

BonoBOT: Um Chatterbot para Interação com Usuários em um Sistema Tutor Inteligente

Rachele Bianchi Sganderla¹, Débora Nice Ferrari^{1,2}, Cláudio F. R. Geyer²

¹Centro Universitário La Salle (UNILASALLE)
Curso de Ciência da Computação
Av. Victor Barreto, 2.288 – 91.501-970 – Canoas – RS – Brasil

²Universidade Federal do Rio Grande do Sul (UFRGS)
Instituto de Informática –
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

kelly.lasalle@terra.com.br, nice@lasalle.tche.br,
{nice,geyer}@inf.ufrgs.Br

Resumo. Na busca incessante pela compreensão e simulação do pensamento e do comportamento humano, a ciência da Inteligência Artificial tem buscado desenvolver os mais diferentes meios de fazer com que um computador possa realizar tarefas de forma racional, não mais prendendo-se a um conjunto pré-programado de instruções. Dentre os diversos artifícios aplicados no estudo da Inteligência Artificial estão os chatterbots – programas desenvolvidos para simular uma conversa através da troca de mensagens de texto num formato de bate-papo virtual. As aplicações de um chatterbot podem ser as mais variadas, podendo servir como um “receptionista” em um site comercial, responder a FAQs (Frequently Asked Questions) ou atuar na área educacional dando suporte ao estudo e pesquisa. Este trabalho vem propor o estudo e a implementação de um chatterbot como uma ferramenta de auxílio educacional, apoiando o processo de ensino-aprendizagem em um Sistema Tutor Inteligente.

Palavras-chave: agentes pedagógicos, processamento de linguagem natural, chatterbot.

Abstract. The science of Artificial Intelligence has been developing ways to make possible that a computer may fulfill different jobs rationally, no more being closed to a set of pre-programmed instructions. Between the different skills applied on the studies of Artificial Intelligence are the chatterbots – software developed to simulate dialogs on natural language by changing text messages like a virtual chat. There are several appliances for a chatterbot, being possible to work as a ‘receptionist’ on a commercial web site, answering FAQs (Frequently sked Questions) or acting on an educational environment, giving the necessary support to studies and research. This work comes to propose the study and implementation of a chatterbot as a tool for educational purposes, supporting the process of teaching and learning on na Intelligent Tutor System.

Key words: pedagogical agents, natural language processing, chatterbot.

1. Introdução

Nos últimos anos, diferentes formas de comunicação entre dois (ou mais) interlocutores foram desenvolvidas. Novos campos se abriram na Tecnologia da Informação, destacando-se o desenvolvimento de interfaces cognitivas e a inteligência artificial. Com estas mudanças, juntamente com as novas tecnologias de comunicação, de forma especial a Internet, surgiram programas para troca de informações entre dois indivíduos ou mesmo entre grupos de interlocutores que visem um mesmo interesse. Dentro deste contexto, destaca-se a Educação a Distância apoiada pelo uso do computador. Assim, com essa tecnologia de ensino, a comunicação virtual permite interações espaço-temporais mais livres, adaptação a ritmos diferentes dos alunos, novos contatos com pessoas semelhantes (ou com interesses semelhantes) fisicamente distantes e maior liberdade de expressão a distância [Moran 2000]. Entre os ambientes computacionais de educação a distância disponibilizados hoje, os Sistemas Tutores Inteligentes (STI) [Giraffa 1999] surgem para centralizar as ações do professor em sua tarefa de auxiliar no processo de construção do conhecimento. Estes sistemas agregam formas de disponibilização do conteúdo, de acompanhamento no desenvolvimento do aluno e meios para troca de mensagens e informações entre os usuários (sejam eles estudantes ou tutores). Seguindo este novo paradigma, temos hoje boas iniciativas, tais como o Semeai que objetiva a disponibilização de um ambiente de educação a distância apoiado por uma sociedade de agentes [Geyer 2001],[Oliveira 2001].

