Interacção Humano-Robô ao Nível da Linguagem

Mário Rodrigues, Luís Seabra Lopes, António Teixeira

mfr@ieeta.pt, lsl@det.ua.pt, ajst@det.ua.pt
Actividade Transversal em Robótica Inteligente
Departamento de Electrónica e Telecomunicações + Instituto de Engenharia Electrónica e Telemática de Aveiro
Universidade de Aveiro, P-3810 Aveiro - Portugal

***Resumo* - O desenvolvimento de um sistema capaz de interagir com um ser humano através de linguagem natural falada é algo que tem sido alvo de muitos estudos e projectos. Este tipo de sistemas são aliciantes pois permitem ao Homem comunicar com a máquina, de uma forma simples e intuitiva. Neste artigo, é apresentado o estado actual da interface de fala desenvolvida para um robô – o Carl. Descreve-se a integração e interligação de todos os subsistemas constituintes da interface desenvolvida, desde o processamento de linguagem natural falada até à própria gestão da interacção (diálogo).**

***Abstract* - Developing a system capable of using spoken language for interaction with humans has been pursued by several studies. Such systems are interesting by allowing humans to communicate with machines in a simple and intuitive way. We present, in this paper, the current development stage of the spoken language interface of the robot Carl.**

I. PROJECTO CARL

O projecto CARL (**C**ommunication, **A**ction, **R**easoning and **L**earning in Robotics) [4,9] visa o desenvolvimento de um robô capaz de entender, através de um interface amigável, instruções expressas em termos de conceitos familiares ao utilizador humano.

A filosofia do projecto é baseada na hipótese de que é possível construir um robô inteligente, delineando a estrutura geral e organização da representação do mundo real e dos módulos de execução de tarefas, desde as funções de controlo de baixo nível até às capacidades de decisão de alto nível [6].

Numa arquitectura deste tipo, os símbolos desempenham um papel muito importante, pois neles se irá basear a representação que a máquina terá do ambiente e das tarefas a desempenhar. Pode mesmo dizer-se que o problema da definição do significado dos símbolos corresponde, no essencial, ao clássico problema da programação de robôs. A incorporação de mecanismos de aprendizagem em arquitecturas robóticas está a ser explorada com o intuito de simplificar o problema da programação. O projecto CARL visa uma aprendizagem do robô, não apenas ao nível de conceitos primitivos, mas também de tarefas.

Para que um utilizador comum possa utilizar um robô inteligente, a única interface suficientemente flexível e amigável é o diálogo em linguagem natural. Na verdade, nenhum outro tipo de interface parece ser suficientemente flexível [2].

II. UM ROBÔ CHAMADO CARL

O projecto CARL está a utilizar a plataforma móvel PIONEER 2-DX da ActivMedia Robotics [8]**.** Esta plataforma, contém entre outras coisas:

- três rodas (duas rodas motrizes e uma direccional);
- 16 sonares repartidos em dois conjuntos de 8 (um conjunto à frente e outro atrás);
- pára-choques com sensores de colisão;
- um computador baseado no processador Pentium 266 MMX da Intel, com 64 MB de memória RAM e com o sistema operativo Linux instalado;

Por cima da plataforma PIONEER 2-DX foi montada uma estrutura em fibra de vidro que aumenta a altura do Carl para aproximadamente 85cm. A estrutura em fibra de vidro contém:

- 10 sensores infravermelhos para detecção de obstáculos;
- 1 sensor infravermelho para detecção de objectos no compartimento superior;
- 1 vector de microfones;
- 1 coluna de som.

O principal motivo para a adição da estrutura em fibra foi o de aproximar o microfone da face do utilizador. Se o microfone estivesse apoiado directamente na plataforma PIONEER 2-DX, o reconhecimento seria mais difícil devido à maior distância entre este aparelho e a boca do utilizador.

Na figura seguinte está uma imagem do corpo actual do Carl.

Figura n.º 1 - Imagem do Carl.

O vector de microfones utilizado é o DA-400 da Andrea Electronics [10]. O DA-400 consegue filtrar o som de modo a que só capte sons que vêm de fontes até 30 graus da perpendicular do vector (ver Figura n.º 2) para fontes de ruído até 1/3 de oitava centradas na frequência 1 kHz. Apesar de existirem outros microfones com estas propriedades, este era o melhor, na altura da sua aquisição.

