

**DESENVOLVIMENTO DE INTERFACES EM
DIFERENTES TECNOLOGIAS PARA
TELEOPERAÇÃO DE ROBÔS MÓVEIS**

LUÍS FELIPE HUSSIN BENTO

**DESENVOLVIMENTO DE INTERFACES EM
DIFERENTES TECNOLOGIAS PARA
TELEOPERAÇÃO DE ROBÔS MÓVEIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Departamento de Ciência da Computação. como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADORA: RAQUEL OLIVEIRA PRATES

CO-ORIENTADOR: LUIZ CHAIMOWICZ

Belo Horizonte, MG

Setembro de 2010

© 2010, Luís Felipe Hussin Bento.
Todos os direitos reservados.

B478d Bento, Luís Felipe Hussin
Desenvolvimento de interfaces em diferentes
tecnologias para teleoperação de robôs móveis / Luís
Felipe Hussin Bento. — Belo Horizonte, MG, 2010
xx, 92 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais. Departamento de Ciência da
Computação.

Orientador: Raquel Oliveira Prates

Coorientador: Luiz Chaimowicz

1. Interação Humano-Robô. 2. Interação
Humano-Computador. 3. Signos. 4. Classes.
5. Engenharia Semiótica. 6. Método de Inspeção
Semiótica. 7. Robótica. 8. e-Puck. 9. Projeto de
interfaces. I. Título.

CDU 519.6*82.9



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Desenvolvimento de interfaces em diferentes tecnologias para teleoperação de robôs móveis

LUÍS FELIPE HUSSIN BENTO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

A handwritten signature in blue ink, reading "Raquel Prates".

PROFA. RAQUEL OLIVEIRA PRATES - Orientadora
Departamento de Ciência da Computação - UFMG

A handwritten signature in blue ink, reading "Luiz Chaimowicz".

PROF. LUIZ CHAIMOWICZ - Co-orientador
Departamento de Ciência da Computação - UFMG

A handwritten signature in blue ink, reading "Alberto Barbosa Raposo".

PROF. ALBERTO BARBOSA RAPOSO
Departamento de Informática - PUC-RJ

A large, stylized handwritten signature in blue ink, reading "Mario Fernando Montenegro Campos".

PROF. MARIO FERNANDO MONTENEGRO CAMPOS
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 23 de setembro de 2010.

Agradecimentos

Primeiramente, gostaria de agradecer aos meus pais, Jaquison e Márcia, por todo o apoio que tive desde sempre. Sem este suporte, certamente não teria chegado onde cheguei, concluindo mais essa etapa. Gostaria de agradecer também à Raquel Prates, minha orientadora, e ao Luiz Chaimowicz, meu co-orientador. Além de me guiarem durante os anos de mestrado, com sabedoria e paciência, foram também amigos que me acompanharam nessa jornada.

Aos velhos amigos, que mantiveram a amizade mesmo com as visitas menos frequentes, e aos novos amigos, que também passaram pelas mesmas experiências acadêmicas e ajudaram a levar o projeto adiante. Por fim, agradeço aos outros moradores da república onde moro (ex e atuais moradores): Hadriel, Leonardo, Henrique e Filipe, e também ao Gabriel e Marcos, principalmente pela distração e companhia, que ajudaram a manter o espírito elevado.

Agradeço ao pessoal do PENS*i* e do VerLab pela companhia durante todo esse tempo em que compartilhamos dificuldades, e em especial ao Wolmar, que sempre faz de tudo pra resolver nossos problemas dentro do VerLab com muita boa vontade. Agradeço também ao prof. Robson e ao Synergia, que me deu a oportunidade de poder trabalhar e me manter em Belo Horizonte durante o período do mestrado.

*“There is an art, it says, or rather, a knack to flying. The knack lies in learning how
to throw yourself at the ground and miss.”*

(Douglas Adams - The Hitchhiker's Guide to the Galaxy)

Resumo

A área de pesquisa de interação humano-robô (*HRI - Human-Robot Interaction*) vem levantando cada vez mais interesses de pesquisadores. É um campo de pesquisa inerentemente multidisciplinar, recebendo contribuições tanto da robótica e da interação humano-computador quanto das ciências humanas, como psicologia, sociologia e antropologia. Este crescente interesse é motivado pelo número cada vez maior de robôs atuando na sociedade (e.g., entretenimento, área médica e utilidades domésticas) e interagindo com pessoas que não são especialistas em robótica, gerando uma preocupação maior com a qualidade da interação entre pessoas e robôs.

O trabalho propõe classes de signos para a interação humano-robô com base na engenharia semiótica, uma teoria de IHC que considera a interação humano-computador como um caso particular da comunicação entre humanos mediada por computador. Através da aplicação de um método de inspeção oriundo da teoria, o método de inspeção semiótica, em interfaces já existentes para controle de um robô *e-Puck*, um pequeno robô para fins acadêmicos, as classes de signos que compõem o sistema de significação foram identificadas. Para representar e avaliar este sistema de significação foi criado um modelo de interação que deu origem a três interfaces para controlar o *e-Puck*: uma para *desktop PCs*, através do uso de mouse e teclado; outra para *tablet PCs*, através da interação com a caneta diretamente na tela; e, por fim, uma para um dispositivo móvel (o *iPod touch*®) que permite a interação através de toques na tela e da movimentação do próprio dispositivo, através de seus sensores de movimento.

As interfaces foram avaliadas com a participação de usuários e os resultados obtidos dessa avaliação foram positivos. Estes indicaram que o sistema de significação é adequado à interação com robôs, sendo que isso é válido não só para os tipos de tarefas e o robô abordados, mas também pode ser extrapolado para tarefas mais complexas e outros tipos de robôs. O trabalho também permitiu identificar algumas limitações e estratégias de interação para cada um dos três tipos de interfaces avaliadas.

Abstract

Human-Robot Interaction (HRI) is a novel field that is attracting researchers from various areas, such as robotics and human-computer interaction as well as from the field of the humanities including psychology, anthropology and sociology. This growing interest is motivated by the increasing number of robots being used in society (e.g., entertainment, health care and home activities) by people who are not specialists, creating demands on interaction quality.

This work proposes the identification of sign classes for the human-robot interaction based on the semiotic engineering, an HCI theory that considers the interaction between people and computers as a particular case of the computer-mediated communication between humans. With the application of an inspection method based on the theory, the Semiotic Inspection Method, the existing interfaces used to control an e-Puck robot, a small robot for academic uses, the sign classes that compose the signification system were identified. Then, a single interaction model was created to represent and evaluate this signification system, resulting in three different interfaces to control the e-Puck: one for desktop PCs, by using the keyboard and mouse; other for tablet PCs, based on the interaction using a pen directly on the screen; and one for a mobile device (an iPod touch®), which allows interaction through its touch screen and the movement of the device itself.

The interfaces were evaluated with user participation and the obtained results were promising. They indicated that the signification system is adequate to the interaction with robots, and not only to the kind of tasks and robots mentioned, but also to more complex tasks and other robot types. The work also allowed to identify some limitations and interaction strategies regarding the three interface types evaluated.

Lista de Figuras

3.1	Robô ePuck	20
3.2	Interface do E-Puck Monitor	22
3.3	Descrição do E-Puck Monitor no site da EPFL	23
3.4	Interface do ePic	30
3.5	Signos, modelagem e interfaces	35
4.1	Representação de uma cena na MoLIC	39
4.2	Representações de processos do sistema na MoLIC	39
4.3	Representações de falas de transição na MoLIC	40
4.4	Representação de acesso ubíquo na MoLIC	41
4.5	Diagrama de metas	42
4.6	Diagrama de interação	43
4.7	Arquitetura do sistema	47
4.8	Interface desktop para controle do ePuck	48
4.9	Signo para representar o estado da conexão com o robô na interface para desktop	48
4.10	Signo para representar a posição e a orientação do robô na interface desktop	49
4.11	Signo para representar a movimentação do robô na interface desktop (robô parado)	50
4.12	Signo para representar a movimentação do robô na interface desktop (robô em movimento)	50
4.13	Signos para representar o trajeto do robô na interface Desktop	51
4.14	Signos que representam os sensores de proximidade na interface Desktop .	51
4.15	Signos que representam a câmera e a imagem fotografada na interface Desktop	52
4.16	Exemplo de uso de um Tablet PC.	52
4.17	Interface tablet para controle do ePuck.	53
4.18	Caminho do robô na interface para Tablet PC	54
4.19	iPod touch®	54

4.20	Interface iPod para controle do ePuck na tela de conexão	55
4.21	Interface iPod para controle do ePuck no modo de controle direto	55
4.22	Interface iPod para controle do ePuck no modo de controle por caminho	56
4.23	Representação do controle do robô por acelerômetro	57
4.24	Exemplo do indicador de inclinação	57
5.1	Arena de testes	60
5.2	ePuck com padrão para localização	60
5.3	e-Puck empurrando um objeto	62
5.4	Local que deve ser fotografado	62
5.5	Imagem obtida pela câmera do <i>e-Puck</i>	63
5.6	Esboço da nova representação para o signo de movimentação nas interfaces para desktop e tablet	72
5.7	Representação do recurso pinch-to-zoom	73
A.1	Termo de consentimento	86
A.2	Script para a avaliação	87
A.3	Questionário pré-teste	88
A.4	Cenário e descrição geral das tarefas	89
A.5	Descrição das tarefas para cada interface	90
A.6	Formulário de acompanhamento do teste	91
A.7	Roteiro básico para a entrevista pós-teste	92

Lista de Tabelas

4.1	Esquema conceitual de signos	45
5.1	Resumo dos testes	71
5.2	Observações gerais nos testes	72

Sumário

Agradecimentos	vii
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Motivação	3
1.2 Objetivos do Trabalho	3
1.3 Organização da dissertação	4
2 Panorama atual e Fundamentação Teórica	5
2.1 Panorama da interação humano-robô	5
2.2 Fundamentação Teórica	14
2.2.1 Engenharia Semiótica	14
2.2.2 Método de Inspeção Semiótica	16
3 Classes de signos para a interação humano-robô	19
3.1 <i>e-Puck</i> , o robô utilizado na pesquisa	19
3.2 Aplicação do MIS	21
3.2.1 Preparação	21
3.2.2 Inspeção dos signos metalinguísticos	22
3.2.3 Inspeção dos signos estáticos	23
3.2.4 Inspeção dos signos dinâmicos	25
3.2.5 Metacomunicação consolidada e apreciação da qualidade de interação	27

3.2.6	Triangulação com o <i>ePic</i>	29
3.3	Classes de Signos	33
4	Desenvolvimento das Interfaces	37
4.1	MoLIC	37
4.2	Modelagem	41
4.2.1	Modelo de Interação	41
4.2.2	Definição dos tipos de signos	44
4.3	Implementação	46
4.3.1	Servidor de controle do <i>e-Puck</i>	46
4.3.2	Interface para Desktop PCs	47
4.3.3	Interface para Tablet PCs	51
4.3.4	Interface para <i>iPod touch</i> ®	53
5	Estudo Comparativo	59
5.1	Preparação	61
5.1.1	Limitações da avaliação	63
5.2	Aplicação dos testes	64
5.3	Resultados da avaliação	65
5.3.1	<i>iPod touch</i> ®	65
5.3.2	<i>Tablet PC</i>	67
5.3.3	<i>Desktop PC</i>	68
5.3.4	Apreciação geral	68
6	Conclusões e Trabalhos Futuros	75
	Referências Bibliográficas	79
A	Documentos utilizados na avaliação com os usuários	85

Capítulo 1

Introdução

A interação entre pessoas e robôs sempre foi uma área ativa na robótica [Burke et al., 2004], mas normalmente restrita aos laboratórios de pesquisa ou em atividades altamente especializadas que demandavam que os desenvolvedores dos robôs estivessem presentes para que as tarefas fossem realizadas. Desta forma não havia necessidade de se preocupar com a qualidade da interação, já que os próprios desenvolvedores interagiam com os robôs, permitindo o uso de interfaces complexas que iriam requerer de outros usuários um treinamento intensivo [Richer & Drury, 2006].

Robôs manipuladores já são amplamente utilizados em parques industriais, principalmente em linhas de montagem (e.g. fábricas de automóveis). Robôs móveis, por sua vez, são aplicados em vários contextos diferentes, como busca e resgate [Drury et al., 2005, Wang & Lewis, 2007], exploração espacial [Fong et al., 2001a], monitoramento aéreo [Richer & Drury, 2006] e detecção de bombas [Humphrey et al., 2007a]. Nesses casos os robôs são tratados como ferramentas avançadas que são controladas remotamente por um supervisor humano [Breazeal, 2004].

Em muitas destas aplicações, a autonomia dos robôs é necessária, pois muitas das decisões que devem ser tomadas excedem a capacidade cognitiva de um operador humano [Wang & Lewis, 2007], *i.e.*, exigem uma quantidade de informações que uma pessoa teria grande dificuldade em processar no tempo necessário. Mas a supervisão humana também se faz necessária para gerenciar mudanças no ambiente e tomar decisões cabíveis.

À medida que os robôs se aproximam da sociedade, seja na execução das atividades já citadas ou em outras como entretenimento (e.g. robô Sony®Aibo™), aplicações domésticas (e.g. aspirador de pó robótico Roomba™), suporte a pessoas com necessidades especiais ou a idosos [Feil-Seifer & Mataric, 2005] ou até mesmo guias turísticos

[Shiomi et al., 2006], mais usuários leigos em robótica passam a interagir com estes robôs, sem obter suporte direto de especialistas. Esse cenário torna necessária a preocupação com a qualidade desta interação.

Essa preocupação deu origem à Interação Humano-Robô (*HRI - Human-Robot Interaction* ou IHR, em português), uma área recente e multidisciplinar, que envolve várias áreas diferentes, tanto no contexto da computação, como da robótica, IHC e inteligência artificial, quanto das ciências humanas, como psicologia, antropologia, ciências cognitivas e sociologia.

O campo de Interação Humano-Computador (IHC) possui várias teorias (e.g. teoria da atividade [Kaptelinin & Nardi, 2006]) e técnicas (e.g. avaliação heurística [Nielsen, 1992] e percurso cognitivo [Rieman et al., 1995]) para se avaliar a interação de usuário com sistemas. O presente trabalho será embasado na teoria da engenharia semiótica [de Souza, 2005b], uma teoria de IHC baseada na semiótica¹ que considera a interação humano-computador como um caso particular da comunicação entre humanos mediada por computador. Nesta interação os projetistas e desenvolvedores comunicam aos usuários, através dos signos² presentes no sistema, as suas intenções e visões do projeto, trazendo à mesma luz os projetistas, o sistema produzido e os usuários. A semiótica define um signo como qualquer entidade que é tomada por alguém para representar alguma outra coisa. Já significação é um processo no qual certos sistemas de signos são estabelecidos através de convenções sociais ou artificiais, como ocorre em IHC, adotadas pelos usuários de tais signos [de Souza, 2005a]. Dessa forma, um sistema de significação é, então, um conjunto destes signos adotados em um determinado contexto. Para a interação humano-computador, um sistema compreende a associação dos elementos de comunicação presentes em uma interface com os significados que os mesmos possuem no contexto onde a aplicação se encontra.

Desta teoria surge o MIS – Método de Inspeção Semiótica – [de Souza et al., 2006][de Souza & Leitão, 2009], que será utilizado para a identificação de signos, ou seja, elementos de comunicação presentes (e.g palavras, sons, comportamentos, elementos de interface), que sejam relevantes para a interação humano-robô, constituindo classes de signos que possam representar essa interação.

¹A semiótica é o estudo de signos, processos de significação e como os signos e a significação participam da comunicação[de Souza, 2005b]

²Um signo é “qualquer coisa que, sob certo aspecto ou modo, representa algo para alguém”, de acordo com [Peirce, 1931]

1.1 Motivação

Um dos principais desafios da robótica é a execução de tarefas em ambientes nos quais estejam presentes tanto robôs quanto humanos [Kemp et al., 2007]. A maioria das soluções apresentadas são restritas a aplicações específicas que são realizadas em ambientes controlados.

A interação com robôs pode se dar de várias formas diferentes: utilizando interfaces de software desktop ou computadores portáteis [Fong et al., 2001a], interfaces de toque (e.g. tablet PCs [Skubic et al., 2006]), interfaces tangíveis [Guo & Sharlin, 2008], comandos de voz e também gestuais [Perzanowski et al., 2001]. Estas variadas formas de interação geram questões em que os métodos tradicionais de IHC não conseguem ser aplicados de forma satisfatória.

[Fong et al., 2001c] diz que a interação humano-robô difere da interação humano-computador porque trata de sistemas que possuem sistemas de controle dinâmicos e complexos, exibem autonomia e cognição e operam em ambientes reais que estão em constante mudança. Além disso, essas diferenças ocorrem nos tipos (papéis) de interação, na natureza física dos robôs, no número de sistemas que o usuário poderá ter que interagir simultaneamente e no ambiente onde esta interação ocorre.

HRI, como dito anteriormente, é um campo de pesquisa recente. Dessa forma há muitas questões ainda em aberto, como as citadas em [Gold et al., 2007]. No Brasil praticamente não há trabalhos na área ([Junior et al., 2008] foi o único encontrado, tratando de um ambiente 3D para interação com manipuladores robóticos), então este trabalho pode contribuir para o surgimento de uma comunidade nacional de HRI e alavancar as pesquisas neste campo.

1.2 Objetivos do Trabalho

Este trabalho pretende, primeiramente, identificar e propor classes de signos para sistemas que promovem a interação entre pessoas e robôs. Esta interação será limitada a pequenos robôs diferenciais, como o *e-Puck* [Mondada et al., 2009]. A proposta do sistema será feita tanto com base em trabalhos relacionados quanto com base na inspeção de sistemas de interação humano-robô existentes para a identificação de signos relevantes nesta interação.

Com estas classes de signos devidamente identificadas, serão implementadas diversas formas de interação para representar o sistema proposto, utilizando diferentes linguagens de interface e tecnologias. Ao longo do trabalho serão desenvolvidas linguagens visuais para trabalhar com interfaces de toque, como PDAs [Fong et al., 2001a]

ou tablet PCs [Skubic et al., 2006], e interfaces *desktop*, cuja interação se dá por mouse e teclado.

Será feito então um estudo comparativo das tecnologias e de suas linguagens de interface com o objetivo avaliar os custos e benefícios da interação com cada uma delas e a qualidade da representação das classes de signos identificadas. É importante ressaltar que não faz parte do objetivo do trabalho comparar as interfaces criadas com as já existentes utilizadas na identificação das classes de signos.

1.3 Organização da dissertação

O restante do trabalho está organizado da seguinte forma: no Capítulo 2 é apresentado um panorama da área de Interação Humano-Robô e alguns trabalhos relacionados. Em seguida é apresentada a fundamentação teórica deste trabalho, aprofundando na teoria da Engenharia Semiótica [de Souza, 2005b] e no Método de Inspeção Semiótica, que foi utilizado para identificar as classes de signos presentes na interação com robôs.

O Capítulo 3 trata da identificação do sistema de significação para a interação com robôs. São mostrados os passos da avaliação de duas interfaces existentes para o *e-Puck* e a triangulação dos resultados das duas avaliações. Por fim são apresentadas as classes de signos identificadas. O Capítulo 4 apresenta o desenvolvimento das interfaces com base nas classes de signos encontradas. Primeiramente é mostrada a modelagem da interação através dos diagramas da MoLIC [Barbosa & de Paula, 2003], depois são definidos os tipos de signos que as interfaces terão e a quais classes estes pertencem. Por fim é descrita a implementação do servidor de controle do robô e cada uma das três interfaces.

No quinto capítulo temos o estudo comparativo das interfaces implementadas. É mostrado como foi feita a preparação para os testes e em seguida é descrita a execução dos testes com os usuários. No fim temos a apresentação dos resultados dos testes. E, finalmente, no Capítulo 6 são apresentadas e discutidas as conclusões obtidas com o trabalho, além das contribuições e dos trabalhos futuros.

Capítulo 2

Panorama atual e Fundamentação Teórica

2.1 Panorama da interação humano-robô

A área de pesquisa da interação humano-robô possui vertentes bastante variadas. Algumas delas incluem, mas não se limitam a, pesquisas com interação social entre pessoas e robôs, supervisão de múltiplos robôs, utilização de robôs em entretenimento e métricas para tais aplicações. Este capítulo apresenta um panorama geral dos trabalhos na área da interação humano-robô e descreve as diferenças em relação a este trabalho.

O trabalho presente em [Breazeal, 2004] define quatro paradigmas de interação em HRI: robô como ferramenta; robô como uma extensão (ciborgue); robô como um avatar; e robô como um parceiro social. Estes paradigmas são diferenciados pelo modelo mental que um humano tem de um robô ao interagir com ele. Mesmo sendo diferentes, os paradigmas possuem alguns desafios em comum, como o controle compartilhado entre o humano e o robô.

O trabalho dela tem foco no último paradigma, onde o robô é um parceiro do humano. Nele também é apontado um trabalho ([Reeves & Nass, 2003]) que mostra que humanos tratam computadores como tratariam outras pessoas se o sistema se portar competentemente de uma maneira sociável. Estudos de IHC aplicados em HRI podem ser úteis para aumentar o entendimento de como as pessoas interagem com este tipo de tecnologia, para ajudar a desenvolver robôs que interajam eficientemente com pessoas. Algumas dificuldades incluem a morfologia do robô, a aparência, aspectos físicos, capacidades perceptuais, expressividade e inteligência. O trabalho sugere que estas dificuldades podem ser trabalhadas através de estudos sobre os seguintes aspectos:

- Comparação com outras mídias: Como a interação com robôs difere da interação com outras mídias? Como é similar?
- Naturalidade: As pessoas têm tendência a interagir naturalmente com este tipo de tecnologia? Elas o fariam como interagiriam com outra pessoa? Se não, quais são as diferenças?
- Expectativas dos usuários: Quais são as expectativas implícitas dos usuários em relação ao robô (como se expressa, como entende)?
- Qualidade: Quais aspectos tornam o robô agradável? Quais fazem o contrário?
- Relacionamento: Qual deve ser a natureza da interação humano-robô (quais paradigmas)?
- Trabalho em equipe: Como os robôs podem ser eficientes em equipes mistas?
- Personalidade: O robô deve ter personalidade? Se sim, quão complexa deve ser?
- Cultura: Como a cultura pode influenciar no projeto?
- Aceitação: A percepção dos robôs é diferente em várias partes do mundo. Como isso impacta na aceitação dos mesmos na cultura humana?

HRI é simplesmente uma adaptação de IHC para robôs? [Breazeal, 2004] diz que não é suficiente avaliar a interação somente no ponto de vista do humano, como em IHC. Como a interação é direta, é importante analisar o ponto de vista do robô. Alguns aspectos são a interação a longo prazo com o mundo, a sobrevivência no mundo real, a integração profunda entre interface e controle no robô, a interação com os humanos e o aprendizado no ambiente dos humanos.

Para o aprendizado dos robôs no mundo real, algumas questões também são levantadas. Como definir o que é importante aprender? Como saber que ações tentar executar durante o aprendizado? Como saber quando se deve aprender e de quem se deve fazê-lo? Como corrigir erros e reconhecer sucessos? Como aprender de forma eficiente?

Em [Xin & Sharlin, 2007] é proposta uma abordagem para testar interfaces humano-robô, principalmente para robôs sociáveis. Uma solução seria realizar os testes no próprio ambiente de aplicação do robô, mas em vários casos, como exploração espacial e suporte em campo de batalha, os desenvolvedores podem não conseguir realizar tais testes. Como a maioria das aplicações para robôs atualmente são difíceis, perigosas e sujas (*DDD: difficult, dangerous and dirty*), testar paradigmas sociais desta forma é

um desafio que deverá esperar até que as plataformas de robôs sejam mais comuns e acessíveis.

A abordagem proposta pelos autores é baseada na interação colaborativa em jogos entre humanos e robôs, e os autores afirmam que, de maneira similar ao conceito de *transferência de habilidade cognitiva* presente na psicologia, a transferência das habilidades robóticas pode ser realizada do ambiente de testes para ambientes reais.

Apesar do crescimento dos jogos eletrônicos, muitos jogos tradicionais são realizados no mundo físico, requerendo interação tangível, o que é interessante para o estudo com robôs, que são entidades físicas presentes no mundo e interagem com pessoas diretamente em aplicações reais. Robôs também têm acesso ao mundo digital e são capazes de agir neste domínio. Esta dualidade é suportada dentro do ambiente de jogos. Estes são jogados dentro de um domínio restrito, possuindo limites de tempo, locais e regras. Esta característica difere de muitas das aplicações reais, que possuem possibilidades ilimitadas de interação. Entretanto, isto torna os jogos favoráveis para a utilização em experimentação, pois ajuda a reduzir o escopo de exploração tanto em termos de implementação quanto do que será investigado.

No trabalho foi implementado o *Sheep and Wolves* (Ovelha e Lobos), um jogo clássico jogado em um tabuleiro de xadrez. São necessários cinco participantes, sendo quatro assumindo o papel de lobos e um de ovelha. A intenção dos lobos é prender a ovelha em alguma região do tabuleiro de modo que ela não possa realizar nenhum movimento legal, i.e. andar nas diagonais em qualquer sentido. Os lobos também só podem se mover nas diagonais, mas somente para frente. Já a ovelha deve conseguir atravessar o tabuleiro até o lado oposto de onde iniciou o jogo. Os lobos e a ovelha se movem em turnos alternados.

Os lobos foram representados por quatro Sony®Aibos™, sendo três autônomos e um controlado por um operador humano. A ovelha foi representada por uma entidade virtual através de realidade aumentada (*augmented reality*) para destacar a natureza multimodal dos robôs. Ganhar o jogo como lobos requer trabalho em equipe, e o jogador humano deve dar sugestões aos outros robôs e considerar as sugestões dadas pelos mesmos para chegar a decisões quanto ao movimento dos lobos.

Os testes foram realizados com dois comportamentos opostos do time de robôs: no primeiro teste estes eram totalmente submissos ao jogador humano, acatando suas decisões e elogiando-o quando as decisões eram certas; no segundo caso os robôs eram mais independentes do humano, tomando as próprias decisões e ignorando o jogador humano, fazendo-o se sentir inferior.

A principal conclusão obtida com o trabalho foi que o conceito de utilização de jogos para testes de robôs foi capaz de produzir uma experiência social imersiva na qual

os jogadores acreditavam estar participando de um jogo colaborativo com membros do time que são realistas e com comportamentos diferenciados, enquanto na verdade os robôs foram implementados com comportamentos simples e iguais para todos.