Para dar suporte a estes sistemas, a Inteligência Artificial apresenta-se como fator importante no relacionamento homem-máquina. Os *chatterbots*, agentes inteligentes desenvolvidos para simular uma conversa através da troca de mensagens de texto (semelhantemente aos bate-papos virtuais), são sistemas desenvolvidos com o intuito de tornar mais familiar a interação entre o homem e a máquina, geralmente usando recursos da Inteligência Artificial. Um *chatterbot* serve-se da idéia básica da interação entre as pessoas para dar a impressão de que o computador possui uma personalidade. Para transmitir tal impressão, o desenvolvimento de um *chatterbot* toma por base o “Jogo da Imitação” (“Imitation Game”) [Turing 1950], cujo propósito é fazer com que um ser humano acredite que esteja conversando com outra pessoa. Isto possibilita ao usuário sentir-se muito mais à vontade na troca de mensagens com o sistema, passando a enxergá-lo não mais como um simples equipamento, mas como um novo “amigo”.

Este trabalho vem apresentar uma ferramenta interativa para auxílio ao processo de ensino e aprendizagem em um STI sob a forma de um *chatterbot* (neste trabalho também denominado *bot* ou robô). Este visa auxiliar o usuário em sua interação com o sistema, incentivando-o nos estudos e pesquisas e colaborando na construção do conhecimento, desempenhando o papel de um monitor no estudo. O STI utilizado é o Semeai [Geyer 2001]. O processo de interação entre usuário e *bot* se efetiva durante seu processo de interação com o STI, fazendo do robô uma ferramenta de auxílio no processo de aprendizagem do estudante, proporcionando uma interação menos formal e mais próxima da linguagem natural. O texto está organizado em 5 seções. Além desta introdução, a seção 2 aborda questões relacionadas a interatividade, educação a distância e *chatterbots*, destacando assim a fundamentação teórica que envolve o trabalho. A seção 3 apresenta o modelo do BonoBOT, um *chatterbot* para auxílio ao processo de ensino e aprendizagem na internet. Na seção 4 são apresentados aspectos relacionados com a implementação do *bot* e com as tecnologias utilizadas. Finalmente, a seção 5 apresenta as conclusões finais do trabalho.

2. Interatividade, Educação a Distância e *Chatterbot*

Conforme [Belloni 1999], as novas tecnologias de informação e comunicação oferecem possibilidades inéditas de interação mediatizada (professor/aluno; estudante/estudante) e de interatividade com materiais de boa qualidade e grande variedade. Vários autores definem interatividade [Lévy 1999a], [Steuer 1992], [Lippmann 1998]. Contudo, para o foco deste trabalho, a interatividade pressupõe interação direta entre participante e máquina, tendo como elemento mediador algum veículo técnico de comunicação para suporte a interação. Atualmente a educação a distância faz uso de tecnologias interativas nas quais os aprendizes estão conectados de forma síncrona (tempo real, ao vivo e conversacional) ou assíncrona (retardada ou adiantada para antes ou depois da aula), por tecnologias que, cada vez mais, procuram simular uma sala de aula. Levando em consideração esta nova realidade, observa-se que no desenvolvimento de programas voltados ao ensino a distância verifica-se a necessidade de uma metodologia própria e adequada para o uso de recursos interativos.

Os *chatterbots* são, por definição, agentes inteligentes desenvolvidos para simular uma conversa através da troca de mensagens de texto, semelhantemente aos bate-papos virtuais. Primo [Primo 1998a]

concebe em sua pesquisa científica que o processo de comunicação constitui parte fundamental no aprendizado, trazendo assim toda a problemática inerente as questões lingüísticas para o ambiente dos *chatterbots*. A primeira experiência no desenvolvimento desta categoria de sistema computacional foi Eliza [Laven 2002b]. Desde então surgiram diversas versões e variantes deste sistema, além de novas técnicas de conversação que permitiram o avanço no estudo e implementação. No Brasil, o primeiro *chatterbot* capaz de conversar em português foi Cybelle [Primo 2002], fruto de um projeto de pesquisa com objetivo de estudar as potencialidades e limites da interação homem-máquina. Hoje, *chatterbots* são utilizados em diversas aplicações educacionais e comerciais.

