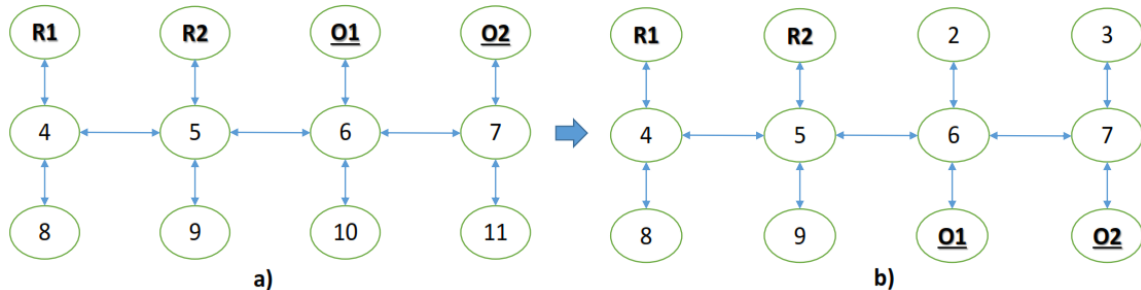


Figura 4. 8 Representação em mapa a) antes da montagem e b) depois da montagem

Seguem as questões: quais os caminhos possíveis para chegar até aos objetos, qual a caminho mais curto? Como levá-los para vértices objetivos evitando colisão com outros robôs em execução ou objetos já posicionados?

A primeira pergunta é respondida com uso do algoritmo de Dijkstra, que resolve o problema de caminho mais curto em um grafo com arestas de pesos não negativos dirigido ou não. A segunda pergunta é respondida usando uma estratégia de modificação no mapa, com modificação na direção do vértice.

Suponha para o exemplo dado, o algoritmo de atribuição de tarefas robóticas foi executado com sucesso e as atribuições são: robô 1 posiciona objeto 1 e robô 2 posiciona objeto 2. Antes de executar o algoritmo de Dijkstra é modificado o mapa original da seguinte forma: vértice onde tem objetos diferentes do vértice destino tem suas arestas direcionadas para fora de si, assim evitando cálculo de trajetória que possa levar a futuras colisões contra objetos. Então a modificação fica como mostrado em vermelho (seta de espessura maior) na Figura 4.9 e a trajetória calculada é mostrado em aresta em negrito (seta de espessura média).

Para capturar objeto o robô deve chegar até o objeto, o caminho para R1 é $0 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 2$ e para R2 é $1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 3$. Para evitar colisão entre robôs durante execução de suas trajetórias os agentes robóticos conversam para saber se o próximo passo da trajetória deles é seguro, ou seja, se não tem um robô na sua frente. Nessa conversa que acontece a colaboração robótica, apresentada na Seção 5.1.2. A troca de mensagem usa a performativa do tipo pedido do FIPA-ACL, a *QUERY-IF*. A estratégia funciona, pois os robôs trabalham em velocidades iguais. Na Figura 4.9 é apresentado o deslocamento até os objetos.

Na Figura 4.10 é apresentado o deslocamento dos objetos até pontos objetivos e na Figura 4.11 é apresentada a volta dos robôs para seus pontos de partida, 0 e 1 consecutivamente.

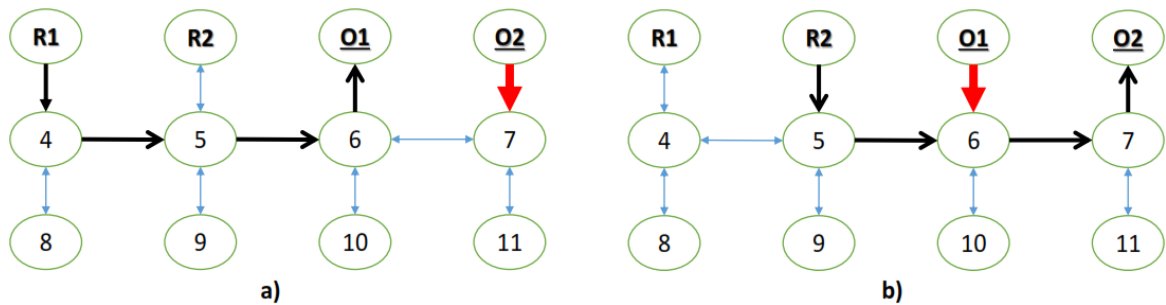


Figura 4.9 Modificação do mapa antes da execução do Dijkstra a) para robô 1 e b) para robô 2

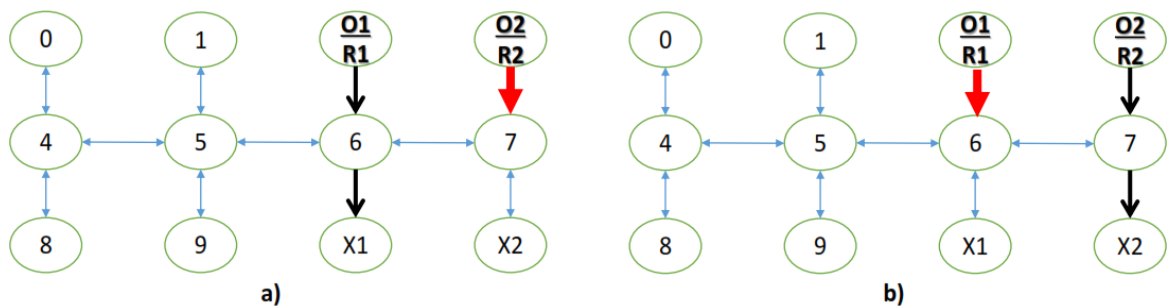


Figura 4.10 Objeto levado até ponto objetivo a) do robô 1 e b) do robô 2

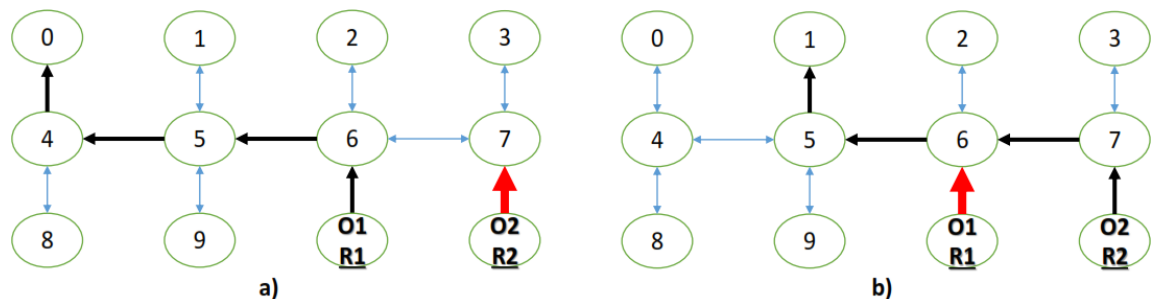


Figura 4.11 Voltar para ponto de partida com a) robô 1 e b) robô 2

3.1.5 Cooperação entre agentes robóticos

Na matriz de tarefas por robô, Tabela 4.1, quando o número de tarefas é menor que o de robôs basta acrescentar linhas de zeros correspondentes à tarefas inexistentes até que a matriz torne-se quadrática, assim podendo resolvê-la pelo Método Húngaro, pois robôs podem realizar tarefa fictícia. Entretanto, quando o número de robôs é menor que o número de tarefas, não pode ser criada uma coluna de robô fictícia, pois será atribuída uma tarefa não fictícia a um

robô fictício. Nesta situação, pela configuração apresentada na Seção 3.3, muda de ST-SR-IA para ST-SR-TE, TE é quando não alocaremos instantaneamente todos os robôs para as tarefas, pois teremos mais tarefas que robôs. Então a solução é criar uma coluna de grupo de robôs, Tabela 4.5.

Tabela 4. 5 Grupo de robôs (G1) adicionada

	Robô Carro	Robô Gancho	G1
T1	10	4	9
T2	6	10	8
T3	6	3	8

Sendo:

T1 - Percorrer um caminho seguindo uma tarja preta;

T2 - Mover objetos de A até B;

T3 - Mapear uma região;

G1 - Grupo 1 de robôs formado por robôs carro e gancho.

Os valores da matriz são uma métrica de estimativa de desempenho que cada robô tem para cada uma das tarefas que podem ser atribuídas a ele. Essa métrica pode variar desde uma simples avaliação de sensor até complexas avaliações baseadas em planejamentos, como os projetos M+ (BOTELHO, 1998) (BOTELHO & ALAMI, 1999) e arquitetura Free Market (DIAS & STENTZ, 2000). Como o foco desse trabalho não é uma métrica robusta, a função de custo é calculada pela soma de todos os desempenhos de cada um dos passos necessários para executar a tarefa por completo, representada função a seguir.

$$D_{x,y} = \sum_{n=1}^{n=m} P_n = P_1 + P_2 + \dots + P_m$$

Onde $D_{x,y}$ é o desempenho do robô x para tarefa y , sendo tarefa y é formada pelos m passos, logo, igual $P_1 + P_2 + \dots + P_m$. O valor de cada passo, nesse primeiro momento, foi calculado de forma manual com experimentos realizados de forma separada em cada um dos robôs, conforme mostrado na Seção 5.1.1. Em versões futuras esses valores serão calculados pelos próprios agentes robóticos.

3.2 AMBIENTE OPERACIONAL

Na arquitetura de hardware é usado plataforma de robótica educacional Lego MindStorms 1.0 e 2.0. A Figura 4.12 mostra as arquiteturas montadas, que são modificações das arquiteturas Ball Hunter e Tribot encontrados na *nxtprograms.com*. A arquitetura Tribot tem modificações para adaptação de Ball Caster encontrados na *Community Lego MindStorms*.

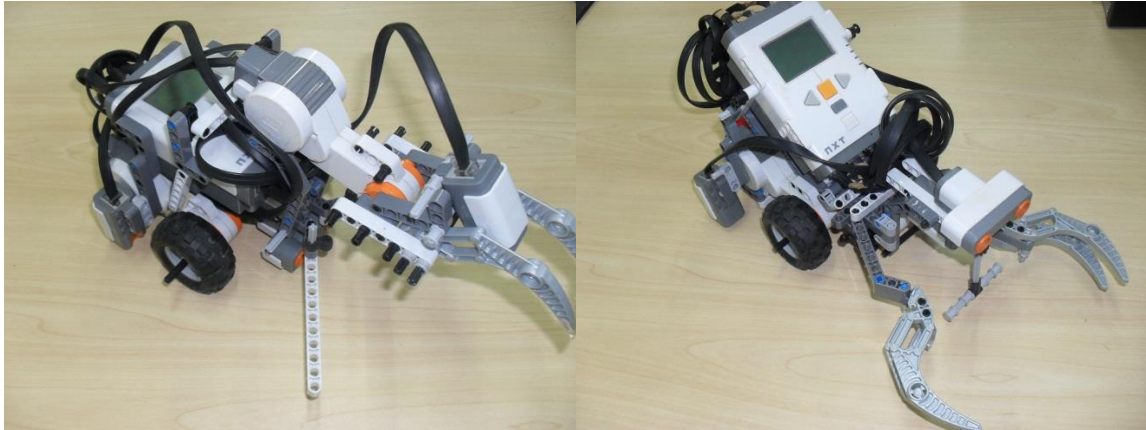


Figura 4. 12 Modelos de robôs móveis

As arquiteturas são robôs móveis de acionamento diferencial, uma roda ou bola castor e com garras mecânicas. Do lado esquerdo da Figura 4.12, chamado de Robô Garra Simples, tem garra de um dedo com contato em forma de “V” onde acontece a retenção da bola. A arquitetura do lado direito da Figura 4.12, chamado Robô Garra Dupla, tem garra de dois dedos onde acontece a retenção da bola, especificação baseada na norma ISO/DIS 14539. Todas as arquiteturas apresentadas tem um grau de capacidade de realizar as tarefas de pegar bola. A Figura 4.13 mostra os objetos que são capturados: uma bola sobre um suporte e um cilindro.



Figura 4. 13 Objetos de captura

São verificadas as relações entre sensoriamento, planejamento e acionamento no ambiente de controle. São usados sensores odométricos para fazer localização e navegação dos robôs com disposições em acionamento diferencial, junto aos robôs têm sensores de cor estrategicamente posicionados para ler os marcos referenciais localizados nos pontos de partida deles. Ainda, toda navegação tem um trajeto calculado em um mapa baseado em grade de ocupação não exploratório e representação topológica, assim, a "arquitetura" operacional do sistema é híbrida, seguindo classificação de Dudek (DUDEK & JENKIN, 2000).

3.3 MODELAGEM DO SISTEMA USANDO METODOLOGIA O-MASE

Usando a metodologia O-MaSE é explicado quais os agentes criados, como serão programados e como eles vão atuar no ambiente. O uso da metodologia O-MaSE é justificada por ser bastante usada e difundida em modelagem de sistema Multiagente. Ainda, o projeto O-MaSE têm a ferramenta aT que facilita design de SMA e permite geração de código do SMA construído (SOUZA 2010) (SOUZA, 2011) (WOOD *et al*, 2001) (WERNECK *et al*, 2011).

No desenvolvimento do SMA, o primeiro passo da modelagem em O-MaSE é criar o processo personalizado em O-MaSE para o SMA LAR. A metodologia O-MaSE define atividades e tarefas e não fases de projeto, porque existem várias maneiras de organizar as atividades e tarefas (DELOACH & GARCIA-OJEDA, 2010). O Engenheiro de SMA que adota ou cria sua organização em fases, pode seguir uma abordagem como, por exemplo, modelo em cascata. A organização de fases deve respeitar as dependências entre atividades que são definidas em diretrizes de construção do método (GARCIA-OJEDA *et al*, 2007). O processo de criação O-MaSE é formado por fases, interações, atividades, tarefas e modelos. Para o processo criado, como mostrado na Figura 4.14, que tem uma única fase, o início - *inception*, não há interações. As atividades são requisitos, análise e design -*requirements*, *analysis* e *design* e as tarefas são especificar requisitos, modelar objetivos, refinando objetivos, modelar interfaces da organização, modelar papéis, modelar classes de agentes, modelar protocolos e modelar planos – *specify requirements*, *model goals*, *goal refinement*, *model organization interfaces*, *model roles*, *model agent classes*, *model protocols* e *model*

plans. E o conjunto de modelos - *workproduct* produzidos em cada uma das tarefas agrupadas em atividades são explicados a seguir.

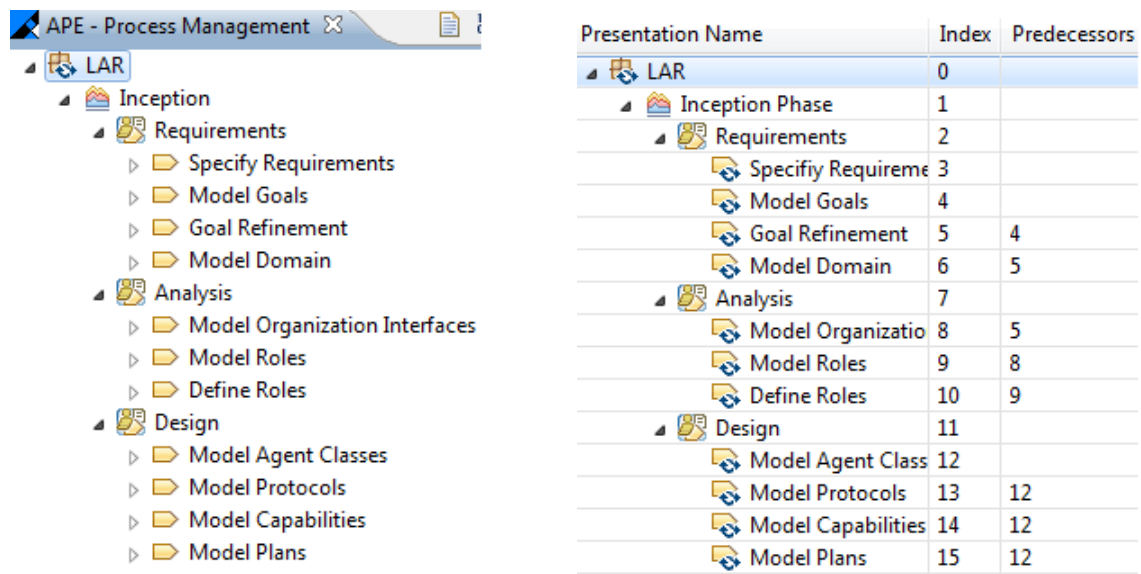


Figura 4. 14 Processo O-MaSE

Com as definições do levantamento dos requisitos do sistema - *requirementspec*, o ponto inicial foi criar modelo de objetivos- *goalmodel* em árvore hierárquica e depois refinando estes objetivos, gerando modelo de objetivo para sistemas dinâmicos (*GMoDS* - *GoalModel for Dynamic Systems*). Os modelos são produtos das tarefas especificar requisitos, modelar objetivos e refinando objetivos respectivamente, na atividade de requisitos - *requirements*. Com *GMoDS* completo definimos o modelo de organização - *organizationmodel* para capturar atores externos. Para criar modelo de papéis - *role model* são usados modelos *GMoDS*, de objetivos e de organização. Os modelos da atividade de análise criados nas tarefas modelar interfaces da organização e modelar papéis são transformados em modelos da atividade de design, onde essa precedência das tarefas é mostrada na Figura 4.14. Para criar o modelo de classes de agentes - *agentclassmodel* é usado modelo de papéis, onde papéis são agrupados em classes. Detalhes do agente e protocolos identificados no modelo de classes de agentes são refinados no modelo de protocolo - *protocolmodel* e modelo de plano de agente - *agentplanmodel*. Os protocolos dos agentes poderiam ser criados também depois de modelo de organização ou modelo de papéis.