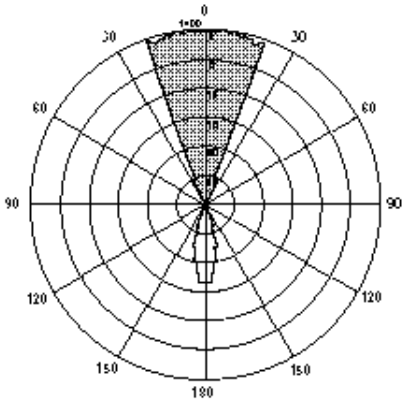

Figura n.º 2 – Zona de filtragem do vector de microfones DA-400

Anteriormente, foram utilizados outros dois microfones. O primeiro foi o modelo LVA-7280 ClearVoice da Labtec [11]. Este modelo é também um vector de microfones, contudo o seu alcance é menor que o DA-400, tendo revelado um fraco desempenho para ser utilizado com o Carl. Depois utilizamos o modelo TCHS-W da Shure [12]. Este modelo é composto por um receptor e um emissor que comunicam por rádio-frequência. O receptor é ligado ao Carl e o emissor fica com o utilizador. O problema é que o utilizador precisa de andar sempre com o emissor. Como tal, é uma má solução, porque pretendemos que o Carl seja autónomo, que interaja com qualquer pessoa e que a sua utilização seja muito fácil e prática.

III. Arquitectura Computacional

O sistema de execução e controlo do Carl está organizado num conjunto de processos executados sobre o sistema operativo Linux. Para o reconhecimento de voz temos dois processos e para a síntese outro processo. Mais processos encarregam-se do processamento de outros tipos de informação vinda do exterior através dos sonares, dos sensores infravermelhos, da visão e dos sensores de colisão. Estes processos foram desenvolvidos em *C*.

Toda a informação de alto-nível converge para um processo, o Gestor de Execução, cujo núcleo central é um sistema de transição de estados [5]. O Gestor de Execução toma todas as decisões de alto-nível relativas às tarefas a executar e à interacção com o utilizador humano. O módulo de aprendizagem recolhe, organiza e fornece informação. Actualmente, os módulos de Gestão de Execução e de Aprendizagem coexistem no mesmo processo Linux, estando ambos implementados em Prolog. Os processos comunicam entre si por troca de mensagens.

A organização e comunicação dos diferentes módulos está representada na Figura n.º 3:

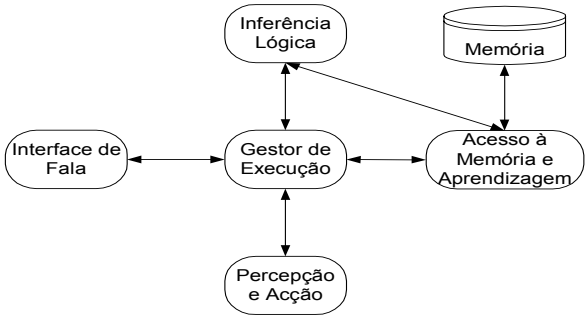

Figura n.º 3 – Arquitectura computacional.

IV. Descrição Geral da Interface

Uma interface tem por função permitir a comunicação entre o Homem e a máquina. No caso dos robôs de serviço, a interface deve parecer natural para o utilizador humano. Como a principal forma de comunicação do Homem é o discurso falado, as interfaces de voz utilizando linguagem natural são as mais intuitivas para os humanos. Infelizmente, o reconhecimento de fala ainda não é muito robusto nos dias de hoje. Em virtude disso, a construção de interfaces de fala robustas é uma tarefa muito difícil.

Podemos dividir a interface de fala em quatro blocos principais, tal como se pode ver na Figura n.º 4.

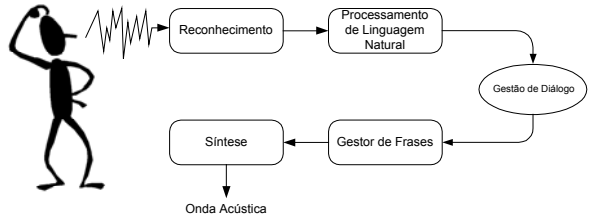

Figura n.º 4 - Módulos de um sistema de diálogo baseado em linguagem natural falada

O bloco de reconhecimento converte o sinal acústico (sinal de voz) numa sequência de fonemas que formam cada palavra e posteriormente a frase. A frase reconhecida apenas nos traz informação disso mesmo - o que foi reconhecido (frase mais provável de ter sido dita pelo interlocutor), não tem correspondência directa na representação interna utilizada pelo processo de decisão do robô. É necessário, numa primeira fase, proceder à extracção da estrutura sintáctica da frase. Para isso, a frase é tratada pelo módulo de processamento de linguagem natural. Este módulo estrutura a frase reconhecida de modo a que esta seja interpretável pela Gestão de Diálogo.