Para simular uma conversação convincente, os programadores de *chatterbots* geralmente utilizam uma técnica de inteligência artificial chamada Processamento de Linguagem Natural (*Natural Language Processing – NLP*) [Russel 1995], [Agustini 1995]. Devido a estas características, *chatterbots* representam um grande potencial como agentes pedagógicos¹, pois possuem autonomia e desenvoltura para direcionar o assunto do estudo de forma natural, sem prender-se a respostas fixas e programadas para serem ativadas em determinados momentos, e talvez esta seja a característica que melhor os diferenciem dos agentes pedagógicos comuns. Além disso, o fato de possibilitar que estes sistemas assumam uma personalidade com a qual o estudante possa identificar-se facilmente aproxima o estudante do computador e de seus estudos. Desta forma, o uso de *chatterbots* em ambientes de Educação a Distância é uma área de estudo e pesquisa com muitas potencialidades. Devido, principalmente, ao uso de técnicas de Inteligência Artificial, os Sistemas Tutores Inteligentes são ambientes que se mostram favoráveis para o uso de ferramentas de interação como os *chatterbots*.

3. BonoBOT – Um Chatterbot para um Sistema Tutor Inteligente

Esta seção tem como objetivo expor um modelo genérico de um *chatterbot* para interação com usuários, mais especificamente para apoiar um STI – o Semeai [Geyer 2001]. O processo de interação usuário-robô se efetiva durante seu processo de interação com o tutor, proporcionando uma interação menos formal e mais próxima da linguagem natural, enriquecendo assim o STI. O diferencial deste reside no fato de permitir uma interatividade natural entre o estudante, auxiliando no processo de ensino e aprendizagem, tal como o papel de um agente pedagógico, inserido num ambiente de ensino e aprendizagem a distância.

3.1. Modelo Funcional do sistema

Chatterbots requerem uma estrutura composta por uma base de conhecimento, onde é armazenada toda a informação pertinente ao domínio de conhecimento do *bot*; uma aplicação responsável por efetuar o tratamento da linguagem natural das mensagens tanto na entrada quanto na saída; além de tecnologias que propiciem a utilização remota paralela do sistema, permitindo que os usuários possam acessá-lo de qualquer local e em qualquer instante.

Para realizar o tratamento das mensagens, optou-se inicialmente por utilizar uma alternativa paralela ao processamento de linguagem natural. Desta forma, o tratamento da linguagem natural é realizado com a utilização de arquivos de definição em XML (*eXtesible Markup Language*) [W3C 2002] como um padrão para representação e troca de dados na Internet. Esta técnica já vem sendo utilizada com sucesso no *chatterbot* ALICE [Wallace 2002]. No contexto deste trabalho, o documento XML contém a formatação das mensagens de entrada (recebidas pelo usuário) e saída (*feedback* ao usuário), ou seja, máscaras com a formatação do texto de forma a permitir que as informações extraídas do banco de dados possam ser expressas em uma linguagem natural. Além disso, contém mensagens de resposta padrão para perguntas freqüentes e não constantes na base de conhecimento, como saudações, por exemplo. O arquivo XML é interpretado por um *parser* (interpretador). Este é responsável por verificar no arquivo de definições qual máscara aplica-se à mensagem recebida do usuário, pesquisar, através das palavras-chave fornecidas pelas definições, a informação requerida no banco de dados e utilizar o resultado da pesquisa na máscara de resposta. O interpretador identifica o comportamento relacionado a cada definição de acordo com as *tags* (marcações).

No BonoBOT (também denominado *bot* neste trabalho), a interação do sistema com o usuário dá-se através da janela de um navegador (*browser*) padrão, permitindo a mobilidade necessária para a utilização do mesmo, acompanhando a proposta do Semeai. A figura 1 apresenta uma arquitetura mais

¹ Entidades que auxiliam o aluno em ambientes educacionais, realizando a função de tutoria, segundo a concepção pedagógica do sistema.

detalhada do sistema. Como se percebe através desta estrutura, o processamento de linguagem natural é substituído por um interpretador de XML desenvolvido em Java, onde serão implementadas máscaras para a interpretação das mensagens e a formatação do conhecimento, permitindo o fornecimento de um *feedback* natural.