O modelo de objetivos é definido para capturar os objetivos principais do SMA, e geralmente vem depois de definir o documento com requisitos do SMA. O modelo tem os objetivos organizados de forma hierárquica onde são usados condições AND/OR para definir

agrupamento obrigatório ou não de objetivos filhos realizados para realização do objetivo pai, condições que podem ser percebidas como generalização e decomposição UML respectivamente. Um exemplo do uso da condição AND no modelo da Figura 4.15, onde para realizar objetivo pai *Gerenciar LAR* primeiro tem que realizar os objetivos filhos *Pegar solicitação de cenário*, *Informar usuário*, *Montar cenário* e *Criar/Editar cenário*. Um exemplo do uso da condição OR no modelo da Figura 4.15, onde para realizar objetivo pai *Informar usuário* basta que um dos objetivos filhos *Informar falha no agendamento/montagem* ou *Informar sucesso no agendamento/montagem* seja realizado.

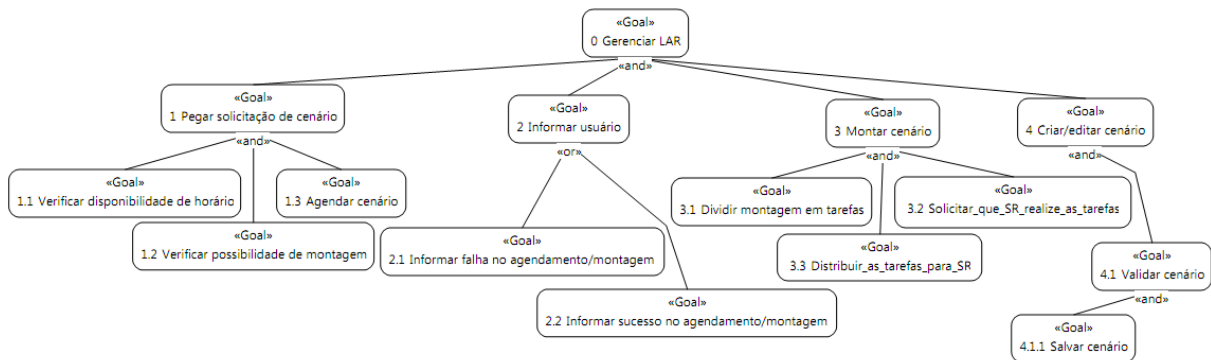


Figura 4. 15 Modelo de Objetivo

Para capturar o dinamismo inicial do SMA é refinado o modelo de objetivos com informações adicionais, gerando o novo modelo *GMoDS*. Além das condições AND/OR, o *GMoDS* aborda sequenciamento, instância e parametrização de objetivos. Eventos chamados *triggers* podem ocorrer durante operação do sistema. Na modelagem do *GMoDS* do LAR, na Figura 4.16, o objetivo *Solicitar que SR realize as tarefas* é preciso que antes seja definido e distribuído as tarefas, por isso é usado o parâmetro <<precedes>>. Outro exemplo desse tipo é o objetivo *Informar usuário* que <<precedes>> objetivo *Pegar solicitação de cenário* e objetivo *Montar cenário*, esse último objetivo é associado ao sequenciamento de objetivos e seus *triggers* *verificar(hora)*, *verificar(cenário)* e *montar(cenário, hora)*.

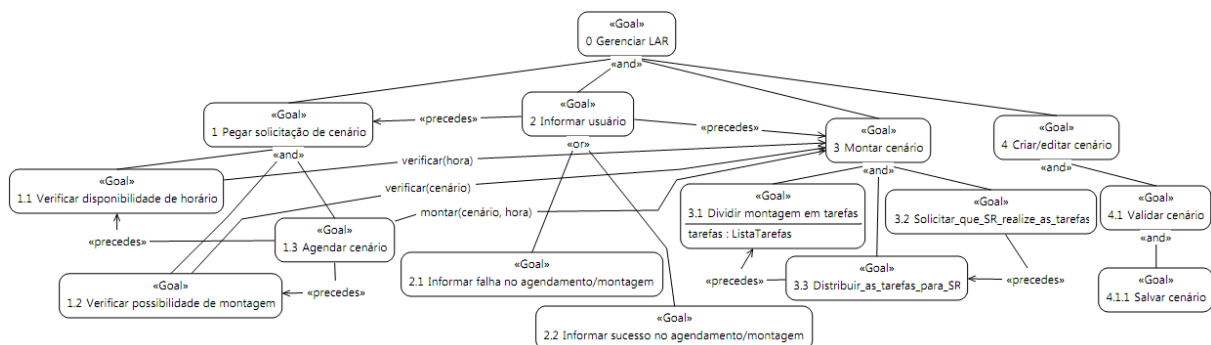


Figura 4. 16 GMoDS

O modelo de domínio tem objetivo de capturar os tipos de objetos, relações e comportamentos que define o domínio em que os agentes vão sentir e agir. A Figura 4.17 mostra o modelo de domínio para LAR.

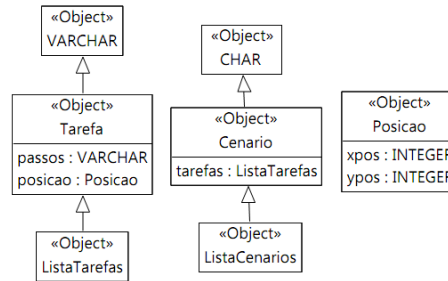


Figura 4. 17 Modelo de Domínio

O modelo de organização tem objetivo identificar interfaces do SMA com os atores externos. *GMoDS* precede o modelo de organização, ficando mais fácil identificar a interface organizacional. Para o LAR a interface organizacional é o próprio sistema LAR com os atores Professor, Estudante e Sistema Robótico (ou SR). A interação com agentes do SMA acontece por meio dos protocolos representados por setas, exemplo do ator Estudante que escolhe e solicita agendamento de cenário por meio do protocolo *escolheAgendaCenario*, Figura 4.18.

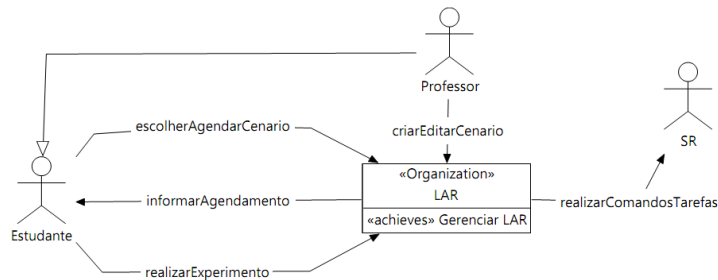


Figura 4. 18 Modelo de Organização

O objetivo do modelo de papéis é identificar os papéis do SMA e suas interações. O *GMoDS* e o modelo de organização são precedentes para esse modelo. Os atores e protocolos do modelo de organização são mapeados para modelo de papéis, assim como os objetivos do *GMoDS*, representados pelo uso pré-fixado de <<achieves>> nesse modelo. O modelo de papéis do LAR tem seis papéis: *Robô móvel*, *BD Agenda*, *Interface Professor*, *Interface Estudante*, *Coordenador Laboratório* e *Extrator tarefas*, como pode ser visto na Figura 4.19 na forma simplificada e em visão geral no Anexo B. O papel *Interface Professor* pode modificar os cenários experimentais que o ator estudante escolher usando o papel *Interface Estudante*. Para agendar o cenário escolhido pelo estudante o papel *BD Agenda* é o

responsável por agendar de maneira correta e respeitando as condições de limite e ordem do LAR. O papel *Coordenador Laboratório* fica responsável pela organização eficiente da montagem do cenário chegando a solicitar ao papel *Extrator Tarefas* o desmembramento da montagem do cenário em tarefas atômicas. O papel mais próximo do SR é o *Robô Móvel* que tem como principal objetivo enviar comandos de tarefas e aguardar respostas.

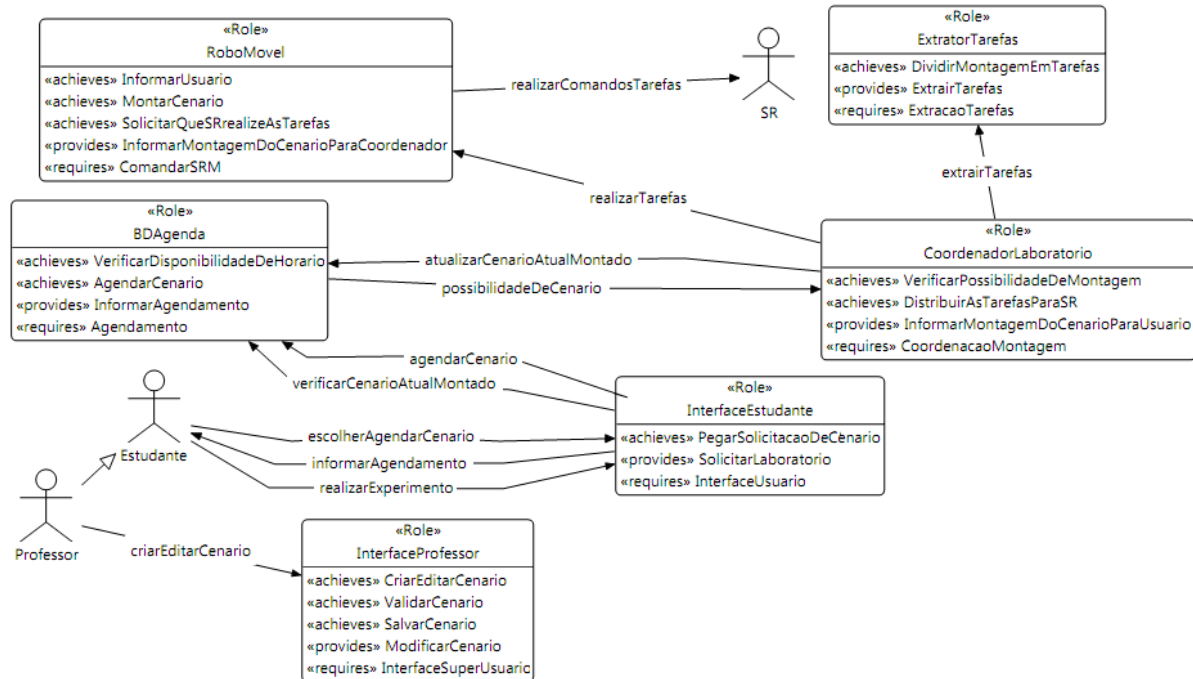


Figura 4. 19 Modelo de Papéis

O objetivo da tarefa *define roles* é definir comportamentos e capacidades para agente desempenhar um papel. A Tabela 4.6 descreve essa tarefa.

Tabela 4. 6 Modelagem da caracterização de experimentos usando tabela

Papel (Role)	Realiza (Achieves) objetivos	Exige (Required) capacidades	Plano (Plan)
InterfaceEstudante	PegarSolicitacaoDeCenario	InterfaceUsuario	SolicitarCenario
InterfaceProfessor	CriarEditarCenario ValidarCenario SalvarCenario	InterfaceSuperUsuario	ModificarCenario
RoboMovel	InformarUsuario MontarCenario SolicitarQueSRrealizeAsTarefas	ComandarSRM	ControlarSRM
BDAgenda	VerificarDisponibilidadeDeHorario AgendarCenario	Agendamento	AgendarCenario
CoordenadorLaboratorio	VerificarPossibilidadeDeMontagem DistribuirAsTarefasParaSR	CoordenacaoMontagem ExecutarTarefas	CoordenarMontagem
ExtratorTarefas	DividirMontagemEmTarefas	ExtracaoTarefas	ExtrairTarefas

O modelo de classes de agentes tem objetivo de aproximar do formato da implementação e para isso são capturadas as funcionalidades básicas do SMA. Os papéis do

modelo de papéis são atribuídos para classes de agentes pelo uso pré-fixado de <<plays>>. Para o LAR há um agente *Coordenador* que desempenha dois papéis e outros agentes desempenham somente um papel cada, como pode ser visto na Figura 4.20.

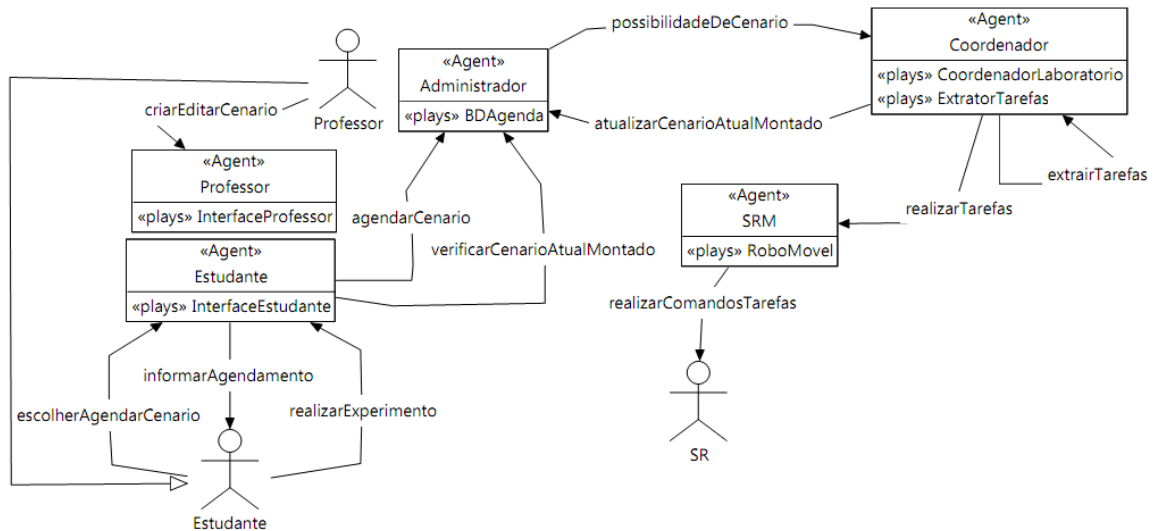


Figura 4. 20 Modelo de Classes de Agentes

As classes de agentes e protocolos foram identificados, porém ainda não definidos. São criados modelos de protocolo e modelos de plano do agente para defini-los.

O modelo de protocolos tem objetivo de detalhar cada protocolo entre agentes ou entre ator externo apresentado no modelo de classes de agentes. O modelo de protocolos usa a abordagem do diagrama de interação AUML (AUML, 2011). Os modelos de protocolos, Figura 4.21 e Figura 4.22, apresentam, respectivamente, como são o agendamento e a montagem dos cenários.

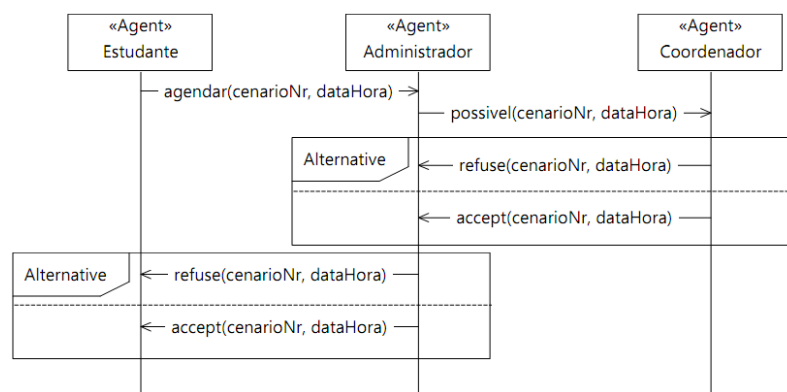


Figura 4. 21 Modelo dos Protocolos de Agendamento

Os modelos apresentado na Figura 4.21 e Figura 4.22 são agrupamentos de protocolos. O Modelo dos Protocolos de Agendamento é formado por 4 protocolos: *escolherAgendarCenario*, *informarAgendamento*, *possibilidadeDeCenario*, e

agendarCenario. O Modelo dos Protocolos de Montagem é formado por 5 protocolos: *atualizarCenarioAtualMontado*, *verificarCenarioAtualMontado*, *realizarTarefas*, *realizarComandosTarefas*, e *extrairTarefas*.