Já o trabalho realizado em [Richer & Drury, 2006] tem como objetivo melhorar a interação com robôs aéreos (*UAVs - Unmanned Aerial Vehicles*), já que muitos erros que resultaram em danos ou perda das aeronaves foram atribuídos a interfaces pobres. Como parte da abordagem para obter possíveis avanços nas interfaces de UAVs, foram analisadas interfaces do campo dos jogos que, de maneira similar a operações com UAVs, em sua maioria requerem que o usuário perceba um ambiente 3D e tome ações que exijam uma interação eficiente.

A abordagem foi feita para examinar a interação com vários tipos de jogos e descrevê-la em termos dos componentes que constituem as interfaces e também expor os vários tipos de interação que são utilizados de forma bem sucedida nos jogos. Os componentes de interface do *framework*, chamado *VGBF - Video Game Based Framework* -, foram divididos em entrada e saída, sendo que cada um destes componentes é dividido em dispositivos e métodos. Ainda há uma outra categoria nos componentes de entrada: a classificação dos mesmos. O *framework* é dividido da seguinte forma:

- Dispositivos de saída: Telas de vídeo; Alto-falantes; Dispositivos hápticos;
- Métodos de saída: Saída gráfica visual; Saída visual adicional; Saída não-visual; Áudio; Hápticas;
- Dispositivos de entrada: Botões; *Joysticks*; Dispositivos apontadores; Multimodal (entrada através de áudio ou vídeo); Controles especializados;
- Métodos de entrada: Comando; Linguagem natural; Cursor; Controle de câmera;
- Classificação das entradas:; Alvo (o que um mecanismo de entrada afeta e se o jogador pode controlá-lo diretamente); Complexidade (simples, contextual, combinatória e sequencial);

São percebidos dois usos para o *VGBF*: avaliação de interfaces existentes e a criação de novas interfaces, sendo que o trabalho compara duas interfaces para UAVs. O *framework* pode ajudar um projeto de HRI fornecendo descrições para os componentes de interface que sejam independentes da aplicação, gerando uma linguagem comum para a análise de diferentes trabalhos que apliquem o *framework*. Uma limitação é o fato do *VGBF* ser baseado no estado atual dos jogos, mesmo que generalizado ao máximo para ser aplicável às próximas gerações.

Há um grande interesse em interfaces que permitam que um único operador possa controlar um time de robôs. Isso pode ser vantajoso para aplicações militares como vigilância e reconhecimento ou para atividades de busca e resgate, avaliação de danos em áreas destruídas e locais perigosos. A utilização de um número maior de robôs pode levar a uma maior cobertura e eficiência, dado que haja um método para coordenar o controle sobre os mesmos.

Em [Humphrey et al., 2007b] é dito que a carga de trabalho (*workload*) e a percepção do ambiente (*situational awareness*) pelo operador são itens fundamentais para determinar a eficiência e a escalabilidade de uma interface. A escalabilidade é definida como a habilidade de uma interface suportar o controle eficiente sobre poucos ou um grande número de robôs. A *situational awareness* vem, em outros trabalhos, sendo tipicamente avaliada para um único robô, e as avaliações para múltiplos robôs não chegaram a resultados conclusivos.

A maioria das interfaces para supervisão de robôs são baseadas em mapas 2D com uma visão de cima. Estas interfaces requerem informações precisas de posicionamento e também um mapa conhecido *a priori*. O foco do trabalho é avaliar a escalabilidade de uma interface para controle de múltiplos robôs, baseado em *workload* e *situational awareness* numa aplicação de busca por bombas.

A interface criada é composta do *video feed* do robô atualmente selecionado, cercada por indicações das posições relativas dos outros robôs. Há um mapa menor, que mostra a posição absoluta de todos os robôs presentes. Numa barra lateral há, para cada robô, uma caixa com seus status. Esta caixa muda de cor (ou pisca) quando um robô está selecionado ou quando o mesmo pede a atenção do usuário. Existem comandos de teleoperação, controlando a movimentação do robô diretamente ou atribuindo comandos de mais alto nível, como ir até um determinado robô ou examinar as redondezas em busca de bombas. Não foram utilizados robôs reais, mas sim um ambiente de simulação que utiliza a *engine* de jogos do *Unreal Tournament 2004*.

Os testes foram realizados com 20 pessoas. Cada um realizou a tarefa de buscar bombas quatro vezes: duas vezes com 6 robôs (2 times) e 2 bombas e outras duas vezes com 9 robôs (3 times) e 3 bombas. Cada time de robôs era composto por dois robôs de reconhecimento e um robô com capacidade de desarmar bombas. Foram utilizados questionários para avaliar o *workload* e o *situational awareness* dos participantes, além de medidos tempos de interação com os robôs.

Com base nos dados obtidos, foram feitas comparações entre os resultados. A variação de *workload* e *situational awareness* encontrada com a alteração do número de robôs foi melhor que o esperado, sendo que para o caso do *situational awareness*, a relação esperada entre o valor obtido e o número de robôs era inversamente proporci-

onal, enquanto a relação observada foi direta. Para os tempos de utilização dos robôs, foi observado que os participantes tendiam a utilizar todos os robôs ao invés de somente focar em alguns. O tempo de intervenção (tempo entre o pedido de atenção do robô e a ação do usuário para este robô) se manteve constante nos dois casos.

No trabalho realizado em [Skubic et al., 2006] é apresentada uma interface na qual um operador pode controlar vários robôs utilizando uma interface *sketch-based* (tradução livre: baseada em esboço) em um tablet PC. Não é necessário um mapa preciso do local, bastando que o usuário faça um desenho aproximado do mesmo. O usuário pode adicionar robôs à interface, que são vinculados a robôs reais. Além do mapa qualitativo, os resultados das leituras dos sensores são mostrados, juntamente das posições atuais dos robôs.

Pode-se dar comandos de movimentação a um ou mais robôs de forma simultânea, marcando locais de destino no mapa e/ou desenhando trajetórias que os robôs tentarão seguir. Os comandos podem ser executados prontamente ou podem ser adiados para um momento futuro (para sincronização, por exemplo). Utilizando o feedback dos sensores dos robôs, o mapa pode ser ajustado para melhorar a correspondência com o mundo real.

A avaliação dos comandos de movimentação é feita analisando a posição do robô em relação ao objeto mais próximo do objetivo e também do mesmo objeto para o ponto de destino do robô. Caminhos são decompostos em vetores, sendo que cada vértice no grafo formado pelos vetores passa a ser um objetivo.

Foi feito um estudo de usabilidade para esta interface. Todos os usuários vieram o local de teste e depois eram removidos da sala. Para metade dos usuários (escolhidos aleatoriamente), o ambiente era ligeiramente alterado depois de visto. Então foram dadas cinco tarefas para serem executadas utilizando a interface. Foram avaliados os tempos necessários para executar cada tarefa para cada grupo.

Os resultados qualitativos não apontaram diferença na execução das tarefas, mas o estudo de usabilidade mostrou que a interface baseada em desenhos é válida no controle de múltiplos robôs.

O trabalho em [Wang & Lewis, 2007] investiga a interação humana com um time de robôs em uma tarefa de busca e resgate, comparando a performance entre equipes autônomas, equipes controladas manualmente e equipes que interagem de forma cooperativa com o operador. A maioria dos trabalhos citados foram realizados em simuladores 2D, mas este realiza uma simulação em ambiente 3D com o USARsim. As simulações foram realizadas com três robôs P2-DX equipados com câmera e laser.

Foi criada uma infraestrutura (chamada de *MrCS - Multi-robot Control System*) para comunicação e controle dos robôs, com uma interface para realização dos estudos.

Através da interface são mostrados os *feeds* de vídeo e as informações obtidas com o laser para cada robô e também é possível teleoperar o robô e movimentar a sua câmera. Os testes foram feitos avaliando o impacto da variação de autonomia dos robôs na realização das tarefas. Além das medidas de performance, como número de vítimas encontradas, foram examinadas também a distribuição da interação entre os robôs e a distribuição dos comandos de controle.

A conclusão obtida foi que a performance foi geralmente melhor quando se tinha uma equipe cooperativa (*mixed-initiative*), encontrando-se mais vítimas e também atingindo uma maior área coberta. Para o modo manual e o modo *mixed-initiative*, não houve diferenças significativas na distribuição da atenção entre os robôs, mas no segundo houve mudanças mais freqüentes de um robô para outro. Três modos de controle foram disponibilizados: teleoperação, *waypoints* e controle de câmera. Também não houve diferenças significativas de tempo entre os modos, sendo que no modo *mixed-initiative* novamente houve mudanças mais freqüentes.

[Fong et al., 2001b] apresenta um sistema para controlar robôs de forma colaborativa que inclui o gerenciamento do diálogo entre humanos e robôs e uma interface pessoal (PdaDriver, apresentado em [Fong et al., 2001a]), para teleoperar múltiplos robôs em terrenos desconhecidos e não-estruturados.

No sistema proposto o diálogo surge da troca de mensagens entre o operador e o robô. Este diálogo não necessita de uma linguagem completa, somente que seja pertinente para a tarefa a ser realizada e que transporte a informação de forma eficiente. Desta forma não é utilizada a linguagem natural e o conteúdo das mensagens é limitada à mobilidade do robô e aspectos relacionados à tarefa em questão.

A interface utilizada foi implementada em um PDA para suportar o controle de múltiplos robôs, fornecendo uma variedade de modos de operação, incluindo controle direto de posição e velocidade, *waypoints* baseados em imagens e *waypoints* baseados em mapas. Ao encontrar certas situações, o robô envia perguntas de como deve proceder para continuar executando suas tarefas. Há também a possibilidade do operador fazer perguntas ao robô, apesar de serem mais freqüentes as perguntas do robô para o operador.

Foram feitos dois testes nos quais um único operador era responsável por dois robôs móveis. O primeiro teste foi feito em ambiente interno, cuja tarefa era vigiar o local e detectar intrusos. O segundo foi executado em ambiente externo e a tarefa em questão era de reconhecimento do local.

Dois fatores cruciais para a colaboração efetiva entre humanos e robôs foram levantados pelo trabalho. Primeiramente, papéis e responsabilidades devem ser atribuídas de acordo com a capacidade tanto dos operadores quanto dos robôs. No caso da

operação de múltiplos robôs é necessário reduzir ao máximo o nível de atenção que o operador deve dedicar a cada robô. Isso pode ser alcançado deixando o operador com tarefas de alto nível (e.g. onde ir) enquanto o robô lida com as tarefas de nível mais baixo (e.g. como chegar lá).

Em segundo lugar, deve ser fácil para o operador perceber e entender a situação atual de um determinado robô. O diálogo é importante porque permite ao usuário rever o que já aconteceu, entender os problemas que cada robô encontrou e ser notificado quando é necessária a sua assistência.

A conclusão obtida é que o controle colaborativo é útil para a teleoperação de robôs, pois permite que a autonomia e a interação entre humano e robô varie de acordo com as necessidades do momento, o que é particularmente útil para ambientes desconhecidos ou que sejam difíceis de se executar algum planejamento. O diálogo permite que o operador seja altamente eficiente, pois foca a atenção somente no que é necessário, ajudando a coordenar e direcionar a solução de problemas.

No trabalho de Steinfeld et al [Steinfeld et al., 2006] são apresentadas métricas para a interação humano-robô orientada a tarefas. Estas métricas são apresentadas em cinco categorias:

- **Navegação (*Navigation*):** Tarefa fundamental para robôs, que é se mover de um ponto A a um ponto B, como chegar e como lidar com fatores do ambiente e dificuldades (como obstáculos).
- **Percepção (*Perception*):** Perceber e entender o ambiente para busca, vigilância, identificação de objetos, etc. Não considera a percepção necessária para as outras categorias (Navegação, por exemplo, necessita de localização).
- **Gerenciamento (*Management*):** Tarefas que têm o propósito de coordenar e gerenciar as ações de humanos e robôs, trabalhando individualmente ou em conjunto.
- **Manipulação (*Manipulation*):** Compreende a interação do robô com o ambiente, tanto em relação a tarefas de *grasping*¹ quanto outros tipos, como empurrar algo.
- **Social (*Social*):** Tarefas que necessitem de alguma interação social, como cuidados, guias ou entretenimento.

O trabalho também cita fatores que podem alterar a avaliação das métricas, como a comunicação (tempo de resposta, largura de banda e interferência), resposta do robô

¹Tarefas nas quais o robô deve segurar algum objeto, possivelmente com um braço mecânico.

(tempo de processamento e resposta) e os próprios usuários (de acordo com a experiência de cada um, a complexidade da tarefa, condições de execução da tarefa, dentre vários outros fatores). As métricas apresentadas para cada categoria são enumeradas a seguir:

- **Navegação**

- **Navegação global:** Entendimento geral do ambiente onde se encontra.
- **Navegação local:** Informações de granularidade mais fina. O robô deve saber qual a situação ao seu redor, como portas, escadas, árvores ou mesmo pessoas.
- **Encontro de obstáculos:** Envolve o planejamento para desvio/sair de obstáculos, necessitando de conhecimento sobre o obstáculo e as ameaças do ambiente.

- **Percepção**

- **Percepção passiva:** Envolve interpretação dos dados dos sensores.
- **Percepção ativa:** Compreende desde a movimentação de câmera até a movimentação do próprio robô, considerando que estas atividades são iniciadas a partir da identificação de um alvo.

- **Gerenciamento**

- **Fan out:** Medida de quantos robôs similares podem ser controlados por uma pessoa.
- **Tempo de resposta de intervenções:** O tempo entre o robô encontrar algum problema e o operador intervir. A intervenção pode ser física (como assistência mecânica) ou cognitiva (como tomada de decisões).
- **Discrepâncias nos níveis de autonomia:** Identificação rápida do nível de autonomia necessária para uma dada tarefa.

- **Manipulação**

- **Grau de computação mental (ou *mental workload*):** Grau da computação mental desempenhada pelo operador, como rotações, projeções de imagens 2D, etc.
- **Erros de contato:** Número de colisões não intencionais entre o manipulador e o ambiente.

- **Social**

- **Características de interação:** Entendimento do estilo de interação ou contexto social.
- **Persuasão:** Quando o robô é utilizado para mudar o comportamento de humanos, como no caso de cuidados com pessoas.
- **Confiança:** Confiança na automação do robô.
- **Participação (*Engagement*):** Eficácia das características sociais do robô (emoção, diálogo, personalidade, etc.) para capturar a atenção e manter o interesse das pessoas.

- **Conformidade (*Compliance*)**: Eficácia de características sociais do robô (aparência, adesão a padrões e normas, etc.) que influenciam no nível de cooperação de um humano.

O estudo de caso apresentado possui poucas informações sobre a aplicabilidade destas métricas, mas elas podem servir de guia para identificar aspectos que devem ser avaliados em interfaces para interação humano-robô.

Como dito anteriormente, os trabalhos em HRI possuem focos bastante variados. Vários procuram abordar interações sociais entre humanos e robôs, seja esta interação direta ou mediada por uma interface de software, enquanto outros buscam avaliar aspectos operacionais desta interação, ou seja, como melhorá-la para que uma determinada tarefa seja executada com mais eficiência e com o mínimo de prejuízos. Existem também trabalhos que buscam definir conceitos para descrever interfaces, como o último citado, mas nenhum deles trabalha com os aspectos que podem emergir da interação, relacionados tanto com a interface quanto às observações geradas pelo resultado desta interação no ambiente.

É justamente essa última vertente que este trabalho aborda. Serão observados aspectos tanto das interfaces de controle de um robô quanto do próprio robô. Estas observações serão utilizadas para gerar uma classificação que servirá de base para o desenvolvimento de novas interfaces em diferentes tecnologias para a interação com robôs.

2.2 Fundamentação Teórica

2.2.1 Engenharia Semiótica

Este trabalho é embasado na teoria da engenharia semiótica, apresentada em [de Souza, 2005b], uma teoria explicativa de IHC fundamentada na semiótica e que considera que os projetistas de sistemas interativos comunicam sua visão de projeto para os usuários através dos seus sistemas. Esta mensagem comunicando a visão do projetista, ou seja, para qual público aquele sistema foi desenvolvido, quais necessidades e preferências dos usuários foram contempladas no projeto, por quê e como, é transmitida através da interface do sistema. Os usuários entendem o conteúdo da mensagem do projetista através da sua própria interação com o sistema. Dessa forma, o sistema é considerado um artefato de metacomunicação. Como os projetistas não estão presentes no momento da interação, essa mensagem é indireta e unidirecional.

A mensagem do projetista para o usuário é composta de signos que, como definidos por Peirce [Peirce, 1931], são “qualquer coisa que, sob certo aspecto ou modo,

representa algo para alguém”. Em outras palavras, é algo que associa um objeto com seu significado e representação, e.g., uma imagem de uma impressora em uma interface é um signo ligando a impressora (o objeto) à ação de imprimir um arquivo (o significado) e ao botão da impressora (a representação). Os signos são parte de um ou mais sistemas de significação, que por sua vez surgem da associação cultural (e artificial, no caso de IHC) entre conteúdos e expressões. A engenharia semiótica divide os signos em três categorias:

- Signos estáticos: são signos cujo significado é interpretado independentemente de relações temporais ou causais, ou seja, a interpretação é limitada aos elementos presentes na interface em um dado momento. Numa interface de software, elementos como botões, caixas de texto, menus e caixas de seleção são signos estáticos, pois seus significados podem ser interpretados sem a necessidade de se saber as ações do usuário e estados do sistema antecedentes ou subsequentes.
- Signos dinâmicos: são signos que dependem das relações causais e temporais entre elementos de interface. Signos dinâmicos são normalmente associados a comportamentos dos sistemas. Não há como interpretá-los sem saber o contexto em que estão inseridos e quais ações levaram a eles. Em formulários de cadastro, o surgimento de mensagens de erro quando o usuário deixa de preencher campos obrigatórios é um exemplo de signo dinâmico.
- Signos metalingüísticos: são signos que se referem a outros signos da interface. São incluídos nessa classe as instruções, explicações, mensagens informativas e dicas, nos quais o projetista comunica explicitamente os significados codificados no sistema e como devem ser utilizados. O texto da mensagem de erro por si só é um signo metalingüístico, pois informa algo (que há um erro) em relação a algum outro signo da interface (o campo obrigatório não preenchido).

É possível notar que os signos dinâmicos e estáticos estão intimamente ligados. Os signos estáticos promovem a interação com o sistema, enquanto os signos dinâmicos representam as transições entre estados do sistema (estes representados pelos signos estáticos). Estas classes de signos juntas comunicam a mensagem de metacomunicação (i.e. mensagem que comunica sobre a comunicação do usuário com o sistema) do projetista para o usuário, que pode ser parafraseada da seguinte forma:

“Este é o meu entendimento de quem você é, o que entendi que você quer ou precisa fazer, de quais maneiras e por quê. Então este é o sistema

que eu projetei para você e esta é a maneira que você pode ou deve usá-lo de forma a atingir os objetivos que foram incorporados na minha visão.”²

Baseado nesse arcabouço teórico em que a interface é percebida como uma comunicação entre projetista e usuário, a qualidade de uma interface é dada pela sua *comunicabilidade*. Comunicabilidade é a propriedade do sistema que apresenta ao usuário o seu projeto e princípios interativos de forma eficiente (organizada e rica) e eficaz (que chega ao resultado esperado)[Prates et al., 2000][de Souza & Leitão, 2009]. Para avaliar a comunicabilidade de uma interface, dois métodos foram propostos: O Método de Avaliação de Comunicabilidade (MAC) [Prates et al., 2000] [de Souza, 2005b] e o Método de Inspeção Semiótica (MIS) [de Souza et al., 2006] [de Souza & Leitão, 2009]. O MAC é um método que envolve a observação das rupturas de comunicação na interação dos usuários com o sistema em um ambiente controlado. O MIS, por sua vez, é baseado na inspeção do sistema por um avaliador especialista, sem a participação de usuários. Este método será utilizado no trabalho e será descrito com mais detalhes na próxima seção.

2.2.2 Método de Inspeção Semiótica

O Método de Inspeção Semiótica é um método de inspeção baseado na engenharia semiótica. O MIS permite a avaliação da comunicabilidade em um sistema interativo baseado na análise dos seus signos estáticos, dinâmicos e metalinguísticos.

Na preparação do MIS, o primeiro passo é definir o propósito da avaliação. Com isso em mente, o avaliador deve estudar o sistema informalmente para identificar o foco da avaliação, além de confirmar quais são os usuários a que o sistema é destinado e os objetivos em alto nível e atividades que o mesmo suporta. Por fim, o avaliador define o escopo da avaliação através de um ou mais cenários³ de interação que descrevem contextos de uso, os usuários a serem considerados durante a inspeção e também a porção do sistema a ser avaliada.

Este último passo é necessário, pois uma inspeção exaustiva tende a ser custosa, dependendo da complexidade e tamanho do sistema. A seleção destes cenários é feita

²Tradução livre do autor. O texto original em [de Souza, 2005b], página 84, é o seguinte: “Here is my understanding of who you are, what I have learned you want or need to do, in which preferred ways, and why. This is the system I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision.”

³Um cenário é definido por como uma narrativa que descreve a interação de um ou mais usuários com um sistema para atingir um ou mais objetivos em uma tarefa. Além disso, um cenário narra os objetivos do usuário, suas expectativas e reações a determinados eventos, se referindo a pessoas, objetos e comportamentos específicos, ao invés de tipos de usuários ou papéis [Rossen & Carroll, 2001].

pelos próprios avaliadores baseada nas partes que são consideradas as mais relevantes para o propósito da inspeção.

Depois de gerar o material necessário da preparação, os avaliadores prosseguem com a execução da inspeção. Esta etapa consiste nos cinco passos descritos a seguir:

- **Passo 1: Inspeção da documentação *online* e *offline* e do conteúdo de ajuda (signos metalinguísticos).** Todos os signos metalinguísticos (i.e., documentação online e offline, sistema de ajuda, instruções e explicações, mensagens de erro e aviso, dicas e fontes relacionadas) são inspecionadas. Com base na inspeção o avaliador reconstrói a mensagem de metacomunicação que é enviada pelos projetistas através dos signos metalinguísticos. Para isso o avaliador utiliza o *template* apresentado na seção anterior para parafrasear a mensagem sendo transmitida.
- **Passo 2: Inspeção dos signos estáticos da interface.** O avaliador analisa os signos estáticos (i.e., elementos de interface das telas e diálogos presentes no sistema) e reconstrói novamente a mensagem de metacomunicação. Note que esta reconstrução não leva em conta a metamensagem gerada no passo 1.
- **Passo 3: Inspeção dos signos dinâmicos.** O avaliador inspeciona os signos dinâmicos, i.e., aqueles que surgem da interação com o sistema e seu comportamento. Novamente o avaliador reconstrói a metacomunicação entre projetista e usuário baseado somente nestes signos.
- **Passo 4: Comparação e contraste das metacomunicações encontradas nos passos anteriores.** O avaliador deve comparar as metamensagens reconstruídas em cada um dos passos anteriores e identificar possíveis ambigüidades e inconsistências que emergem da interação com o sistema.
- **Passo 5: Apreciação da qualidade da metacomunicação consolidada obtida no passo 4.** Neste passo é feita a avaliação da comunicabilidade do sistema unificando as mensagens de metacomunicação obtidas em cada passo. O avaliador verifica se os custos, benefícios e a adequabilidade das estratégias de comunicação adotadas. Neste momento o avaliador também considera o efeito de inconsistências, brechas e falta de redundância (caso haja) encontrados no passo 4.

O propósito da análise é então reconstruir uma versão integrada da metacomunicação, levantando casos de inconsistências e ambigüidades, caso hajam. Para a execução do método, os avaliadores devem assumir a postura de defensores do usuário.

Assim, apesar de não serem usuários reais, levarão em consideração os interesses dos usuários e os possíveis benefícios para os mesmos numa grande variedade de possíveis cenários de interação através de seu sólido conhecimento de IHC.

Como qualquer método baseado em teoria, quanto mais conhecimento o avaliador possuir, melhor serão os resultados da sua avaliação. Também por causa da base teórica, o MIS herda da engenharia semiótica a capacidade de considerar diferentes interlocutores e suas respectivas intenções, além de classificar os vários sistemas de significação utilizados na comunicação usuário-sistema em diferentes contextos de interação. O MIS pode ser utilizado tanto para analisar questões técnicas quanto científicas [de Souza & Leitão, 2009] relacionadas à interação com sistemas, teoricamente podendo ser aplicado independente de tecnologia.

Aplicações técnicas do método estão relacionadas à melhoria do projeto de interação de um determinado sistema. A análise tem foco em aspectos deste sistema em particular, detectando problemas de comunicabilidade. Já as aplicações científicas possuem um foco diferente: elas pretendem obter avanços no conhecimento de IHC, contribuindo para um ou mais dos seguintes objetivos:

- Uma nova visão em relação a problemas já conhecidos;
- Identificação de novas soluções para problemas e desafios existentes;
- Identificação de novos problemas e desafios; e
- Formulação de novas teorias, conceitos, modelos ou métodos.

Quando aplicado cientificamente, um passo final de triangulação é adicionado ao método. Este é um importante passo para aplicações científicas, pois permite que os avaliadores verifiquem a validade das suas descobertas através de evidências obtidas de outras fontes, que podem ser endógenas ou exógenas.

Fontes endógenas são outras interpretações geradas pelo uso do mesmo sistema em diferentes contextos ou diferentes sistemas que compartilham o mesmo domínio. Já as fontes exógenas não possuem o mesmo domínio, mas compartilham aspectos relevantes com o sistema inspecionado. Em outras palavras, eles compartilham aspectos comuns que são diretamente relacionados à interpretação que o avaliador está tentando validar.

Capítulo 3

Classes de signos para a interação humano-robô

Para a identificar classes de signos para a interação humano-robô foi conduzida uma avaliação de interfaces existentes para controle de robôs. Esta avaliação tem o intuito de identificar aspectos presentes na interação com os mesmos. Além disso, alguns trabalhos existentes serviram para auxiliar nesta identificação, ajudando a levantar aspectos que as interfaces avaliadas não contemplaram.

Esta avaliação foi realizada através da aplicação do Método de Inspeção Semiótica [de Souza et al., 2006] no *E-Puck Monitor* [EPFL, 2009a] e no *ePic* [EPFL, 2009b], duas interfaces utilizadas para controlar e-Pucks [Mondada et al., 2009]. A avaliação do *ePic* foi realizada para a triangulação com os resultados da avaliação do *E-Puck Monitor*. O restante deste capítulo apresenta a aplicação do MIS nas interfaces citadas e as classes de signos que foram identificadas através da análise e da comparação com alguns trabalhos relacionados.