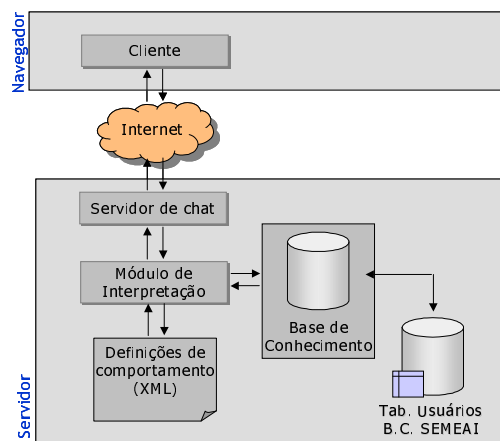


Figura 1. Modelo Funcional do Sistema

O Servidor de *Chat* responde pelo processo de envio e recebimento de mensagens. O Módulo de Interpretação realiza toda a interpretação das mensagens, desde a comparação da mensagem de entrada (enviada pelo usuário) com o conteúdo do arquivo de definições em XML, efetuando as pesquisas necessárias ao banco de dados, até a formatação do resultado obtido em uma nova mensagem seguindo a formatação provida pelo arquivo XML. Para exemplificar o funcionamento do Módulo de Interpretação, consideremos a seguinte mensagem enviada por um usuário em interação com o *chatterbot* - *Usuário*: "Olá, como vai?". Consideremos também o seguinte trecho do arquivo XML de definições:

```
<mensagem> como vai </mensagem>
<retorno> Eu vou bem, e você? </retorno>
<mensagem> fale-me sobre <objeto/> </mensagem>
<retorno><objeto/> é <conceito/>. </retorno>
```

Ao receber a mensagem, o Módulo de Interpretação deverá comparar a *string* do usuário às palavras-chave definidas entre as *tags* <pergunta> e </pergunta>, e retornar a resposta definida para a mensagem de entrada definida entre as *tags* <resposta> e </resposta>. Desta forma, para a mensagem de entrada indicada no exemplo, o *chatterbot* devolverá ao usuário o seguinte *feedback*:

```
Usuário: "Olá, como vai?"
Chatterbot: "Eu vou bem, e você?"
```

Da mesma forma, para uma mensagem que necessite pesquisa no banco de dados, como por exemplo a solicitação de uma explicação sobre um determinado assunto, poderemos ter o seguinte diálogo:

```
Usuário: "Por favor, fale -me sobre inteligência artificial."
```

```
Chatterbot: "Inteligência Artificial é o estudo que busca possibilitar aos computadores executar tarefas tipicamente humanas, como solucionar um problema de forma eficiente ou mover-se de forma autônoma."
```

No caso acima, a *tag* <objeto/> informa ao Módulo de Interpretação que ele deve procurar esta informação na base de conhecimento e que o resultado da pesquisa deverá trazer como resultado o conceito, que será utilizado no *feedback* através da *tag* <conceito/>.

3.2. Base de Conhecimento

Para a representação do conhecimento do *chatterbot*, optou-se pela utilização da técnica de *Frames* [Russel 1995]. Esta abordagem apresenta-se como melhor solução para aplicação neste sistema, especialmente devido à sua similaridade com o paradigma de programação orientada a objetos. Além disso, as informações representadas em *frames* podem ser facilmente associadas umas às outras, gerando uma árvore de inter-relacionamento que permitirá ao sistema um gerenciamento eficiente do conhecimento. Os dados relativos ao domínio de conhecimento do sistema serão armazenados em um

banco de dados relacional. Desta forma, o conhecimento do *chatbot* é representado através de quatro *frames* básicos, que podem ser relacionados entre si, possibilitando ao sistema manter e manipular o conhecimento a respeito do domínio. A figura 2 mostra os *frames* básicos utilizados.