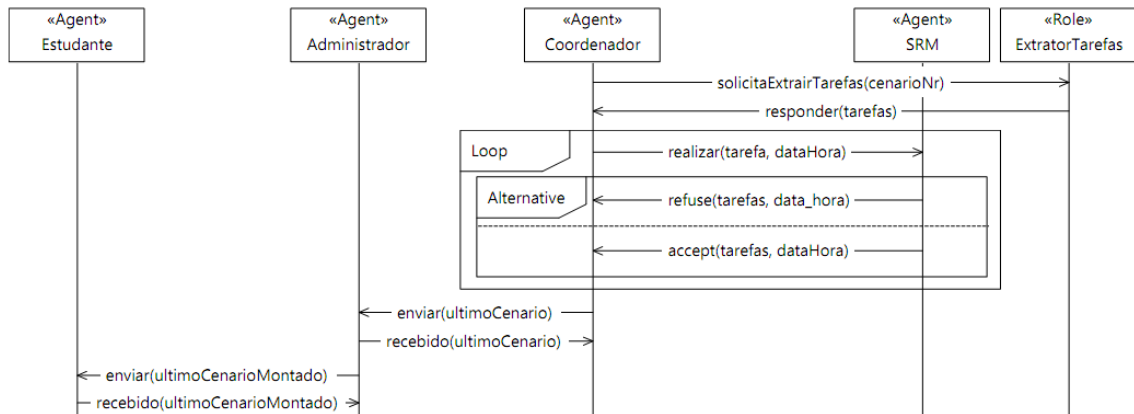


Figura 4. 22 Modelo dos Protocolos de Montagem

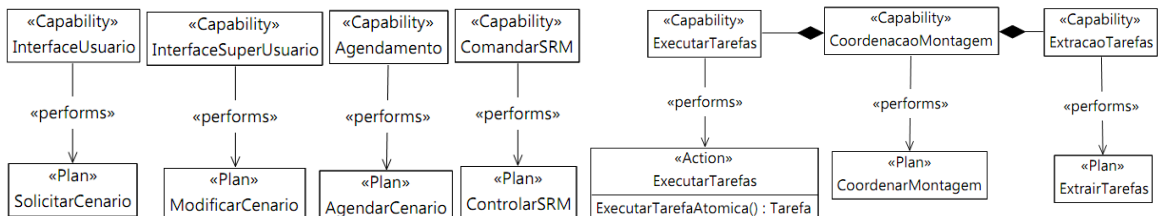


Figura 4. 23 Modelo de Capacidades

O modelo de plano do agente é o algoritmo para alcançar objetivo do agente na organização. A quantidade de modelos de planos deve ser pelo menos a mesma da quantidade de agente no modelo de classes de agentes, um para cada agente. Para definir comportamento do agente com modelo de plano é usando um autômato finito e cada transição corresponde às mensagens recebidas ou enviadas. A sintaxe da transição é $[guard] receive(message, sender) / send(message, receiver)$, o parâmetro $[guard]$ determina habilitação da transição. O modelo do plano do agente deve ser desenvolvido em coerência e consistência com os protocolos identificados no modelo de classe de agentes e modelos de protocolos. Foram modelados 6 diagramas de plano, Figuras 4.24, 4.25, 4.26, 4.27 e 4.28. O Agente Coordenador possui dois planos, um para coordenar a montagem e outro para extrair tarefas do cenário escolhido.

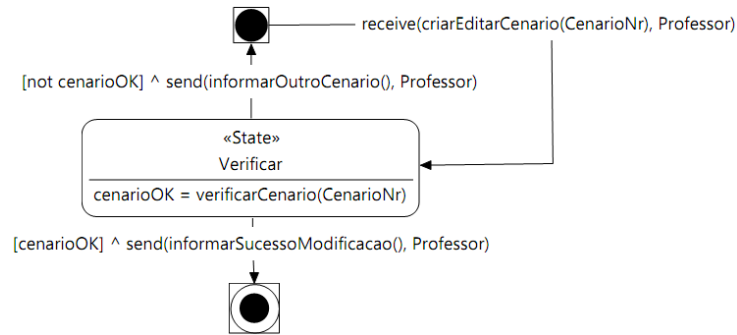


Figura 4. 24 Modelo de Modificar Cenário do Agente Professor

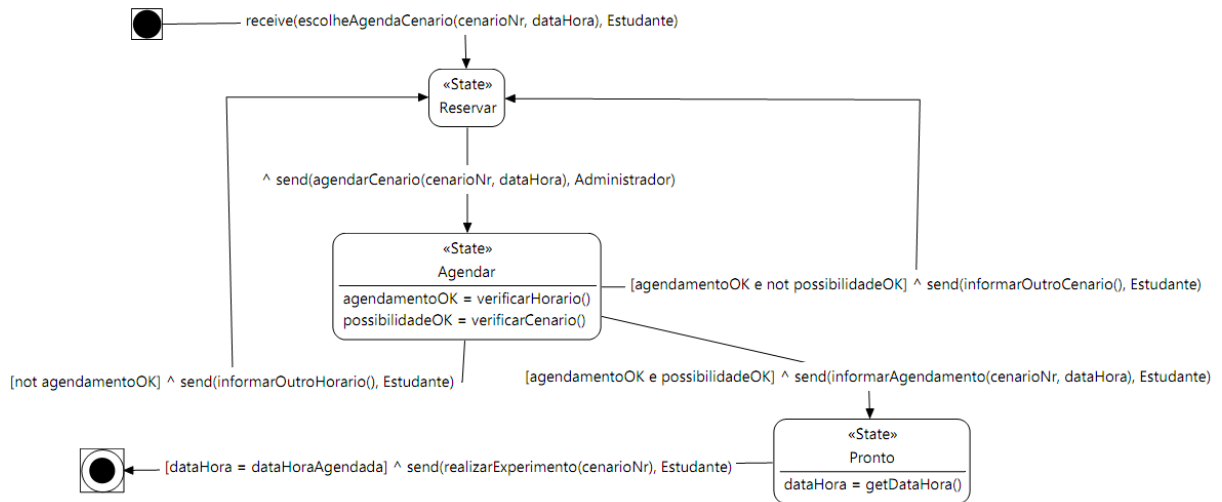


Figura 4. 25 Modelo de Plano Solicitar Cenário do Agente Estudante

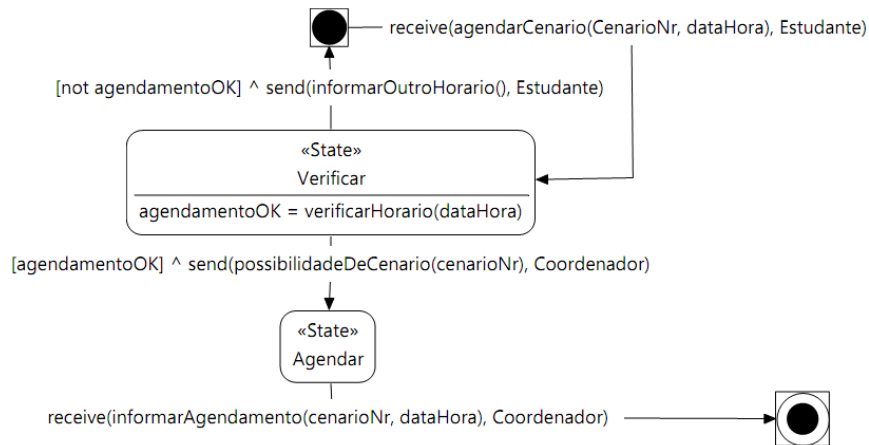


Figura 4. 26 Modelo de Agendar Cenário do Agente Administrador

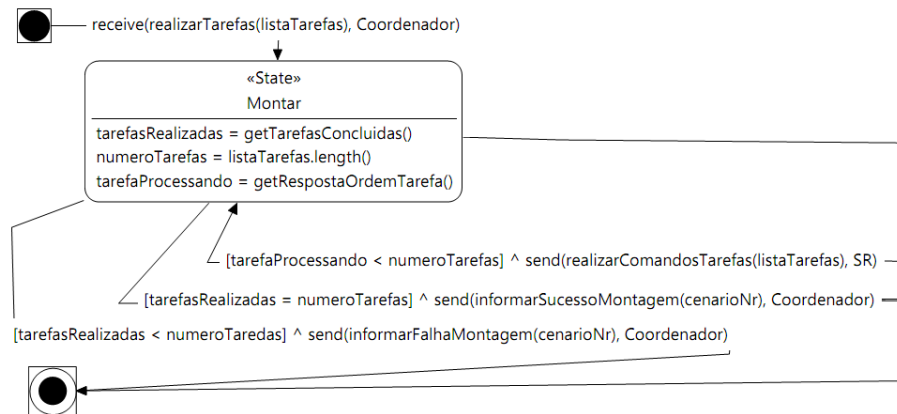


Figura 4.27 Modelo de Plano Controlar SRM do Agente SRM

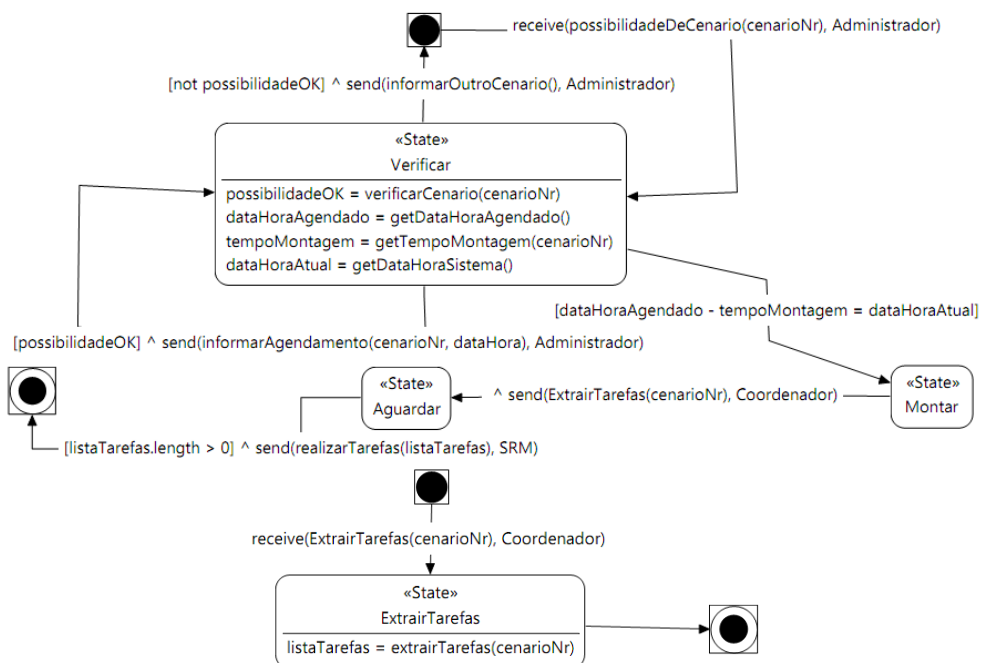


Figura 4.28 Modelo de Plano Coordenar Montagem e Extrair Tarefas do Agente Coordenador

3.4 MODELAGEM DO SISTEMA USANDO REDE DE PETRI

Usando o meta-modelo RdP é explicado quais os agentes robóticos, como serão programados e como eles vão atuar no ambiente. A modelagem em RdP foi uma das primeiras etapas desse trabalho, depois observou-se a necessidade de atualizar a RdP quando a modelagem MaSE apresentava uma visão macroscópica do comportamento dos agentes robóticos, onde não foi o suficiente para entender a natureza concorrente e/ou colaborativo deles. O modeloRdP oferece uma abstração do sistema distribuído e discreto característicos de um SMA. Não modelamos

com autômatos por serem modelos baseados em estados, diferentes da RdP que é um modelo heterogêneo que considera estados e ações (MACIEL, 2008), o que se apresenta ser o mais indicado para modelar sistemas como o discutido nessa ilustração, onde os agentes robóticos são também reativos.

Para entendermos como o robô vai atuar em certas tarefas, utilizamos o exemplo simples de tarefas dos robôs em posicionarem bolas, como mostrado Figura 4.29, representando os passos para tarefa de capturar bola no modelo de RdP a seguir.

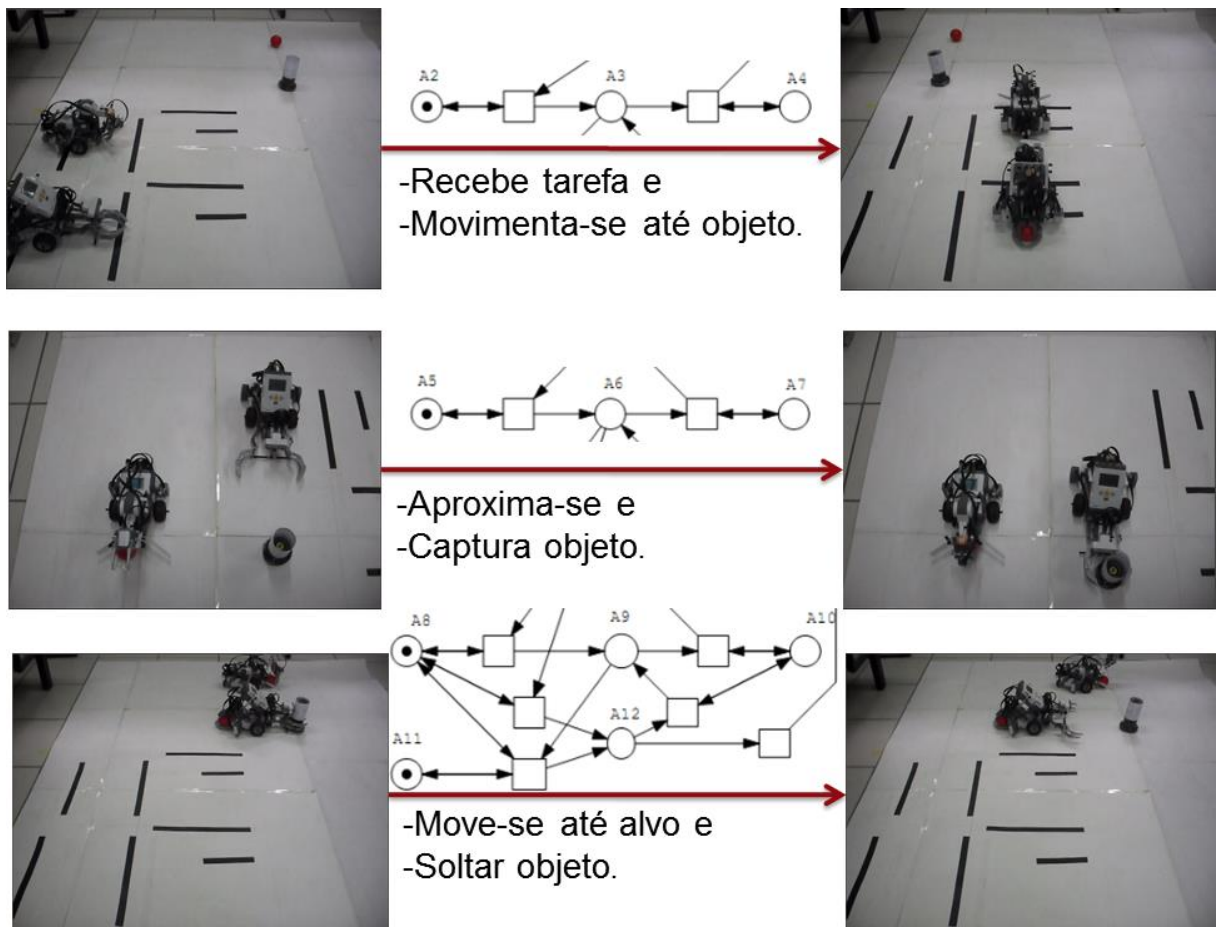
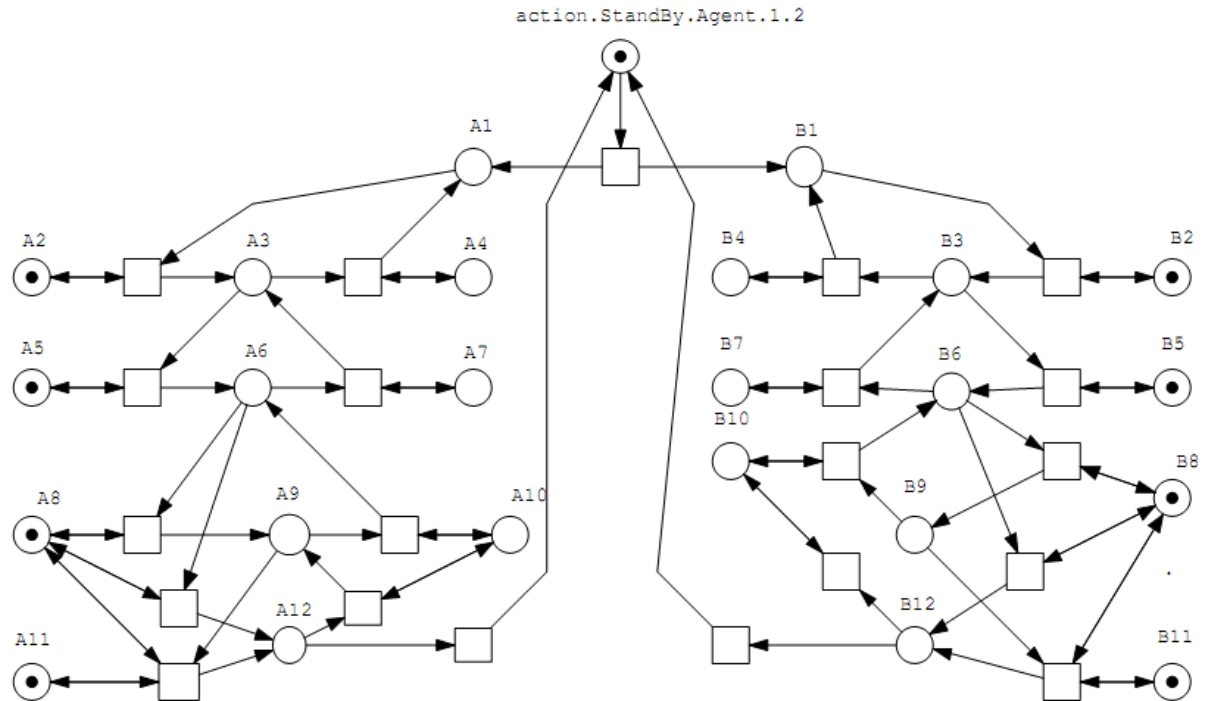


Figura 4. 29 Passos da RdP na tarefa posicionar objetos

Para primeira situação os agentes robóticos têm comportamento concorrente, observando-se a causalidade de concorrência no lugar *action.StandBy.Agent.1.2* e a transição entre os dois lugares *action.Active.Agent.Exclusivo_OR* na Figura 4.30 do modelo concorrente. Os agentes robóticos atuam em bolas diferentes para cumprir o objetivo de montar o cenário.