A actividade da Gestão de Diálogo consiste em interpretar a mensagem vinda do utilizador e reagir em conformidade. Recebe e envia informação para a interface de fala. O Gestor de Frases recebe a mensagem que o robô quer transmitir ao utilizador e constrói uma frase que a represente.

Quando o robô necessita de transmitir informação ao utilizador tem de o fazer por voz. Por isso existe o bloco de Síntese que tem por função gerar um sinal acústico a partir de texto. O sinal acústico tem de ser perceptível pelo ser humano.

V. Reconhecimento

O bloco de reconhecimento é o mais crítico e onde se nos depararam a maioria dos problemas inerentes à interface de fala. As dificuldades surgem quando temos um elevado nível de ruído ambiente: conversas paralelas à frase a ser reconhecida; mudança súbita de contexto por parte do orador. O ruído dos motores do robô também podem afectar o processo de reconhecimento. Também ocorrem problemas quando os modelos de voz não são os mais adaptados a um determinado orador, devido ao seu tom de voz ou à sua pronuncia.

No primeiro ano deste projecto o reconhecedor utilizado foi o *grapHvite*, desenvolvido pela Entropic. Foi igualmente utilizada uma ferramenta de processamento de linguagem natural, *CPK-NLP*, desenvolvida na Universidade de Aalborg, Dinamarca, pela equipa de Tom Brøndsted [1]. Esta foi utilizada para a construção da gramática de reconhecimento.

Infelizmente, o *grapHvite* não permitia o retreino dos modelos de voz. A vantagem de treinar os modelos não é só de os melhor adaptar a determinados utilizadores, mas também de os adaptar às condições de ruído existentes no meio ambiente mais usual para o Carl. Assim, optámos por mudar de reconhecedor. Actualmente estamos a utilizar o *IBM ViaVoice*.

Já procedemos ao treino dos modelos de voz (HMM – Hidden Markov Models) do *ViaVoice*. Verificamos, através do uso, algumas melhorias no reconhecimento, contudo ainda não foi feita uma avaliação objectiva e sistemática à diferença de desempenho. Um dos próximos passos a dar no interface de fala do Carl é precisamente quantificar a melhoria obtida com o retreino dos modelos de voz.

O *ViaVoice* fornece um conjunto de rotinas em *C*, o *Speech Manager Application Interface* (*SMAPI*), que possibilitam a interacção com o reconhecedor. Este reconhecedor é tolerante a hesitações do orador e permite retreinar os modelos de voz. Construímos o nosso sistema de reconhecimento de fala, utilizando rotinas do SMAPI para modelar e interagir com o reconhecedor de voz.

Para que o resultado do reconhecimento seja conhecido tão cedo quanto possível, à medida que o utilizador fala, o reconhecedor vai processando o sinal de voz. Como resultado, o reconhecedor pode devolver, para além da frase reconhecida, um nível de confiança nessa frase.

As frases que podem ser aceites estão definidas numa gramática. Para comodidade do programador, esta gramática é escrita no formato *Augmented Phrase Structure Grammar* (APSG). Uma ferramenta desenvolvida pelo projecto converte-a para BNF. Finalmente, o *ViaVoice* converte-a para o formato *Finite State Grammar* (FSG), utilizado pelo reconhecedor.

Cada palavra é definida, no *ViaVoice*, pela concatenação de elementos de um conjunto de fonemas.

Em virtude disso, para garantir que qualquer palavra possa ser reconhecida, como por exemplo qualquer nome próprio, adaptámos rotinas do *CPK*, que geram um dicionário, por forma a trabalhar com a representação dos fonemas e estrutura do dicionário usados pelo *ViaVoice*. O dicionário é um ficheiro onde estão contidas as palavras da gramática e a respectiva representação em fonemas.

No formato APSG, definem-se diversos atributos para cada palavra, através de *lexical rules* (L-rules). Existem dois atributos obrigatórios: o *cat* (categoria) e o *lex* (elemento lexical ou palavra). Os restantes são definidos pelo utilizador. Cada palavra pode ser definida várias vezes, com diferentes atributos. Alguns exemplos de L-rules podem ser vistos na Figura n.º 5.