Área de Estudo	Pessoa	Notícia	Ajuda
Descrição Conceito Links Abreviaturas	Nome Apelido Sexo E-mail Última Interação	Título Informação Data Validade	Recurso Descrição Dicas

Figura 2 . Frames para representação do conhecimento

- **Área de Estudo:** refere-se a qualquer assunto passível de estudo. O nível de generalização (mais genérico ou mais específico) é dado através da hierarquia proporcionada pelos relacionamentos. São itens desta classe: descrição: o nome da área de estudo; conceito: o conceito relativo a esta área; links: dicas de links onde o estudante poderá obter mais informações sobre o assunto; abreviaturas: vícios de linguagem relacionados a esta área, como abreviações e símbolos frequentemente utilizados para referenciá-la.
- **Pessoa:** esta classe representa o papel do usuário no sistema, contendo informações pessoais e a respeito de sua interação com o *chatbot*. Contém como itens: nome: nome completo do usuário; apelido: apelido comum do mesmo; sexo: identifica o sexo masculino ou feminino; e-mail: endereço de correio eletrônico; última interação: data da última interação do usuário no sistema, permitindo uma saudação mais adequada ou a confirmação do usuário como novo interlocutor;
- **Notícia:** esta classe representará informações sobre notícias relacionadas a determinadas áreas de estudo ou informações generalizadas. São itens desta classe: título: o título da notícia; informação: conteúdo da notícia; links: dicas de endereços eletrônicos onde poderão ser encontradas maiores informações sobre o evento; validade: prazo de validade para a notícia;
- **Ajuda:** esta classe representará informações de ajuda para o Semeai, tais como: recurso: o item de ajuda; descrição: descreve sobre o item, seu objetivo e utilidade; dicas: dicas para melhor utilizar o recurso;

Toda a hierarquia entre as classes definidas acima acontece através das relações entre elas. O armazenamento do conhecimento do sistema será realizado em tabelas de um banco de dados relacional de acordo com os *frames* estabelecidos acima. O escopo de conhecimento do *chatbot* deve limitar-se a informações sobre o projeto (fornecidas diretamente do arquivo de definições em XML) e o conteúdo relacionado às áreas de estudo que integram o Semeai.

3.3. Aquisição de conhecimento

A aquisição de conhecimento em um sistema *chatbot* tem sido tema de diversas pesquisas, tais como os trabalhos de [Primo 2000], [PixelBot 2002] e [Wallace 2002]. Os autores admitem que seus sistemas não aprendem através das interações e que o conhecimento do *chatbot* é ampliado através da edição direta da base de conhecimento, geralmente baseado nos estudos dos registros de interações com usuários, atualizando ou incluindo os assuntos mais abordados pelos interlocutores. Durante o projeto deste sistema, foram concebidas duas idéias para a aquisição de conhecimento do sistema: a primeira, à qual chamaremos **aquisição natural de conhecimento**, beneficia-se do processo de interação em linguagem natural do *chatbot* para inserir novos conhecimentos em sua base de dados e a segunda, à qual chamaremos **aquisição impositiva de conhecimento**, permite a inserção de novos conteúdos através de um editor com permissão de inclusão direta no banco de dados.

Aquisição Natural de Conhecimento - uma vez que o *chatbot* visa trazer ao usuário o conforto da interação pela linguagem natural, também deve-se considerar a aquisição de conhecimento como parte deste processo. No contexto em que este trabalho está inserido, o problema da confiabilidade da fonte de informação pode ser solucionado através da autenticação do usuário, dando poder de “ensino” ao *chatbot* apenas para usuários cadastrados no STI com *status* de “professores”. Esta idéia vem ao encontro a um comportamento análogo ao dos seres humanos, onde se reconhecem como fontes confiáveis de conhecimento os pais ou professores. A aquisição de conhecimento natural acontecerá quando o usuário informar ao *chatbot* uma mensagem contendo palavras-chave que indiquem ao sistema a intenção de ensino de um novo conceito (por exemplo, “Vamos aprender um novo assunto?”). Neste momento, o Módulo de Interação realizará a autenticação do usuário, retornando *feedback* positivo

no caso de sucesso na identificação do professor (“Claro! O que vamos aprender hoje? Um novo conceito ou uma nova notícia?”), ou negativo, no caso de não-permissão (“Obrigado, mas devo aprender apenas com professores.”). Após a autenticação, um novo objeto do tipo ‘Área de Estudo’ ou notícia (de acordo com a solicitação do usuário) é instanciado na memória, e através de perguntas realizadas pelo *chatterbot* o professor informa os dados que comporão o novo assunto. A gravação da nova instância é realizada no momento da confirmação dos dados por parte do usuário.