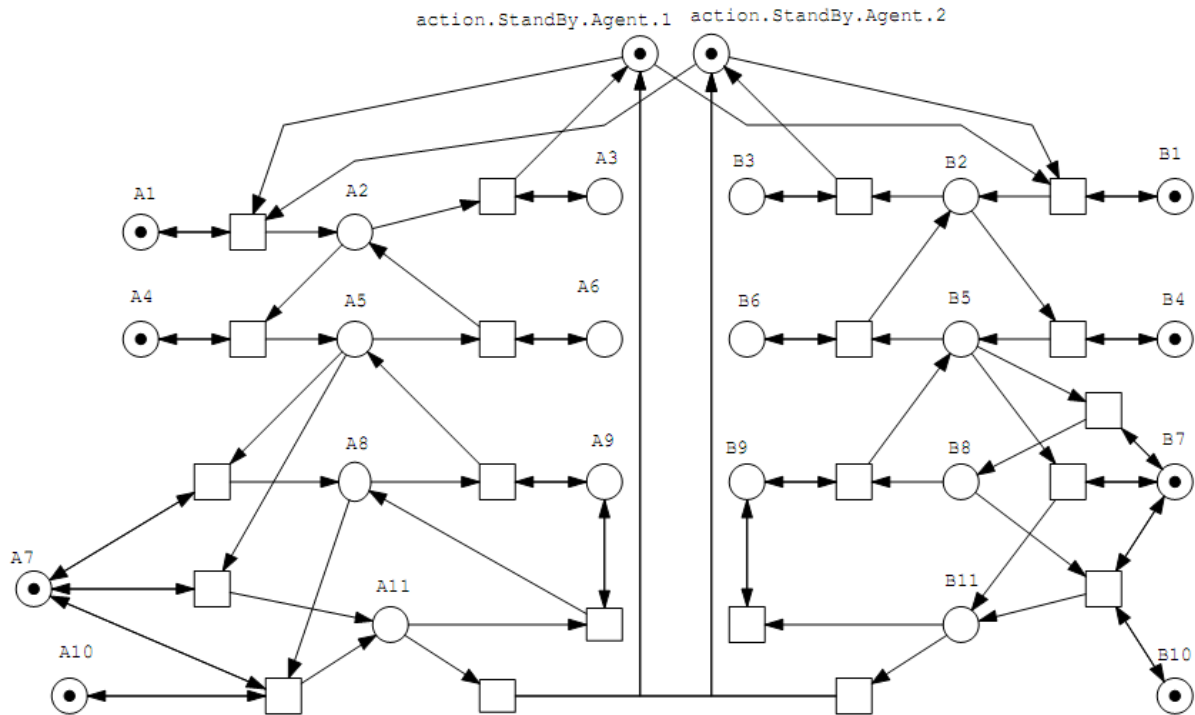


A1= action.Active.Agent.Exclusive_OR
 A2= predicate.SeeBall.1
 A3= action.MoveToBall.1
 A4= predicate.NOT_seeBall.1
 A5= predicate.CloseToBall.1
 A6= action.CatchBall.1
 A7= predicate.NOT_CloseToBall.1
 A8= predicate.HasBall.1
 A9= action.MoveToTarget.1
 A10 = predicate.NOT_HasBall.1
 A11 = predicate.NearTarget.1
 A12 = action.o.Unhitch.Ball.1

B1 = action.Active.Agent.Exclusive_OR
 B2= predicate.SeeBall.2
 B3= action.MoveToBall.2
 B4= predicate.NOT_seeBall.2
 B5= predicate.CloseToBall.2
 B6= action.CatchBall.2
 B7= predicate.NOT_CloseToBall.2
 B8= predicate.HasBall.2
 B9= action.MoveToTarget.2
 B10 = predicate.NOT_HasBall.2
 B11 = predicate.NearTarget.2
 B12 = action.o.Unhitch.Ball.2

Figura 4. 30 RDP para montagem do cenário com agentes robóticos concorrentes

Para a segunda situação os agentes têm comportamento sincronizado e observa-se a causalidade de sincronização nos lugares *action.StandBy.Agent.1* e *action.StandBy.Agent.2* com a transição que começa a atividade na bola 1 ou bola 2 na Figura 4.30 do modelo sincronizado. Os agentes robóticos realizam cooperação para atingir objetivo de montar o cenário.



A1= predicate.SeeBall.1
 A2= action.MoveToBall.1
 A3= predicate.NOT_seeBall.1
 A4= predicate.CloseToBall.1
 A5= action.CatchBall.1
 A6= predicate.NOT_CloseToBall.1
 A7= predicate.HasBall.1
 A8= action.MoveToTarget.1
 A9= predicate.NOT_HasBall.1
 A10 = predicate.NearTarget.1
 A11 = action.o.Unhitch.Ball.1

B1= predicate.SeeBall.2
 B2= action.MoveToBall.2
 B3= predicate.NOT_seeBall.2
 B4= predicate.CloseToBall.2
 B5= action.CatchBall.2
 B6= predicate.NOT_CloseToBall.2
 B7= predicate.HasBall.2
 B8= action.MoveToTarget.2
 B9= predicate.NOT_HasBall.2
 B10 = predicate.NearTarget.2
 B11 = action.o.Unhitch.Ball.2

Figura 4. 31 RdP para montagem do cenário com agentes robóticos sincronizados

3.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi apresentada a arquitetura geral do sistema e suas características de software e de hardware na solução do problema. Ainda, foram apresentados os modelos de RdP e MaSE para o sistema SMA proposto com objetivo de resolver o problema dos cenários de experimentação, afim de não serem multi-configuráveis e não terem montagem autônoma, apresentado na Seção 1.1.

4 TESTES E AVALIAÇÃO DO SISTEMA

Neste capítulo são apresentados os testes realizados no sistema para estudo de caso. Foram realizados testes para detectar a presença de falhas no sistema, bem como avaliar sua eficácia. Os testes são divididos em dois grupos, o primeiro grupo apresentado na Seção 5.1.1 e o segundo grupo na Seção 5.1.2. A implementação foi em prototipação e a cada funcionalidade implementada foram realizados testes específicos de caixa preta e com isso a qualidade do *software* foi sendo melhorada. Esta Seção está subdividida em subseção 5.1, que descreve os testes com cada robô de forma separada e com os robôs de forma colaborativa. Na Seção 5.2 são apresentadas discussões e conclusões sobre os testes empregados.

4.1 DESCRIÇÃO DOS TESTES EMPREGADOS

O primeiro grupo de testes foi em cada robô móvel de forma separada. O objetivo desse primeiro grupo de testes é saber se o ambiente de controle é apropriado e conseguiria realizar as tarefas. Além disso, foi ponto de partida para calcular os desempenhos de cada robô para todas as tarefas. O segundo grupo de testes foi realizado com os robôs de forma colaborativa. O objetivo do segundo grupo de testes é saber se aconteceria a montagem sincronizada do cenário escolhido, e também, montaria mais de um sem intervenção humana para realizar setup inicial.

4.1.1 Testando os agentes robóticos separadamente

Para planejar os testes do sistema é definido Ciclo de Desenvolvimento: é a combinação da arquitetura robótica mais ambiente de atuação dos robôs mais codificação em JADE e LeJOS. Essa definição também pode ser chamada de ambiente de controle robótico quando são mostrados aspectos relacionados à interação entre sensoriamento, planejamento e acionamento (DUDEK & JENKIN, 2000). Ainda, é definido Ciclo de Experimentação: é a experimentação de um Ciclo de Desenvolvimento. A experimentação de um Ciclo de Desenvolvimento termina quando os objetivos de uma tarefa não são alcançadas.

A Figura 5.1 mostra os ciclos de desenvolvimento do robô garra simples, formado por cinco ciclos. As mudanças no robô e no cenário foram gradativas. Na Figura 5.1a o robô tem configuração de garra em um dedo com contato em forma de “V” onde acontece a retenção da bola, um sensor de luz na garra para verificar se ao capturar a bola ele ficou retida entre a garra e o contato em “V”, um sensor ultrasônico para realizar alinhamento angular e aproximação linear com a bola. A configuração da superfície de contato com robô, na Figura 5.1a, é um papel quadriculado, onde foi colado na mesa com fita dupla face para ser usado como referência para medir a qualidade da montagem do cenário. Com o desenvolver dos ciclos aconteceram incrementos buscando um sistema mais eficiente. No segundo ciclo de desenvolvimento, Figura 5.1b, foi colocado um sensor de cor embaixo do carro para fazer alinhamento das rodas com ponto de partida do robô depois que ele acaba de realizar suas tarefas e está voltando para se preparar para próxima tarefa. No terceiro ciclo de desenvolvimento, Figura 5.1c, o sensor de cor é trocado por dois sensores de luz, um em cada uma das rodas. No quarto ciclo de desenvolvimento, Figura 5.1d, foram realizados ajustes nos marcadores sobre o papel quadriculado para melhorar a identificação pelos sensores dos robôs. No quinto ciclo de desenvolvimento, Figura 5.1e, foram feitos ajustes na fixação das rodas usando uma liga.

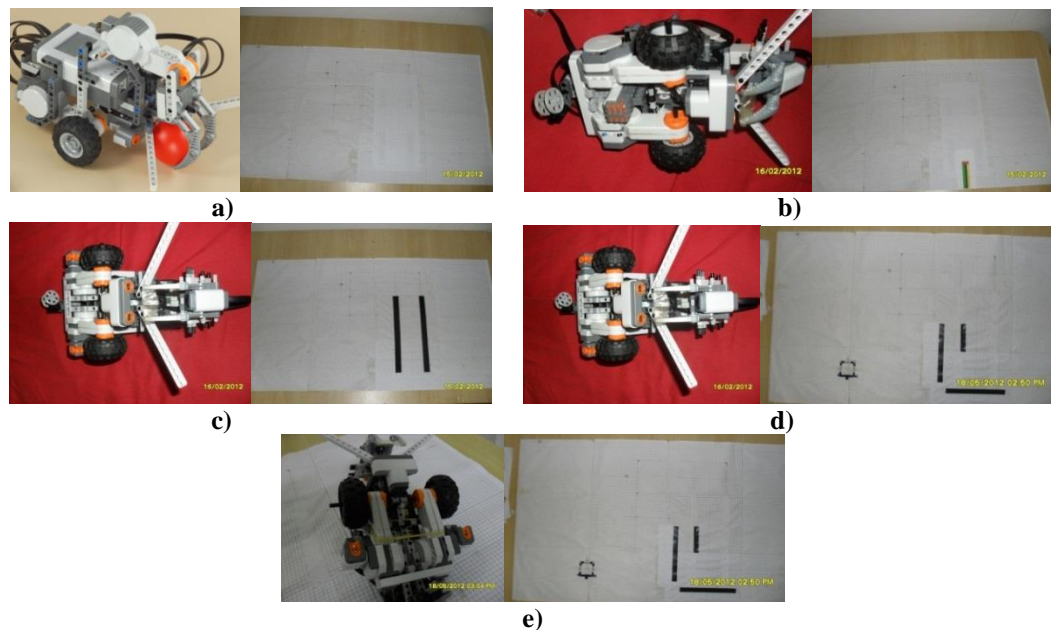


Figura 5. 1 Ciclos de desenvolvimento do robô garra simples

A Figura 5.2 mostra os ciclos de desenvolvimento do robô garra dupla, formado por cinco ciclos. As mudanças no robô e no cenário foram gradativas, como na etapa anterior.

Na Figura 5.2a o robô tem configuração garra de dois dedos onde acontece a retenção da bola, um sensor de toque na garra para verificar a distância de captura da garra, um sensor ultrassônico para realizar alinhamento angular e aproximação linear com a bola. A configuração da superfície de contato com robô, na Figura 5.2a, é um papel quadriculado, onde foi colado na mesa com fita dupla face para ser usado como referência para medir a qualidade da montagem do cenário. Assim como foi feito com robô garra simples, com o desenvolver dos ciclos aconteceram incrementos buscando um sistema mais eficiente. As mudanças nos ciclos de desenvolvimento foram parecidas com as do robô garra simples. São mostrados as mudanças em cada ciclo de desenvolvimento nas Figuras 5.2b, 5.2c, 5.2d e 5.2e.

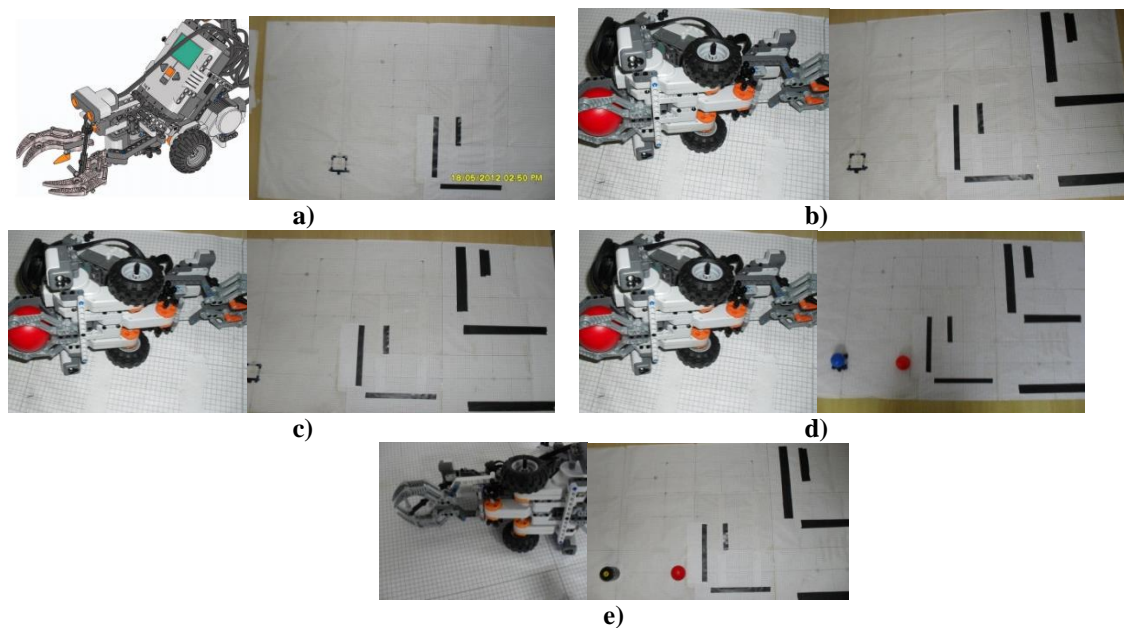


Figura 5. 2 Ciclos de desenvolvimento do robô garra dupla

A Figura 5.3 refere-se ao robô garra simples, com o desvio padrão da população foi de 13,623 mm para x e 9,77 mm para y e ainda é observado que em cada ciclo de experimentação a taxa de sucesso em realizar a montagem do cenário melhorava, chegando a atingir 100% na taxa de execução com sucesso no ciclo de experimentação 5. Considerando-se todas as quantidades de execuções com sucesso de todos os ciclos de experimentação obtêm-se 77% de sucesso na montagem do cenário, considerando somente tarefa realizada pelo robô garra simples.

A Figura 5.4 refere-se ao robô garra dupla, com o desvio padrão da população foi de 46,296 mm para x e 31,129 mm para y e ainda observamos que em cada ciclo de

experimentação a taxa de sucesso em realizar a montagem do cenário melhorava, chegando a atingir 100% na taxa de execução com sucesso no ciclo de experimentação 5.