3.1 e-Puck, o robô utilizado na pesquisa

O e-Puck, mostrado na figura 3.1, é um robô diferencial de 7 centímetros de diâmetro projetado pela EPFL – *École Polytechnique Fédérale de Lausanne* – para uso educacional e em pesquisas. O robô é equipado com um processador dsPIC de 16 bits, comunicação via *bluetooth*, uma câmera VGA, um acelerômetro 3D, oito sensores infravermelhos, três microfones e um anel de LEDs em volta do mesmo.

O projeto *e-Puck* foi criado para desenvolver um robô que tenha uma estrutura mecânica limpa e simples de entender; seja capaz de cobrir uma larga gama de atividades; possua sensores variados; seja pequeno e fácil de operar; resista ao uso por



Figura 3.1. Robô *e-Puck*.

estudantes; tenha manutenção simples e barata e; tenha um preço baixo para utilização em larga escala [Mondada et al., 2009].

Além disso, o *e-Puck* foi desenvolvido sob a licença de “*open hardware*”, permitindo a qualquer um usar a documentação e desenvolver extensões para o mesmo. Para programar para o *e-Puck*, uma biblioteca de código aberto escrita em C está disponível, permitindo o controle sobre as funcionalidades do robô.

Existem diferentes maneiras de se interagir com um robô de acordo com a variação no nível de autonomia. Considerando o espectro de autonomia, um robô pode ser totalmente autônomo em um lado ou teleoperado no outro. Existem vários níveis de semi-autonomia entre estes dois níveis. Robôs completamente autônomos são normalmente controlados por programas embutidos, não necessitando de interfaces elaboradas com os usuários. Por outro lado, robôs teleoperados não possuem nenhuma autonomia, então os usuários devem entrar com todos os comandos, fazendo-se necessária uma interface.

Neste trabalho o *e-Puck* é utilizado como um robô teleoperado e a sua avaliação considera a interface de teleoperação, além avaliar também os aspectos que podem ser observados no próprio robô.

3.2 Aplicação do MIS

Nesta seção é apresentada a aplicação do MIS no contexto de interação com robôs. Este estudo tem dois objetivos: verificar a aplicabilidade do MIS para o contexto de Interação Humano-Robô e também identificar aspectos relevantes e específicos relacionados à interação com robôs teleoperados, de modo a identificar classes de signos para este tipo de interação. Essa identificação é importante para que se tenha conhecimento das classes de signos que são relevantes para a interação com esse tipo de robôs, o que pode guiar o projeto de interfaces, uma vez que apontam para aspectos que devem ser representados. Os sistemas citados – *E-puck Monitor* e *ePic* – foram selecionados porque foram criados e disponibilizados pelos desenvolvedores do *e-Puck* na EPFL e porque não foi encontrada nenhuma outra interface disponível para o controle de um *e-Puck*.

O estudo seguiu os passos para a aplicação científica do MIS descritos na Seção 2.2.1. Nesta seção será primeiro apresentada a preparação executada, o escopo de interação e o cenário gerado. Então, os principais pontos levantados pela inspeção (passos 1 a 3 do método) serão discutidos. Em seguida será apresentada a consolidação da inspeção dos três níveis de signos e os principais pontos que surgiram desta análise.

Por fim será apresentada a triangulação dos resultados da inspeção do *E-puck Monitor* com outro sistema relacionado, o *ePic*. Para isso uma aplicação científica do MIS também foi executada para a interface do *ePic*. A triangulação vai permitir identificar os signos representados nas duas diferentes interfaces e as classes a que esses signos pertencem, sendo usada como insumo para a identificação do sistema de significação.

3.2.1 Preparação

A intenção da inspeção é avaliar uma interface usada para a teleoperação de um *e-Puck* para identificar aspectos que devem ser comunicados para que possa haver a interação humano-robô. O sistema escolhido não é complexo e possui somente uma tela, então todo o sistema foi avaliado. Um cenário de interação foi criado para definir o contexto de uso a ser considerado na avaliação. O cenário utilizado para inspecionar o sistema foi o seguinte:

“João é um especialista em robótica e seu laboratório acaba de adquirir alguns e-Pucks novos para conduzir pesquisas. Sua primeira tarefa é testar um dos e-Pucks e ver qual tipo de tarefa ele é capaz de realizar. Para tal ele utilizará o software ”E-puck Monitor”, disponibilizado pela universidade que desenvolveu os robôs. João já

sabe como iniciar a conexão entre o robô e o computador, pois leu um tutorial presente no site da universidade. Agora ele deve utilizar o software para executar uma simples tarefa de navegação, movendo o robô de um dado ponto a outro sobre uma mesa. O caminho possui alguns objetos que serão utilizados como obstáculos para observar a resposta do robô enquanto a tarefa é executada.”

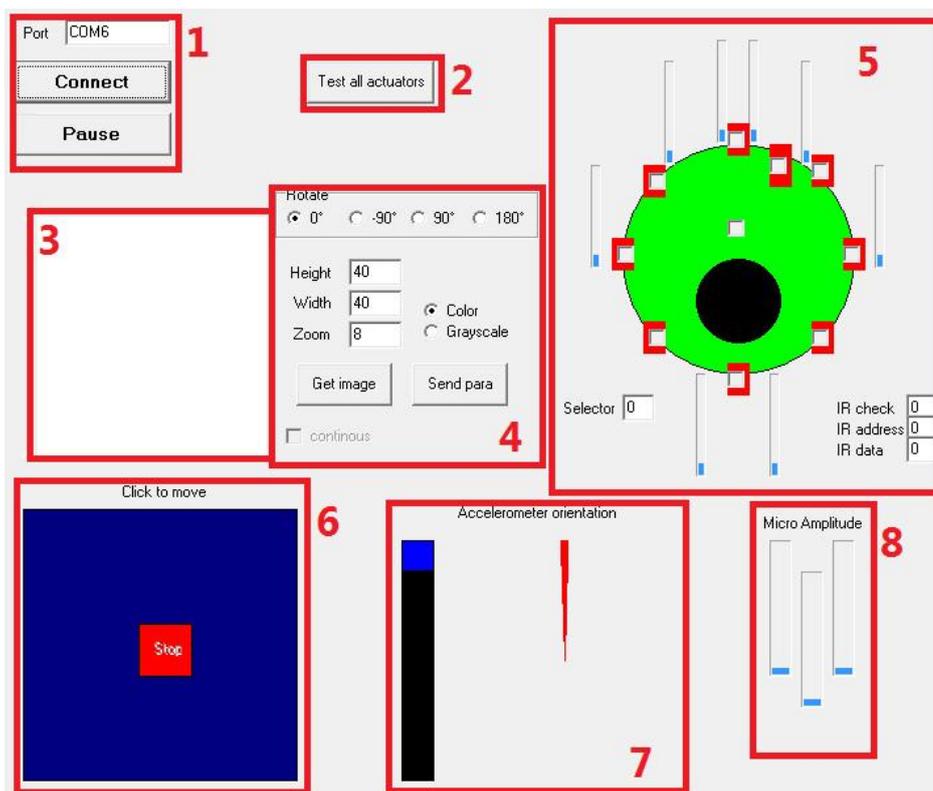


Figura 3.2. Interface do *E-Puck Monitor*.

3.2.2 Inspeção dos signos metalinguísticos

O sistema contém somente um signo metalinguístico relevante representado por uma mensagem no site do projeto *e-Puck* (figura 3.3). Não há outras indicações de como usar o software além da necessidade de haver um certo programa, chamado *sercom.hex*, carregado na memória do robô. O site também diz que o sistema é utilizado para visualizar seus sensores e manipular seus atuadores. Com poucos signos metalinguísticos, a metacomunicação reconstruída neste passo é bastante limitada.

A falta de signos metalinguísticos, especialmente de um sistema de ajuda, pode indicar que o sistema é destinado a especialistas em robótica que já possuem experiência no uso deste tipo de sistema e também conhecimento da estrutura do *e-Puck*. Isso pode

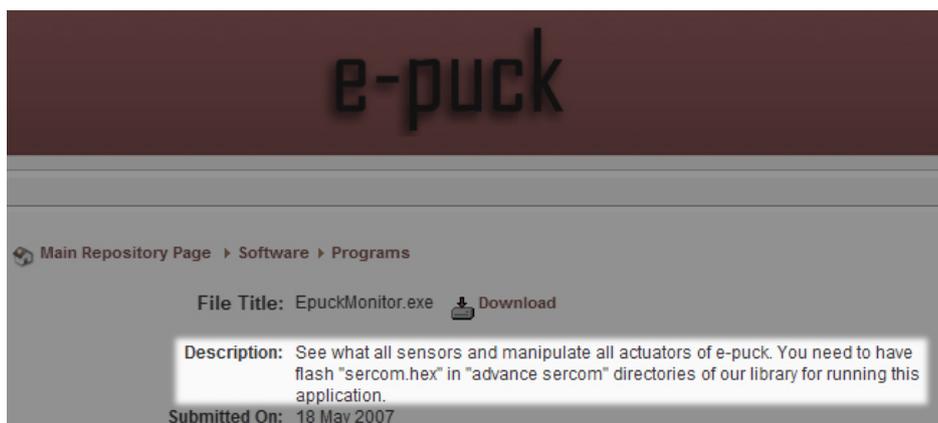


Figura 3.3. Descrição do *E-Puck Monitor* no site da EPFL.

ser suposto porque o usuário já deve saber de antemão como carregar um programa na memória do *e-Puck*, já que isso não é explicado. Pode, também, significar que o projetista acredita que a interface é fácil de usar e maiores explicações não seriam necessárias.

Através da inspeção dos signos metalinguísticos chegamos à seguinte mensagem de metacomunicação:

*“Caro usuário do **E-puck Monitor**, este é o software que criei para você testar seu e-puck, no qual você pode tanto ver as leituras dos sensores como também controlar os atuadores do robô. Para conseguir executar o software, você deve já ter carregado na memória do e-puck o programa 'sercom.hex', disponível juntamente com o software.”*

3.2.3 Inspeção dos signos estáticos

Pela análise dos signos estáticos é possível notar que a interface pode controlar somente um *e-Puck*, já que há somente um campo “port” (área 1 da Figura 3.2), relacionado à porta de conexão do robô. É possível perceber que a interface permite a leitura de dados de todos os sensores. Mas para entender a leitura dos sensores infravermelhos (IR), o usuário precisa primeiro inferir que o círculo representa o *e-Puck* (marcado como 5 na fig. 3.2) e que cada barra em volta do mesmo representa um dos seus sensores. Algumas caixas de seleção estão localizadas também em volta do círculo, mas não é trivial identificar a sua função, que é ligar é desligar os LEDs, somente observando a interface. Ele pode pensar que estes elementos são relacionados aos sensores IR em volta do robô. As outras caixas de texto (*IR check*, *IR address*, *IR data* e *Selector*) não tiveram seus significados identificados nesta etapa.

Os signos relacionados à câmera estão agrupados em uma área da tela (marcada

como 4), mas não existe nenhuma marcação explícita na tela que indique isso e algumas das funções dos signos, como o botão *"Send para"*, não são identificadas imediatamente. Os campos de texto relacionados com a câmera são facilmente interpretados, mas não há indicação de uma faixa de valores válidos a serem inseridos.

Há um signo relacionado à movimentação do robô (mostrado na área marcada como 6 na Figura 3.2), representado por um quadrado grande com um outro menor no seu centro. Sobre o quadrado maior há uma instrução (um signo metalinguístico) que diz *"Click to move"*, tentando explicar como os usuários devem interagir com o elemento. Entretanto não é claro onde deve ser clicado para se obter o movimento desejado. À primeira vista, este signo parece representar um *joystick* virtual.

A leitura do acelerômetro (área 7) pode ser interpretado por especialistas em robótica. O ponteiro indica a aceleração nos eixos X e Y simultaneamente, como uma bússola, assumindo que a orientação deste ponteiro é a mesma do círculo representando o robô, enquanto a barra vertical indica a aceleração no eixo Z. Este último signo pode ser mais difícil de interpretar, já que não remete o usuário a representações usuais do eixo Z, e nem tem uma indicação do que está sendo representado.

Outros signos relacionados à leitura de sensores são as barras de intensidade dos microfones (área 8). Olhando para estes signos, o usuário pode presumir que a intensidade dos sons em volta do robô podem ser representados pelas barras. Ele deve perceber que o posicionamento das barras está relacionado à posição dos sensores no robô.

O cenário de inspeção considera o contexto de teleoperação no qual o robô pode ser visto pelo usuário. Então, signos disponíveis no robô também devem ser inspecionados. Para o *e-Puck*, qualquer padrão de LEDs é um signo estático que poderia representar um estado do robô (apesar da ação de ligar e desligar os LEDs não constituir um signo estático). Outro signo estático é o posicionamento do robô no mundo real, que é percebido pelo usuário quando o robô está no seu campo de visão.

Os signos estáticos reforçam a afirmação de que o sistema é destinado a usuários com experiência em robótica e conhecimento sobre o *e-Puck*, já que a interface demanda que o usuário entenda a leitura dos sensores e o comportamento esperado do robô ao entrar com um comando. Um usuário inexperiente pode ter problemas de interação, com a possibilidade de causar danos ao robô, por exemplo, fazendo o mesmo cair de uma mesa devido a comandos errados de movimentação. Mas o número de signos de entrada indicam que o *e-Puck* não é muito complexo de se operar.

É interessante notar que, neste passo, dois conjuntos de signos claramente emergem: os signos do robô e os signos da interface de controle. Dos signos do robô, somente os LEDs estão representados na interface. Para localizar o robô é necessário

que o mesmo esteja no campo de visão do usuário.

Com a inspeção dos signos estáticos obtemos a seguinte metagemagem:

*“Caro usuário do E-puck Monitor, este é o software que criei para você testar as funcionalidades do seu **e-puck**. Através do sistema você poderá conectar-se a um **e-puck** e realizar várias ações com ele, como movimentação e controle de câmera. Poderá também obter informações dos sensores do robô. Estão disponíveis informações de proximidade, através dos infravermelhos, informações de direção do movimento, através dos acelerômetros, e também informações sobre a detecção de sons no ambiente, através dos microfones. Além disso, você poderá também pausar a conexão do robô a qualquer momento. A interface não irá mostrar onde o robô está localizado, então você precisará manter o **e-Puck** no seu campo de visão a todo momento. É importante notar que você deverá ter um conhecimento básico de robótica e sobre o próprio **e-Puck** para que possa utilizar a interface.”*

3.2.4 Inspeção dos signos dinâmicos

Através da inspeção dos signos dinâmicos, ou seja, da interação com o *E-puck Monitor*, é possível observar que os usuários têm a opção de desconectar o robô que está sendo controlado pela interface. Também é possível parar/reiniciar a conexão. O botão *“Test all actuators”* (área 2 da Figura 3.2) está consistente com o seu texto: o robô testa todos os seus atuadores ativando cada um deles. O robô move, liga e desliga seus LEDs e emite sons.

O signo na área marcada como 3 na Figura 3.2 mostra a imagem da câmera e os signos da área 4 são utilizados para configurá-la. A função do botão *“Send para”* pode ser identificada através dos signos dinâmicos após interagir um pouco na base da tentativa e erro. O usuário deverá inferir que deve clicar neste botão primeiro, antes de clicar no *“Get image”* para obter a imagem da câmera. Assim, o botão *“Send para”* é relacionado ao envio de parâmetros para a câmera do robô.

É possível interpretar o signo de movimento (área 6) através da interação. Quando o usuário clica num ponto dentro da grande área quadrada, o robô começa a se mover. A movimentação pode ser observada olhando diretamente para o robô, pois a interface dá poucas indicações através das leituras dos sensores, somente com a leitura do acelerômetro e com os sensores de proximidade. O ponto onde o usuário clica define a velocidade linear e angular do movimento do robô. A velocidade linear é dada pela distância vertical entre o ponto clicado e o centro do quadrado menor, enquanto a velocidade angular é definida pela distância horizontal entre os mesmos dois pontos. O usuário necessita de algum tempo experimentando para aprender e entender o compor-

tamento do robô quando interage com este signo. Não é uma abordagem simples, pois definir a velocidade angular diretamente, através do clique do mouse, não é intuitivo. O fato de a interação com este elemento não ser clara e dever ser aprendida pelo usuário por tentativa e erro pode ter sérias consequências, já que um erro na interação pode causar danos físicos ao *e-Puck*.

Quando o robô está se movendo e se aproxima de um obstáculo, as barras dos sensores de IR (área 5) mostram a proximidade do robô ao mesmo. Quanto mais cheia está uma barra, mais próximo do obstáculo está o respectivo sensor. Desta maneira é possível detectar e prevenir colisões pela leitura dos sensores de infravermelho.

A leitura do acelerômetro se comporta como esperado. Quando o robô está se movendo horizontalmente com aceleração diferente de zero, o ponteiro vermelho mostra a direção para onde o robô está se movendo enquanto houver aceleração naquela direção. Se o robô se move verticalmente, i.e., subindo uma rampa ou caindo, a barra do lado esquerdo do ponteiro, que corresponde ao eixo Z, mostra a leitura correspondente. Caso esteja sendo testado em uma superfície plana não será possível notar a variação na barra, dificultando o entendimento do seu significado. O mesmo ocorre quando o robô está se movimentando horizontalmente com velocidade constante, zerando a leitura do acelerômetro e fazendo com que o signo não aponte para a direção do movimento, pois não há aceleração.

Os signos relacionados aos microfones (área 8 na fig. 3.2) também se comportam como esperado. Quando um som é emitido e captado pelo robô, a barra de intensidade age de acordo, se enchendo para indicar a intensidade do som que chega ao respectivo microfone.

Também é possível identificar signos dinâmicos no robô. Estes signos são relacionados com a resposta à entrada dos usuários no “*E-Puck Monitor*”. Quando interage com a área descrita como “*Click to move*”, o robô se move de acordo com as velocidades linear e angular correspondentes ao local onde o usuário clica. O movimento do robô só será percebido plenamente se o mesmo estiver no campo de visão do usuário pois, como dito anteriormente, a interface dá poucas indicações sobre a movimentação do robô. Outro signo é relacionado a padrões de LEDs: o usuário pode ligar e desligar alguns dos LEDs (através das caixas de seleção) para criar um padrão de luz. A interface não permite fazer com que o *e-Puck* emita sons por outro meio a não ser o botão “*Test all actuators*”. Mesmo assim a emissão de sons pode ser considerada outro signo dinâmico do robô, pois supõe-se que seja possível emitir outros tipos de sons através do *e-Puck*.

A análise dos signos dinâmicos mostra que a interação com o *e-Puck* é baseada principalmente em tentativa e erro. Esta é uma outra evidência de que os usuários devem possuir alguma experiência prévia em robótica e conhecimento do próprio *e-*

Puck, pois sem esse conhecimento prévio o robô pode ser danificado devido a uma interação mal-sucedida (derrubando o robô de uma mesa, por exemplo).

Por fim, a metamensagem obtida através da inspeção dos signos dinâmicos é apresentada a seguir:

“Caro usuário do E-puck Monitor, este é o software que criei para você testar as funcionalidades do seu e-puck. Através do sistema você poderá conectar-se a um e-puck e realizar várias ações nele, como movimentá-lo, controlar a câmera para obter imagens e também acender/apagar os leds presentes no robô. Poderá também obter informações dos sensores existentes. Estão disponíveis informações de proximidade a objetos, através dos sensores infravermelhos, informações de direção de movimento e aceleração, através dos acelerômetros, e também informações sobre a detecção de sons no ambiente, através dos microfones. Além disso, você poderá pausar ou terminar a conexão do robô a qualquer momento. Você deverá ter conhecimento do funcionamento do robô e da disposição dos seus sensores para poder utilizar o sistema sem problemas. Caso contrário terá que dedicar alguns minutos para explorar o sistema e entendê-lo. Esses testes preliminares para compreender a interface devem ser feitos em locais seguros para o robô, como uma mesa com barreiras em volta, para evitar que o robô sofra danos, principalmente por quedas.”

3.2.5 Metacomunicação consolidada e apreciação da qualidade de interação

Depois de executar os três primeiros passos do MIS, obtivemos três mensagens de metacomunicação relacionadas a cada nível de signo. Nesta etapa, uma metacomunicação unificada foi gerada e os resultados de cada passo foram comparados.

A comparação da análise de cada nível de signo mostra que a comunicabilidade do *E-Puck Monitor* é bastante pobre. A ausência de signos metalinguísticos cria uma interação custosa para os usuários. Este alto custo vem da interação via tentativa e erro, sem saber se suas ações podem levar a danos ao robô. Esta é também uma indicação de que o sistema foi criado para usuários que já possuem um bom conhecimento da estrutura e das funcionalidades do robô. Também não há mecanismos de prevenção de erro, somente uma mensagem simples de erro mostrada quando o sistema não consegue se conectar ao robô.

O signo de entrada para controle de movimento através da interface não é adequado. A análise mostra que o comportamento esperado – de um *joystick* – é somente verificado parcialmente pelos signos dinâmicos, devido ao fato de que o usuário deve clicar em algum ponto para definir as velocidades, o que pode ser confuso. Cliques

distintos indicam ações distintas, mas dirigir um robô é uma ação contínua, o que pode levar o usuário a desenvolver interpretações errôneas quando notar que o robô está movendo de maneira diferente da esperada. Como exemplo, ele pode pensar que o robô não está funcionando corretamente quando o mesmo começa a girar sobre o próprio eixo porque o usuário clicou do lado direito do signo, definindo somente a velocidade angular. Mudar o estilo de interação de “clicar-e-soltar” para “clicar-e-segurar” pode ser uma abordagem mais direta para a simulação de um *joystick*.

Os signos relacionados à câmera também possuem problemas. Os valores-padrão definidos na interface funcionam, mas não há nenhum signo estático ou metalinguístico para indicar aos usuários a faixa de valores válida se os mesmos tentarem mudar estes valores. Caso o usuário entre valores fora da faixa válida e acione o botão “*Get image*”, o sistema não lhe dará nenhum retorno, somente voltando os campos para os valores-padrão. Quando isto acontece, os usuários podem notar que não interagiram de forma correta com a interface, pois nada muda na imagem da câmera. Mas eles podem não notar que devem primeiro acionar o botão “*Send para*” para definir os parâmetros da câmera. Também é possível notar que o rótulo do botão deve ser alterado, pois “*para*” não remete claramente a “*parameters*”. Assim, “*Send params*” poderia ser um rótulo mais adequado.

Muitas das funções da interface podem ser identificadas somente através dos signos dinâmicos, já que não há signos metalinguísticos relevantes associados a elas e os signos estáticos não são claros o suficiente. Desta forma é reforçada a observação de que a interação deve ser por tentativa e erro e pode causar problemas às entidades físicas (os *e-Pucks*) controladas pelo sistema.

Os signos identificados pela análise mostram que a interação com o sistema consiste basicamente de um ciclo representado pelos seguintes passos: a) Observar a leitura dos sensores na interface; b) Enviar comandos ao robô baseado nas informações obtidas; e c) Monitorar a resposta à execução dos comandos.

Este *feedback* é transmitido através da interface do sistema e/ou diretamente pelo robô. A interface pode prover o *feedback* através dos signos que representam a leitura dos sensores, como a proximidade a um obstáculo mostrado pelos sensores de IR e pela câmera. O *feedback* fornecido diretamente pelo robô somente pode ser utilizado se o mesmo estiver dentro do alcance da visão de forma que seus signos possam ser percebidos (como considerado no cenário de inspeção).

Baseado na inspeção semiótica realizada utilizando o MIS foi possível classificar os signos identificados em categorias que caracterizam os aspectos da interação com o robô a que estão associados. As classes de signos são: movimentação, posicionamento e ambiente. Signos de movimentação compreendem alguns signos que indicam infor-

mação de entrada ou saída relativa à movimentação do robô, como o quadrado grande que serve para mover o robô, o movimento do próprio robô, a leitura do acelerômetro, que mostra a direção do movimento, a imagem da câmera, que mostra se o robô está se movendo. Signos de posicionamento são signos que se referem à posição do robô em um ambiente. No caso do *E-Puck monitor* o signo de posicionamento identificado é o próprio posicionamento do robô, enquanto pode ser visto pelo usuário. Signos relacionados ao ambiente são aqueles que dão informações sobre os arredores do robô, como a leitura dos microfones e também a imagem da câmera e a leitura dos sensores de IR. A câmera pertence a duas classes (movimentação e ambiente), isso ocorre porque, a movimentação pode ser percebida pelas mudanças ocorridas na imagem, mas o ambiente e suas mudanças também são percebidos através da mesma imagem, como no caso de algum objeto cair na frente do robô e ser captado pela câmera.

Com isso chegamos à metamensagem consolidada para o E-puck Monitor:

“Caro usuário do E-puck Monitor, este é o software que criei para você testar as funcionalidades do seu e-puck. Através do sistema você poderá conectar-se a um e-puck e realizar várias ações nele, como movimentá-lo, controlar a câmera para obter imagens e também acender/apagar os leds presentes no robô. Poderá também obter informações dos sensores existentes. Estão disponíveis informações de proximidade a objetos, através dos sensores infravermelhos, informações de direção de movimento e aceleração, através dos acelerômetros, e também informações sobre a detecção de sons no ambiente, através dos microfones. Além disso, você poderá pausar ou terminar a conexão do robô a qualquer momento. Você deverá ter conhecimento do funcionamento do robô e da disposição dos seus sensores para poder utilizar o sistema sem problemas. Caso contrário terá que dedicar alguns minutos para explorar o sistema e entendê-lo. Esses testes preliminares para compreender a interface devem ser feitos em locais seguros para o robô, como uma mesa com barreiras em volta, para evitar que o robô sofra danos, principalmente por quedas. Para conseguir executar o software, você deve já ter carregado na memória do e-puck o programa 'sercom.hex', disponível na biblioteca do software”

3.2.6 Triangulação com o ePic

A inspeção semiótica do *E-Puck Monitor* permitiu a identificação de algumas classes de signos que emergiram da interação com o sistema. Estas classes identificadas são relacionadas não só com a interface de controle, mas também com o próprio robô. Para reforçar a análise, a interface *ePic* também foi inspecionada através do MIS. O *ePic* é uma interface similar ao *E-Puck Monitor* que deve ser utilizada junto ao MatLab®.