Aquisição Impositiva de Conhecimento - Auxiliando o professor em sua tarefa de alimentação da base de conhecimento do sistema, este projeto contempla a alteração do módulo de inserção de conteúdo do Semeai, adaptado-o para a inclusão das informações nas duas bases de dados: a do STI e a do *chatterbot*. Desta forma, evita-se o que o professor tenha mais trabalho em incluir duplamente as informações em cada um dos bancos de dados. Este processo de inclusão nas duas bases é possível porque existem semelhanças estruturais no domínio de conhecimento do *chatterbot* e do Semeai. A diferença reside no fato de que entre os *frames* utilizados para a representação do conhecimento no *chatterbot* há uma simplificação para as estruturas ‘Áreas’, ‘Disciplinas’ e ‘Tópicos’ da base de conhecimento do Semeai em um único *frame*, denominado ‘Área de Estudo’ no *chatterbot*.

3.4. Marcações de Definição

As marcações XML abaixo foram sugeridas para a manipulação do conteúdo do arquivo de comportamento e dos dados do sistema durante a etapa de projeto:

- `<bonobot>` e `</bonobot>` - denotam, respectivamente, a abertura e o fechamento de um arquivo de definições em XML para *chatterbot*;
- `<categoria>` e `</categoria>` - denotam, respectivamente, a abertura e o fechamento de um bloco de mensagens e retornos similares. Mensagens que recebam a mesma resposta e diversas respostas para a(s) pergunta(s) especificadas;
- `<mensagem>` e `</mensagem>` - marcam o início e o fim de uma *string* para comparação com a mensagem enviada para o usuário;
- `<retorno>` e `</retorno>` - indicam o início e fim de um padrão para *feedback*, podendo conter uma *string* simples (no caso de respostas padrões) ou uma combinação com o uso de outras *tags* que indiquem a necessidade de consulta ao banco de dados;
- `<objeto/>` máscara para uma expressão a ser pesquisada na base de conhecimento, podendo ser uma descrição ou abreviatura de uma Área de Estudo, um nome de recurso da ajuda ou um título de notícia;
- `<conceito/>` denota a posição na *string* onde deverá ser aplicado o resultado da pesquisa pelo conceito de uma área de estudo;
- `<autentica/>` marca a necessidade de autenticação do usuário como professor no STI. Será utilizado nos casos de mensagens que indiquem a intenção de ‘ensinar’ ao *chatterbot* um novo conceito;
- `<autenticacao>` e `</autenticacao>` - denotam a abertura e fechamento de um bloco de demarcações que só podem ser utilizados através de autenticação do usuário como professor do STI;
- `<aprendizagem>` e `</aprendizagem>` - indicam um bloco de aprendizagem de uma nova Área de Estudo ou de um novo conceito;
- `<bonobot>` e `</bonobot>` - denotam, respectivamente, a abertura e o fechamento de um arquivo de definições em XML para *chatterbot*;
- `<personalidade/>` declaração de características armazenadas como constantes no *chatterbot*, utilizadas para a definição de personalidade do robô, sendo constituído por dois atributos obrigatórios: atributo – identifica a característica; valor – define o valor da característica. Exemplo: `<personalidade atributo="idade" valor="dez meses"/>`;
- `<bot/>` marca o ponto onde deve ser inserido o valor da personalidade identificado pelo atributo. Pode ser utilizado dentro das marcações de mensagem e retorno;
- * (asterisco) máscara para mensagens do usuário onde qualquer expressão informada naquela posição será aceita. Dependendo das marcações do retorno, o valor obtido através desta máscara pode ser utilizado na resposta ou como informação a ser pesquisada na base de dados. Exemplo: `<mensagem>Você gosta de *? </mensagem> <retorno>Eu poderia gostar de * se fosse humano.</retorno>`;
- `<conceito/>` denota a posição na *string* de retorno onde será aplicado o resultado da pesquisa pelo conceito de uma área de estudo;