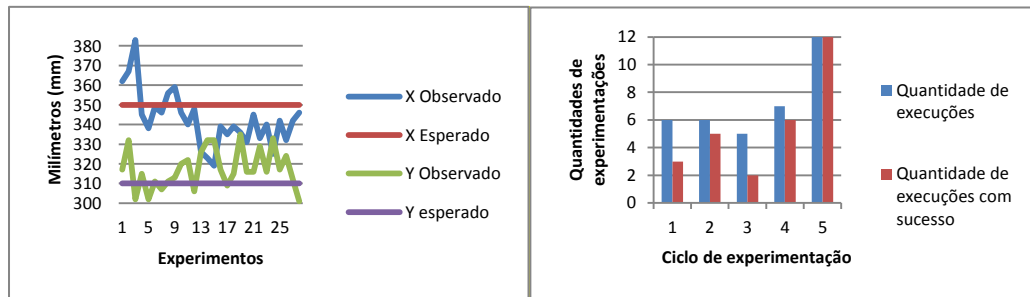


Figura 5.3 Qualidade do cenário montado pelo robô garra simples

Juntando todas as quantidades de execuções com sucesso de todos os ciclos de experimentação obtêm-se 44% de sucesso na montagem do cenário, considerando somente tarefa realizada pelo robô garra dupla. As conclusões e discussões são feitas na Seção 5.2.

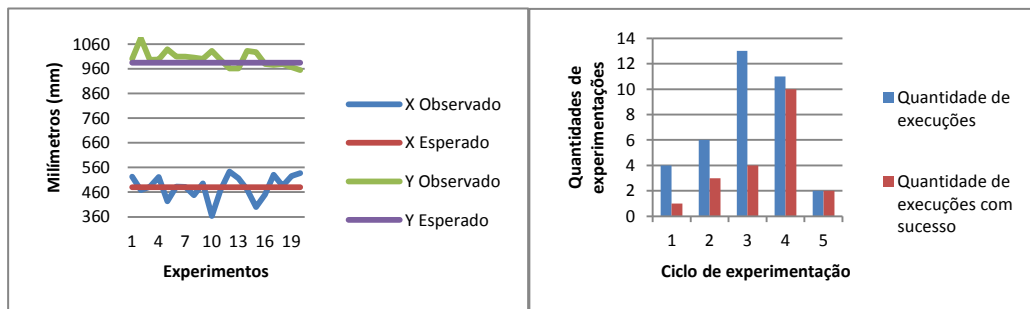


Figura 5.4 Qualidade do cenário montado pelo robô garra dupla

Os dados coletados para calcular desvio padrão são apresentados em anexo D. Nessa primeira bateria de testes foi coletada energia inicial dos robôs antes de realizar a tarefa recebida. A estratégia inicial era usar energia dos robôs como parâmetros para a função de cálculo de desempenho, porém, foi observada imprecisão na leitura da energia do dispositivo robótico, então, os valores, nesse primeiro momento ficaram como sendo fixos, seguindo a Tabela 5.1.

Tabela 5.1 Desempenhos dos robôs para cada um dos passos em cada objeto

Passos	Robô garra dupla	Robô garra simples	Robô garra dupla	Robô garra simples
	Objeto 0	Objeto 1	Objeto 0	Objeto 1
P1	18	11	11	21
P2	16	5	8	14
P3	18	12	12	21
P4	15	5	7	13
P5	18	12	12	21

P1 – Aproximar do objeto.

P2 – Capturar objeto.

P3 – Deslocamento até ponto objeto.

P4 – Soltar objeto.

P5 – Voltar ao ponto de partida.

Os passos apresentados de P1 a P5 são os mesmos apresentados no fluxograma da tarefa de posicionar objeto, Figura 4.6.

4.1.2 Testando os agentes robóticos colaborativos

Para esse conjunto de testes foram definidos três cenários de montagem. Nesses testes não foram coletados dados de precisão na montagem, pois o objetivo é de verificação da colaboração na execução concorrente das tarefas.

- Cenário 1: colocar objeto 1 na posição 10 e objeto 2 na posição 11.
- Cenário 2: colocar objeto 1 na posição 9 e objeto 2 na posição 10.
- Cenário 3: colocar objeto 1 na posição 8 e objeto 2 na posição 9.

Foram executados sete testes. Primeiro foi testado cenário 1 seguido do cenário 2, então em seguida, foram reiniciadas as posições dos objetos e dos robôs para testar cenário 1 seguido cenário 3, depois, foram reiniciados para testar cenário 3 seguido cenário 2. Para cada teste foi também testado o inverso, por exemplo, testado cenário 2 seguido cenário 1 para o teste cenário 1 seguido cenário 2. E para finalizar, último teste com montagem do cenário 1, seguido do 2 e 3.

A Figura 5.5, Figura 5.6 e Figura 5.7 mostram os caminhos das trajetórias calculadas pelos robôs para se aproximarem do ponto objetivo, deslocar objeto até ponto alvo e retornar para ao ponto de partida para os cenários 1, 2 e 3, respectivamente.

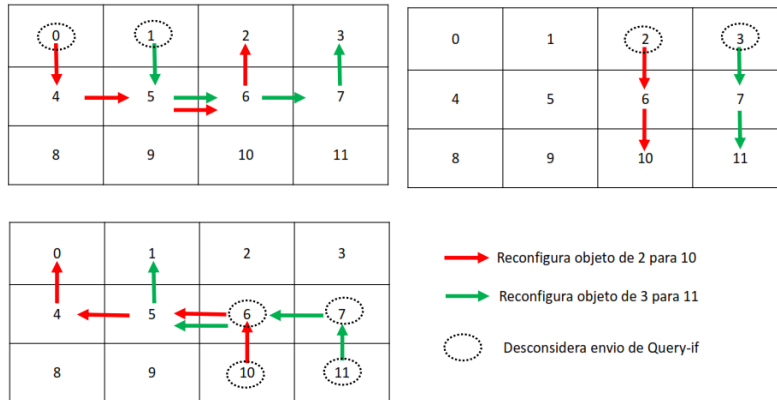


Figura 5. 5 Montagem do cenário 1

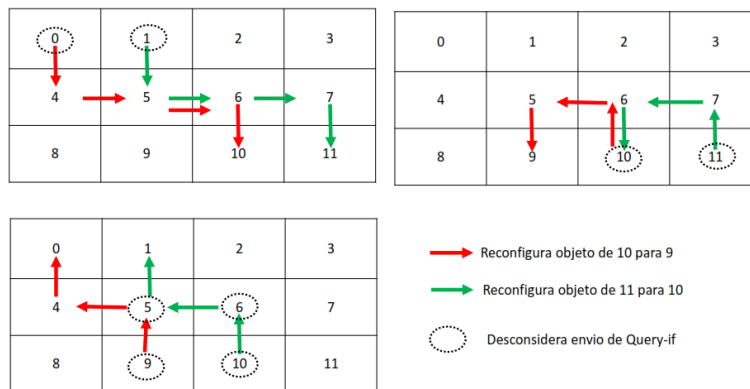


Figura 5. 6 Montagem do cenário 2 depois de ter montado o cenário 1

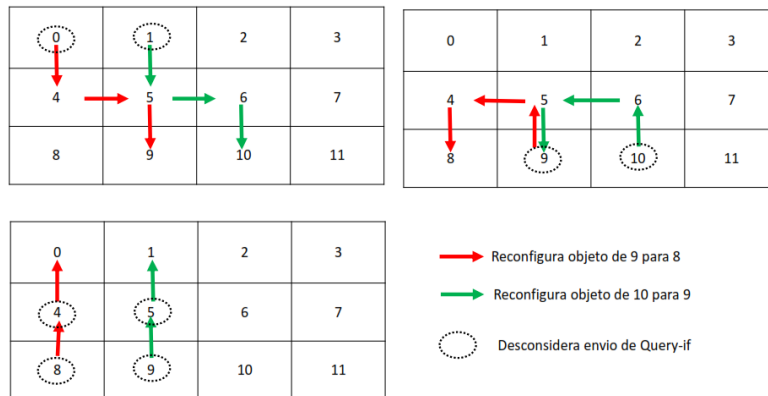


Figura 5. 7 Montagem do cenário 3 depois de ter montado o cenário 2

A Figura 5.8 apresenta os cenários finais no ambiente real depois de montado, é visível que os objetos são corretamente posicionados nas células que eram que ficar de acordo com arquivo *xml* escolhido.

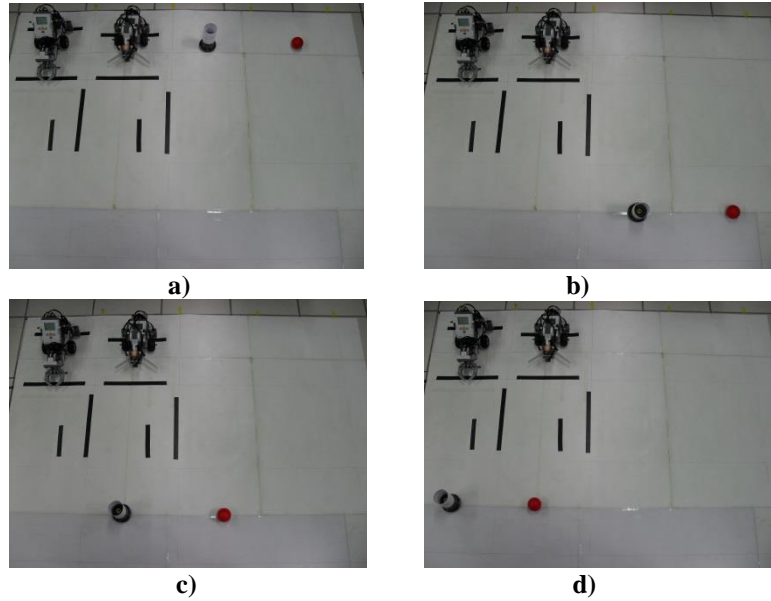


Figura 5. 8 Estados finais do a) cenário inicial, b) cenário 1, c) cenário 2, d) cenário 3

A Figura 5.9 apresenta o primeiro grupo de troca de mensagem para garantir uma montagem segura. A mensagem com performativa INFORM vem do agente usuário (retângulo RMA) que escolheu um cenário (arquivo XML) e enviou para Agente Coordenador. Em seguida, os Agentes Robôs participantes da montagem são informados quando Agente Coordenador manda um REQUEST para Agente de Páginas Amarelas para saber quais robôs são capacitados para aqueles tipos de tarefas. O Agente Coordenador extrai as tarefas do cenário e pelas performativas de negociação (CPF – Calls For Proposals, PROPOSES, ACCEPT-PROPOSAL) realiza negociação com os Agentes Robôs participantes da montagem. Depois da negociação e das devidas atribuições de tarefas para os robôs, o Agente Coordenador envia um INFORM com mapa geral e posições dos objetos para os Agentes Robôs participantes da montagem.

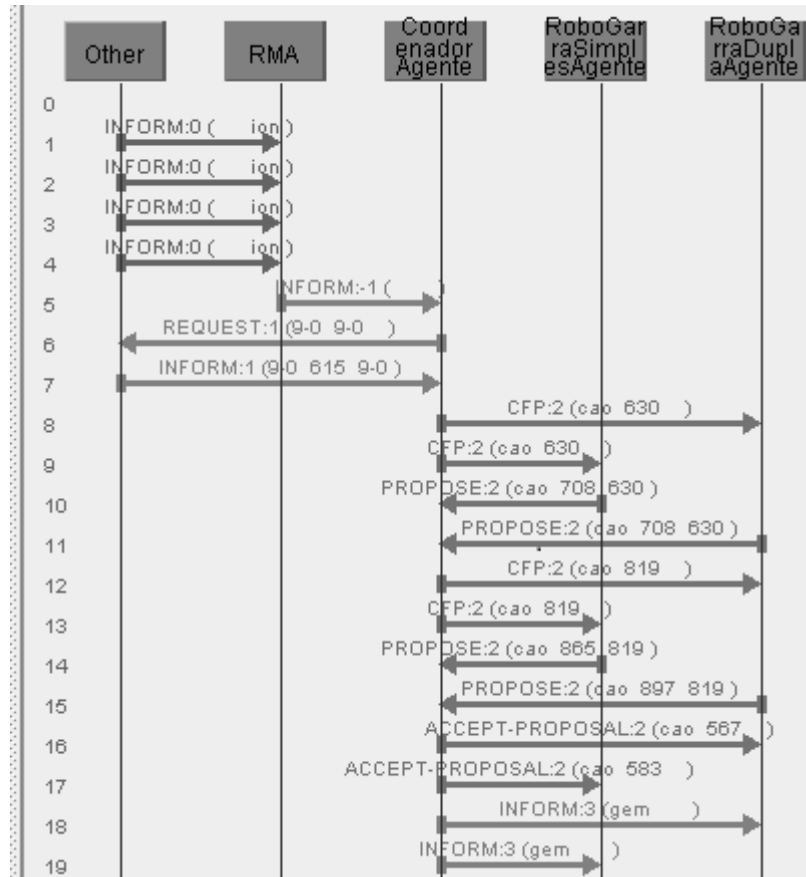


Figura 5. 9 Primeiro grupo de mensagens para montagem do cenário

A Figura 5.10 mostra o segundo grupo de troca de mensagem para garantir uma montagem segura. Com o mapa global e posições dos objetos os Agentes Robóticos tratam o mapa e em seguida é calculado as suas trajetórias. Usando de performativas QUERY-IF os Agentes robóticos participantes perguntam se o próximo passo deles está livre, depois de receber todos os INFORM eles continuam com suas trajetórias enviando QUERY-IF e esperando os informes. Caso a resposta do INFORM seja que a próxima posição não estará livre, o critério de decisão é a favor do robô que tiver menor caminho seguido de menor energia, fazendo outro robô esperar próxima rodada de QUERY-IF para avançar com sua rota. Isso funciona porque os robôs trabalham em velocidades iguais. Esse grupo de mensagens vai até quando os agentes soltam os objetos e em seguida informam ao Agente Coordenador que atualizem novas posições dos objetos.

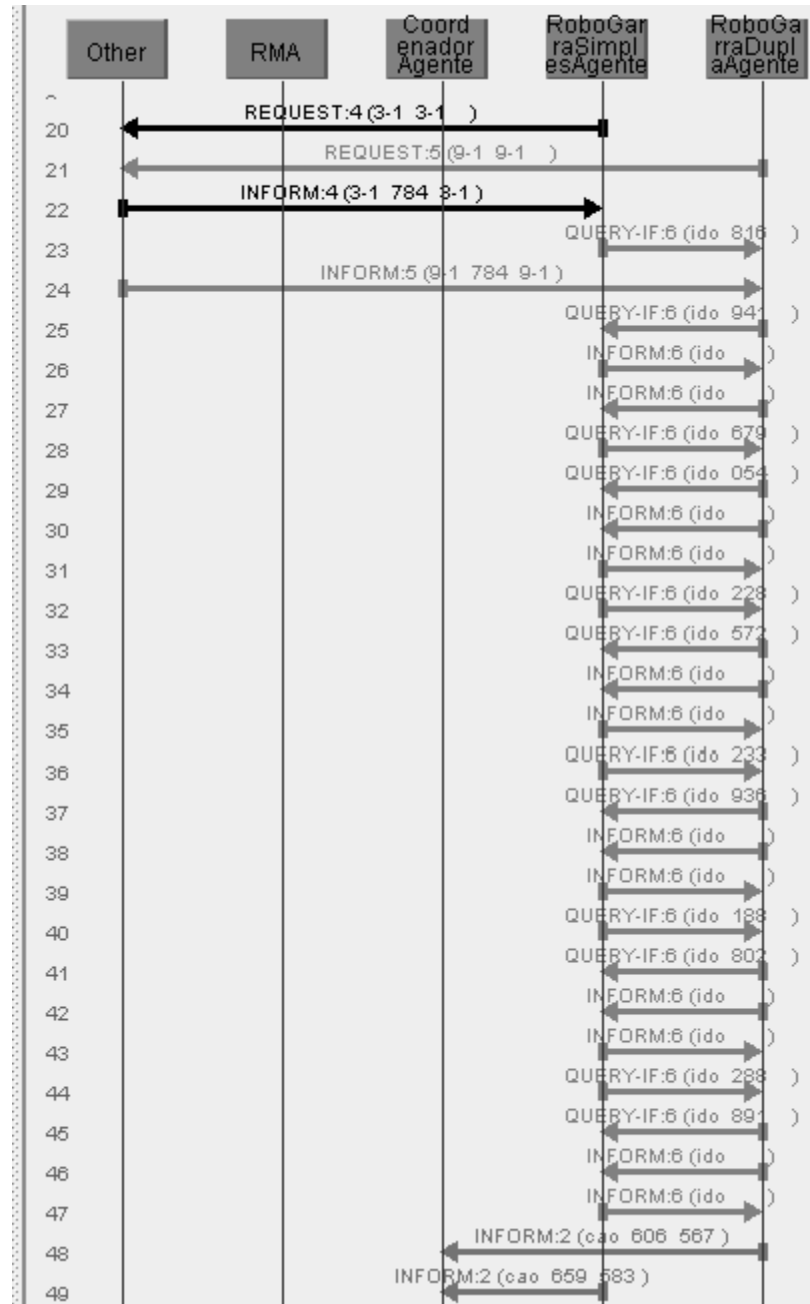


Figura 5. 10 Segundo grupo de mensagens para montagem do cenário

A Figura 5.11 mostra o terceiro grupo de troca de mensagem para garantir uma montagem segura. Com os objetos posicionados e suas posições atualizadas, o Agente coordenador manda um INFORM para os agentes robóticos com novas posições de objetos no mapa, em seguida, os agentes robóticos recalculam suas trajetórias levando em consideração essas novas atualizações. Então, segue sua mesma estratégia do grupo de mensagens anterior, usando de performativas QUERY-IF os Agentes robóticos participantes perguntam se o próximo passo deles está livre, depois de receber todos os INFORM eles continuam com suas

trajetórias enviando QUERY-IF e esperando os informes. Assim, os robôs terminam em suas respectivas posições iniciais mandando mensagens de tarefa realizada com sucesso pela performativa CONFIRM para Agente Coordenador.