Com base nos atributos das palavras, são construídas as estruturas das frases, utilizando *structure building rules* (B-rules). As B-rules também podem incluir referências a outras B-rules, como se pode ver no exemplo da Figura n.º 6. Após a definição da gramática, o conversor gramatical gera todas as frases permitidas pela conjunção das L-rules com as B-rules.

```
{lex=is,cat=verb,verb=be,vform=present,
  person=third}.
{lex=tony,cat=noun,type=person,art=no,
  person=third}.
{lex=where,cat=prep,type=askprep,subject=place}.
```

Figura n.º 5 – Exemplos de L-rules

```
ask_phrase5={cat=ask5,stype=ask}
[
    (
       {cat=prep,type=askprep,subject=place};
       {cat=prep,type=askprep,subject=person}
    ),
    {cat=verb,verb=be,vform=present,person=$P},
    {cat=np,type=just_noun,person=$P}
].

np_noun = {cat=np,type=just_noun,person=$P}
[
     {cat=pron,type=personal,person=$P};
     {cat=noun,type=person,art=no,person=$P}
].
```

Figura n.º 6 – Exemplos de B-rules

A gramática actual é uma versão muito melhorada da gramática referida em publicação anterior [3]. Actualmente a gramática contém cerca de 50 estruturas de frases (B-rules) e aproximadamente 100 palavras (L-rules) diferentes. Com estas estruturas e palavras são geradas mais de 12000 frases. A representação interna da gramática é um grafo, onde cada nó é uma palavra da gramática. Os tipos de frases aceites (ver Tabela 1) são um subconjunto da *Human Robot Communication Language* (*HRCL*) [2].

Tipo de frase	Descrição
Register(S,R)	**S** anuncia a sua presença a **R**
Achieve(S,R,C)	**S** pede a **R** para fazer a acção **C** no seu meio físico
Tell(S,R,C)	**S** diz a **R** que a preposição **C** é verdadeira
Ask(S,R,C)	**S** pergunta a **R** uma instância da preposição **C**
Ask_if(S,R,C)	**S** pergunta a **R** se a preposição **C** é verdadeira
Thanks(S,R)	**S** agradece a **R**
Bye(S,R)	**S** despede-se de **R**
Dye(S,R)	**S** (humano) pede a **R** (robô) para terminar todos os seus processos

Tabela 1 – Tipos de frases aceites (S=sender, R=receiver)

A gramática actual não pode ser muito mais aumentada devido à capacidade de processamento requerida. E, no entanto, esta gramática apenas cobre uma pequena parte da gramática inglesa. Por exemplo, nas frases do tipo Tell, Ask e Ask_if só são aceites frases cujo verbo seja *to be*, por exemplo: "*The professor is in Portugal*"; "*The car of Peter is at the University*"; ou "*The chairman of the conference is a professor*".

Uma forma de alargar o conjunto de frases reconhecíveis, sem degradar o desempenho do reconhecimento, é através do ajuste dinâmico da gramática ao contexto. Para isso foram construídas ferramentas para adicionar gramáticas dinamicamente, isto é, à medida que o diálogo se desenrola, a gramática pode variar.

Um cenário típico onde a adição dinâmica de gramáticas é vantajosa, surge quando o robô coloca questões ao utilizador. Para cada pergunta, é definido um conjunto de respostas possíveis. À medida que o robô coloca uma questão, o conjunto de respostas permitidas para essa questão, é definido como a gramática para reconhecimento a utilizar naquele instante. Assim, em vez de a gramática conter todas as respostas permitidas em qualquer contexto, só contem as permitidas para aquela pergunta em particular. Este método, além de exigir menos recursos da máquina, também reduz a probabilidade de erro no reconhecimento.

Também no caso de uma frase não ser completamente reconhecida, é gerada uma gramática com todos os finais possíveis para essa frase (gramática de completação). Para isso é feita uma pesquisa no grafo da gramática para determinar qual é a sequência de nós que representam a frase reconhecida. Se o último nó da sequência encaminha para um estado final, a frase foi totalmente reconhecida,

senão é gerada uma gramática com todos os percursos possíveis a partir desse nó até um estado final. O robô pede então para o utilizador completar a frase anterior a partir de um certo ponto.

Estas ferramentas encontram-se em fase final de desenvolvimento.

VI. Processamento de Linguagem Natural

A frase reconhecida é passada ao módulo de Processamento de Linguagem Natural. Este módulo constrói uma representação da estrutura sintáctica da frase e identifica qual o tipo de mensagem HRCL em que essa frase se enquadra. O resultado deste processamento é passado ao módulo de Gestão de Diálogo.