- <links/> marca a posição na *string* de retorno onde será aplicado o resultado da pesquisa pelos *links* de páginas de Internet de uma área de estudo;
- <ajuda/> denota a posição na *string* de retorno onde será aplicado o resultado da pesquisa pela descrição de ajuda de um recurso do STI;
- <dica_ajuda/> marca a posição na *string* de retorno onde será aplicado o resultado da pesquisa pela dica de ajuda de um recurso do STI;
- <noticia/> define a posição na *string* de retorno onde será aplicado o resultado da pesquisa por uma notícia;
- <relacionamento/> marca a posição na *string* de retorno onde será aplicado o resultado da pesquisa por um relacionamento. Possui um atributo obrigatório: *item* – identifica o tipo de item de relacionamento, podendo ter um dos seguintes valores: “pessoa”, “área” ou “notícia”. Exemplo: <relacionamento *item*=’área’/>. O resultado desta marcação é um dos relacionamentos entre a área identificada pela máscara e um item do tipo indicado no atributo *item*.

4. Implementação do Modelo

A escolha da plataforma para desenvolvimento e implementação do protótipo do BonoBOT baseou-se fundamentalmente em três aspectos: a troca de mensagens a distância, via Internet, a utilização de *software* livre, devido ao baixo custo, e a facilidade de integração com o ambiente do Semeai. Sob este enfoque são usadas tecnologias como *Java* (linguagem de programação) [Sun 2002b], *JSP* (*Java Server Pages*)² [Sun 2002d] e *Apache Jakarta Tomcat* (servidor de páginas *JSP* e *servlets*) [Sun 2002c]. [Sun 2002b]. Os dados utilizados para o funcionamento do sistema são divididos em dois grupos. Aqueles que dão a fundamentação da “fala”, composto das máscaras para formatação dos textos do robô, estão armazenados em um arquivo que utiliza marcações XML desenvolvidas especialmente para esta aplicação. Estas informações são armazenadas no arquivo *Arquivo de Definições de Comportamento*, conforme apresentado no item 3.1. Para a interpretação das marcações o sistema faz uso de *JAXP* (*Java API for XML Processing*). Os dados que representam o conhecimento são armazenados em tabelas especiais que compartilham a base de dados do Semeai. Por se tratar de uma aplicação em que o cliente acessa o serviço a distância, intermediada pela Internet, a interface desenvolvida foi voltada a usuários que utilizem navegadores de páginas da Internet, tais como o *Internet Explorer*, *Netscape* ou *Mozilla*.

Para melhor representar a implementação do projeto e manter fidelidade ao modelo proposto, divide-se a implementação em três grandes partes: a interpretação do arquivo de definições de comportamento e seu armazenamento em memória, o processo de bate-papo (interpretação das mensagens do usuário e geração de respostas) e o processo de aquisição de conhecimento.

4.1. Módulo Interpretador das Definições de Comportamento

Para que as mensagens enviadas pelo usuário sejam compreendidas, o BonoBOT realiza comparações com as definições do arquivo de comportamento, em XML. Estas definições são carregadas para a memória durante a ativação do *webserver*, quando o *servlet* BonoBot é disparado. Na inicialização do *servlet* é definida a classe onde está implementada a manipulação do XML, a classe que manipula os erros no processo de interpretação e o caminho do arquivo de definições do comportamento. Após a identificação do arquivo são feitas chamadas que realizam o processamento (*parsing*) através da classe *ManipulaConteudo*. Essa classe é responsável por interpretar as marcações do arquivo XML. Para cada categoria definida no XML, é instanciado um objeto do tipo *Categoria* que a classe *ManipulaConteudo* utiliza para inserir os dados interpretados. As mensagens encontradas dentro desta categoria são incluídas em um vetor de *strings*. Os retornos são armazenados em instâncias de objetos do tipo *Retorno*, que são colocados em outro vetor. Quando a leitura da categoria é concluída, a instância é passada para a classe *Comportamento*, onde é inserida em um vetor, que durante a troca de mensagens com o usuário será percorrida para comparação de *strings*. É durante o processamento do XML que as marcações <bot/> são identificadas e substituídas pelo conteúdo da respectiva *tag* <personalidade/>. Esta substituição é possível porque as informações de personalidade são constantes e não variam no decorrer da interação com o usuário. Após a interpretação das definições de comportamento o *servlet* instancia estes dados em uma variável de contexto do *web application*, onde estará disponível para todas as sessões *web*. Neste momento *chatbot* está pronto para interagir com os usuários.