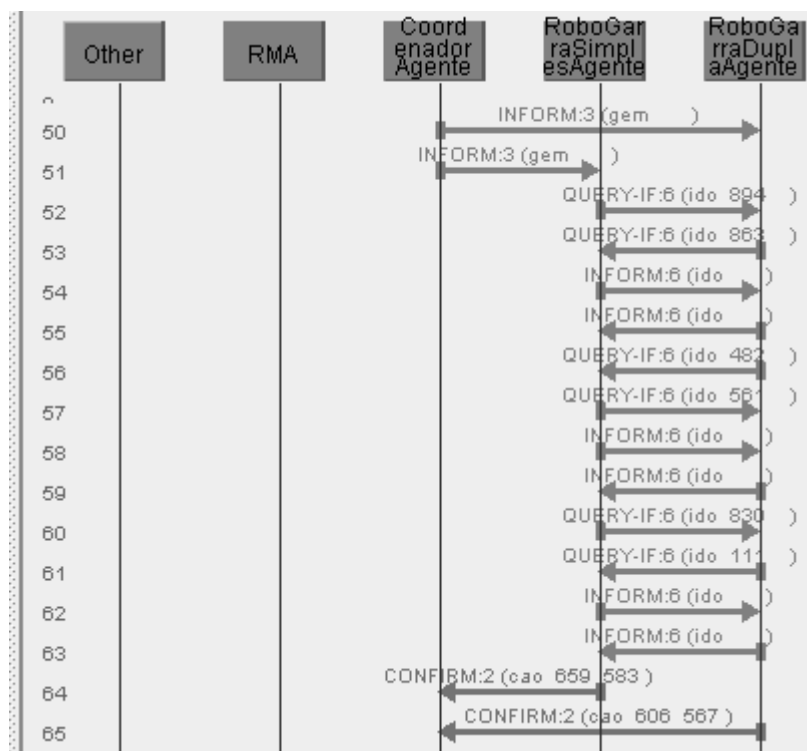


Figura 5. 11 Terceiro grupo de mensagens para montagem do cenário

Todas as trocas de mensagens seguem o padrão FIPA. As performativas usadas foram escolhidas de acordo com suas características para as necessidades de comunicação para resolver o problema, o uso correto das performativas é detalhado no trabalho que discute as linguagens para SMA (GLUZ & VICARI, 2003) e desenvolvimento de SMA usando JADE (WILEY, 2007).

4.2 CONCLUSÕES E DISCUSSÕES SOBRE OS TESTES

Foram realizados testes em dois grupos. No primeiro grupo de testes, trabalhou-se com os robôs móveis de forma separada, onde foi verificado que ambiente de sistema de controle realizou suas tarefas com precisão aceitável, em até 12 execuções com sucesso e desvio de até 46 mm. Ainda, foi estimado o desempenho de cada robô para todas as tarefas baseado em observações de suas precisões em posicionar os objetos. Já no segundo grupo realizaram-se

testes com os robôs trabalhando de forma colaborativa, permitindo assim, verificar a cooperação dos agentes participantes da montagem, além disso, averiguou-se que eles conseguiram montar com sucesso mais de um cenário. Sendo assim, este trabalho realizou todos os testes necessários para validar a estratégia na solução do problema. Contudo, identificou-se que a solução é limitada a precisão da técnica de mapeamento e navegação, além de existirem técnicas melhores que a de odometria com marco referencial. Como por exemplo, as técnicas que usam visão computacional para achar a localização de objetos e suas poses. Outra limitação que influenciou na precisão da montagem, foram os hardwares utilizados, que são voltados para aplicações educacionais e utilizam hardwares não profissionais ou industriais.

5 CONCLUSÕES E TRABALHOS FUTUROS

Nesta dissertação foi apresentada uma abordagem que utiliza Sistema Multiagente desenvolvido em Framework JADE, integrado com robótica móvel combinado com representação de cenários em XML para resolver o problema de configuração e montagem autônoma de cenário em Laboratórios de Acesso Remoto. Os Laboratórios de Acesso Remoto atuais não são multi-configuráveis, e esta característica tira a oportunidade do estudante de escolher o cenário de experimentação, tornando-se um problema que o Sistema Multiagente resolve. Ele foi desenvolvido no framework JADE, integrado com framework LeJOS, que são usados para programar a plataforma de robótica educacional Lego Mindstorms.

Antes das primeiras implementações do sistema, foram elaborados modelos iniciais em Rede de Petri que validaram os princípios da ideia de usar robótica móvel para solucionar o problema. Em seguida, realizou-se a modelagem na metodologia O-MaSE e foi verificado que a modelagem em Rede de Petri precisava ter melhor visualização do comportamento sincronizado e concorrente dos agentes robóticos móveis, então, o modelo Rede de Petri foi remodelado. Os modelos em Rede de Petri e O-MaSE contribuíram para se ter um entendimento mais profundo dos objetivos hierárquicos, domínios, organizações, papéis dos agentes e protocolos, planos e classes de agentes implementados e como seria a troca de mensagens, ou seja, quais performativas devem ser utilizadas por eles. Além disso, foi possível perceber o comportamento concorrente e sincronizado dos agentes robóticos.

Para validar a abordagem os testes foram realizados em dois grupos. O primeiro grupo de testes foi realizado com cada robô de forma isolada, solicitando que cada robô executasse uma tarefa por vez. Durante os testes foram capturadas informações de precisão na montagem. Esses dados juntamente com a variação da quantidade de energia antes e depois de montar o cenário foram usados para definir o desempenho de cada um em cada uma das tarefas. O segundo grupo de testes com os robôs foi realizado de forma colaborativa, solicitando para que eles montarem um cenário e decidirem quem ficaria com qual tarefa da montagem baseado em desempenhos pré-definidos. Depois das tarefas atribuídas, os robôs trocam mensagens para garantir que não acontecerão colisões de robôs entre si e colisões de robôs com objetos. Com os testes realizados foram obtidos resultados de eficiência da utilização dessa estratégia de solução do problema.

O resultado do uso da abordagem de Sistema Multiagente com robótica móvel para configurar cenários de experimentação foi comprovado com montagem de cenários de boa qualidade, em até doze execuções seguidas com sucesso, com desvio de até 46 mm e ainda comprovou-se que a montagem colaborativa do cenário mostrou-se eficiente e segura, ao garantir que não iria ocorrer colisão de robôs e de objetos com robôs, graças aos conjuntos de troca de mensagens. O sistema garante montagens consecutivas de cenários com sua continuidade e consistência, provado no último grupo de testes, com até três cenários.

Este trabalho apresentou uma abordagem funcional de qualidade que ainda não foi encontrado em Laboratórios de Acesso Remoto. De todos os trabalhos levantados no referencial bibliográfico de Laboratórios de Acesso Remoto o único encontrado que usa uma abordagem de Sistema Multiagente foi o LARRE, que foi desenvolvido em uma dissertação de Mestrado do LaboREAM e apresentado na Seção trabalhos correlatos.

Destarte, o trabalho produziu as seguintes contribuições: a primeira é o uso de Sistema Multiagente para construção de Laboratórios de Acesso Remoto é mais robusto, escalável e flexível do que as encontradas em outras pesquisas. Uma contribuição importante é a solução para laboratórios multi-configuráveis por meio da combinação de Sistema Multiagente com robótica móvel, possibilitando o usuário escolher qual cenário quer experimentar no Laboratório de Acesso Remoto. Portanto, cenários que ocupavam espaços físicos distintos agora podem ocupar um mesmo espaço, sendo necessário fazer apenas as configurações e montagens, reduzindo assim, o custo de construções de vários laboratórios de cenários fixos.

Constatou-se que apesar da eficiência da solução, ela ainda apresenta limitações de precisão associadas às técnicas de mapeamento e navegação adotadas e ainda aos dispositivos de hardware usados para atuar no ambiente físico, limitações essas que serão tratadas em trabalhos futuros, mas servirão sem dúvida para instigar a realização de trabalhos futuros. Os robôs podem futuramente ser substituídos por outros compostos de hardware e sensores profissionais. Para melhorar a navegação, futuramente, serão usadas câmeras no ambiente para identificar os marcos e fazer localização dos robôs, usando técnicas mais sofisticadas de Navegação Robótica.

Ainda como trabalhos futuros tem que ser tratada a possibilidade de *deadlock* do sistema quando os robôs participantes da montagem são dependentes das respostas dos outros para finalizar sua montagem. Por outro lado, a arquitetura de software apresentada na Seção

3.2 generaliza o uso de Sistema Multiagente para configurar ambientes, independentes de serem para finalidade laboratorial didática ou não.

Neste trabalho conclui-se que o uso da abordagem de Sistema Multiagente para configurar cenários é factível possibilitando a montagem autônoma de diferentes cenários, dependendo da escolha do usuário.

REFERÊNCIAS

- (AL-JUFOUT et al, 2011) Al-Jufout, Saleh; Jeschke, Sabina; Al-Zoubi, Abdullah Y.; Nsour, Jarir e Pfeiffer, Olivier. **Application of Remote Technology to Electrical Power System Laboratories**. Título do livro: Automation, Communication and Cybernetics in Science and Engineering, 2011. ISBN: 978-3-642-16208-4 DOI: 10.1007/978-3-642-16208-4_18
- (AWARD ICELIE, 2009) **Best Educational Tool Award at the ICELIE 2009**. URL: <http://www.uninova.pt/icelie2009/diplomas.htm>. Último acesso em: 22 de junho de 2012.
- (AUML, 2011) **Projeto AUML**. URL: <http://www.auml.org/auml/>. Último acesso em 2011.
- (BEZERRA et al, 2004) Bezerra, Gleisson de S.; Costa, Danielle F.; Favero, Eloi L.; Brito, Silvana R. de e Silva, Aleksandra do S. da. **FASE - Ferramenta de Autoria e Simulação de Experimentos**. In: World Congress on Engineering and Technology Education (CET) - WCETE 2004. Março de 2004, São Paulo, Brasil.
- (BOTELHO, 1998) Botelho, Silvia S. C. **A distributed scheme for task planning and negotiation in multi-robot systems**. 1998.
- (BOTELHO & ALAMI,1999) Botelho, Silvia S. C. e Alami, R. **M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement**. In: Proceedings of the 1999 IEEE International Conference on Robotics & Automation. Detroit, Michigan, May 1999. DOI: IEEE/ 0-7803-51 80-0-5/99
- (BRUCKER, 2006) Brucker Peter. **Scheduling Algorithms**. Berlin, 2006 : Springer. - Fifth : p. 376.
- (CAGILTAY et al, 2009) Cagiltay, N.E.; Aydin, E.; Oktem, R. e Kara, A.; **Requirements for Remote RF Laboratory Applications: An Educators' Perspective**. In: IEEE Transactions on Education Issue: 2009 Volume: 52 Issue:1 ISSN: 0018-9359 DOI: 10.1109/TE.2008.919806 Data de Publicação: 09 de Dezembro 2009.
- (CAGILTAY et al, 2010) Cagiltay, N.E.; Aydin, E. e Kara, A. **Principles for the design of a remote laboratory: A case study on ERRL**. In: IEEE Education Engineering (EDUCON). Data: 14-16 April 2010 Print ISBN: 978-1-4244-6568-2 DOI: 10.1109/EDUCON.2010.5492484 Data da versão corrente: 24 Junho 2010.
- (CAGILTAY et al, 2011) Cagiltay, N.E.; Aydin, E.; Aydin, C.C.; Kara, A. e Alexandru, M. **Seven Principles of Instructional Content Design for a Remote Laboratory: A Case Study on ERRL**. In: IEEE Transactions on Education Data: May 2011 Volume: 54 Issue:2 ISSN: 0018-9359 DOI: 10.1109/TE.2010.2058115 Data de Publicação: 03 Agosto 2011.
- (CAPES, 2011) O Portal de **Periódicos Capes** divulga e dá acesso aos recentes artigos de pesquisadores brasileiros indexados em diversas bases. URL: <http://www.periodicos.capes.gov.br/>

- (CASINI et al, 2009) Casini, Marco; Prattichizzo, Domenico e Vicino, Antonio. **A Matlab-Based Remote Lab for Control and Robotics Education**. Título do livro: Web-Based Control and Robotics Education Book Series Title: Intelligent Systems, Control and Automation: Science and Engineering ISBN: 978-90-481-2505-0 Volume: 38 DOI: 10.1007/978-90-481-2505-0_6 Copyright: 2009.
- (CHEN et al, 2010) Chen, Xuemin; Song, Gangbing; Zhang, Yongpeng. **Virtual and Remote Laboratory Development: A Review**. In: Proceedings of the 12th International Conference on Engineering, Science, Construction, and Operations in Challenging Environments. Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments ASCE Conf. Proc. DOI:10.1061/41096(366)368.
- (COSTELHA & LIMA, 2010) Costelha, Hugo e Lima, Pedro. **Petri Net Robotic Task Plan Representation :Modelling , Analysis and Execution**. In: InTechOpen Pages: 65-90 ISBN: 978-953-307-089-6 URL: <http://www.intechopen.com/articles/show/title/petri-net-robotic-task-plan-representation-modelling-analysis-and-execution>
- (CARVALHO et al, 2006) Carvalho, Sandro R.S.; Penatti, Tiê L. T. de; Roisenberg, Mauro. **Arquitetura Hierárquica de Comportamentos como Ferramenta de Projeto para Robôs Didáticos**. In: Anais do XXVI Congresso da SBC e III Encontro de Robótica Inteligente – EnRI. Campo Grande, MS, 2006. Pages: 309-317. URL <http://www.natalnet.br/sbc2006/pdf/arq0099.pdf>
- (CHOREN & LUCENA, 2003) Choren, R. e Lucena, C. J. P. **A Note: A Modeling Language for Agent-Based Systems**. ACM Transactions on Software Engineering and Methodology. 2003.
- (DAPONTE et al, 2010) Daponte, P.; Grimaldi, D. e Rapuano, S. **Recent Progresses of the Remote Didactic Laboratory LA.DI.RE “G. Savastano” Project**. Título do livro: Remote Instrumentation and Virtual Laboratories ISBN: 978-1-4419-5597-5 DOI:10.1007/978-1-4419-5597-5_34 Copyright: 2010.
- (DEITEL & DEITEL, 2003) Deitel, H. M.; Deitel, P. J. e Nieto, T. R. **Internet & World Wide Web como programar**. 2nd Edição. 2003. Página 612 Seção 20.4.1 - Definições de tipo de documento (DTDS) e esquemas.
- (DENIZ et al, 2003) Deniz, Derviş Z.; Bulancak, Atila; Özcan, Gökhan. **A Novel Approach to Remote Laboratories**. In Proceedings: 33 rd ASEE/IEEE Frontiers in Education Conference, pages 5-8. Doi:10.1109/FIE.2003.1315161 ISSN: 0190-5848 ISBN: 0-7803-7961-6. Data de versão corrente: 09 Agosto 2004. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1315161>
- (DOĞANE ERDAL, 2008) Doğan, Barış e Erdal, Hasan. **System Control Through the Internet and a Remote Access Laboratory Implementation**. Título do livro: Advances in Web Based Learning – ICWL 2007 Book Series Title: Lecture Notes in Computer Science ISBN: 978-3-540-78138-7 Volume: 4823 DOI: 10.1007/978-3-540-78139-4_47 Copyright: 2008