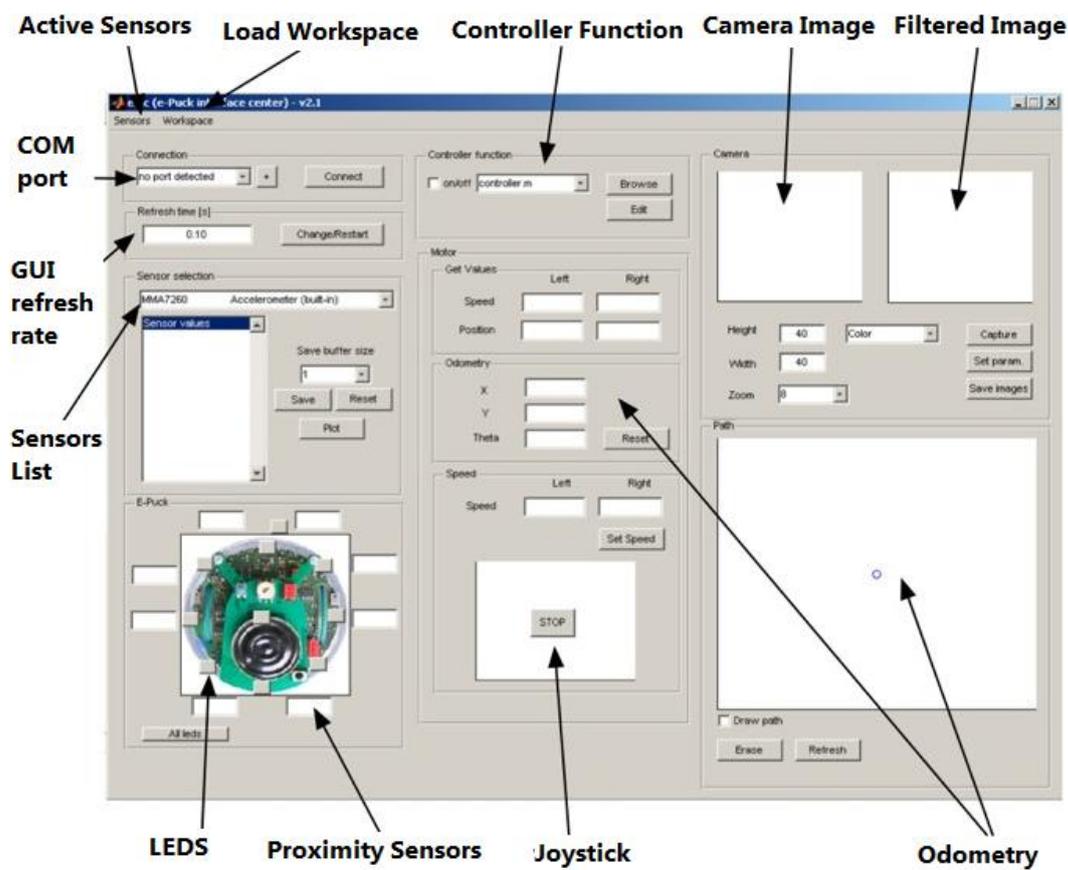


Figura 3.4. Interface do *ePic*.

Esse sistema é um pouco mais complexo que o *E-Puck Monitor*, mas também contém somente uma tela. Assim, como na inspeção anterior, todo o sistema foi inspecionado e o cenário utilizado também foi o mesmo. Nesta seção não será apresentada a análise detalhada do *ePic* como foi feito com o *E-Puck Monitor*. O foco será somente nos aspectos que são relevantes para as classes de signos identificadas e os signos utilizados para transmitir estes aspectos. Alguns detalhes técnicos também serão apresentados para que as diferenças entre os dois sistemas sejam entendidas.

Existe uma documentação disponível para o *ePic* no site do projeto *e-Puck*. Esta documentação explica de forma sucinta como usar a interface, como lidar com a leitura dos sensores, com o movimento do robô e o controle da câmera. Também mostra que é possível adicionar um controlador personalizado ao e-Puck, permitindo algum comportamento autônomo por parte do mesmo. A documentação também descreve muitos aspectos da interação com o sistema cujo entendimento foi deixado a cargo do usuário na interação com a interface anterior (o *E-Puck Monitor*). Mesmo que não seja esperado que os usuários leiam a documentação antes de usar o sistema, ela está disponível para consulta caso haja alguma ruptura comunicativa. Algumas

partes do sistema também possuem *tooltips*, mostrando o significado de alguns signos e ajudando os usuários a interagirem com eles. Estas mensagens podem ser vistas no signo de controle dos LEDs, no signo relativo ao sensor de proximidade e na seleção de controlador.

Alguns aspectos da interação com o *ePic* são os mesmos da interação com o *E-Puck Monitor*, mas com diferentes elementos de interface para representá-los. O controle dos LEDs não são caixas de seleção, mas sim botões com estado. A leitura dos sensores não é representada por barras de intensidade, mas pelos seus valores numéricos. Excetuando o sensor de proximidade, a leitura dos outros sensores não pode ser feita de forma simultânea: o usuário deve selecionar um sensor de cada vez e somente seus valores são mostrados. Esta seleção é feita através da área “*Sensor list*”, mostrada na Figura 3.4. Também é possível gerar gráficos com as leituras de cada sensor através do botão “*Plot*”, localizado abaixo da lista de sensores.

A decisão de mostrar somente o sensor selecionado pode causar problemas aos usuários porque eles só podem estar cientes de uma das leituras por vez. Isto significa que se um sensor obtém uma leitura importante e o mesmo não está selecionado, o usuário não tomará conhecimento desta leitura, provavelmente afetando a interação com o robô.

Para movimentar o robô, o usuário pode interagir de maneira similar ao *E-Puck Monitor*, clicando em uma área retangular para definir as velocidades do robô de acordo com o ponto clicado. A diferença é que agora existem também dois campos de texto localizados logo acima da área do joystick que mostram a velocidade linear e angular quando o usuário clica nesta área. Desta forma os usuários podem identificar mais facilmente como este signo está relacionado à movimentação do robô. Eles também podem inserir os valores diretamente nos campos de texto, tornando-os signos tanto de entrada quanto de saída.

Novos signos são introduzidos pelo *ePic*: existem signos relacionados à odometria que se referem à pose do robô. Através destes signos os usuários podem ver o caminho do robô e sua pose relativa em um plano, dados por X, Y e *theta* (orientação). Estes signos também comunicam ao usuário que o *e-Puck* possui outro sensor (a odometria) que não foi mencionado no *E-Puck Monitor*.

Com a adição de um controlador personalizado, é possível implementar um comportamento novo no robô através de um script de MatLab. Este comportamento criado é um novo signo dinâmico no *e-Puck*. Através do controlador o projetista permite aos usuários adicionarem novos signos dinâmicos ao robô utilizando os signos estáticos já existentes.

A inspeção mostrou que o *ePic* é um sistema que pretende ser uma alternativa

ao *E-Puck Monitor* através de uma melhora na interface existente e permitindo uma interação mais complexa através do controlador personalizado, mas os usuários deverão possuir conhecimento de scripts em MatLab para tal. A comunicabilidade do ePic também é melhor, permitindo aos usuários interagir com o sistema com menos tentativas e erros.

Como na inspeção do *E-Puck Monitor*, na inspeção do *ePic* também surgiram signos do robô. Estes signos dependem dos atuadores que o mesmo possui e sua função primária é dar *feedback* para o usuário. Estes signos podem aparecer como signos visuais, tais quais os LEDs ao redor do *e-Puck* que podem ser utilizados para criar padrões luminosos indicando diferentes estados do robô, ou o movimento do robô, e.g., quando ele muda de direção por causa de um obstáculo. Existem também signos sonoros, no caso do robô poder emitir sons (o *e-Puck* tem saídas de som, então é possível utilizar tais signos). Observa-se que os signos do robô são na sua maioria dinâmicos, surgindo do processo de interação. Existem signos estáticos, como os padrões de LEDs, mas estes não serão mostrados até que o robô mude de estado.

Foi observado que os signos do robô podem ser independentes do contexto, i.e., estes signos aparecem em qualquer aplicação do robô, como a própria movimentação. Estes signos passam a ser dependentes do contexto quando são associados a significados específicos, como por exemplo um padrão de LEDs para mostrar que o robô encontrou um objeto num contexto de busca. Existem signos (dependentes de contexto ou não) cuja percepção pelo usuário dependerá da aplicação. Um exemplo é a emissão de sons: se a aplicação não necessitar deste tipo de signo, o usuário não terá como interagir com o mesmo, apesar do signo existir mesmo assim.

Contrastando os resultados das duas inspeções foi possível notar que apesar de usarem diferentes signos, os sistemas geraram interpretações similares. Como apresentado, um novo signo surgiu da inspeção do *ePic*: o signo relacionado à odometria, mostrando a posição do *e-Puck* relativo a um dado ponto de referência. Este signo também pode ser relacionado à movimentação, já que mostra o caminho do robô. É importante ressaltar que, conceitualmente, o odômetro é um sensor proprioceptivo¹, mas a informação transmitida pelo signo (o caminho) tem relação com a movimentação do robô. A triangulação também permitiu verificar que as classes de signos identificadas foram consistentes.

¹Relativo ao estado interno do robô.

3.3 Classes de Signos

Através da aplicação do MIS, mostrada neste capítulo, foi possível identificar três classes de signos na interação com o e-Puck: Movimentação, Posicionamento e Ambiente. As interfaces inspecionadas não são direcionadas a nenhuma tarefa específica. Assim, não há signos que representem a situação atual da tarefa sendo executada.

Observando interfaces presentes em outros trabalhos como o de Scholtz et al. [Scholtz et al., 2004] e Richer e Drury [Richer & Drury, 2006], pode-se perceber que existem pelo menos mais duas classes de signos comumente usadas em interface para controle de robôs: signos relacionados com a tarefa sendo executada e também relacionados ao estado do robô. No trabalho de Scholtz et al [Scholtz et al., 2004] temos signos nas interfaces apresentadas que mostram o status da tarefa de busca e resgate, como a possível presença de vítimas e a proximidade das mesmas. Também eram mostrados os locais onde vítimas já foram encontradas.

No trabalho de Richer e Drury [Richer & Drury, 2006] temos interfaces que mostram o estado do robô e seu funcionamento, representando o estado da bateria e também da comunicação. Estes signos são importantes para robôs operados à distância, pois sabendo como está o nível da bateria do robô é possível prever até quando o robô pode prosseguir com a tarefa antes de chegar ao nível crítico de bateria necessária pra voltar para um lugar seguro. Sabendo o estado da comunicação (nível do sinal captado pelo robô) pode-se evitar áreas onde o robô tem chances de ficar incomunicável.

Signos que seriam relacionados ao estado do robô e da tarefa foram citados na fase de análise da inspeção com o MIS, como no caso de padrões luminosos para identificar níveis de bateria ou sinais sonoros/luminosos para mostrar que o robô realizou alguma tarefa, e.g., chegou ao local de destino. Mas estas classes não foram contempladas nos sistemas avaliados, tanto pela simplicidade dos sistemas quanto pelo cenário de avaliação.

Dessa forma, as duas classes de signos mencionadas acima se juntam às identificadas pela inspeção, totalizando cinco classes de signos: Movimentação, Posicionamento, Ambiente, Funcionamento e Tarefa. Com base nos resultados da aplicação do MIS e nas métricas para tarefas apresentadas por Steinfeld et al. [Steinfeld et al., 2006], que possuem elementos em comum com os identificados na inspeção, temos a descrição das classes apresentadas:

- **Movimentação:** Classe que representa os signos relacionados à movimentação do robô. Compreende principalmente o destino da movimentação do robô, a direção do movimento e a velocidade do deslocamento.

- **Posicionamento:** Representa os signos relacionados ao posicionamento do robô. Este posicionamento pode ser relativo a um dado ponto ou absoluto dentro de um mapa. Compreende o posicionamento do robô com granularidade mais fina, como o local dentro de uma sala, ou mais grossa, como em que sala o robô se localiza dentro de um prédio ou em que local do mundo ele está (com o uso de GPS, por exemplo). Também faz parte desta classe a orientação do robô num dado momento e a distância que o robô está de pontos de referência arbitrários.
- **Ambiente:** São signos que representam o estado do ambiente ao redor do robô, como a presença de obstáculos, a detecção de mudanças no ambiente (e.g. temperatura, umidade) ou mudanças que modifiquem a execução de tarefas (e.g. portas que estão fechadas ou o aparecimento de obstáculos não previstos no mapa, caso utilizado).
- **Funcionamento:** Signos que representam o estado interno do robô e o seu funcionamento, como nível de bateria, nível de sinal da comunicação, estado dos atuadores e também dos sensores (se há algum problema ou não).
- **Tarefa:** Esta classe compreende os signos relacionados à execução de tarefas pelo robô. Os signos presentes dependem do tipo de tarefa que será executada. No caso de busca e resgate temos a detecção de vítimas, a avaliação de locais perigosos, a distância do robô a essas vítimas, dentre outros. Em tarefas de reconhecimento ou vigilância temos o nível de cobertura proporcionado pelo robô, se ele detectou elementos estranhos durante a execução, etc.

A identificação das classes acima permitiu observar que o MIS é aplicável no contexto de teleoperação de robôs móveis. Um aspecto que deve ser notado é não terem sido necessárias modificações no método para a aplicação nesse contexto. Com estas classes definidas as interfaces para controles de robôs utilizando diferentes tecnologias foram modeladas e desenvolvidas. O próximo capítulo apresentará a modelagem da interação utilizando a MoLIC [Barbosa & de Paula, 2003] e o desenvolvimento das interfaces para cada uma das tecnologias abordadas: *desktop*, *Tablet PCs* e o *iPod touch*®, como representado na figura 3.5.

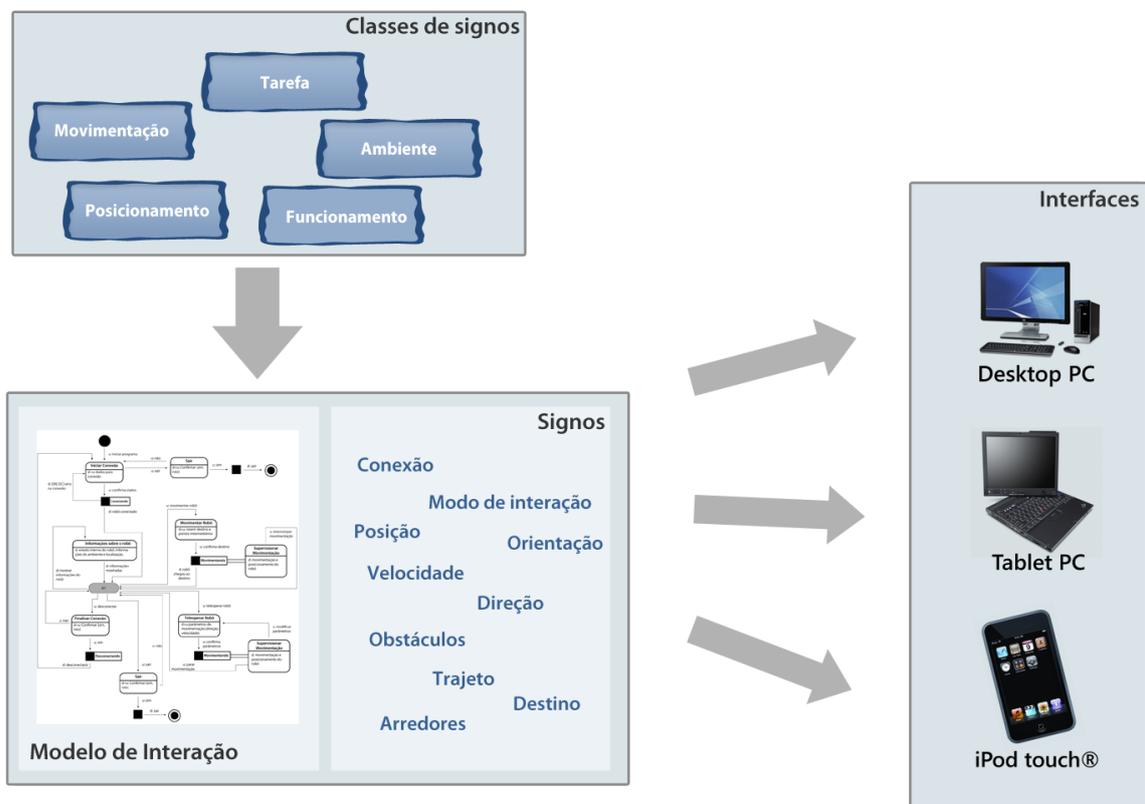


Figura 3.5. Signos, modelagem e interfaces. A partir dos signos foi gerado o modelo de interação, que deu origem às três interfaces implementadas.

Capítulo 4

Desenvolvimento das Interfaces

Foram desenvolvidas três interfaces para controle de robôs, considerando tecnologias diferentes. Foi desenvolvida uma interface para *desktops*, uma para *Tablet PCs* e outra para dispositivos móveis (no caso, o *iPod touch*® da *Apple*®). Todas as interfaces foram desenvolvidas com base nas classes de signos identificadas e também com base em um único modelo de interação. O modelo de interação foi criado utilizando a MoLIC [Barbosa & de Paula, 2003], uma linguagem para modelagem de interação baseada na Engenharia Semiótica. Depois da criação do modelo de interação o passo seguinte foi definir os signos que seriam utilizados em cada interface, considerando as classes de signos identificadas e as diferentes tecnologias utilizadas.

Além disso, para simplificar o desenvolvimento e permitir a utilização do robô com interfaces criadas em diferentes sistemas operacionais, foi implementado um servidor que é responsável pela comunicação com o *e-Puck*. Dessa maneira as interfaces podem ser desenvolvidas em qualquer linguagem de programação que suporte comunicação de rede via *sockets*.

4.1 MoLIC

A MoLIC - *Modeling Language for Interaction as Conversation* - [Barbosa & de Paula, 2003, da Silva, 2005, da Silva & Barbosa, 2007] é uma linguagem de modelagem que representa a interação humano-computador como um conjunto de conversações que os usuários podem ou devem ter com um sistema para atingir seus objetivos. Um modelo da MoLIC representa as possíveis conversações interativas que um usuário poderá ter com um sistema, ou seja, os caminhos da interação, inclusive os caminhos alternativos para um mesmo objetivo, e caminhos para a recuperação de rupturas na interação ou erros.

Para que os usuários entendam melhor o interlocutor, também chamado de *preposto do projetista*, a interface deve comunicar de forma adequada o que o sistema fez ou não, o que está fazendo ou não, o que permite ou não fazer, como fazer e por quê.

Quatro artefatos compõem a MoLIC: um diagrama de metas, um diagrama de interação, um esquema conceitual de signos, e uma especificação dos signos.

O diagrama de metas apresenta o que pode ser feito com o sistema. Este diagrama é diferente de um modelo de tarefas, sendo usado somente para definir **o que** o usuário quer realizar, sem considerar **como** isso vai ser feito. O diagrama de interação define como as metas podem ser atingidas na interação, o esquema conceitual de signos define e organiza os conceitos envolvidos no sistema e a especificação de signos detalha os signos utilizados no diagrama de interação para servir de base entre a modelagem e o projeto de interfaces.

Diferentemente da conversa entre interlocutores humanos, que é um processo dinâmico e sem passos previsíveis, a conversa que ocorre em IHC (a metacomunicação entre o usuário e o projetista mediada pelo sistema, de acordo com a engenharia semiótica) possui todos os seus passos definidos, por ter sido criada pelo projetista do sistema. Ou seja, todas as falas que o usuário e o projetista (através do sistema) podem emitir fazem parte do projeto do discurso interativo.

Geralmente uma conversa é composta de tópicos que, por sua vez, são compostos de um conjunto de diálogos. Os diálogos são compostos de falas, sendo que cada fala transmite uma mensagem composta por um conjunto de signos (de interface, no caso de IHC). A MoLIC possui um conjunto de elementos para representar esta conversa no diagrama de interação. Alguns destes elementos, que foram utilizados na modelagem dos sistemas deste trabalho, são: cenas, processos do sistema, falas de transição e acesso ubíquo. Estes elementos serão descritos brevemente a seguir¹.

Uma cena representa uma conversa entre o usuário e o projetista (novamente, através do sistema) sobre algum tópico. Um tópico geralmente está associado a uma meta ou tarefa do usuário (ou parte delas) e pode conter diálogos. A conversa descrita por uma cena pode ocorrer em algum elemento da interface, como uma janela ou tela do sistema. Um exemplo de cena é mostrado na Figura 4.1.

Processos do sistema são representações para momentos durante a interação onde há algum processamento do sistema para definir qual o próximo tópico da conversa. A notação de “caixa preta” foi utilizada porque o usuário não nota que o processamento ocorre (e nem como ocorre), a não ser que o sistema o comunique. Um exemplo de processo de sistema é mostrado na Figura 4.2-a. O processo pode comunicar ao usuário

¹A notação utilizada se refere à segunda versão da MoLIC[da Silva & Barbosa, 2007]



Figura 4.1. Representação de uma cena na MoLIC. A letra “d” representa uma fala do sistema e a letra “u” representa uma fala do usuário.

o estado atual do processamento. Isto passa a ser importante no caso de processamentos longos, nos quais o usuário pode não entender o que está acontecendo e ter uma ruptura na conversa. O processo com indicação de progresso é mostrado na Figura 4.2-b. Um processo também pode possuir uma cena associada, permitindo que uma conversa seja conduzida enquanto o sistema está em processamento (Figura 4.2-c).

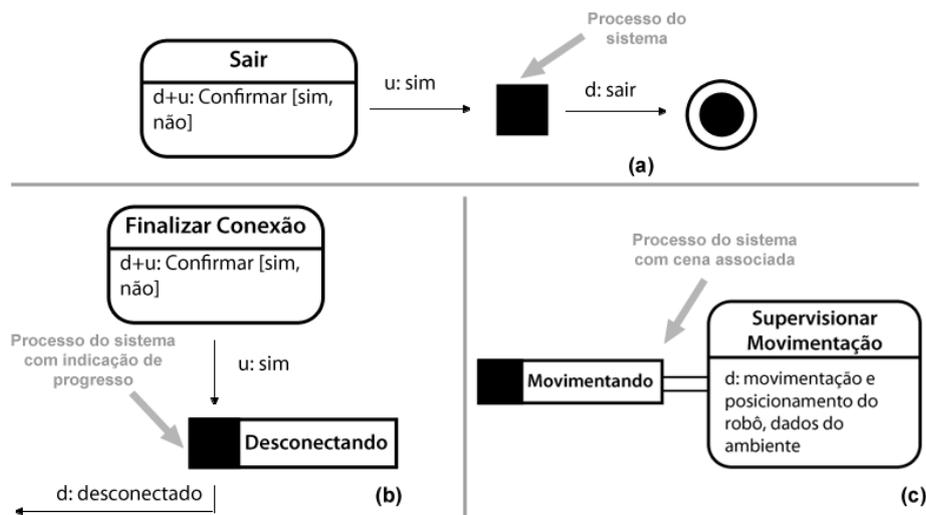


Figura 4.2. Representações de processos do sistema na MoLIC.

As falas de transição são utilizadas para mudar de tópico em uma conversa e dispara um processo de sistema que vai definir o novo rumo da conversa. Uma fala de transição é representada por uma seta preta indicando a direção da transição e um rótulo (Figura 4.3-a). Se esta fala for consequência de uma ruptura, i.e., é uma fala de recuperação de ruptura, a seta é representada pontilhada (Figura 4.3-b). O rótulo é composto por uma indicação do interlocutor, sendo utilizado “u” para usuário ou “d” para o sistema², como visto na Figura 4.3.

²A letra “d” vem de *deputy* (*designer’s deputy*, ou preposto do usuário). Na versão em português

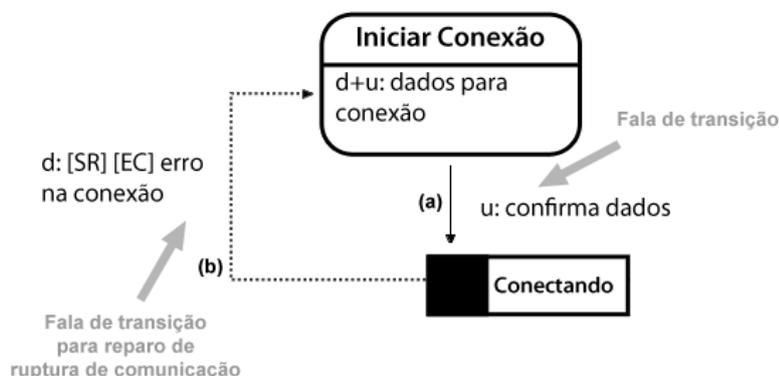


Figura 4.3. Representações de falas de transição na MoLIC.

Quando a fala é de recuperação de ruptura, temos a classificação destas adicionada ao rótulo, como visto na Figura 4.3. Estas falas podem ser classificadas da seguinte maneira:

- **Prevenção passiva (PP, de *passive prevention*):** Rupturas que podem ser indicadas por documentação ou instruções do sistema. Ex: comunicar previamente que o formato do campo de telefone é “(xx) xxxx-xxxx” (onde x representa um número).
- **Prevenção ativa (AP, de *active prevention*):** Rupturas que podem ser evitadas pelo sistema, como, por exemplo, desabilitar um botão da tela de acordo com o resultado de um processamento do sistema.
- **Prevenção apoiada (SP, de *supported prevention*):** Situações detectadas como possíveis rupturas, mas que a decisão de ser ou não uma ruptura deve ser tomada pelo usuário. Um exemplo é a mensagem de “Arquivo já existe. Deseja sobrescrever?” apresentada quando o usuário deseja salvar um arquivo com o nome de um outro que já exista. Ele pode querer sobrescrever ou pode ter colocado o mesmo nome por engano, cabendo a ele mesmo a decisão.
- **Recuperação apoiada (SR, de *supported recovery*):** Rupturas que devem ser tratadas pelo usuário com o apoio do sistema. Ex: mensagem de erro para um campo obrigatório de um formulário deixado em branco, dando a oportunidade para o usuário preenchê-lo.

da MoLIC é usada a letra “p”, de preposto, para o mesmo propósito.

- **Captura de erro (EC, de *error capture*):** Situações em que não há uma ruptura comunicativa, mas erros do sistema. Quando estes erros são identificados, devem ser notificados ao usuário e, se possível, sugerir formas de recuperação do erro. Como exemplo temos a mensagem de erro de comunicação em um cliente de e-mail que ocorre quando o servidor de mensagens não está acessível.