A mensagem *HRCL* é construída utilizando regras definidas na gramática APSG, as *semantic rules* (M-rules). Deste modo, garantimos que todas as frases da gramática têm uma representação coerente no formato *HRCL*.

As M-rules podem incluir referências a outras M-rules e são constituídas por duas partes distintas. A parte que está abaixo do símbolo "/", indica a estrutura das frases às quais uma determinada M-rule é aplicada. A parte acima do símbolo "/", define a estrutura sintáctica desse tipo de frases. As M-rules correspondentes às B-rules da Figura n.º 6, estão na figura seguinte:

```
(ask(phrase(askprep($P),verb($V),#NP)))
/
{cat=ask5,stype=ask}
[
    {cat=prep,lex=$P},
    {cat=verb,verb=$V},
    {cat=np}#NP
].

(np(propernoun($N)))
/
{cat=np}
[
    {cat=noun,lex=$N}
].
```

Figura n.º 7 – Exemplos de M-rules

Um exemplo de uma frase reconhecida e da mensagem *HRCL* correspondente, gerada a partir das M-rules apresentadas na Figura n.º 7, é apresentado de seguida:

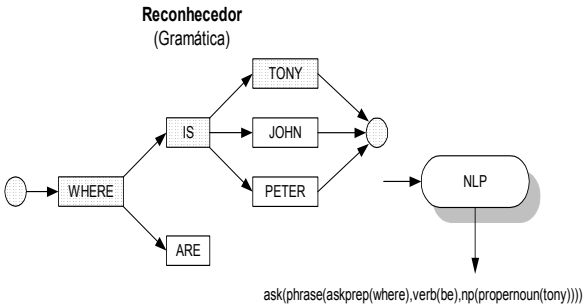

Figura n.º 8 – Composição da mensagem *HRCL*

O interlocutor diz a frase "Where is Tony". Esta informação, ao passar pelo módulo de Processamento de Linguagem Natural, fica com o seguinte aspecto: `ask(phrase(askprep(where),verb(be),np(propernoun(tony))))`, onde é indicado ao robô que o interlocutor está a perguntar algo (ask) e que a frase é constituída por uma preposição "where", por um verbo (be) e, por fim, um sintagma nominal (np) com um nome próprio (propernoun) "tony".

A mensagem *HRCL* é então enviada para o módulo de Gestão de Diálogo que extrai a respectiva semântica. Esta tarefa está implementada em *Prolog*. A semântica da frase é uma descrição relacional. A título de exemplo, a regra escrita em Prolog ilustrada na Figura n.º 9 extrai a semântica de frases declarativas (fases do tipo Tell), baseadas no verbo *to be* e que incluam uma preposição. A chamada recursiva da regra extrai a semântica do objecto directo, originando NP1sem e relações adicionais na lista L1. Pelo mesmo processo é extraída a semântica do objecto indirecto. Finalmente a semântica da frase é expressa pela relação is_(NP1sem, What), com relações complementares na lista L3. Mais regras desta natureza processam outros tipos de frases presentes na gramática.

```
semantics(
  tell(phrase(NP1,verb(be),prep(P),NP2)),
  is_(NP1sem,What),
  L3
):- semantics(NP1,NP1sem,L1),
    semantics(NP2,NP2sem,L2),
    What=..[P,NP2sem],
    append(L1,L2,L3),
```

Figura n.º 9 – Exemplo de uma regra para extrair a semântica de uma frase do tipo Tell

Como exemplo, a semântica da frase "*Professor Carlos is at the University of Aveiro*", será representada, depois de processada segundo a regra da Figura n.º 9, pela seguinte lista de relações: `[is_(X,at(Y)), is_(X,professor), obj_name_(X,carlos), of_(Y,Z), is_(Y,university), obj_name_(Z,aveiro)].`

VII. SÍNTESE

Quando o robô pretende transmitir alguma informação ao utilizador, entrega essa informação ao Gestor de Frases. Este módulo gera, consoante a informação recebida, uma frase perceptível pelo utilizador. No fundo este módulo efectua a operação inversa da do bloco de Processamento de Linguagem Natural.

A frase gerada tem de ser gramaticalmente correcta e deve estar em conformidade com o contexto. Deve também variar, embora sem alterar a informação a transmitir, de forma a que o robô não diga sempre as mesmas frases nas mesmas situações. É, depois, passada ao módulo de síntese.