- (DELOACH & WOOD, 2001) Deloach, S. A. e Wood, M. F. **Developing Multiagent Systems with agentTool**. In: Proceedings of Lecture Notes in Artificial Intelligence. Springer – Verlag. Berling.2001.
- (DELOACH & GARCIA-OJEDA, 2010) DeLoach, Scott A. e Garcia-Ojeda, Juan Carlos. **O-MaSE: a customizable approach to designing and building complex, adaptive multiagent systems**.In: International Journal of Agent-Oriented Software Engineering. Volume 4, no. 3, 2010, pp. 244 - 280.
- (DIAS & STENTZ, 2000) Dias, M Bernardine e Stentz, Anthony. **A Free Market Architecture for Distributed Control of a Multirobot System**.In: 6th International Conference on Intelligent Autonomous Systems (IAS-6), July, 2000, pp. 115-122.
- (DUDEK & JENKIN, 2000) Dudek, Gregory e Jenkin, Michael. **Computational Principles of Mobile Robotics**. First published, 2000. Cambridge University Press. ISBN 0521568765. URL: <http://bit.ly/WskWNT>
- (E-LAB, 2011) **E-Lab** é um espaço onde podem ser realizadas experiências reais através da Internet.URL: <http://www.e-escola.pt/elab.asp>
- (IRMAK et al, 2011) Irmak, Erdal; Bayindir, Ramazan; Colak, Ilhami e Soysal, Mustafa. **A remote laboratory experiment for 4-quadrant control of a DC motor**.In: Computer Applications in Engineering Education Volume 19, Issue 4, pages 747–758, DOI: 10.1002/cae.20361, December 2011.
- (FIPA, 2011) <http://www.fipa.org/>
- (HONGYAN, 2010) Hongyan, Gao. **Development of a remote laboratory for process control experiments**. In: International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT). Issue Data: 17-18 Abril 2010 Print ISBN: 978-1-4244-5514-0 DOI: 10.1109/EDT.2010.5496474 Data de versão corrente: 28 Junho 2011.
- (GAGO et al, 2009) Gago, I. S. B.; Werneck, V. M. B. e Costa, R. M. **Modeling an Educational Multi-agent System in MaSE**.In: 5th international conference, AMT 2009, Beijing, China, Outubro 22–24. Proceedings. Berlin: Springer. Lecture Notes in Computer Science 5820, 335-346. Liu, Jiming (ed.) et al., Active media technology.
- (GARCÍA-ZUBIA et al, 2009a) García-Zubia, J.; Ângulo, I.; Orduña, P.; Irurzun, J.; Garibay, J. Ruiz de.; Hernandez, U. e Gonzalez, J.M. **Innovative Autonomous Hardware for Remote Experimentation with Microcontrollers**.In: 3rd IEEE International Conference on e-Learning in Industrial Electronics - ICELIE'2009.
- (GARCIA-OJEDA et al, 2007) Garcia-Ojeda,J. C.; DeLoach, Scott A.; Robby, Walamitien H. Oyenán e Valenzuela, Jorge. **O-MaSE: A Customizable Approach to Developing Multiagent Development Processes**.In: Proceedings of the 8th International Workshop on Agent Oriented Software Engineering, Honolulu HI, Maio 2007.
- (GARCÍA-ZUBIA et al, 2009b)Garcia-Zubia, J.; Orduna, P.; Lopez-de-Ipina, D. eAlves, G.R.**Addressing Software Impact in the Design of Remote Laboratories**.In: IEEE

- Transactions on Industrial Electronics. Issue Data: Dec. 2009 Volume: 56 Issue:12 ISSN: 0278-0046 DOI: 10.1109/TIE.2009.2026368 Data de publicação: 30 Junho 2009.
- (GARCÍA-ZUBIA et al, 2010) García-Zubia, J.; Ângulo, I.; Hernandez, U.; Castro, M.; Sancristobal, E.; Orduña, P.; Irurzun, J.; Garibay, J. Ruiz de. **Easily Integrable platform for the deployment of a Remote Laboratory for microcontrollers.** IEEE EDUCON 2010 (ISBN: 978-1-4244-6570-5). 2010.
- (GAUTHIR, 2007) Gauthir, Fernando Ostuni. **Disciplina Web Semântica - EGC6006, Material Aula 2 - Definição de Tipos de Documentos.** Ano: 2007. URL: [http://www.inf.ufsc.br/~gauthier/EGC6006/material/Aula%202/Definicao%20de%20Tipos%20de%20Documentos%20\(DTD\).pdf](http://www.inf.ufsc.br/~gauthier/EGC6006/material/Aula%202/Definicao%20de%20Tipos%20de%20Documentos%20(DTD).pdf)
- (GERKEY, 2003) Gerkey, B.P. **On Multi-Robot Task Allocation:** Dissertação de Doutorado / University Of Southern California. - Southern California : (s.n.).2003a.
- (GERKEY & MATARIC, 2003) Gerkey B.P. e Mataric, M.J. **Multi-robot task allocation: analyzing the complexity and optimality of key architectures.**In: IEEE International Conference on Robotics and Automation. 2003b.
- (GLUZ & VICARI, 2003) Gluz, J. C. e Vicari, Rosa Maria. **Linguagens de Comunicação entre Agentes: Fundamentos, Padrões e Perspectivas.** In: Renata Vieira; Fernando Santos Osorio; Solange Resende. (Org.). Livro Texto da III Jornada de Mini-Cursos de Inteligência Artificial - XXIII Congresso da SBC. 1 ed. SBC: Campinas, 2003, v. 8, p. 53-102.
- (GOMES, 2009) Gomes, L. Bogosyan, S. **Current Trends in Remote Laboratories.**Fac. ofSci. & Technol., Univ. Nova de Lisboa, Caparica, Portugal. In: IEEE Electronics Industrial Electronics Data: Dec. 2009 Volume: 56 page(s): 4744 - 4756 ISSN: 0278-0046 DOI: 10.1109/TIE.2009.2033293 Data de Publicação: 2009-10-06 11:07:57.0 Data da versão corrente: 06 Novembro 2009.
- (GRAVIER et al, 2008) Gravier, C.; Fayolle, J.; Bayard, B.; Ates, M. e Lardon, J. **State of the Art About Remote Laboratories Paradigms – Foundations of Ongoing Mutations.**In: International Journal of Online Engineering (iJOE). Volume 4, February 2008. ISSN: 1861-2121. URL: <http://www.online-journals.org/index.php/i-joe/article/view/480>
- (GUIMARAES et al, 2011) Guimaraes, E.G.; Cardozo, E.; Moraes, D.H. e Coelho, P.R. **Design and Implementation Issues for Modern Remote Laboratories.** In: IEEE Transactions on Learning Technologies. Data: Feb. 2011 Volume: 4 Issue:2 ISSN: 1939-1382 DOI: 10.1109/TLT.2010.22 Data de Publicação: 19 Agosto 2011.
- (HENDERSON-SELLERS & GIORGINI, 2005) Henderson-Sellers, Brian e Giorgini, Paolo. **Agent-oriented Methodologies.** Ed. 2005. 1ed: Idea Group Inc, London, UK, ISBN 1-59140-581-5, p412.
- (IVI, 2011) **IVI - Interchangeable Virtual Instruments Foundation.** URL: <http://www.ivifoundation.org/>

- (JING MA & NICKERSON, 2006) Jing Ma e Nickerson, Jeffrey V. **Hands-On, simulated, and remote laboratories: A comparative literature review.** In: ACM Computing Surveys, Vol. 38, No. 3, Article 7, Publication date: Setembro 2006. Doi:10.1145/1132960.1132961 URL: <http://portal.acm.org/citation.cfm?doid=1132960.1132961>
- (KUTLU & AYDOGAN, 2009) Kutlu, A. e Aydogan, T. **Performance Analysis of MicroNet: A Higher Layer Protocol for Multiuser Remote Laboratory.**In: IEEE Transactions on Industrial Electronics. Data: Dec. 2009 Volume: 56 Issue:12ISSN: 0278-0046 DOI: 10.1109/TIE.2008.2006228 Data de Publicação: 31 Outubro 2009.
- (CHOI et al, 2011) Choi, Kwansun; Han, Saeron; Kim,Dongsik; Jeon, Changwan; Lim, Jongsik; Lee, Sunheum; Park, Doo-soon e Jeon, Heunggu. **The Construction of Remote Microcontroller Laboratory Using Open Software.** Título do livro: Future Information Technology ISBN: 978-3-642-22333-4 Volume: 184 DOI: 10.1007/978-3-642-22333-4_16 Copyright: 2011.
- (LEVA & DONIDA, 2008) Leva, A. e Donida, F. **Multifunctional Remote Laboratory for Education in Automatic Control: The CrAutoLab Experience.**In: IEEE Transactions on Industrial Electronics. Data: June 2008 ISSN: 0278-0046 DOI: 10.1109/TIE.2008.922590 Data da versão corrente: 28 Maio 2008.
- (MACIEL, 2008) Maciel, Paulo. **Introdução aos Formalismos para Modelagem de Sistemas Concorrentes.** Slides do grupo de pesquisa MoDCS-ModellingofDistributedandConcurrent Systems do Centro de Informática Universidade Federal de Pernambuco. 2008. URL: <http://www.modcs.org/wp-content/uploads/2008/09/Formalismos2Handouts.pdf>
- (MARCO et al, 2011) Guerra, Marco André; Francisco, Cláudia Mariline e Madeira, Rui Neves. **Portablelab: Implementation of a Mobile Remote Laboratory for the Android Platform.**Technicalreport, Telecommunicationinstitute, Lisboa,Portugal, 2011.
- (MARRANGHELLO, 2005) Marranghello, Norian. **Redes de Petri: Conceitos e Aplicações.** No Curso Modelagem de Sistemas. Nível de Graduação ministrada em Março de 2005 pela DCCE/IBILCE/UNESP URL: <http://www.dcce.ibilce.unesp.br/~norian/cursos/mds/ApostilaRdP-CA.pdf>
- (MASÁR et al, 2011) Masár, I.; Bahník, P.; Batora, V. eHudacek, M.; **Design of an easy re-configurable remote laboratory.** In: 14th International Conference on Interactive Collaborative Learning (ICL). Date: 21-23 Sept. 2011 Print ISBN: 978-1-4577-1748-2 DOI: 10.1109/ICL.2011.6059605 Data da versão corrente: 27 Outubro 2011.
- (MDA, 2011) **MDA - Model Driven Architecture.** URL: <http://www.omg.org/mda/>
- (BORRERO & MÁRQUEZ, 2011) Borrero,A. MejíaseMárquez,J. M. Andújar. **A Pilot Study of the Effectiveness of Augmented Reality to Enhance the Use of Remote Labs in Electrical Engineering Education.** Journal Name: Journal of Science Education and Technology ISSN: 1059-0145 DOI: 10.1007/s10956-011-9345-9.

- (MITTAL et al, 2009) Mittal, Ankush; Pande, Amite Kumar, Praveen. **Content-based network resource allocation for real time remote laboratory applications.** Journal Name: Signal, Image and Video Processing Publisher: Springer London ISSN: 1863-1703 Volume: 4 Issue: 2 DOI: 10.1007/s11760-009-0116-5 Cover Date: 2009-06-01.
- (MOHTAR et al, 2008) Mohtar, A.; Nedic, Z. e Machotka, J. **A remote laboratory for microelectronics fabrication.** In: Frontiers in Education Conference, 2008. FIE 2008. 38th Annual Data: 22-25 Oct. 2008 ISSN: 0190-5848 Print ISBN: 978-1-4244-1969-2 DOI: 10.1109/FIE.2008.4720477 Data da versão corrente: 22 Dezembro 2008.
- (MOLLER et al, 2008) Moller, T.; Rosin, A.; Hoimoja, H. e Muur, M. **PLC based control of remote laboratory experiments.** In: 11th International Biennial Baltic Electronics Conference, 2008. BEC 2008. Data: 6-8 Oct. 2008 ISSN: 1736-3705 Print ISBN: 978-1-4244-2059-9 DOI: 10.1109/BEC.2008.4657533 Data da versão corrente: 24 Outubro 2008.
- (NAEF, 2006) Naef, O. **Real laboratory, virtual laboratory or remote laboratory: What is the most efficient way?** In: Proceedings. International Journal of Online Engineering (iJOE), Vol 2, No 3 (2006). ISSN 1861-2121. URL: <http://online-journals.org/index.php/i-joe/article/view/349>
- (NETTO, 2006) Netto, J. F. M. **Uma Arquitetura para Ambientes Virtuais de Convivência: uma Proposta Baseada em Sistemas Multiagente.** Tese de Doutorado, Universidade Federal do Espírito Santo, Vitória, 2006.
- (O'HARE & JENNINGS, 1996) O'Hare, Gregory J. M. e Jennings, Nick R. Preface. In: **Foundations of Distributed Artificial Intelligence.** New York: John Wiley & Sons, 1996, p. xii-ix (edited by G. M. P. O'Hare and N. R. Jennings). URL google livros: <http://migre.me/6180v>
- (LOPES, 2007) Lopes, Sara Patrícia de Medeiros Lacerda. **Laboratório de Acesso Remoto em Física.** Tese Mestrado. Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Departamento de Física.
- (LUHUA, 2009) Luhua, Li Pengfei Nie. **Remote Control Laboratory Based On LabVIEW.** In: Second International Conference on Intelligent Computation Technology and Automation, 2009. ICICTA '09. Data: 10-11 Oct. 2009 Volume: 4 Print ISBN: 978-0-7695-3804-4 DOI: 10.1109/ICICTA.2009.737 Data da versão corrente: 16 Outubro 2009.
- (QRBG, 2011) **Quantum Random Bit Generator Service - QRBG** é um serviço motivado pela comunidade científica em gerar números aleatórios para simulação de experimentos. URL: <http://random.irb.hr/signup.php> Acesso em Janeiro 2012.
- (RAJASEKHAR et al, 2008) Rajasekhar, Y. Kritikos; Schmidt, W.V. e R., A.G. Sass. **Teaching FPGA system design via a remote laboratory facility.** This paper appears in: International Conference on Field Programmable Logic and Applications, 2008. FPL 2008. Data: 8-10 Sept. 2008 Print ISBN: 978-1-4244-1960-9 INSPEC

Accession Number: 10250349 DOI: 10.1109/FPL.2008.4630040 Data da versão corrente: 23 Setembro 2008.

- (REAL, 2011) **REAL** é um laboratório que permite testar padrões de respostas de antenas. URL: <http://www.preal.ece.cmu.edu/>
- (REIS, 2003) Reis, Luís. Paulo. **Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico (Coordination in Multi-Agent Systems: Applications in University Management and Robotic Soccer)**, PhD Tese, FEUP, July of 2003. <http://paginas.fe.up.pt/~lpreis/Tese/Capitulo3.PDF>
- (RUSSELL & NORVIG, 2004) Russell, S. e Norvig, P. **Inteligência Artificial**. Ed. Campus, 2004.
- (SALIHBEGOVIC & RIBIC, 2008) Salihbegovic, Adnan e Ribic, Samir. **Development Of Online Internet Laboratory (Online I-Lab)**. Título do livro: Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education. ISBN: 978-1-4020-8739-4 DOI: 10.1007/978-1-4020-8739-4_1 Copyright: 2008.
- (SAMUELSEN & GRAVEN, 2011) Samuelsen, D.A.H. e Graven, O.H.; **Remote laboratory for advanced servo control of electric machines**. In: 14th International Conference on Interactive Collaborative Learning (ICL). Data: 21-23 Sept. 2011 Print ISBN: 978-1-4577-1748-2 DOI: 10.1109/ICL.2011.6059554.
- (SASAHARA, 2007) Sasahara, L.R.Hajime. **Uma nova abordagem em robótica educacional**. 2007. Rio de Janeiro: UFRJ e Universidade Estácio de Sá.
- (SECCHI, 2008) Secchi, Humberto Alejandro. **Una Introducción a los Robots Móviles**. Instituto de Automática – INAUT. Universidade Nacional de San Juan – UNSJ – Argentina. Edição: Agosto de 2008. Tradução: “Uma Introdução aos Robôs Móveis” Cynthia N. de Almeida e Felipe N. Martins. NER@ – Núcleo de Estudos em Robótica e Automação IFES – Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo Edição: Abril de 2012.
- (SCORM, 2011) **SCORM - Sharable Content Object Reference Model** URL: <http://scorm.com/pt/> Data Acesso em Janeiro 2012.
- (SILVA, 2008) Silva, Luciana Leal da. **Classificação e Caracterização de Ambientes para Navegação de Robôs Móveis Baseada em Mapas**. Dissertação Mestrado em Ciência da Computação. São José do Rio Preto, 2008. URL: <http://bit.ly/XQGnHS>
- (SOWA, 1999) Sowa, John F. **Knowledge Representation: Logical, Philosophical, and Computational Foundations**. Brooks Cole Publishing Co., Pacific Grove, CA, 2000. Actual publication date, 16 August 1999.
- (SOUZA & NETTO, 2010) Souza, M. B. e Netto, J. F. M. **Modelagem do Laboratório de Acesso Remoto de Robótica Educacional Utilizando a Metodologia MaSE**. Escola Regional de Informática - ERIN 2010. Manaus.