Existem cenas que podem ser o destino de falas de transição provenientes de qualquer outra cena, não importando o tópico atual da conversa. Estas cenas são chamadas de *acessos ubíquos* e são representadas como na Figura 4.4. Este acesso ubíquo representa qualquer cena do diagrama de interação de onde o usuário pode sair e para onde ele e o sistema podem voltar utilizando falas de transição.



Figura 4.4. Representação de acesso ubíquo na MoLIC.

4.2 Modelagem

4.2.1 Modelo de Interação

Para ter uma mesma base na implementação das interfaces foi desenvolvido um único modelo de interação utilizando a MoLIC. O diagrama de metas é descrito a seguir e apresentado na figura 4.5. O diagrama contém as seguintes metas dos usuários:

- **Conectar/desconectar a um *e-Puck*:** O usuário pode iniciar a conexão com o *e-Puck* assim que quiser. Uma vez conectado, deve ser possível finalizar a conexão a qualquer momento.
- **Obter informações do robô:** O usuário pode recuperar informações obtidas pelo robô. Esta meta se divide em três:
 - **Saber o que está ao redor do robô:** Saber o que há ao redor do *e-Puck*, como obstáculos, paredes e outros objetos de interesse.
 - **Saber onde o robô está:** Obter informações sobre a localização do robô em relação a um dado referencial (dentro de uma sala, por exemplo).
 - **Obter informações do estado do robô:** Recuperar informações do estado interno do robô, como o nível de bateria.

- **Movimentar o robô através de pontos de destino *waypoints*:** O usuário poderá movimentar o robô informando ao mesmo um trajeto a ser percorrido. Uma vez definido os pontos do trajeto, o robô se movimentará de forma autônoma até chegar ao destino ou ser interrompido.
- **Teleoperar o robô:** O usuário poderá controlar o robô manualmente, guiando-o pelo ambiente.

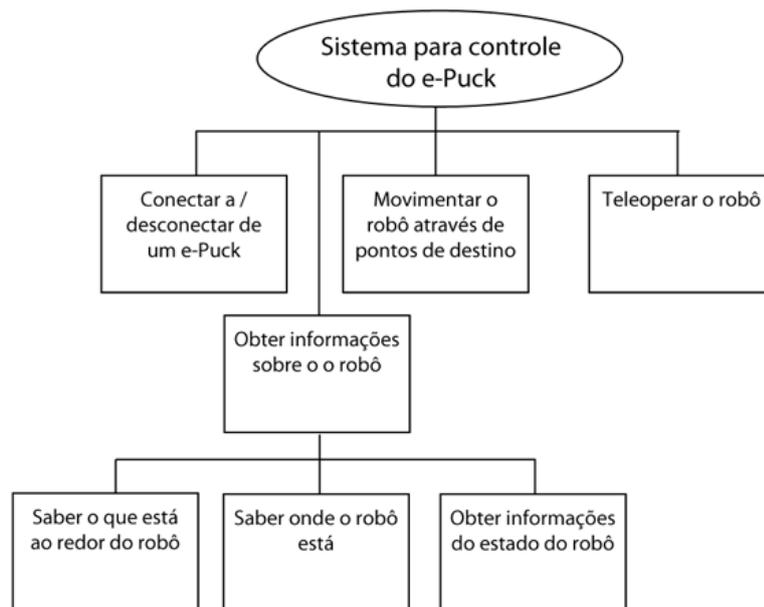


Figura 4.5. Diagrama de metas.

A Figura 4.6 apresenta o diagrama de interação para as interfaces a serem implementadas. O acesso ubíquo “U1” corresponde ao estado do sistema depois da conexão ao robô. Deste acesso é possível partir para qualquer outra cena do diagrama, enquanto as informações do robô são mostradas pela cena “Informações do robô”, acessada por uma fala de transição do sistema.

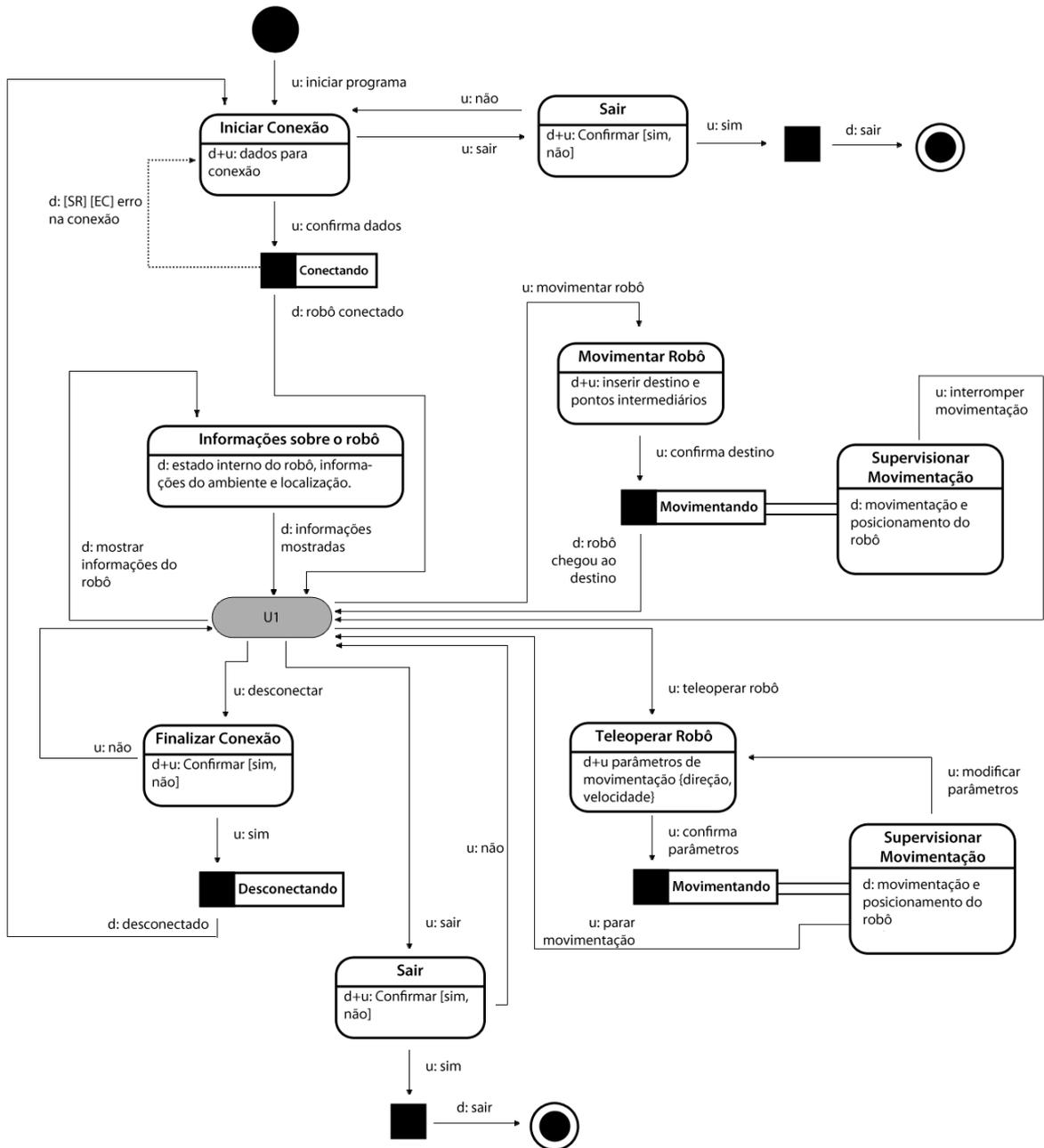


Figura 4.6. Diagrama de interação.

4.2.2 Definição dos tipos de signos

O próximo passo é definir o esquema conceitual de signos. Este esquema é definido na forma de tabela, contendo atributos referentes aos signos. Todos os atributos e descrições completas se encontram no guia prático da MoLIC[da Silva & Barbosa, 2007].

A tabela apresentada aqui contém somente os campos que são relevantes para o trabalho. Foi adicionado o campo “classe” à tabela para classificar os signos de acordo com as classes identificadas no capítulo anterior. Além da classe, temos os seguintes campos:

- Identificador: Identificação do signo.
- Descrição: Descrição do signo.
- Fonte: Qual a origem do signo.
- Conteúdo: Tipo de conteúdo do signo.
- Prevenção: Formas de prevenção de rupturas (*PP*, *AP* ou *SP*).
- Recuperação: Formas de recuperação de rupturas (*SR* ou *EC*).
- Emissor: Quem emite o signo (“d” para preposto, “u” para usuário e “d+u” para ambos).

Nota-se que não há signos relacionados à classe “Tarefa”. Isso ocorreu porque o objetivo atual era somente controlar o *e-Puck* e a tarefa era genérica, i.e., qualquer tarefa que pudesse ser executada por um *e-Puck*. Dessa forma, nenhuma representação de aspectos da tarefa foram incluídas na interação.

Se não fosse a intenção, o fato de que nenhum signo da classe “Tarefa” foi representado poderia levar o projetista a considerar se esse era realmente o caso ou não de se representar tal classe, dado o presente contexto. Se não fosse, ele então iria refletir que signos deveriam fazer parte da interação entre o sistema e o usuário e quais tópicos da conversação (diagrama de interação) devem estar relacionados.

Este exemplo mostra como classes de signos podem levar o projetista a refletir sobre aspectos específicos do modelo de interação humano-robô. Dessa forma, podemos argumentar que adicionar as classes de signos incrementa a habilidade da MoLIC de dar suporte a projetistas de interação humano-robô nas suas reflexões e, conseqüentemente, decisões sobre o sistema sendo projetado.

O passo seguinte é a especificação dos signos, que não foi feita de forma tabular de acordo com o guia da MoLIC [da Silva & Barbosa, 2007], mas sim de forma descritiva

Tabela 4.1. Esquema conceitual de signos.

Identificador	Descrição	Fonte	Conteúdo	Prevenção	Recuperação	Emissor	Classe
Status da conexão	Signo(s) que deve(m) indicar o status da conexão da interface com o robô	aplicação similar	texto ou imagem			d, d+u	Funcionamento
Modo de interação	Signo(s) para indicar se o modo atual de interação é por teleoperação ou por trajeto.	aplicação	texto ou imagem	PP: Mudança do modo é de forma alternada, sempre com um modo ativo.		d, d+u	Funcionamento
Posição	Signo(s) para indicar onde o robô está.	domínio	imagem			d	Posicionamento
Orientação	Signo(s) para indicar qual a orientação do robô.	domínio	imagem			d	Posicionamento
Velocidade	Signo(s) para indicar a velocidade do robô.	domínio / aplicação similar	imagem			d, d+u	Movimentação
Direção	Signo(s) para mostrar a direção do movimento do robô.	domínio	imagem			d, d+u	Movimentação
Trajeto	Signo(s) para indicar se o trajeto a ser percorrido pelo robô.	domínio	imagem		SR: Remover pontos do trajeto	d, d+u	Movimentação
Destino	Signo(s) para indicar se o robô chegou ao seu destino.	domínio	texto ou imagem			d	Posicionamento
Proximidade a obstáculos	Signo(s) para mostrar a proximidade do robô a obstáculos.	domínio	imagem			d	Ambiente
Arredores	Signo(s) para mostrar o que há ao redor do robô.	domínio	texto ou imagem			d	Ambiente
Imagem fotografada	Signo(s) para mostrar ao usuário o que foi fotografado pelo robô.	aplicação similar	imagem			d	Ambiente

para cada uma das interfaces desenvolvidas. Esta especificação detalha os signos de acordo com o esquema mostrado aqui e é apresentada na próxima seção.

4.3 Implementação

4.3.1 Servidor de controle do e-Puck

Como dito anteriormente, um servidor foi implementado para centralizar e isolar a comunicação com o robô, permitindo a criação de interfaces em diferentes linguagens de programação. Toda a lógica de controle do robô está implementada no servidor, sendo que as interfaces passam apenas comandos através de uma conexão de rede e recebem respostas sobre a execução destes e sobre o estado atual do robô. Assim é possível ter liberdade na escolha da linguagem de programação mais adequada para cada interface e não é necessário portar as bibliotecas de controle para cada uma delas.

Outro motivo para o desenvolvimento do servidor é a questão da localização do robô. O *e-Puck* possui somente o cálculo da sua posição de acordo com a odometria, gerando erros cumulativos que tornam o uso deste dado inviável. A solução utilizada foi através do uso de câmeras situadas por cima da área onde o robô vai atuar. Estas câmeras, com o uso da biblioteca *silver* [Garcia & Chaimowicz, 2009], localizam o robô com maior precisão e menos erros que a odometria. Esta biblioteca localiza o robô através de um padrão gráfico colocado sobre o mesmo. O uso das câmeras limita onde o robô poderá ser utilizado, pois o custo de se montar a estrutura para a localização não é baixo. Para os propósitos deste trabalho a limitação imposta pelo uso das câmeras não é relevante, pois é possível fazer testes como os descritos no cenário das inspeções sem muitas dificuldades.

O servidor é utilizado para fornecer a imagem do mapa e as definições de tamanho e escala deste para que a interface possa apresentá-lo corretamente. Assim o mapa deve ser fornecido *a priori* para o sistema, sendo que nesta implementação não caberá ao robô mapear o ambiente ou adicionar informações ao mapa. Outra função é recuperar as imagens da câmera do *e-Puck* e transmití-las para as interfaces. Por restrições do hardware do robô, a imagem recuperada da câmera possui baixa resolução e é monocromática.

O servidor foi desenvolvido em ambiente Linux, na linguagem C++, e se comunica com o robô diretamente através do *driver* [Garcia & Chaimowicz, 2009] para utilização do *e-Puck* no *player* [Gerkey et al., 2003], um servidor para controle de robôs que permite abstrair a camada de conexão e outros aspectos de cada tipo de robô. Desta forma, pode-se, com pouca alteração no código, trocar o tipo de robô que está sendo

usado ou até substituí-lo por um robô simulado no ambiente *stage*[Gerkey et al., 2003]. As interfaces desenvolvidas se comunicam com o servidor através da troca de mensagens de texto na rede enviando comandos ao robô e solicitando informações sobre seu estado atual. A arquitetura é mostrada na Figura 4.7 .

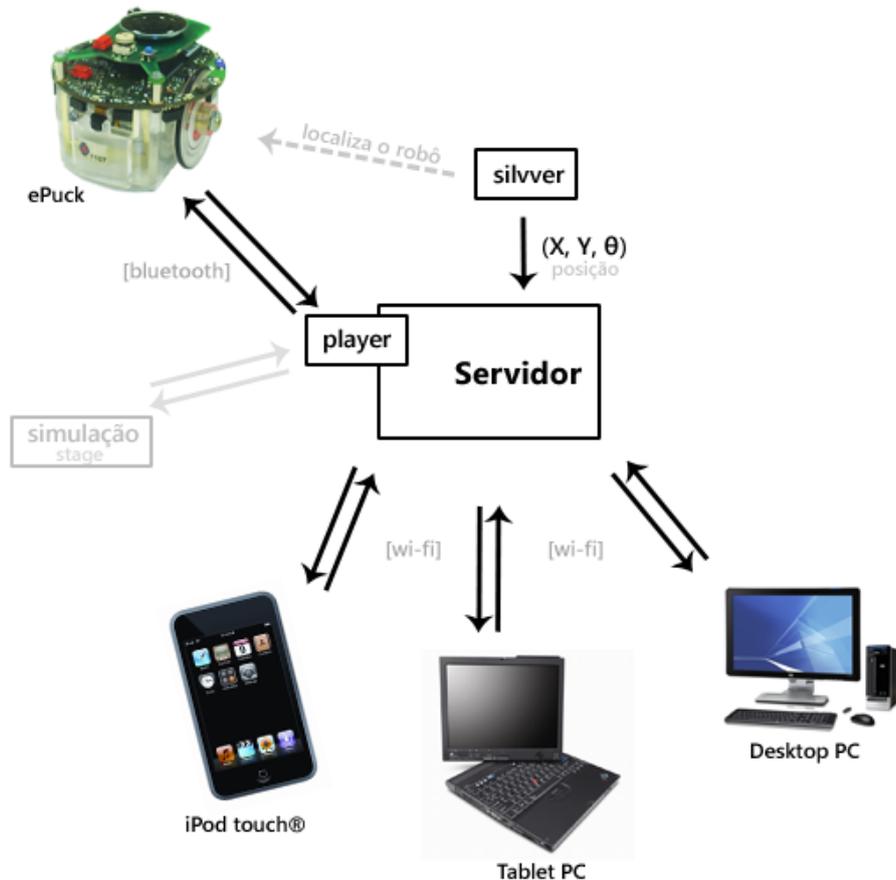


Figura 4.7. Arquitetura do sistema.

O *driver* para o uso do *ePuck* no *player* ainda não contempla todas as funcionalidades do robô. Somente os motores, a odometria, a câmera e os sensores de proximidade podem ser utilizados. A escolha dos signos das interfaces desenvolvidas levaram em conta esta limitação.

4.3.2 Interface para Desktop PCs

A interface para Desktop, que pode ser vista na figura 4.8, foi desenvolvida em ambiente Windows®, utilizando a plataforma *.NET* e a linguagem C#. A interação com esta interface se dá através de mouse e teclado.

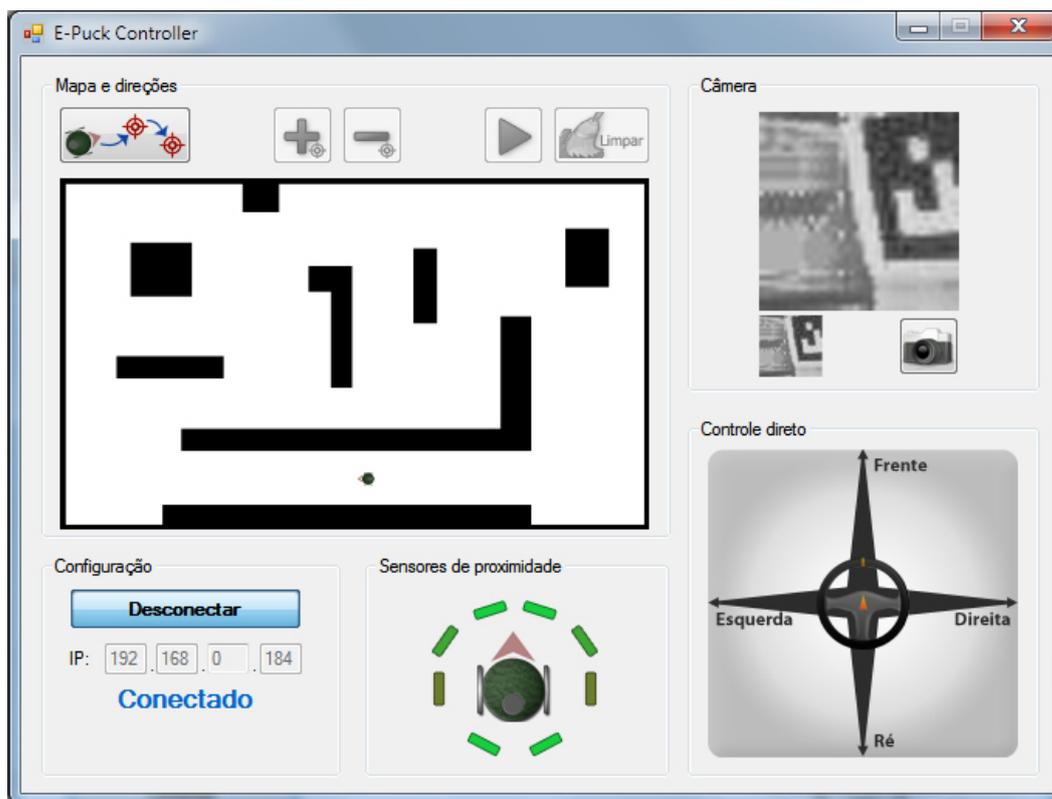


Figura 4.8. Interface para *Desktop*.

Para representar o estado da conexão do sistema com o robô foi escolhido o signo mostrado na figura 4.9. Quando desconectado, as caixas de texto contendo o endereço IP do servidor de controle se tornam editáveis, é mostrado o texto “Desconectado” na cor cinza e o botão apresenta o rótulo “Conectar”. Após conectar ao servidor os campos que contém o endereço IP passam a ser somente-leitura, o texto “Conectado” na cor azul substitui o “Desconectar” e o botão é mantido no estado pressionado com o rótulo “Desconectar”.



Figura 4.9. Signo para representar o estado da conexão com o robô na interface para *Desktop*.

O modo de interação com o robô é representado por um botão localizado acima

do mapa. Quando está no modo de controle direto, o botão apresenta este estado: . Neste caso o botão indica que se o usuário clicá-lo, o modo será mudado para controle por pontos de destino. Se clicado, o botão fica no estado pressionado e passa a ser apresentado desta maneira: . Outra indicação de qual estado o sistema se encontra é a habilitação/desabilitação dos botões para inserir o caminho do robô: . Quando desabilitados, o sistema está no modo de controle direto. Se habilitados, significa que o modo atual é o de pontos de destino.

A posição e orientação do robô mostrada no mapa, exemplificado na figura 4.10. Nele é representado todo o ambiente onde o robô está, mostrando onde o mesmo se encontra e para onde está virado através da colocação do símbolo  sobre o mapa. Este símbolo se movimenta de acordo com a movimentação do o robô e também é rotacionado para representar a orientação. O mapa também indica a velocidade e a direção do robô através do signo dinâmico que é a movimentação do  sobre ele.



Figura 4.10. Signo para representar a posição e a orientação do robô na interface para *Desktop*.

Outro signo para representar a velocidade e a direção do robô é o mostrado nas figuras 4.11 e 4.12. Na esquerda temos o signo no modo de controle direto. Neste modo o usuário irá interagir com a área para definir a velocidade linear e angular do robô. Enquanto o usuário mantiver o botão do mouse clicado, o volante no meio da figura será rotacionado para indicar a velocidade angular e uma barra laranja se movimentará verticalmente para indicar qual a velocidade linear. Ao soltar o botão do mouse o signo volta ao seu estado inicial (Figura 4.11) e o robô para. A imagem da direita representa o signo no modo de pontos de destino, não permitindo que o usuário defina as velocidades do robô. No modo em questão, a barra se movimentará verticalmente e o volante será rotacionado de acordo com as velocidades que o robô reportar durante o seu movimento.



Figura 4.11. Signo para representar a movimentação do robô na interface para *Desktop* (robô parado).

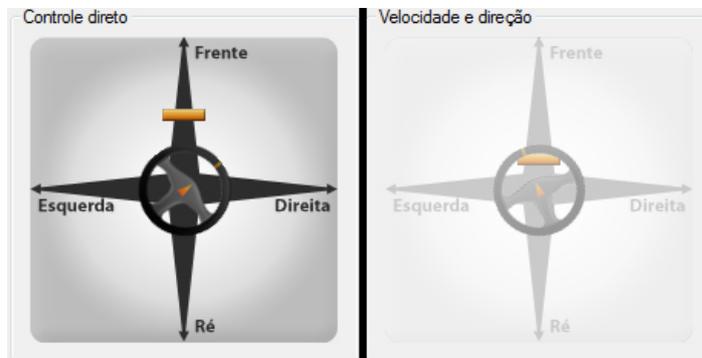


Figura 4.12. Signo para representar a movimentação do robô na interface para *Desktop* (robô em movimento).

O trajeto do robô é representado no mapa por pequenas imagens em forma de alvo (🎯), acompanhado de um número que indica a ordem do ponto no caminho do robô. O trajeto pode ser alterado utilizando os botões de “+” e “-”, que permitem, respectivamente, adicionar e remover pontos ao mesmo. Ao clicar no botão de “play” (▶), o robô começa a se mover e o botão muda de estado (⏹), para indicar que o usuário pode parar a movimentação do robô. O botão limpar (🧹) permite que o usuário remova todos os pontos do trajeto de uma vez. Este conjunto de signos pode ser observado na figura 4.13. Para mostrar que o robô chegou ao destino, os pontos são removidos do mapa e a tela volta para o estado onde é possível inserir novamente pontos de destino.

A proximidade do robô em relação a obstáculos pode ser percebida de duas maneiras: através do mapa, visualizando qual a distância do robô para as paredes, e através dos sensores de proximidade, mostrados na figura 4.14. A figura no meio representa o robô e cada barra ao redor, um sensor de proximidade. Quando não há obstáculos perto de um determinado sensor, a barra terá a cor verde, ficando gradativamente vermelha

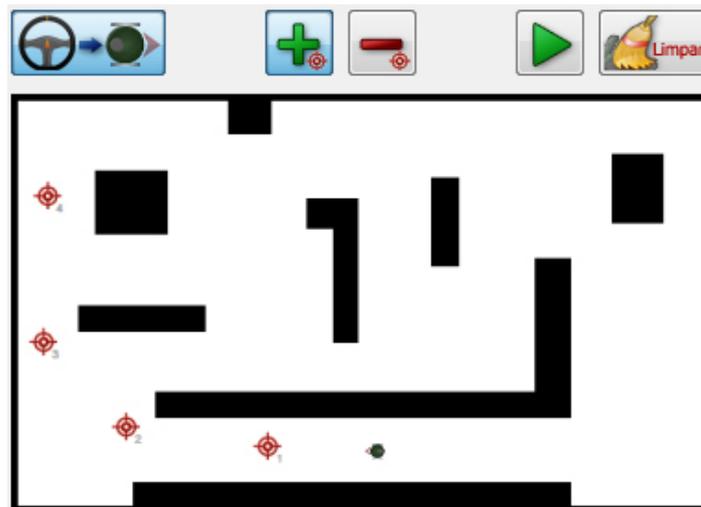


Figura 4.13. Signos para representar o trajeto do robô na interface para *Desktop*.

à medida que o robô se aproxima de um obstáculo.

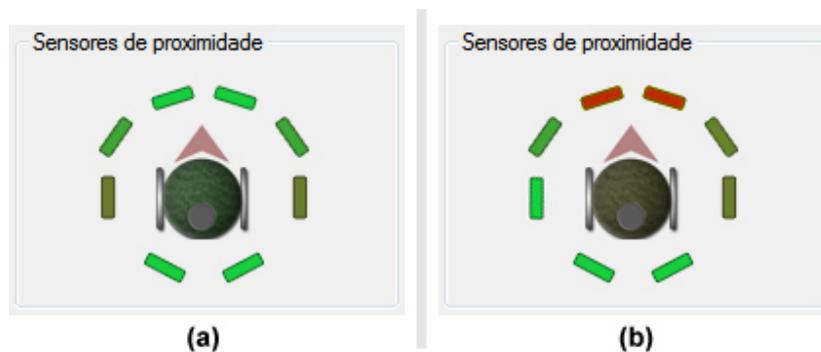


Figura 4.14. Signos que representam os sensores de proximidade na interface para *Desktop*. Em (a) o robô está em uma área livre. Em (b) ele se encontra com a frente encostada em um obstáculo.