O módulo de síntese de voz transforma uma mensagem proveniente do gestor de frases numa onda acústica. Além da habitual conversão de texto para voz, é competência deste módulo transmitir informação paralinguistica, como a emoção. Pretende-se que, consoante o tipo de mensagem a sintetizar (informação, pedido, mensagem urgente, ..) sejam utilizados parâmetros do sintetizador (como o tom, intensidade e velocidade) adequados.

O som sintetizado deve ser facilmente perceptível. Para tal é necessário utilizar um bom sintetizador e uma boa coluna de som.

O sintetizador em uso é o *IBM ViaVoiceTTS* (*TTS*-Text To Speech). Este sintetizador permite alteração de inúmeros parâmetros da voz sintetizada, conseguindo assim atribuir emotividade e identidade à voz do robô.

A síntese ainda não foi alvo de grande pesquisa neste projecto. No que diz respeito à geração do texto da frase, dada uma semântica, são utilizados mecanismos muito simples (frases pré-definidas para certas situações, respostas a perguntas segundo um estrutura pré-definida).

Quanto à síntese de fala, limitamo-nos a utilizar uma configuração *standard* do *ViaVoiceTTS*. Presentemente, um dos trabalhos em curso consiste em desenvolver uma voz específica e mais natural (humana) para o Carl.

VIII. CONCLUSÃO

O trabalho descrito é um trabalho típico de integração. Um conjunto de ferramentas disponíveis no domínio público (*IBM ViaVoice* e *ViaVoiceTTS*, *CPK-NLP*) e outras desenvolvidas pelo projecto foram integradas numa plataforma física, ela própria resultante da integração de componentes adquiridos com outros desenvolvidos por nós. O nosso trabalho consistiu em desenvolver módulos de software para cobrir certas funcionalidades bem como o sistema global de gestão da execução e gestão do diálogo. O trabalho de integração é hoje reconhecido como um dos mais complexos em robótica [6,7].

O Carl participou no *AAAI Robot Competition*: *Hors d'Oeuvres Anyone?* organizado pela American Association for Artificial Intelligence, que decorreu em Seattle, EUA, em Agosto de 2001. O objectivo era servir aperitivos aos congressistas, interagir com as pessoas e com os outros robôs concorrentes. Esta exibição foi uma óptima ocasião para testar, entre outras coisas, o desempenho do interface de voz num ambiente real. O desempenho verificado não foi o melhor devido sobretudo ao grande ruído de fundo que existia no pavilhão onde decorreu a prova. Como tal o reconhecimento de voz foi muito fraco. Raramente o robô conseguiu reconhecer o que lhe diziam. Convém notar, no entanto, que este fraco desempenho se deve considerar normal, no estado actual de desenvolvimento das tecnologias de reconhecimento de fala. De resto, os restantes robôs presentes no concurso nem sequer apresentaram capacidades de reconhecimento de fala.

A classificação final obtida pelo Carl na competição da AAAI foi o terceiro (3º) lugar.

IX. REFERÊNCIAS

[1] T. Br∅ndsted, "The CPK NLP Suite for Spoken Language Understanding", em *Proc. EuroSpeech,* 1999.

[2] L. Seabra Lopes A. Teixeira, "Teaching Behavioral and Task Knowledge to Robots through Spoken Dialogues", em *My Dinner with R2D2: Natural Dialogues with Practical Robotic Devices*, AAAI-2000 Spring Symposium Series, Stanford, Março 2000.

[3] L. Seabra Lopes e A. Teixeira, "Human-Robot Interaction through Spoken Language Dialogue", em *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Takamatsu, Japão, 2000.

[4] L. Seabra Lopes, K. L. Doty, F. Vaz e J. A. Fonseca. "Service Robotics and the Issue of Integrated Intelligence", em *Proc. ECAI Workshop on Service Robotics - Applications and Safety Issues in an Emerging Market*, Berlim, pp. 35-44, 2000.

[5] L. Seabra Lopes, "Carl, a Learning Robot, serving Food at the AAAI Reception", *Proc. of the 11th AAAI Mobile Robot Competition and Exhibition Workshop*, Seattle, Washington, 2001.

[6] L. Seabra Lopes and J.H. Connell, edrs., *Semisentient Robots* (special issue of *IEEE Intelligent Systems*, 16 (5)), IEEE Computer Society, 2001.

[7] L. Steels, "Language Games for Autonomous Robots", in [6], 2001.

[8] http://www.activrobots.com/robots/p2dx.html

[9] http://www.ieeta.pt/CARL

[10] http://www.andreaelectronics.com/

[11] http://www.labtec.com/

[12] http://www.shure.com/