- (SOUZA, 2011) Souza, M.B. **Arcabouço de um Ambiente Telerobótico Educacional Baseado em Sistemas Multiagentes**. Dissertação Mestrado Universidade Federal do Amazonas, Julho 2011.
- (SOUZA et al, 2011) Souza, M. B.; Netto, J. F. M. ; Alencar, M. A. S. e Silva, M. M. . **Arcabouço de Um Ambiente Telerobótico Educacional Baseado em Sistemas Multiagente**. In: XXII Simpósio Brasileiro de Informática na Educação e XVII Workshop de Informática na Escola, 2011, Aracaju. Ambientes de Aprendizagem, 2011. p. 680-689.
- (STRAATSMA et al, 2009) Straatsma, M.; Cox, D.; Cstis, C. eBartz, R. **Development and Enhancement of RLab - A Remote Laboratory System**. In: Fourth International Conference on Systems and Networks Communications, 2009. ICSNC '09. Data: 20-25 Sept. 2009 Print ISBN: 978-1-4244-4772-5 DOI: 10.1109/ICSNC.2009.97 Data da versão corrente: 06 Outubro 2009.
- (VICENTE et al, 2010) Vicente, A.G.; Muñoz, I.B.; Galilea, J.L.L. edel Toro, P.A.R. **Remote Automation Laboratory Using a Cluster of Virtual Machines**. In: IEEE Transactions on Industrial Electronics. Data: Oct. 2010 Volume: 57 Issue:10 ISSN: 0278-0046 DOI: 10.1109/TIE.2010.2041131 Data de Publicação: 08 Fevereiro 2010.
- (WEBLAB-DEUSTO, 2012) **Weblab-Deusto -Research group of the University of Deusto**. URL: <https://www.weblab.deusto.es/web/>. Último acesso em: 22 de junho de 2012.
- (WENSHAN et al, 2011) Wenshan Hu; Hong Zhou e Qijun Deng. **Design of web-based 3D control laboratory**. This paper appears in: 2nd International Conference on Intelligent Control and Information Processing (ICICIP). Data: 25-28 July 2011 Print ISBN: 978-1-4577-0813-8 DOI: 10.1109/ICICIP.2011.6008318 Data da versão corrente: 01 Setembro 2011.
- (WERNECK et al, 2011) Werneck, Vera Maria B.; Costa, Rosa Maria E. Moreira e Cysneiros, Luiz Marcio. **Modelling Multi-Agent System using Different Methodologies**. In: InTechOpen. Multi-Agent Systems - Modeling, Interactions, Simulations and Case Studies, ISBN 978-953-307-176-3, publishing date: April 2011
- (WILEY, 2007) Wiley. **Developing Multi Agent Systems with JADE**. Livro publicado em Abril de 2007, página 40.
- (WONSHIL et al, 2010) Wonshil, Kang; Ho-Gil, Jo; Hyunchul, Ku e Yong, Kim. **Electronics & Telecommunications Remote Laboratory for RF transmitter and power amplifier test**. This paper appears in: Asia-Pacific Microwave Conference Proceedings (APMC). Data: 7-10 Dec. 2010 Print ISBN: 978-1-4244-7590-2 Data da versão corrente: Março 2011.
- (WOOD et al, 2001) Wood, Mark F. e DeLoach, Scott A. **An Overview of the Multiagent Systems Engineering Methodology**. In: Proceedings of the 1st International Workshop on Software Engineering. Springer Verlag, Berlin, Janeiro 2001.
- (WOOLDRIDGE, 2002) Wooldridge, M. **An Introduction to Multiagent Systems**. John Wiley and Sons Ltd, Fev. 2002.

- (WOOLDRIDGE E JENNINGS, 1995) Wooldridge, M. e Jennings, N. R. **Intelligent agents: Theory and practice**. 1995. Knowl. Eng. Rev. 10,2,115–152. URL: <http://www.csc.liv.ac.uk/~mjw/pubs/ker95/ker95-html.html>
- (XML SPY, 2013) **XML SPY - Altova XML editor**. 2013. XML editor and the industry standard XML development environment for modeling, editing, transforming, and debugging XML-related technologies. URL: <http://www.altova.com/simpledownload2c.html>
- (ZAMBONELLI et al, 2003) Zambonelli, F.; Jennings, N. R.e Wooldridge, M. (2003). **Developing multiagent systems: The gaia methodology**. ACM Trans. Software Eng. Methodol., 12(3):317–370.

ANEXOS

ANEXO A – EXEMPLOS DE LAR

TÍTULO DO ARTIGO:	(MOHTAR et al, 2008) A Remote Laboratory for Microelectronics Fabrication.
NOME DO PROJETO:	Extensão do NetLAB.
EM OPERAÇÃO (sim/não):	Informação não encontrada - N.E
CAMPO CIENTÍFICO:	Eletrônica (construir circuitos microeletrônicos).
ARQUITETURA DE SOFTWARE:	Usuário, servidor e dispositivo remoto.
TECNOLOGIA DA ARQUITETURA:	LabView Server.
FERRAMENTA DE COMUNICAÇÃO:	N.E
RESUMO:	A <i>University of South Australia</i> (UniSA) vem desenvolvendo um laboratório remoto onde usuário pode inspecionar e testar circuitos microeletrônicos sobre um microscópio. Trabalhos semelhantes: WebLab (2002), AIM-Lab (1999), LAB-on-WEB (2002), Next-Generation (2002) e RETWINE (2002).

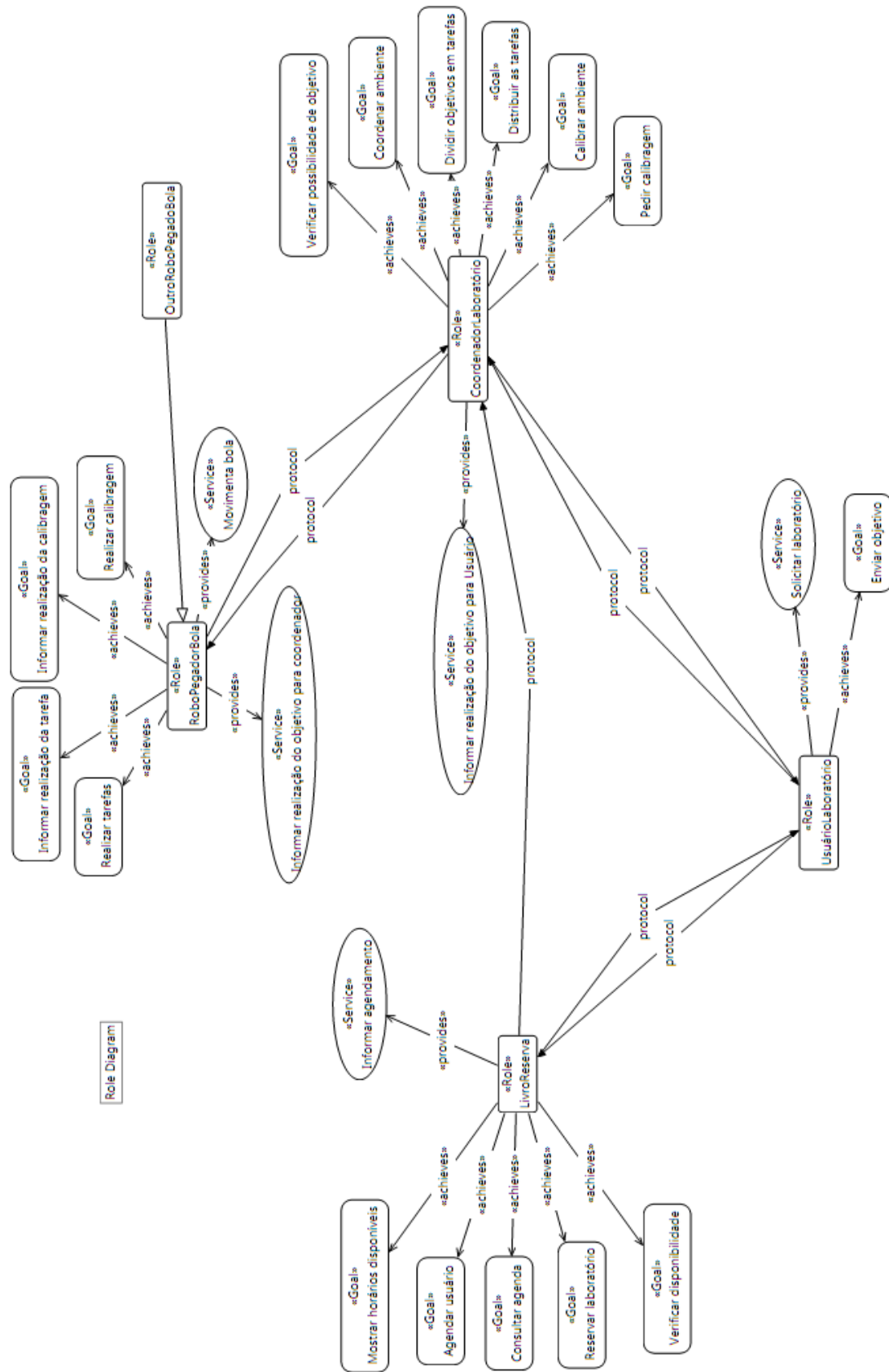
TÍTULO DO ARTIGO:	(RAJASEKHAR et al, 2008) Teaching FPGA System Design VIA A Remote Laboratory Facility.
NOME DO PROJETO:	System Design Using FPGA Board.
EM OPERAÇÃO (sim/não):	N.E
CAMPO CIENTÍFICO:	Computador.
ARQUITETURA DE SOFTWARE:	Usuário, servidor e dispositivo remoto.
TECNOLOGIA DA ARQUITETURA:	VHDL e Verilog
FERRAMENTA DE COMUNICAÇÃO:	N.E
RESUMO:	A <i>University of North Carolina</i> desenvolve uma 'plataforma FPGA (também conhecido Systems-on-a-Reconfigurable-Chip) para design como laboratório remoto.

TÍTULO DO ARTIGO:	(LEVA & DONIDA, 2008) Multifunctional Remote Laboratory for Education in Automatic Control: The CrAutoLab Experience.
NOME DO PROJETO:	CrAutoLabexperiments.
EM OPERAÇÃO (sim/não):	Não.
CAMPO CIENTÍFICO:	Multidisciplinar.
ARQUITETURA DE SOFTWARE:	Usuário, servidor e dispositivo remoto
TECNOLOGIA DA ARQUITETURA:	LabView Web Server.
FERRAMENTA DE COMUNICAÇÃO:	N.E
RESUMO:	Laboratório remoto localizado em Cremona na <i>Politecnico di Milano</i> . Vários experimentos podem ser realizados como: PID control of a LTI SISO system, Control of a LTI SISO system with a 2-d.o.f. LTI regulator, Single-loop temperature control, Temperature control with measured disturbance compensation, Temperature control with cascade structure, Multivariable temperature control, decentralised or with decoupling, Speed control with flexible transmission e Speed control with eccentric mechanical load and gravity effect.

TÍTULO DO ARTIGO:	(MOLLER et al, 2008) PLC Based Control of Remote Laboratory Experiments.
NOME DO PROJETO:	N.E
EM OPERAÇÃO (sim/não):	N.E
CAMPO CIENTÍFICO:	Eletrônica.
ARQUITETURA DE SOFTWARE:	Usuário, servidor e dispositivo remoto.
TECNOLOGIA DA ARQUITETURA:	Java Applets.
FERRAMENTA DE COMUNICAÇÃO:	Serviço de e-mail.
RESUMO:	Laboratório de elétrica onde pode programar, controlar e verificar um PLC (<i>Programmable Logic Controller</i>) a distância.

Tabela completa: <http://bit.ly/TWqYrs>

ANEXO B – HIERARQUIA DE PAPEIS EM VISÃO GERAL



ANEXO C – PUBLICAÇÕES REALIZADAS

An Educational Robotic Game for Transit Education Based on the Lego MindStorms NXT Platform

Marckson Monteiro da Silva José Francisco de Magalhães Netto

Universidade Federal do Amazonas, Departamento de Ciência da Computação, Brasil

Abstract

This paper describes the development of an educational game using Lego MindStorms NXT 1.0 platform of educational robotics and Java technology, seeking the construction of knowledge about the Transit Education for children, in a fun and interesting way. With this game children can experience situations of urban traffic, leading to reflections on what attitude to take in view of typical situations. The game seeks to simulate urban traffic close to real-world situation.

Keywords: Educational robotics, Educational games, Lego MindStorms NXT.

Authors' contact:

marcksonms@gmail.com.br
jnetto@dcc.ufam.edu.br

development; Section 6 discusses evaluation, verification and validation. Section7 presents our conclusions and future works.

2. Motivation

According to the latest DENATRAN¹'s yearbook, there were 653,827 transit accidents with 33,996 fatalities, 105,337 of these were motorists. The punishment for transit violations have become tougher, but still were not enough to reduce accidents.

Much of the motorists have an aggressive and violent behavior in transit. These attitudes are associated with lack of Transit Education. For this reason new laws were added to the Brazilian Traffic Code in order to promote Transit Education and the development of new supportive tools.

[doi>](#) **SILVA, M. M.** ; **NETTO, J. F. M.** . An Educational Robotic Game for Transit Education Based on the Lego MindStorms NXT Platform. In: Brazilian Symposium on Games and Digital Entertainment, 2010, Florianopolis. computing track, 2010. p. 82-88.

Desenvolvimento de um Ambiente Virtual para Apoio a Experimentos em Telerobótica

Marckson M. Silva, José F. M. Netto

Universidade Federal do Amazonas, Dept. de Ciência da Computação
Manaus – AM – Brasil

marcksonms@gmail.com.br, jnetto@dcc.ufam.edu.br

Abstract. *This article describes the methodology, proposed and results of implementation of work-PIBIC UFAM year 2009/2010 about the development of a virtual environment telerobotic. In this project we propose a telerobotic environment with the purpose of allowing students from our region have access to telerobotic experiments, thus allowing access this technology to regions where we lack laboratory robotics.*

SILVA, M. M. ; **NETTO, J. F. M.** . Desenvolvimento de um Ambiente Virtual para Apoio a Experimentos em Telerobótica. In: Escola Regional de Informática, 2010, Manaus. Informática e os Desafios Regionais, 2010.

Configuração Automática de um Laboratório de Acesso Remoto por Meio de Agentes e Framework O-MaSE

Marckson M. da Silva, José Francisco M. Netto

Instituto de Computação (IComp/PPGI) – Universidade Federal do Amazonas (UFAM)
Av. Gen. Rodrigo Otávio, 3.000, Coroado – Manaus – AM – Brasil

{marcksonms, jnetto}@icompu.ufam.edu.br

Resumo. *Nesse artigo é apresentado o problema de configuração de cenários experimentais em Laboratórios de Acesso Remoto (LARs) e como esse problema é resolvido usando a abordagem de Sistemas Multiagente (SMA). Para apresentar como resolver o problema, o SMA é modelado usando a metodologia O-MaSE. Esta metodologia permite modelar SMA a partir da abordagem organizacional e é focada em objetivo do SMA e não em cada agente de forma isolada.*

SILVA, M. M. ; **NETTO, J. F. M.** . Configuração Automática de um Laboratório de Acesso Remoto por Meio de Agentes e Framework O-MaSE.. In: Simpósio Brasileiro de Informática na Educação, 2012, Rio de Janeiro. Tecnologias da Informação e a Integração da Academia, Escolas, Governo e Empresas para uma Educação Sustentável, 2012. v. 23.

ANEXO D – DADOS COLETADOS

Coleta de dados para robô garra simples

Número da amostra ou Ciclo de experimentação	Quantidade de experimentos	Quantidade experimento realizado com sucesso	Contador de experimento realizado com sucesso	Observado		Esperado	
				X	Y	X	Y
				1	6	3	1
			2	367	332	350	310
			3	383	302	350	310
2	6	5	4	345	315	350	310
			5	338	302	350	310
			6	349	311	350	310
			7	346	307	350	310
			8	356	311	350	310
3	5	2	9	359	313	350	310
			10	346	320	350	310
4	7	6	11	340	322	350	310
			12	348	306	350	310
			13	326	326	350	310
			14	323	332	350	310
			15	319	332	350	310
			16	339	317	350	310
5	12	12	17	335	309	350	310
			18	339	315	350	310
			19	336	335	350	310
			20	331	316	350	310
			21	345	316	350	310
			22	333	329	350	310
			23	340	316	350	310
			24	325	333	350	310
			25	342	317	350	310
			26	332	324	350	310
			27	342	312	350	310
			28	346	301	350	310

Coleta de dados robô garra dupla

Número da amostra ou Ciclo de experimentação	Quantidade de experimentos	Quantidade experimento realizado com sucesso	Contador de experimento realizado com sucesso	Observado		Esperado	
				X	Y	X	Y
				1	4	1	1
2	6	3	2	468	1084	480	985
			3	482	997	480	985
			4	522	998	480	985
3	13	4	5	423	1039	480	985
			6	483	1009	480	985
			7	481	1009	480	985
			8	447	1006	480	985
4	11	10	9	495	1000	480	985
			10	363	1033	480	985
			11	468	997	480	985
			12	543	961	480	985
			13	516	961	480	985
			14	471	1033	480	985
			15	399	1027	480	985
			16	450	979	480	985
			17	531	976	480	985
			18	485	979	480	985
5	2	2	19	525	966	480	985
			20	537	955	480	985