Os arredores do robô são mostrados pelo mapa, pelos sensores de proximidade e também pela imagem da câmera, localizada no canto superior direito da interface. Junto com ela aparecem os signos que tratam do que o robô fotografa. Ao clicar no botão  a imagem da câmera no momento do clique é armazenada em disco e exibida do lado esquerdo do botão, como é mostrado na Figura 4.15.

4.3.3 Interface para Tablet PCs

A interface para *Tablet PCs*, mostrada na figura 4.17, foi desenvolvida sob as mesmas condições da interface para *Desktop*, para ambiente Windows e usando C#. As duas



Figura 4.15. Signos que representam a câmera e a imagem fotografada na interface para *Desktop*.

interfaces são visualmente semelhantes, mas a interação se dá de forma diferente, através do uso da caneta diretamente sobre a tela ao invés de mouse e teclado (como na Figura 4.16). Sendo as interfaces bastante semelhantes, somente serão mostrados os signos que são diferentes nesta interface.

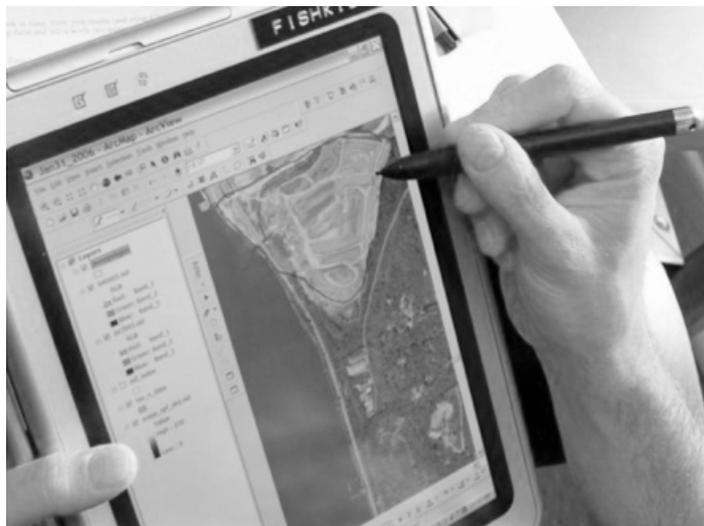


Figura 4.16. Exemplo de uso de um *Tablet PC*.

Com a nova forma de interação, em vez de clicar para adicionar pontos ao caminho do robô o usuário desenhará este caminho com a caneta e os pontos serão gerados a partir da linha desenhada. Um novo signo é apresentado na interface: o botão . Acionando-o o usuário passa a poder desenhar o caminho do robô na tela. A Figura 4.18 mostra a representação desta maneira de se informar o caminho para o robô. O usuário desenha o trajeto, que é então transformado automaticamente

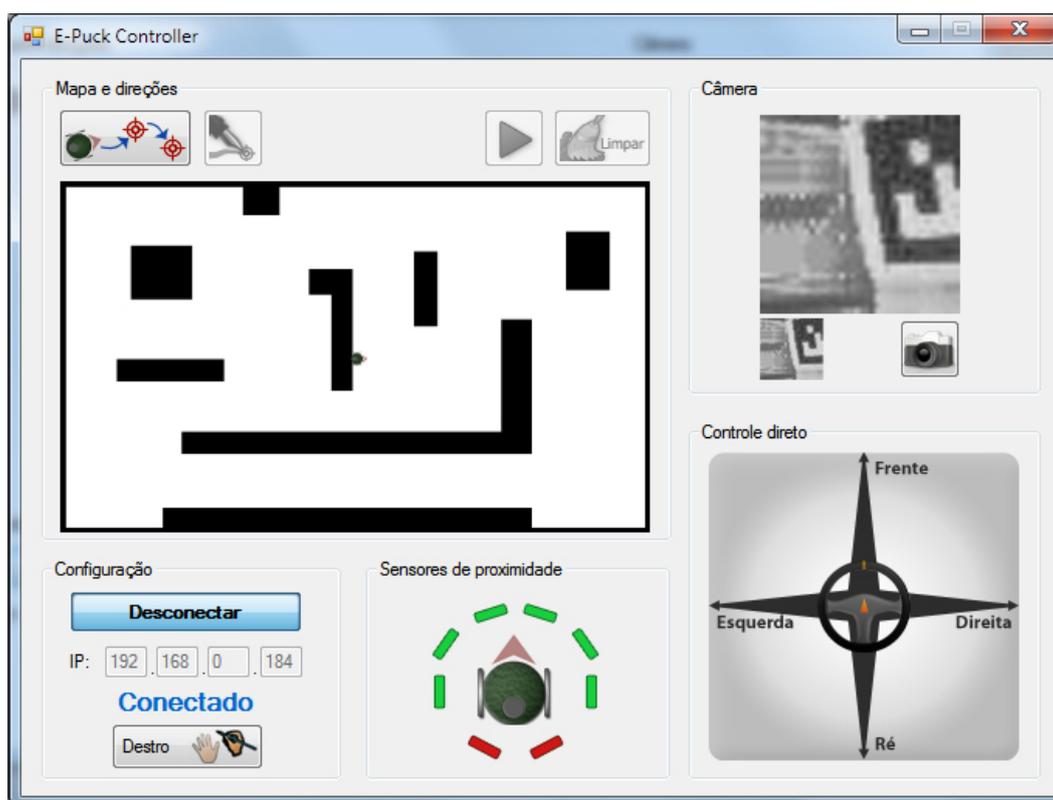


Figura 4.17. Interface para *Tablet PC*.

numa sequência simplificada de pontos, através do algoritmo de redução de pontos de Douglas-Peucker [Douglas & Peucker, 1973]. Estes pontos formam então o caminho que o robô irá percorrer da mesma forma que na interface para desktop.

4.3.4 Interface para iPod touch®

O dispositivo móvel escolhido para o desenvolvimento de uma interface foi o *iPod Touch*®, da Apple®. O *iPod Touch*®, além de um tocador de músicas, é um dispositivo poderoso, possuindo interface *touchscreen* multi-toque, conexão de rede sem fio e acelerômetro interno, além de outros recursos. A interface para o *iPod Touch*® foi desenvolvida em ambiente *Mac OS X*®, da própria Apple®, na linguagem *Objective-C* e utilizando a IDE (*Integrated Development Environment*) fornecido também por ela, o *Xcode* [Apple®, 2010].

O estilo de interação com o *iPod Touch*® possui grandes diferenças em relação às duas interfaces anteriores. Primeiramente, ele é um dispositivo pequeno, possuindo 11cm de comprimento, 6cm de largura e menos de 1cm de espessura (Figura 4.19). Este tamanho reduzido limita a quantidade de elementos que podem ser mostrados na

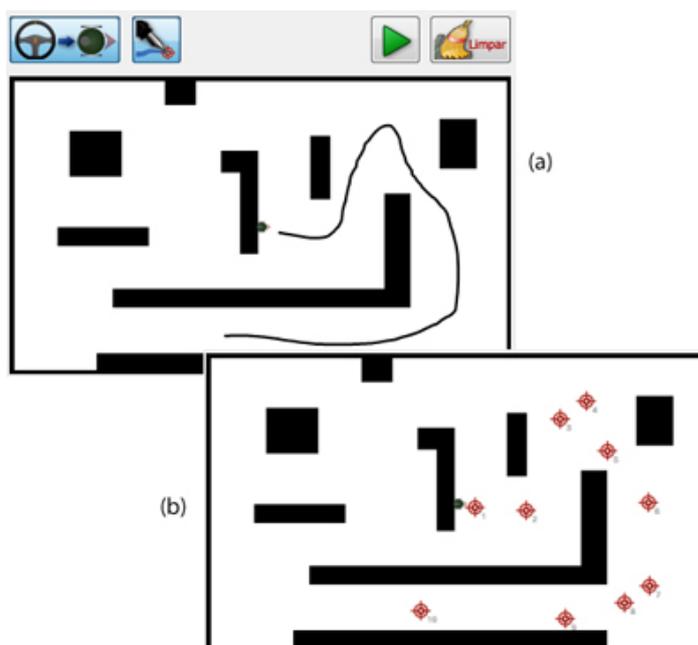


Figura 4.18. Caminho do robô na interface para *Tablet PC*: (a) caminho desenhado pelo usuário; (b) pontos gerados a partir do caminho.



Figura 4.19. *iPod touch*®.

tela ao mesmo tempo, o que resultou na divisão do sistema em três interfaces: uma para conexão com o robô (Figura 4.20), uma para dirigir o robô diretamente (Figura 4.21) e outra para passar um trajeto ao robô (Figura 4.22).

A segunda diferença é a interação com a tela do dispositivo. O *Tablet PC* permite a interação direta com a tela com a utilização da caneta, mas o *iPod Touch*® basicamente só permite a interação por meio do toque com os dedos, que são bem menos precisos e variam de acordo com a pessoa que estiver utilizando o sistema. Existem



Figura 4.20. Interface para *iPod Touch*® na tela de conexão.

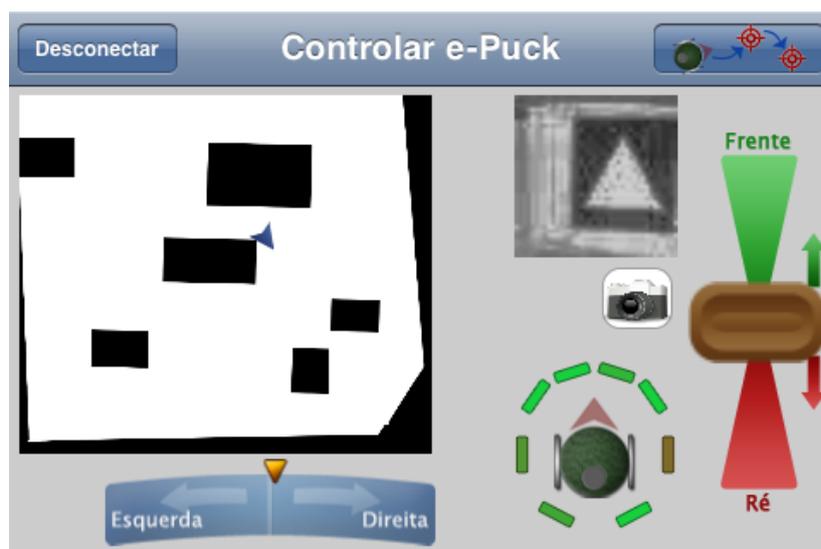


Figura 4.21. Interface para *iPod Touch*® no modo de controle direto do robô.

canetas para o *iPod touch*®, mas não são muito fáceis de se encontrar no Brasil. Dessa forma, a interface deve possuir elementos que compensem a falta de precisão, como botões grandes e tolerância maior à diferença do ponto entre o toque e a área desejada da tela. Algumas representações também devem ser simplificadas na tela pelo nível menor de detalhes que é possível exibir. Esse foi o caso da representação do robô e dos pontos do caminho no mapa. Nesta interface o robô é representado por uma seta azul (▶) e os pontos por círculos verdes (●).

O estado da conexão com o robô é representado pela troca de telas. O sistema

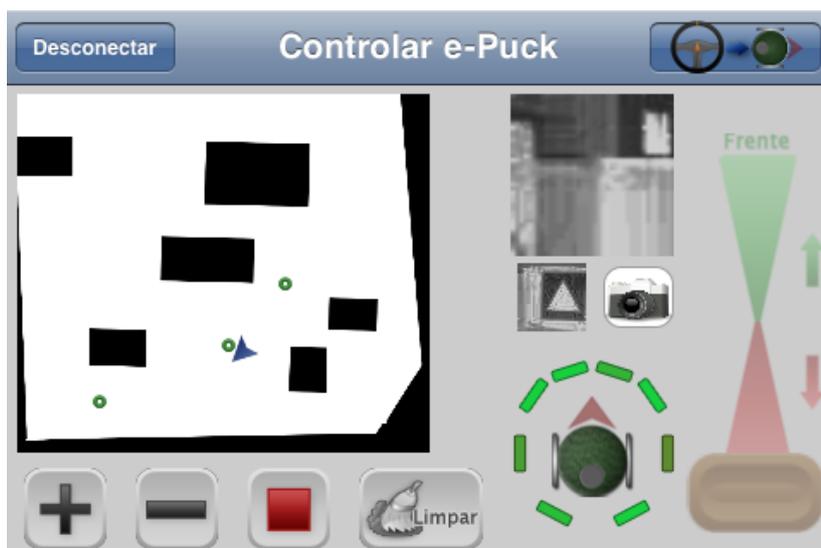


Figura 4.22. Interface para *iPod Touch*® no modo de controle por caminho.

só sai da tela de conexão e vai para a tela de controle do robô quando a conexão é realizada. Se desconectado, através do botão “Desconectar” no topo esquerdo da tela, o sistema volta à tela de conexão.

Por fim, a última diferença é a presença de um acelerômetro no dispositivo. Com isso é possível realizar ações através da mudança de orientação do iPod. Este recurso foi utilizado para dirigir o robô diretamente, sendo que o usuário deve utilizá-lo como se fosse um volante. Ao inclinar o aparelho para a esquerda ou para a direita enquanto acelera, o robô irá virar para o lado da inclinação, como mostrado na figura 4.23. Ao inclinar o iPod, o indicador logo abaixo do mapa aponta para qual lado e o quanto está inclinado, para mostrar o quanto o robô vai virar, assim como indicado na figura 4.24.

No lado direito das figuras 4.21 e 4.22 existe o signo que representa a velocidade linear do robô. Assim como nas interfaces anteriores, no modo de dirigir diretamente o usuário deve interagir com o signo para acelerar o robô. A interação é feita empurrando o botão do meio para cima (para fazer o robô andar para frente) ou para baixo (para fazer o robô andar de ré). Já no modo de controle por caminho o signo corresponderá à velocidade reportada pelo robô.

Para movimentar o robô o usuário deverá então realizar duas ações simultâneas: acelerar o robô através do controle de velocidade na tela e inclinar o iPod para os lados para que ele consiga realizar curvas.

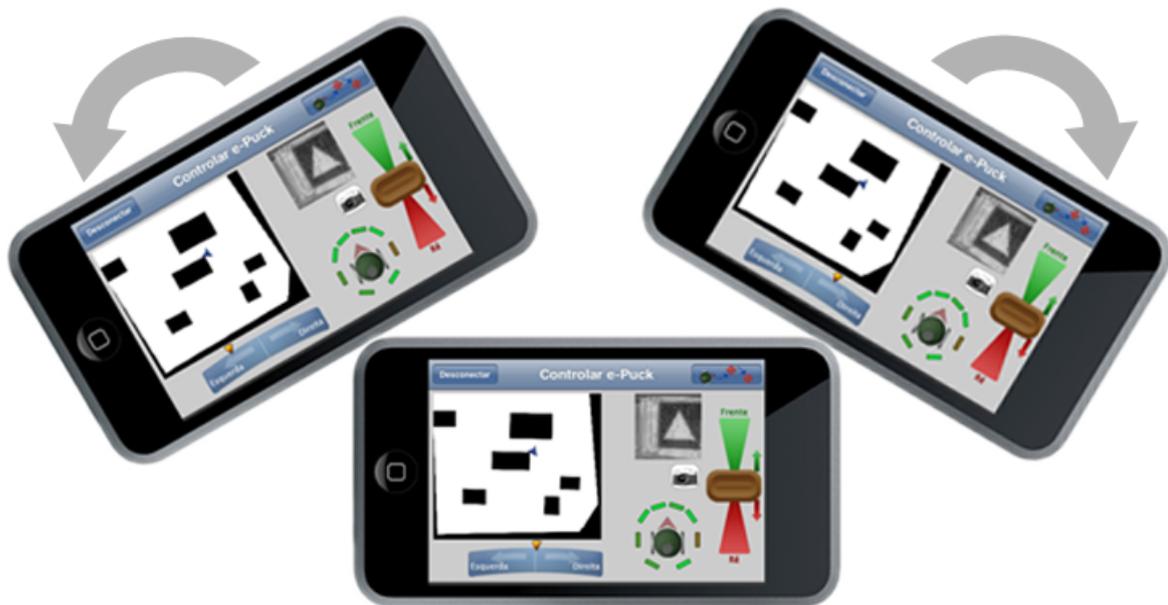


Figura 4.23. Representação do controle do robô por acelerômetro.



Figura 4.24. Exemplo do indicador de inclinação, mostrando que o robô irá virar para a direita.

Capítulo 5

Estudo Comparativo

A avaliação das interfaces criadas foi realizada a partir de testes comparativos com a participação de usuários. Os testes têm como objetivo observar se as classes de signos e as suas representações são adequadas para a teleoperação de robôs e avaliar a interação do usuário com os diferentes tipos de interfaces e formas de interagir com as mesmas, gerando resultados qualitativos. Não era objetivo da avaliação obter validação estatística para os resultados encontrados nem obter dados quantitativos. Dessa forma, não foi conduzida uma Avaliação de Usabilidade propriamente dita.

Apesar do trabalho ser baseado na engenharia semiótica, a avaliação também não foi feita de acordo com o Método de Avaliação de Comunicabilidade (MAC) [de Souza & Leitão, 2009]. Isso ocorreu principalmente devido às dificuldades técnicas para se gravar a interação utilizando o *iPod touch*®. Na época da realização dos testes não existiam softwares de captura de tela para o dispositivo e, para o MAC, se faz necessária a captura em vídeo para a etiquetagem posterior.

Para os testes comparativos foram selecionados nove participantes com experiências variadas em relação a robótica e os diferentes tipos de interação que foram empregados nas interfaces. Estes participantes foram divididos em três grupos para variar a ordem de uso das interfaces e minimizar a influência que o uso anterior de uma interface poderia ter sobre o uso de outra. O primeiro grupo (grupo A) realizou primeiramente a tarefa com a interface para *desktop PCs*, depois a tarefa para a interface para *iPod touch*® e, por fim, a tarefa com a interface para *Tablet PCs*. O segundo grupo (grupo B) realizou as tarefas na ordem *iPod touch*® - *Tablet* - *Desktop* e o terceiro (grupo C) na ordem *Tablet* - *iPod touch*® - *Desktop*. O número de usuários escolhidos foi baseado na curva de problemas encontrados em avaliações de usabilidade em relação ao número de usuários, proposta por Nielsen em [Nielsen, 2000].

Os testes foram realizados no **Laboratório de Visão Computacional e Ro-**

bótica do DCC-UFMG (VeRLab) devido à dificuldade de se transportar toda a estrutura necessária para a utilização do robô para uma sala especializada para a realização de testes com usuários. Foi montada uma pequena arena (mostrada na figura 5.1) sobre uma mesa do laboratório para que os testes pudessem ser realizados. Acima da mesa foram posicionadas três câmeras que, utilizando a biblioteca *silvver* [Garcia & Chaimowicz, 2009], identificam a posição e a orientação do robô através da identificação de um padrão geométrico colocado sobre ele (figura 5.2).

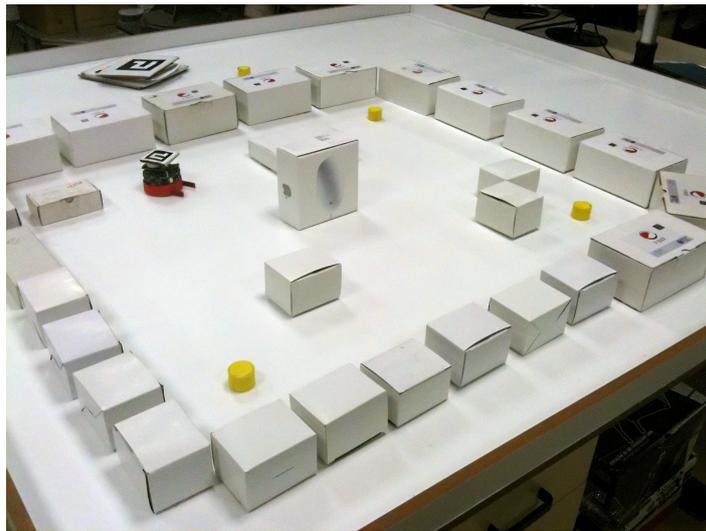


Figura 5.1. Arena de testes.



Figura 5.2. *ePuck* com padrão para localização.

5.1 Preparação

Primeiramente foram gerados o termo de consentimento, que informa ao participante as condições da avaliação e também solicitar a autorização para uso dos dados obtidos na avaliação para a pesquisa, o questionário pré-teste, utilizado para conhecermos o perfil dos participantes, e o script da avaliação, contendo os passos que deveriam ser executados pelo avaliador para que todos os testes fossem realizados com as mesmas condições e nenhuma informação fosse esquecida.

Para a realização do teste foi considerado como cenário uma situação de busca e resgate em um galpão com risco de desmoronamento. A descrição do cenário foi feita da seguinte forma:

“Esta é uma simulação em escala reduzida de uma situação de busca e resgate em um local utilizando robôs. A arena montada sobre a mesa representa um galpão onde houve uma explosão que abalou a estrutura do local, com risco de desabamento. Existem alguns objetos que estão dentro do galpão que devem ser recuperados, mas não se sabe se é seguro para uma pessoa entrar nele. Por isso será enviado um robô para verificar alguns locais críticos do galpão, fotografando-os para verificar se o local é seguro. É desejável que os objetos sejam recuperados também, para que não haja a necessidade de pessoas entrarem no local. Sua tarefa será controlar o robô para fotografar as três áreas de risco do galpão e recuperar os três objetos, trazendo-os de volta à entrada.”

O teste é composto por três tarefas, sendo que cada uma consiste em fotografar um ponto crítico identificado pelo símbolo dado na tarefa e recuperar um objeto de dentro do galpão que possui o mesmo símbolo da área fotografada. Cada tarefa deve ser executada usando uma interface diferente (*desktop*, *Tablet PC* e *iPod touch®*), sendo que o caminho de ida para os locais de interesse (local onde se encontra o objeto ou o local a ser fotografado) deve ser feito informando no mapa os pontos do trajeto que o robô deve percorrer (navegação por *waypoints*) e a volta para o local de início, trazendo o objeto, deve ser realizada controlando o robô diretamente.

Os locais críticos que devem ser fotografados estão marcados cada um com um símbolo diferente: uma cruz (✚), um triângulo (▲) e um círculo (●). Para cada um dos símbolos também existe um objeto correspondente a ser recuperado. A figura 5.3 mostra o robô transportando um destes objetos, enquanto a figura 5.4 mostra um dos locais da arena que devem ser fotografados.

Uma vez gerado o material para os testes com os usuários, foi realizado um teste piloto para verificar a validade do roteiro de testes e tarefas e também prevenir alguns problemas que possam ocorrer durante a execução das tarefas no teste real. Os dados do teste piloto foram descartados da avaliação, mas o teste serviu para reformular o

questionário pré-teste e também identificar problemas com a filmagem (a necessidade de mais fitas para a câmera) e o áudio das entrevistas, pois o equipamento usado para a gravação não era adequado, gerando arquivos de áudio com qualidade baixa. As entrevistas passaram a ser gravadas utilizando um *iPhone*® e o software *SpeakEasy Voice Recorder*®¹.

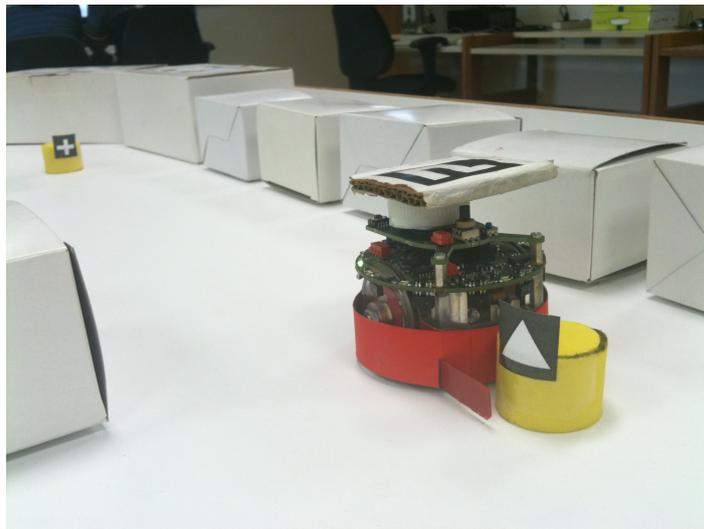


Figura 5.3. *e-Puck* transportando um objeto.



Figura 5.4. Um dos locais a serem fotografados pelo usuário com o *ePuck*.

Em cada tarefa o participante deveria recuperar e fotografar os elementos marcados com uma das figuras. Com a interface desktop deve ser fotografada a área com **+**

¹Disponível para compra em: <http://itunes.apple.com/us/app/speakeasy-voice-recorder/id284652710?mt=8>

e recuperado o objeto com o mesmo símbolo. O mesmo deve ser feito para a interface tablet com os elementos marcados com ▲ e para a interface do iPod, com os elementos marcados com ●. Os locais que devem ser fotografados são sempre os mesmos, enquanto os objetos eram dispostos em locais diferentes a cada teste.

5.1.1 Limitações da avaliação

A avaliação teve algumas limitações, principalmente devido aos recursos oferecidos pelo *e-Puck* e também devido a problemas apresentados por ele. O primeiro deles é a imagem da câmera, que é monocromática e possui uma resolução baixa. A câmera filma em cores e possui resolução VGA (640x480 pixels), mas o robô não possui capacidade de processamento e espaço em memória para processar uma imagem desse tamanho. Além disso, certos tamanhos de imagem tornavam o robô mais lento para responder a comandos podendo inclusive causar travamentos em algumas situações. Por esse motivo a câmera foi configurada, após testes, para gerar uma imagem de 40x40 pixels e sem cores, para que o robô não tivesse sua performance (tempo de resposta aos comandos) prejudicada e ainda fosse possível identificar os símbolos que deveriam ser fotografados.



Figura 5.5. Imagem obtida pela câmera do *e-Puck*.

A segunda limitação é em relação à localização no mapa, que não era muito precisa devido às pequenas variações na identificação da posição do robô, principalmente nas áreas onde mais de uma câmera conseguia ver o padrão para localização e nas bordas das imagens obtidas. Acreditamos que isso poderia ser melhorado utilizando mais câmeras sobre a arena de forma a diminuir o uso das bordas e também poder possibilitar que as câmeras fiquem mais próximas do robô para identificar o padrão com mais facilidade. No entanto, estas opções não foram testadas, pois a calibração das câmeras é uma tarefa custosa.

Por fim, o *e-Puck* às vezes para de responder aos comandos por motivos que ainda não foram descobertos. Este problema gerou alguns atrasos na realização das tarefas

pela necessidade de reiniciar a comunicação com o robô, mas nenhum teste teve que ser descartado por causa disso. As tarefas podiam ser retomadas do exato ponto onde foram interrompidas devido ao travamento do robô.

5.2 Aplicação dos testes

Os testes foram realizados no período de 03 a 11 de agosto de 2010 ocorrendo separadamente para cada participante. As avaliações foram conduzidas da seguinte maneira (o material utilizado na avaliação está disponível no Anexo A):

- Recepção do participante e explicação sobre o teste a ser realizado, apresentando o laboratório, o robô, a arena de testes, as interfaces e uma breve explicação sobre as tarefas.
- Entrega e assinatura do termo de consentimento, que tem como objetivo informar o participante sobre as condições da avaliação e solicitar a autorização para uso dos dados obtidos na avaliação para a pesquisa.
- Preenchimento do questionário pré-teste, para identificar os perfis dos usuários e suas experiências anteriores com as tecnologias utilizadas no teste.
- Entrega e explicação do cenário e das tarefas que devem ser executadas.
- Execução do teste, filmando a interação e fazendo anotações, se necessário. Ao final do teste com cada interface, uma pequena entrevista foi conduzida para avaliar a experiência do usuário com a mesma.
- Realização de uma entrevista pós-teste para avaliar comparativamente a experiência do usuário com as três interfaces e levantamento da opinião do usuário em relação a elas.

Dos nove usuários selecionados, três eram alunos de graduação ou recém graduados, quatro eram mestrandos e dois já tinham o mestrado completo, todos dentro da área de computação ou áreas relacionadas (Ciência da Computação, Sistemas de Informação e Ciência da Informação). Somente três deles tinham algum conhecimento sobre robótica, sendo que um desses já havia lidado superficialmente com robôs em um curso técnico. A idade dos participantes variou entre 19 e 34 anos, sendo duas mulheres e sete homens.

Em relação ao uso de *Tablet PCs*, quatro usuários nunca tinham usado um, quatro já o tinham usado pelo menos uma vez e um deles já usou o tablet para tarefas mais

complexas, como desenho ou escrita. Já para o *iPod touch*®², dois usuários já possuíam um e tinham experiência com o uso, um outro usuário não possuía o dispositivo, mas já tinha usado e conhecia os recursos de detecção de movimento utilizando o acelerômetro, dois outros já tinha usado o dispositivo pelo menos uma vez antes e os quatro restantes não tinham tido contato anterior com o *iPod*®.

Para a realização das tarefas os participantes ficavam localizados sempre em um mesmo local, para que a orientação do mapa na interface fosse invertida (i.e., ficasse de cabeça para baixo) em relação à disposição da arena e dos objetos na mesa. Isso foi feito com a intenção de verificar se os usuários teriam muita dificuldade de orientação ao executar as tarefas e movimentar o robô e para observar se os participantes conseguiam executar as tarefas normalmente com a orientação invertida ou se seria necessário que a orientação fosse sempre direta em relação aos mesmos. Os participantes poderiam se levantar da cadeira em que estavam, mas não poderiam sair do lugar onde estavam para melhorar o campo de visão ou alterar a orientação dos mesmos. Os resultados obtidos foram analisados em conjunto com as anotações feitas durante a execução dos testes e serão apresentados na próxima seção.

5.3 Resultados da avaliação

Como dito anteriormente, os objetivos dos testes eram “observar se as classes de signos e as suas representações são adequadas para a teleoperação de robôs e avaliar a interação do usuário com os diferentes tipos de interfaces e formas de interagir com as mesmas”. Desta forma, não estava no escopo dos testes avaliar as interfaces para obter indicações para o redesenho das mesmas, apesar de tais indicações terem sido obtidas. Primeiramente vamos mostrar os resultados obtidos para cada interface e no final apresentaremos os resultados consolidados.

5.3.1 iPod touch®

O primeiro ponto observado foi que a experiência anterior com o *iPod touch*® não teve influência significativa na forma com que os usuários interagiram com a interface para esse dispositivo para guiar o robô diretamente. Dois dos participantes que já tinham experiência tiveram dificuldades em relação ao controle do robô utilizando a inclinação do *iPod*®, enquanto outros dois que não tinham experiência conseguiram guiar o robô sem dificuldades. Um destes conseguiu guiar o robô de forma contínua, mantendo a

²O questionário considerou a experiência tanto o *iPod touch*® quanto o *iPhone*®, pois não há diferenças entre os dois em relação à forma de interação requerida para os testes.

aceleração e virando o *iPod*®), com facilidade e por quase todo o trajeto entre um objeto a ser recuperado e o local para onde o robô deve retornar. Os outros participantes não conseguiam dirigir o robô deste modo, e o controlaram através de movimentos discretos (e.g., acelera sem virar, para o robô, acelera virando, para o robô, acelera sem virar, etc).

A maioria dos participantes justificou a movimentação discreta pelo medo de perder o controle do robô e bater ou enganchar em algum obstáculo. Nas entrevistas, três motivos foram apresentados pelos usuários para esse problema: a sensibilidade do movimento com o *iPod*®, que não dava muita segurança mesmo com o indicador de direita/esquerda, e a dificuldade de se executar duas tarefas ao mesmo tempo (acelerar com o dedo indicador e guiar o robô através da inclinação do dispositivo). Às vezes também o participante tentava mover em linha reta, mas esquecia o *iPod*® inclinado, resultando em um caminho em curva. O terceiro motivo foi a dificuldade de orientação que alguns tiveram ao guiar o robô olhando para a mesa ou para o mapa. Este problema, como veremos posteriormente, foi comum às três interfaces.

Em relação à marcação de pontos, a experiência anterior com o dispositivo fez alguma diferença, pois como o *iPod*® é controlado por toques na tela com os dedos (não há caneta como o tablet), a precisão é prejudicada de acordo com a área de contato do dedo com a tela. As pessoas que tinham dedos mais grossos e não tinham experiência tiveram alguns problemas ao marcar pontos, pois os mesmos não apareciam exatamente no mesmo local onde esperavam. Isso acontecia porque esses participantes tocavam a tela de forma mais pesada, com uma área de contato maior. Dois deles fizeram uso da remoção de pontos para conseguir ajustar o caminho do jeito que queriam. Os que tinham experiência tocavam levemente na tela para minimizar a área de contato e aumentar a precisão na definição dos pontos e não tiveram problemas ao marcar o caminho.

Todos conseguiram levar o robô aos pontos de interesse através da marcação de pontos para definir o trajeto do robô. Isso indica que conseguiram lidar com os problemas apresentados acima rapidamente e entender como realizar a ação de passar o caminho para o robô. Todos também conseguiram dirigir o robô de volta ao início carregando o objeto, mas alguns tiveram mais problemas por causa da dificuldade de orientação enquanto o *e-Puck* se movia. Um dos usuários disse ter conseguido se orientar somente pelo mapa na hora de movimentar o robô, tanto diretamente quanto passando o caminho.

Um dos participantes citou a vantagem de uso do *iPod*® pelo tamanho e mobilidade do dispositivo. Isso facilitaria a portabilidade do robô e de sua interface de controle para os locais onde fosse necessário seu uso. O mesmo participante disse que

gostou do fato de no *iPod*® os controles ficarem mais próximos uns dos outros, agilizando seu uso e facilitando a observação de mais de um signo ao mesmo tempo, pois mais elementos ficariam simultaneamente dentro do campo de visão do usuário.

Dessa forma, apesar das dificuldades apontadas, todos os participantes conseguiram compreender o que a interface comunicava e realizaram as tarefas com sucesso.

5.3.2 Tablet PC

Poucos participantes tinham experiência com *Tablet PCs*, mas o que se notou pelo teste é que a falta de experiência anterior não influenciou na execução das tarefas. Salvo um participante que não conseguiu identificar o símbolo procurado na câmera, todos os participantes conseguiram realizar as tarefas com sucesso e com mais facilidade do que com o *iPod touch*. Todos os usuários citaram a facilidade de uso da caneta pela precisão do toque da mesma na tela e pela forma intuitiva de interação, como se fosse uma caneta sobre um caderno. A maioria dos participantes (seis pessoas) preferiu a utilização do tablet para realizar as tarefas. Outro detalhe a ser observado em relação ao uso da caneta é o mapeamento um-para-um do movimento dela com o movimento do cursor na tela, i.e., o deslocamento feita com a caneta é o mesmo que será feito pelo cursor na tela, e na mesma escala, permitindo que o usuário tenha mais precisão que com o mouse.

Somente um dos participantes teve problemas em traçar o caminho do *e-Puck* no mapa com a caneta, pois eram gerados mais pontos que ele esperava depois de desenhar o trajeto. Isso ocorreu porque o participante não percebeu que ao iniciar um novo traçado, os pontos anteriores não são removidos a não ser que o robô já tenha terminado a movimentação ou o botão “Limpar” seja acionado. O participante encostou na tela sem querer, em cima do signo que representa o robô e gerou alguns pontos. Estes foram ocultados pelo desenho do *e-Puck* no mapa, fazendo com que não fossem percebidos.

Alguns participantes tiveram uma dificuldade inicial ao guiar o robô diretamente. O desenho do volante no meio foi entendido pela grande maioria como o único local onde clicar para virar o robô, tendo que clicar e arrastar dentro da área do volante para isso. Poucos perceberam que o volante era somente a indicação do quanto o robô iria virar de acordo com a área clicada dentro do signo indicado pela figura 4.11. O resultado disso é que os participantes só conseguiam girar o *e-Puck* a uma velocidade baixa em relação à máxima que o mesmo pode girar, pois o ponto acionado com a caneta ficava perto do centro. Os que conseguiram perceber a intenção do signo de controle direto (dois que iniciaram com o *Tablet PC* e todos que utilizaram a interface

para desktop antes) guiaram o robô com muita facilidade pela arena. Esta diferença na quantidade de participantes que perceberam a função do signo de movimentação tendo começado ou não com a interface para *Tablet PC* é um indício que a ordem de uso das interfaces altera a experiência do usuário.

5.3.3 Desktop PC

A interface para *desktop PCs* não apresentou dificuldades para os que usaram a interface para *Tablet PCs* antes, devido ao fato de as duas serem bem parecidas. Esta foi a interface que apresentou menos problemas iniciais de interação nos testes. Todos os usuários conseguiram adicionar os pontos corretamente, mesmo que em uma das vezes tenha sido necessário mandar o *e-Puck* parar e refazer o caminho, pois o mesmo ficou preso em um dos cantos da arena e não conseguiu sair. Este problema foi causado por que o suporte das câmeras usadas para localização tinham saído da posição original, fazendo com que o robô fosse mostrado um pouco deslocado da posição onde realmente estava. Os mesmos problemas em relação ao controle direto do robô encontrados na interface para *Tablet PCs* também foram encontrados aqui.

Dois usuários preferiram esta interface em relação às outras. Um desses participantes é o que teve problemas com o traçado do caminho na interface do *Tablet PC*. No entanto, durante a entrevista final ele disse que se tivesse percebido a razão da geração de pontos indevidos teria optado por ela ao invés da interface para *desktop*. O segundo participante preferiu esta interface por achar mais preciso gerar o caminho com cliques do mouse que com a caneta ou com o *iPod*®.

Os que preferiram a interface do *Tablet PC* a essa o fizeram principalmente pela diferença na precisão entre a caneta e o mouse. Com o mouse não temos o mapeamento um-para-um da caneta e o deslocamento do cursor na tela quando usado o mouse é maior que o deslocamento do próprio mouse.

5.3.4 Apreciação geral

Na avaliação geral, seis participantes preferiram a interface para *Tablet PCs*, dois preferiram a interface para *desktops* e somente um preferiu a interface para o *iPod touch*®. Como mostrado, a interface para *Tablet PCs* foi a preferida pelos participantes por causa da interação utilizando caneta. Eles sentiram que tinham mais controle sobre o robô e que tinham mais precisão tanto para desenhar o caminho quanto para guiar o *e-Puck* diretamente. O mapeamento um-para-um da caneta e a sua forma de uso tornou a interação mais intuitiva para os participantes.

O *iPod touch*® foi o que os usuários menos gostaram, principalmente por causa da movimentação direta do robô. Muitos participantes tiveram dificuldades com a orientação pelo mapa e com a sensibilidade do movimento para guiar o robô, além da dificuldade em acelerar e virar ao mesmo tempo. Um dos principais detalhes notados na interação com esta interface foi o fato dos usuários muitas vezes esquecerem o *iPod*® inclinado ao acelerar, gerando uma trajetória curva, diferente da esperada em linha reta.

Uma dificuldade comum às três interfaces foi a orientação pelo mapa e pela mesa, como citado anteriormente. Os usuários demoraram um pouco para se acostumarem com a diferença na representação, pois o mapa estava girado em 180° em relação à arena. Assim os participantes giravam o *e-Puck* para um lado esperando que ele girasse para o outro. Isso fez com que quase todos os participantes executassem as tarefas olhando para a mesa e não para a interface para se orientar. Mesmo assim, um dos usuários tentou se orientar somente pelo mapa e reportou ter conseguido sem problemas, apesar de ter sido menos confiável que olhar para a arena porque a posição do robô não era informada com muita precisão pelas câmeras, fazendo com que a sua representação no mapa ficasse um pouco instável, variando a posição mesmo quando o *e-Puck* estava parado.

Cinco participantes citaram que não utilizaram o mapa com mais frequência porque não havia indicações de onde os objetos se localizavam no mesmo. Isso os forçava a olhar para a mesa para aproximar o robô corretamente do objeto e recuperá-lo, pois também era muito difícil fazê-lo utilizando a câmera, devido à qualidade da imagem obtida.

Outra dificuldade foi em relação à câmera do *e-Puck*. Por ser bastante limitada, com imagem em baixa resolução, em preto e branco, com ângulo de visão reduzido e com taxa de atualização (número de quadros por segundo) baixa, os participantes não conseguiram usá-la para se orientar enquanto movimentavam o robô. A única situação em que fizeram uso dela foi para fotografar os símbolos, que foram criados com bastante contraste para que pudessem ser identificados na câmera. Os participantes reportaram que acreditavam que teriam conseguido se orientar somente através da interface se a câmera fosse melhor, pois conseguiriam reduzir o problema da confusão com a orientação do robô se usassem a câmera em conjunto com o mapa.

O signo dos sensores de proximidade só foi utilizado por um dos participantes e em somente em uma das tarefas, a do desktop, ao trazer o robô de volta. O participante não conseguia ver o robô atrás de uma das caixas e não quis levantar da cadeira para isso. Então observou os sensores para saber se o *e-Puck* estava muito próximo das paredes e se ainda estava carregando o objeto. Em todos os outros casos os participantes disseram

que a observação dos sensores não foi necessária para guiar o robô, pois era possível vê-lo na mesa com facilidade e saber com mais confiabilidade se o robô estava preso em alguma parede e se ainda estava carregando o objeto. Mesmo assim somente três dos participantes disseram que achavam que os sensores seriam necessários se não pudessem ver o *e-Puck*, pois os outros achavam que com o mapa e a câmera (se a mesma fosse melhor) já teriam informações suficientes para saber o que acontece com o robô.

Apesar das limitações do teste devido aos recursos utilizados, foi observado que o que foi representado nas interfaces foi suficiente para que os participantes pudessem realizar as tarefas, mesmo com algumas dificuldades. A falta da indicação da localização dos objetos no mapa dificultou, mas não impediu a execução das tarefas pelo fato de que os usuários podiam olhar para a mesa. Isso é um indício de que as classes de signos foram suficientes para representar a interação com o robô.

A ordem de uso das interfaces teve influência no desempenho dos participantes com cada uma delas, sendo que a primeira interface utilizada sempre apresentou a dificuldade inicial de se adaptar ao ambiente, conhecer o funcionamento do robô e a execução das tarefas e se acostumar com a orientação do *e-Puck* no mapa e na mesa em relação ao participante. Para as interfaces utilizadas em seguida, notou-se que a forma de execução das tarefas já estava mais clara para os participantes. Então podemos concluir que as dificuldades encontradas a partir da segunda interface foram relacionadas ao próprio estilo de interação do dispositivo usado. Na Tabela 5.1 é apresentado o resumo do que foi observado no teste para cada interface utilizada, enquanto a Tabela 5.2 contém observações gerais dos testes com cada participante.

Por fim, alguns aspectos para o redesenho das interfaces foram identificados, tanto através da observação da interação dos participantes com os sistemas quanto a partir de sugestões dos próprios usuários. O primeiro aspecto diz respeito à dificuldade de orientação com o mapa. Um dos participantes sugeriu, enquanto interagia com a interface *desktop*, que o mapa pudesse girar para acompanhar a orientação do robô. Em outras palavras que o mapa fosse apresentado sempre com a direção em que o robô seguia voltada para cima. Isso facilitaria o entendimento da localização do *e-Puck* no mapa e também a tarefa de movimentá-lo, pois agora a representação do robô no mapa seguiria sempre o mesmo referencial de orientação do usuário. Esse item poderia ser aplicado às três interfaces para gerar melhores resultados.

Outro aspecto, observado durante a interação dos usuários com as interfaces para *Tablet PC* e *desktop*, é sobre o signo de controle direto do robô. A metáfora do volante utilizada no signo causou confusão na maioria dos usuários, podendo ser substituída por um outro elemento que indicasse o quanto o robô está virando. Faltaram também indicações mais precisas para mostrar ao usuário que é possível movimentar o robô e

Tabela 5.1. Resumo dos testes.

Grupo / Partici-pante	Desktop	Tablet	iPod
A01	Gostou do controle direto.	Melhor tempo de resposta e mais fácil para dar caminho ao robô	Mais difícil de usar, mas mais prático. Turn-in-place mais difícil de fazer.
A02	Controle direto mais fácil.	Passar pontos para o robô foi mais simples. Achou a caneta mais fácil de usar que o mouse.	Toque na tela menos preciso.
A03	Passou o caminho para o robô com um ponto de cada vez. Conseguiu guiar o robô sem dificuldades.	Achou o desenho do caminho mais preciso e fácil de fazer.	Conseguiu guiar o robô diretamente sem dificuldades e de forma contínua.
B01	Menor controle de movimento usando o mouse em relação à caneta.	Maior facilidade e precisão no uso do mapa para desenhar o caminho em relação às outras.	Dificuldade de acostumar com as ações distintas (acelerar e virar).
B02	Possibilidade de marcar pontos mais próximos.	Dirigibilidade do robô foi mais simples. Turn-in-place mais fácil de fazer.	Orientação gerou confusão para guiar o robô diretamente. Achou a interface mais difícil de usar. Esquecia o dispositivo inclinado ao acelerar.
B03	Conseguiu guiar o robô e marcar pontos com facilidade por já ter usado a interface do tablet.	Desenho do trajeto mais fácil que marcar pontos. Sentiu mais segurança com a caneta.	Dificuldade com a precisão do toque na tela. Não conseguia manter o robô sempre em linha reta por causa da sensibilidade do acelerômetro.
C01	Mais fácil de controlar o robô diretamente.	Mais fácil para guiar o robô dando o caminho.	Tinha experiência com iPod, mas teve dificuldades para guiar o robô diretamente por causa da sensibilidade do acelerômetro.
C02	Sentiu maior precisão ao marcar pontos com mouse.	Achou que o caminho marcou mais pontos que deveria. Foi mais fácil dirigir diretamente.	Gostou da interface pela proximidade dos comandos. Usou mais o mapa pelo mesmo motivo.
C03	Mais simples de usar por ser uma forma de interação já conhecida.	Conseguiu desenhar o caminho com facilidade.	Teve dificuldade para marcar e excluir pontos por causa da precisão do toque. Achou mais fácil dirigir usando o ipod.

Tabela 5.2. Observações gerais nos testes.

Grupo / Participante	Observações gerais
A01	Confuso com orientação, não conseguia guiar o robô só com mapa quando ele estava no campo de visão. Teve dificuldade com orientação inicial. Sentiu falta da indicação da posição dos objetos no mapa.
A02	Dificuldade com orientação invertida. Imagem da câmera atrapalhou (mas não impediu) a identificação de objetos. Conseguiu usar informação do sensor de proximidade para perceber que o robô estava preso. Realizava ações discretas: andar em linha reta ou virar.
A03	Conseguiu usar o mapa para movimentar, apesar de ter mais noção de espaço olhando para a mesa.
B01	Sentiu falta da identificação dos objetos no mapa. Teve dificuldade para mover o robô quando o mesmo se encontrava preso nas paredes da arena.
B02	Olhou para o mapa somente para marcar pontos. Não sentiu diferença entre marcar pontos no desktop ou no tablet.
B03	Gostaria que a orientação do mapa pudesse acompanhar a orientação do robô. Dirigiu sempre olhando a mesa.
C01	Dificuldade com orientação. Sentiu falta de um aviso claro (som ou alerta na interface) para indicar que o robô chegou ao destino.
C02	A qualidade da imagem da câmera impediu a identificação de um dos locais a fotografar. Não conseguiu guiar o robô diretamente da maneira correta. Teve dificuldades para perceber que o robô estava preso e retirá-lo da situação. A imprecisão do posicionamento no mapa dificultou o seu uso.
C03	Não sentiu necessidade de usar o sensor de proximidade. Usaria em conjunto com o mapa se não enxergasse o robô.

girá-lo ao mesmo tempo. Uma sugestão dada por um dos participantes foi a inclusão de círculos com gradações de cores diferentes para indicar que é possível clicar fora dos eixos para executar a movimentação e a troca do volante por um botão que pudesse ser arrastado como na interface para o *iPod*®. Um esboço para uma possível representação deste signo é mostrado na figura 5.6.

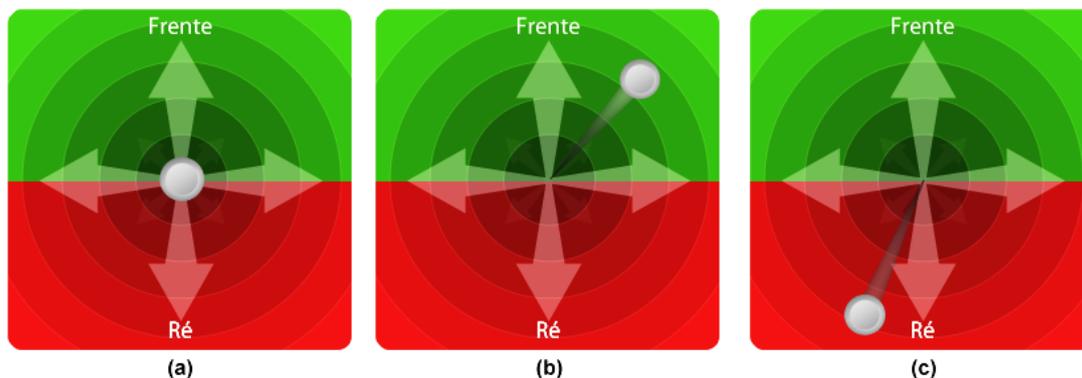


Figura 5.6. Esboço da nova representação para o signo de movimentação nas interfaces para *desktop* e *tablet*. Em (a) temos o robô parado, em (b) temos o robô acelerando para frente e virando para a direita e, em (c) temos o robô acelerando para trás e virando um pouco para a esquerda.

Na interface para o *iPod*® foi observado o problema da precisão do toque no mapa para inserir os pontos. Esse problema é agravado pelo tamanho reduzido da tela e o tamanho do mapa na mesma. Uma das opções sugeridas por um participante para solucionar o problema é colocar o mapa ocupando toda a tela e esconder os comandos, deixando à mostra somente um botão para voltar a mostrá-los. Outra opção é adicionar a funcionalidade de *zoom* ao mapa, permitindo que o usuário aproxime a região onde quer adicionar o ponto e marcar com maior precisão, aproveitando o conceito de *pinch-to-zoom* que é bastante utilizado em aplicativos para *iPod touch*® e *iPhone*®. O *pinch-to-zoom* consiste em tocar com dois dedos na tela e movimentá-los em direções opostas, afastando um dedo do outro (Figura 5.7-a) para aumentar o *zoom* e juntando-os (Figura 5.7-b) para diminuir a aproximação.

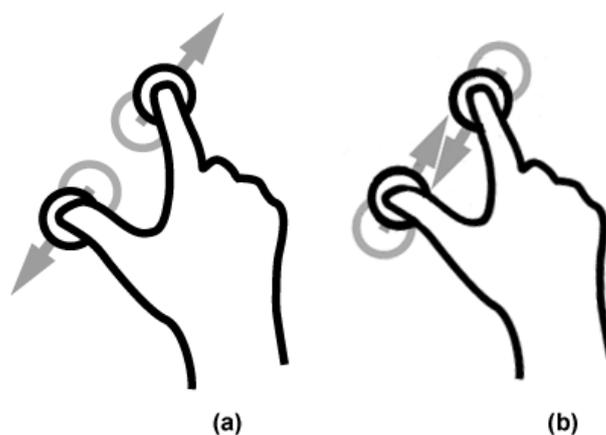


Figura 5.7. Representação do recurso *pinch-to-zoom*. Em (a) é representado o movimento para aumentar a aproximação e em (b) o movimento para diminuí-la.

Com a utilização da caneta na interface para o *tablet* e dos dedos na interface do *iPod touch*® foi notada uma limitação destas duas formas de interação em relação à exibição de textos auxiliares, como *tooltips*. No *tablet* é difícil para o usuário manter o cursor parado sobre um elemento da interface para a exibição de um *tooltip*. Já no *iPod touch*® não existe o conceito de manter o cursor sobre um objeto, pois tocar a tela tem funcionamento análogo ao clique do mouse. Isso trouxe dificuldades aos usuários quando havia alguma dúvida sobre o funcionamento de algum comando. Uma estratégia que pode ser adequada para contornar este problema é ter um comando de “Ajuda”, pelo qual todos os comandos da tela exibiriam o texto auxiliar quando acionados ao invés de executar a ação prevista. Saindo do modo de ajuda o usuário voltaria a executar os comandos normalmente. Este conceito já existe em algumas aplicações para *desktop*, sendo que ao clicar na ajuda e depois em um comando é

exibido o texto de ajuda, mais completo que o texto de um *tooltip*, relacionado àquele item.

De acordo com os resultados obtidos, não foram percebidos problemas em relação às classes de signos. As dificuldades encontradas foram associadas principalmente à expressão dos signos presentes nas classes, como no caso do volante e do mapa, que provocaram dúvidas iniciais nos participantes, ou do estilo de interação, como no caso do *iPod touch*®[®], com o qual os usuários tiveram mais problemas para guiar o robô que nas outras interfaces.

Capítulo 6

Conclusões e Trabalhos Futuros

O trabalho investigou a identificação de classes de signos para interfaces de controle de robôs e a utilização destas classes para desenvolver diferentes interfaces para o controle de um robô *e-Puck*. Isso foi realizado através da aplicação científica do método de inspeção semiótica em interfaces já existentes de forma a obter classes de signos que representem esta interação e do estudo de classificações e métricas já existentes na literatura. As classes obtidas serviram de base para a criação de um modelo de interação que deu origem a três interfaces com estilos diferentes de interação: *desktop*, *Tablet PC* e *iPod touch*®. Com as três interfaces desenvolvidas, foi conduzida uma avaliação destas com usuários. A avaliação permitiu observar as classes de signos e suas diferentes formas de representação para cada uma das interfaces. Também foram identificados problemas com a representação de alguns signos e sugestões para solucioná-los através da avaliação.

A identificação das classes de signos através da aplicação do MIS permitiu observar que o método pode ser aplicado no contexto da interação humano-robô sem sofrer alterações e ainda produzir resultados relevantes. Isso reforça a hipótese de que o MIS pode ser aplicado independentemente da tecnologia utilizada, permitindo o seu uso em diferentes contextos de interação, contribuindo para a consolidação do método. Pelo fato de o método focar na comunicação através da interface e não nos elementos presentes na mesma, acreditamos que o método possa ser aplicado a outros tipos de robôs além do que é tratado neste trabalho.

O modelo de interação criado utilizando a MoLIC serviu de base para três diferentes interfaces sem sofrer alterações. Esse fato é um indício de que a linguagem permite abstrair a tecnologia que será utilizada para implementar o modelo, mantendo o foco somente na comunicação através da interface do sistema.

A independência de tecnologia do MIS e o poder de abstração da MoLIC ob-

servados representam contribuições para a engenharia semiótica, mostrando benefícios obtidos pela aplicação dos seus métodos e do uso da teoria. O presente trabalho pode ser considerado um ponto de início para a aplicação da engenharia semiótica e dos seus métodos na interação humano-robô. Dessa forma, temos uma contribuição relevante para a área de IHC como um todo.

A avaliação das interfaces levantou questões referentes aos três tipos de interfaces e suas particularidades. Pudemos observar quais as principais dificuldades dos usuários com cada forma de interação e quais aspectos destas traziam melhoras para a experiência do usuário. Estes resultados são relevantes principalmente para as interfaces para *Tablet PCs* e para o *iPod touch*® , que contemplam maneiras de interagir diferentes do usual e que possuem potencial para serem exploradas mais a fundo.

Os resultados deste trabalho podem ser utilizados por desenvolvedores de sistemas para interagir com robôs. Foram identificados os aspectos comunicados pelas interfaces e pelo próprio robô e como estes signos se relacionam. Assim, desenvolvedores podem guiar seu trabalho para criar interfaces para robôs com maior comunicabilidade e também com a utilização de diferentes dispositivos para proporcionar novas formas de interação.

O desenvolvimento das interfaces de controle também foi útil para propor uma arquitetura para testes com robôs *e-Puck* dentro do laboratório, que permitiu integrar diferentes tecnologias de forma transparente para o controle do robô, abstraindo detalhes de comunicação, comandos de níveis baixos e a utilização conjunta do sistema de localização por câmeras. Considerando as limitações da avaliação com os usuários, a execução desta permitiu que a validade da arquitetura fosse observada. Trabalhos futuros podem utilizar a arquitetura proposta ou extendê-la para permitir aplicações mais complexas.

A partir deste trabalho a linha de pesquisa pode caminhar para a aplicação em diferentes contextos dentro da robótica. Um caminho é tratar de aplicações reais, com robôs mais robustos, com diferentes níveis de autonomia e tarefas críticas, nas quais quem estiver interagindo com o robô deve sofrer o mínimo de rupturas de comunicação possível, pois estas rupturas teriam um custo muito alto. Em tarefas de desarmamento de bombas, por exemplo, um erro pode detonar uma bomba e causar danos sérios ao robô e ao ambiente que o rodeia.

Outro caminho é tratar de interfaces para controle e supervisão de múltiplos robôs, em ambientes em que o usuário deve ficar responsável por um certo número de robôs que realizam tarefas simultaneamente. Esse caminho permitiria verificar se são necessárias novas classes de signos (ou se as existentes já bastam) e quais representações poderiam ser utilizadas para as interfaces para múltiplos robôs.

Um ponto observado e que não pôde ser investigado a fundo é como o robô se encaixa dentro da ontologia da engenharia semiótica. O robô é parte do preposto do projetista? É um artefato intelectual linguístico? A resposta para estes questionamentos depende de uma análise mais profunda da teoria e do impacto da presença de um terceiro elemento além dos projetistas e dos usuários.

Por fim, a investigação de novas formas de interação é uma área que possui muito a ser investigado, tanto dentro da robótica quanto em outros contextos. Além da possibilidade de explorar a fundo as formas de interação apresentados neste trabalho, como o uso da caneta do *Tablet* e os toques na tela (inclusive o multi-toque¹) e sensores de movimento no *iPod*®), temos ainda a possibilidade de explorar outras formas, como a interação por voz, imagens, gestos, dentre outras.

¹No qual o dispositivo reconhece mais de um toque simultâneo na tela.

Referências Bibliográficas

- [Apple®, 2010] Apple® (2010). Xcode integrated development environment. Último acesso em agosto de 2010. <http://developer.apple.com/technologies/tools/xcode.html>.
- [Barbosa & de Paula, 2003] Barbosa, S. D. J. & de Paula, M. G. (2003). Designing and evaluating interaction as conversation: a modeling language based on semiotic engineering. *Springer Verlag Lecture Notes in Computer Science LNCS*, 2844.
- [Breazeal, 2004] Breazeal, C. (2004). Social interactions in hri: the robot view. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):181–186.
- [Burke et al., 2004] Burke, J.; Murphy, R.; Rogers, E.; Lumelsky, V. & Scholtz, J. (2004). Final report for the darpa/nsf interdisciplinary study on human-robot interaction. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):103–112.
- [da Silva, 2005] da Silva, B. S. (2005). MoLIC Segunda Edição: revisão de uma linguagem de modelagem da interação humano-computador. Dissertação de Mestrado, PUC-Rio, Brasil.
- [da Silva & Barbosa, 2007] da Silva, B. S. & Barbosa, S. D. J. (2007). Designing Human-Computer Interaction With MoLIC Diagrams – A Practical Guide. In *C.J.P. de Lucena (ed.), Monografias em Ciência da Computação*, Departamento de Ciência da Computação, PUC-Rio, Brasil. 12/07, ISSN 0103-9741.
- [de Souza, 2005a] de Souza, C. S. (2005a). Semiotic engineering: bringing designers and users together at interaction time. *Interacting with Computers*, 17:317–341.
- [de Souza, 2005b] de Souza, C. S. (2005b). *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press.

- [de Souza & Leitão, 2009] de Souza, C. S. & Leitão, C. F. (2009). *Semiotic Engineering Methods for Scientific Research in HCI*. Morgan & Claypool Publishers.
- [de Souza et al., 2006] de Souza, C. S.; Leitão, C. F.; Prates, R. O. & da Silva, E. J. (2006). The semiotic inspection method. Em *Procs of IHC '06*, pp. 148--157. ACM.
- [Douglas & Peucker, 1973] Douglas, D. H. & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112--122.
- [Drury et al., 2005] Drury, J. L.; Yanco, H. A. & Scholtz, J. (2005). Using competitions to study human-robot interaction in urban search and rescue. *Interactions*, 12(2):39-41.
- [EPFL, 2009a] EPFL (2009a). E-Puck Monitor. http://www.e-puck.org/?option=com_remository&Itemid=71&func=fileinfo&id=51.
- [EPFL, 2009b] EPFL (2009b). ePic. http://www.e-puck.org/?option=com_remository&Itemid=71&func=fileinfo&id=63.
- [Feil-Seifer & Mataric, 2005] Feil-Seifer, D. & Mataric, M. (2005). Defining socially assistive robotics. *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 465--468.
- [Fong et al., 2001a] Fong, T.; Cabrol, N.; Thorpe, C. & Baur, C. (2001a). A personal user interface for collaborative human-robot. Em *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Montreal, Canada.
- [Fong et al., 2001b] Fong, T.; Grange, S.; Thorpe, C. & Baur, C. (2001b). Multi-robot remote driving with collaborative control. Em *IEEE International Workshop on Robot-Human Interactive Collaboration*.
- [Fong et al., 2001c] Fong, T.; Thorpe, C. & Baur, C. (2001c). Collaboration, dialogue, human-robot interaction. Em Jarvis, R. & Zelinsky, A., editores, *Robotics Research, The Tenth International Symposium, ISRR 2001, Lorne, Victoria, Australia, November 9-12 2001*, volume 6 of *Springer Tracts in Advanced Robotics*, pp. 255--266. Springer.

- [Garcia & Chaimowicz, 2009] Garcia, R. & Chaimowicz, L. (2009). Uma infraestrutura para experimentação com enxames de robôs. Em *Anais do IX SBAI – Simpósio Brasileiro de Automação Inteligente*.
- [Gerkey et al., 2003] Gerkey, B. P.; Vaughan, R. T. & Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. Em *In Proceedings of the 11th International Conference on Advanced Robotics*, pp. 317-323.
- [Gold et al., 2007] Gold, K.; Fasel, I.; Freier, N. G. & Torrey, C. (2007). Young researchers' views on the current and future state of hri. Em *HRI '07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 357--364, New York, NY, USA. ACM.
- [Guo & Sharlin, 2008] Guo, C. & Sharlin, E. (2008). Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. Em *Procs of CHI '08*, pp. 121--130. ACM.
- [Humphrey et al., 2007a] Humphrey, C. M.; Henk, C.; Sewell, G.; Williams, B. W. & Adams, J. A. (2007a). Assessing the scalability of a multiple robot interface. Em *HRI '07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 239--246, New York, NY, USA. ACM.
- [Humphrey et al., 2007b] Humphrey, C. M.; Henk, C.; Sewell, G.; Williams, B. W. & Adams, J. A. (2007b). Assessing the scalability of a multiple robot interface. Em *Procs of HRI '07*, pp. 239--246. ACM Press.
- [Junior et al., 2008] Junior, J. M.; Junior, L. C. & Caurin, G. A. P. (2008). Scara3d: 3-dimensional HRI integrated to a distributed control architecture for remote and cooperative actuation. Em *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pp. 1597--1601, New York, NY, USA. ACM.
- [Kaptelinin & Nardi, 2006] Kaptelinin, V. & Nardi, B. (2006). *Acting with Technology: Activity Theory and Interaction Design (Acting with Technology)*. The MIT Press.
- [Kemp et al., 2007] Kemp, C.; Edsinger, A. & Torres-Jara, E. (2007). Challenges for robot manipulation in human environments [grand challenges of robotics]. *Robotics & Automation Magazine, IEEE*, 14(1):20–29.
- [Mondada et al., 2009] Mondada, F.; Bonani, M.; Raemy, X.; Pugh, J.; Cianci, C.; Klaptocz, A.; Magnenat, S.; Zufferey, J.-C.; Floreano, D. & Martinoli, A. (2009). The

- e-puck, a Robot Designed for Education in Engineering. Em Gonçalves, P. J.; Torres, P. J. & Alves, C. M., editores, *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pp. 59--65, Portugal. IPCB: Instituto Politécnico de Castelo Branco.
- [Nielsen, 1992] Nielsen, J. (1992). Finding usability problems through heuristic evaluation. Em *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 373--380, New York, NY, USA. ACM.
- [Nielsen, 2000] Nielsen, J. (2000). Why you only need to test with 5 users. Último acesso em outubro de 2010. <http://www.useit.com/alertbox/20000319.html>.
- [Peirce, 1931] Peirce, C. S. (1931). *Collected Papers of Charles Sanders Peirce*. Harvard University Press.
- [Perzanowski et al., 2001] Perzanowski, D.; Schultz, A. C.; Adams, W.; Marsh, E. & Bugajska, M. (2001). Building a multimodal human-robot interface. *Intelligent Systems*, 16(1):16--21.
- [Prates et al., 2000] Prates, R. O.; de Souza, C. S. & Barbosa, S. D. J. (2000). Methods and tools: a method for evaluating the communicability of user interfaces. *interactions*, 7(1):31--38.
- [Reeves & Nass, 2003] Reeves, B. & Nass, C. (2003). *The Media Equation : How People Treat Computers, Television, and New Media Like Real People and Places*. CSLI Lecture Notes. Center for the Study of Language and Inf.
- [Richer & Drury, 2006] Richer, J. & Drury, J. L. (2006). A video game-based framework for analyzing human-robot interaction: characterizing interface design in real-time interactive multimedia applications. Em *Procs of HRI'06*, pp. 266--273. ACM.
- [Rieman et al., 1995] Rieman, J.; Franzke, M. & Redmiles, D. (1995). Usability evaluation with the cognitive walkthrough. Em *CHI '95: Conference companion on Human factors in computing systems*, pp. 387--388, New York, NY, USA. ACM.
- [Rossen & Carroll, 2001] Rossen, M. B. & Carroll, J. M. (2001). Scenarios, objects, and points of view in user interface design. pp. 39--69.
- [Scholtz et al., 2004] Scholtz, J.; Young, J.; Drury, J. L. & Yanco, H. A. (2004). Evaluation of human-robot interaction awareness in search and rescue. Em *ICRA*, pp. 2327--2332. IEEE.

- [Shiomi et al., 2006] Shiomi, M.; Kanda, T.; Ishiguro, H. & Hagita, N. (2006). Interactive humanoid robots for a science museum. Em *HRI '06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 305--312, New York, NY, USA. ACM.
- [Skubic et al., 2006] Skubic, M.; Anderson, D.; Blisard, S.; Perzanowski, D. & Schultz, A. (2006). Using a qualitative sketch to control a team of robots. Em *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 3595--3601.
- [Steinfeld et al., 2006] Steinfeld, A.; Fong, T. W.; Kaber, D.; Lewis, M.; Scholtz, J.; Schultz, A. & Goodrich", M. (2006). Common metrics for human-robot interaction. Em *2006 Human-Robot Interaction Conference*. ACM.
- [Wang & Lewis, 2007] Wang, J. & Lewis, M. (2007). Human control for cooperating robot teams. Em *HRI '07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 9--16, New York, NY, USA. ACM.
- [Xin & Sharlin, 2007] Xin, M. & Sharlin, E. (2007). Playing games with robots - a method for evaluating human-robot interaction. Em *Human-Robot Interaction*, p. 522. Itech Education and Publishing.

Anexo A

Documentos utilizados na avaliação com os usuários

Termo de consentimento e participação

Título: Avaliação de interfaces para controle de um robô ePuck

Data: ____ / ____ / 2010

Instituição: DCC/UFMG

Pesquisadores responsáveis:

Luis Felipe Hussin Bento (lfhb@dcc.ufmg.br), Raquel Oliveira Prates (rprates@dcc.ufmg.br) e Luiz Chaimowicz (chaimo@dcc.ufmg.br)

Este Termo de Consentimento contém informações sobre a pesquisa indicada acima. Para assegurar que você esteja informado sobre a sua participação nesta pesquisa, pedimos que leia este Termo de Consentimento. Caso tenha alguma dúvida, não hesite em perguntar ao pesquisador responsável. Você também deverá assinar o termo do qual receberá uma cópia.

Objetivo da avaliação: O objetivo desta avaliação é identificar a influência das diferentes tecnologias e formas de interação no controle de um robô.

Informação geral sobre a pesquisa: Você será solicitado a realizar algumas tarefas simples utilizando cada uma das três interfaces. A realização destas tarefas será gravada para posterior análise pelos pesquisadores. Ao fim da execução das tarefas será executada uma entrevista sobre sua experiência com os sistemas.

Utilização dos dados coletados: Os dados coletados durante a avaliação serão utilizados no trabalho de mestrado do pesquisador Luis Felipe Hussin Bento, que trata de interação com robôs. Quaisquer dados utilizados para publicação serão apresentados de forma a garantir o anonimato dos participantes da avaliação.

Privacidade: Informações que possam identificar os participantes desta avaliação não serão divulgadas. Seu nome não constará em nenhum dos relatórios. Caso deseje, poderá solicitar uma cópia dos dados gerados por você.

Se você decidir não participar desta pesquisa: Você é livre para decidir, a qualquer momento, se quer participar ou não da pesquisa. Sua decisão não afetará sua vida estudantil e nenhum relacionamento com os avaliadores, professores ou a instituição por trás destes.

Compensação: A participação nesta pesquisa é voluntária e não será oferecido nenhum tipo de remuneração aos participantes.

Problemas ou questionamentos: Se você tiver algum problema ou questionamento que julga estar relacionado à pesquisa e a sua participação, poderá entrar em contato com os pesquisadores a qualquer momento pelo e-mail lfhb@dcc.ufmg.br.

Novas condições: Caso deseje, você poderá especificar abaixo condições adicionais que devam ser atendidas para que você participe dessa avaliação.

Consentimento livre e esclarecido (acordo voluntário)

O documento mencionado acima descrevendo os benefícios, riscos e procedimentos da pesquisa "Avaliação de interfaces para controle de um robô ePuck" foi lido e explicado. Eu tive a oportunidade de fazer perguntas sobre a pesquisa, que foram respondidas satisfatoriamente. Eu estou de acordo em participar como voluntário.

Data: _____ Assinatura do participante: _____
 Nome do participante: _____
 Assinatura do avaliador: _____
 Nome do avaliador: _____

Figura A.1. Termo de consentimento.

Script para avaliação dos sistemas para controle do ePuck

1) Recepção do participante

- a. Boas vindas e agradecimento.

2) Sistemas

- a. Apresentar os sistemas a serem usados.
- b. Explicar os três tipos de interação.

3) Explicar o objetivo da pesquisa

- a. Avaliar as três interfaces e estilos de interação diferentes. Não esquecer de deixar claro que não é o usuário que vai ser avaliado, mas sim os sistemas.

4) Apresentação da avaliação

- a. Explicar sobre a sala de teste, onde o usuário ficará sentado, onde o robô ficará e dizer que ele pode se levantar para observar a mesa enquanto realiza o teste. Explicar a questão do anonimato do teste.
- b. Apresentar a arena de testes e o robô. Mostrar o que são os objetos a serem recuperados e os padrões a serem fotografados.
- c. Explicar os passos do teste (leitura e assinatura do termo de consentimento, entrevista pré-teste para identificação do perfil do usuário, duração média do teste, realização das tarefas pelo usuário e entrevista pós-teste).
- d. Reforçar que o objetivo é avaliar as interfaces e não o usuário.
- e. Tirar dúvidas.

5) Leitura/assinatura do termo de consentimento.

6) Aplicação do questionário pré-teste.

7) Realização da avaliação.

- a. Antes de cada etapa:
 - i. Apresentar a interface a ser usada, dizendo como é a interação, como fazer para o robô andar e como fotografar. Explicar a limitação da câmera.
 - ii. Entregar a tarefa e tirar dúvidas.
 - iii. Posicionar os objetos da tarefa
 - iv. Iniciar gravação.
 - v. Entrevista sobre a interação.

8) Entrevista pós-teste.

9) Agradecimento pela participação voluntária

10) Configuração do ambiente para o próximo teste

- a. Reiniciar o servidor e o sistema de câmeras.
- b. Conferir gravação de vídeo.

Figura A.2. *Script* para a avaliação.

Questionário pré-teste

Data: ____ / ____ / 2010

Identificador: _____

Nome: _____ Idade: _____ Sexo: () M () F

Por favor, preencha os campos com a opção que melhor representa a sua resposta a cada pergunta.

1) Qual seu nível de escolaridade?

- Graduando
- Graduado
- Mestrando
- Mestre
- Doutorando
- Doutor
- Outro. Qual? _____

2) Em que área?

- Ciência da computação
- Sistemas de informação
- Matemática computacional
- Engenharia elétrica
- Engenharia de controle e automação
- Engenharia mecânica
- Outro. Qual? _____

3) Qual sua experiência com robôs?

- Nenhuma
- Tenho um conhecimento básico sobre robôs e seu funcionamento
- Já cursei uma disciplina de robótica
- Já desenvolvi trabalhos em robótica além dos exigidos na disciplina
- Minha área de pesquisa envolve robótica

4) Qual sua experiência com tablet PCs?

- Nenhuma
- Já vi ou ouvi falar, mas nunca usei
- Já usei pelo menos uma vez
- Já usei pelo menos uma vez para algo relacionado a desenho ou escrita
- Possuo um e/ou uso com frequência

5) Qual sua experiência com iPhone® / iPod touch®?

- Nenhuma
- Já vi ou ouvi falar, mas nunca usei
- Já usei pelo menos uma vez
- Já usei e conheço os recursos de detecção de movimento
- Possuo um e/ou uso com frequência

Figura A.3. Questionário pré-teste.

Tarefa para os sistemas de controle para o ePuck

Cenário

Esta é uma simulação em escala reduzida de uma situação de busca em um local utilizando robôs. A arena montada sobre a mesa representa um galpão onde houve uma explosão que abalou a estrutura do local, com risco de desabamento. Existem alguns objetos que estão dentro do galpão que devem ser recuperados, mas não se sabe se é seguro para uma pessoa entrar nele. Por isso será enviado um robô para verificar alguns locais críticos do galpão, fotografando-os para verificar se o local é seguro. É desejável que os objetos sejam recuperados também, para que não haja a necessidade de pessoas entrarem no local. Sua tarefa será controlar o robô para fotografar as três áreas de risco do galpão e recuperar os três objetos, trazendo-os de volta à entrada.

Instruções para a execução da tarefa

A tarefa é composta por três subtarefas, sendo que cada uma consiste em fotografar um ponto crítico identificado pelo padrão dado na tarefa e recuperar um objeto de dentro do galpão.

Cada subtarefa deverá ser executada usando uma interface diferente (**desktop**, **tablet** e **iPod touch®**), sendo que a ida para os locais de interesse deve ser feita passando no mapa os pontos do trajeto que o robô deve percorrer e a volta deve ser realizada dirigindo o robô diretamente.

Cada ponto crítico está identificado por um símbolo diferente: uma cruz (**+**), um triângulo (**▲**) e um círculo (**●**). Quando alcançar o local com o robô, você deverá fotografá-lo de modo que o símbolo fique visível. Os objetos são pequenos cilindros que possuem os mesmos símbolos dos locais críticos.

Figura A.4. Cenário e descrição geral das tarefas.

Subtarefa Desktop

Esta tarefa consiste em fotografar o local marcado com **+** e recuperar o objeto com a mesma identificação. Para isso você deverá utilizar a **interface para desktop**. Você deverá chegar até eles passando os pontos do trajeto no mapa. Quando chegar ao local, deverá ajustar o robô e utilizar a câmera para fotografá-lo de forma que o símbolo que o identifica seja visível. O próximo passo é recuperar e trazer o objeto com o mesmo símbolo de volta, dirigindo o robô diretamente.

Subtarefa Tablet

Esta tarefa consiste em fotografar o local identificado com **▲** e recuperar o objeto com o mesmo símbolo. A busca deverá ser feita usando a **interface para tablet PCs**. Você deverá chegar até eles passando os pontos do trajeto no mapa. Quando chegar ao local, deverá ajustar o robô e utilizar a câmera para fotografá-lo de forma que o símbolo que o identifica seja visível. O próximo passo é recuperar e trazer o objeto com o mesmo símbolo de volta, dirigindo o robô diretamente.

Subtarefa iPod touch

Esta tarefa consiste em fotografar o local identificado com **●** e recuperar o objeto com o mesmo símbolo, devendo ser realizada utilizando a interface para **iPod touch®**. Os passos da busca são os seguintes: Você deverá chegar até eles passando os pontos do trajeto no mapa. Quando chegar ao local, deverá ajustar o robô e utilizar a câmera para fotografá-lo de forma que o símbolo que o identifica seja visível. O próximo passo é recuperar e trazer o objeto com o mesmo símbolo de volta, dirigindo o robô diretamente.

Figura A.5. Descrição das tarefas para cada interface.

Formulário de acompanhamento do teste

Pesquisa: Avaliação de interfaces para controle de um robô ePuck

Avaliador: Luís Felipe Hussin Bento

Data: ____ / ____ / 2010

Identificação do participante: _____

Tarefa desktop, **+**

Tarefa	Executou corretamente?	
	Sim	Não
Levar o robô à área de risco		
Fotografar a área		
Chegar ao objeto a ser recuperado		
Encaixar o objeto no robô		
Trazer o objeto para a entrada		

(a)

Tarefa tablet, **▲**

Tarefa	Executou corretamente?	
	Sim	Não
Levar o robô à área de risco		
Fotografar a área		
Chegar ao objeto a ser recuperado		
Encaixar o objeto no robô		
Trazer o objeto para a entrada		

Tarefa iPod, **●**

Tarefa	Executou corretamente?	
	Sim	Não
Levar o robô à área de risco		
Fotografar a área		
Chegar ao objeto a ser recuperado		
Encaixar o objeto no robô		
Trazer o objeto para a entrada		

(b)

Figura A.6. Formulário de acompanhamento do teste. A primeira página do formulário corresponde à parte marcada com (a). A parte marcada com (b) se refere às duas páginas seguintes, cujo conteúdo é bastante semelhante à primeira (diferenciando pelo título da tarefa). Por esse motivo as duas páginas não foram reproduzidas em seus tamanhos originais.

Roteiro para entrevista

Após cada etapa do teste

- 1) O que achou fácil de usar na interface? Por quê?
- 2) O que achou difícil? Por quê?
- 3) O que mais gostou e o que menos gostou? Por quê?
- 4) Sentiu falta de alguma informação na interface? Quais e por quê?

Ao final do teste

- 1) Qual das 3 interfaces gostou mais? Por quê?
- 2) Qual gostou menos? Por quê?
- 3) Teve alguma dificuldade comum a mais de uma interface? Quais e por quê?

Figura A.7. Roteiro básico para a entrevista pós-teste.