² De forma simplificada, tecnologia que aproveita todos os recursos da linguagem *Java* para geração de páginas dinâmicas. O código *JSP* é armazenado em arquivos com a terminação *.jsp*.

4.2. Módulo de Troca de Mensagens com o Usuário

A troca de mensagem com o usuário é realizada em duas etapas distintas. A primeira consiste na obtenção da mensagem enviada pelo usuário e sua comparação com as definições de comportamento já carregadas na memória do servidor. A segunda etapa realiza a escolha do retorno apropriado, fazendo as substituições de texto nos locais onde são aplicadas as máscaras, realizando a consulta ao banco de dados quando necessário e formatando a mensagem de resposta. O acesso ao BonoBOT é realizado através da página inicial do projeto, que pode ser acessada através do *link* existente na barra de tarefas do Semeai. Quando o usuário clica sobre o *link* para iniciar a interação com o *chatterbot* a página *principal.jsp* é chamada, carregando na seção *web* do usuário uma instância da classe *SessaoChat*. Esta classe é responsável por controlar toda a troca de mensagens entre o robô e o interlocutor. O *chat* é então iniciado em uma janela dividida em dois *frames*: o *frame* superior (*chat.jsp*), onde é exibido o diálogo, e o *frame* inferior (*controle.jsp*), com um formulário onde o usuário deverá digitar suas mensagens. Quando o usuário envia sua mensagem, os dados são submetidos à página de diálogo, onde o método *recebeEntrada* da classe *SessaoChat* é acionado. Este método adiciona a mensagem no histórico e inicia o processo de tratamento da *string*, chamando o método *procuraMensagem* na classe *Comportamento* instanciada na inicialização do serviço.

O método *procuraMensagem* realiza a identificação da mensagem do usuário, que consiste na comparação da *string* com as definições de mensagens de cada uma das categorias armazenadas no vetor de comportamento. A cada comparação, o método verifica os casos de mensagens previstas nas definições de comportamento onde possa haver uma máscara, sinalizada na definição por um * (asterisco). Nestes casos, esta definição é dividida em *substrings*, que são comparadas com a mensagem enviada pelo interlocutor. Quando a mensagem do usuário é identificada, o método *getRetorno* do objeto *Categoria* é acionado, solicitando um retorno. Este método faz a escolha randômica entre as opções de retorno disponíveis para a categoria, realizando consultas ao banco de dados de acordo com a necessidade, ou seja, quando dentro da mensagem de retorno é encontrada uma das marcações apresentadas no item 3.1. Cada marcação possui um método próprio para obter o resultado da pesquisa no banco de dados. Se a consulta ao banco de dados não retornar nenhum resultado, a comparação da mensagem do usuário com as definições é continuada, e todo o processo de obtenção do retorno é reiniciado. Se ao final das comparações o robô não encontrar um retorno que se aplique à mensagem enviada pelo interlocutor, então ele refaz o método *procuraMensagem* passando a palavra reservada “%*incompreensao*”, que deve estar definida em uma categoria a parte, com diferentes possibilidades de retorno que façam com que o robô informe ao usuário que ele não possui uma resposta mais adequada. O processo de troca de mensagem é finalizado quando o retorno é incluído no histórico do diálogo, que é atualizado na janela de *chat* do usuário. A figura 3 apresenta a página inicial do BonoBOT e a interface de bate-bapo.

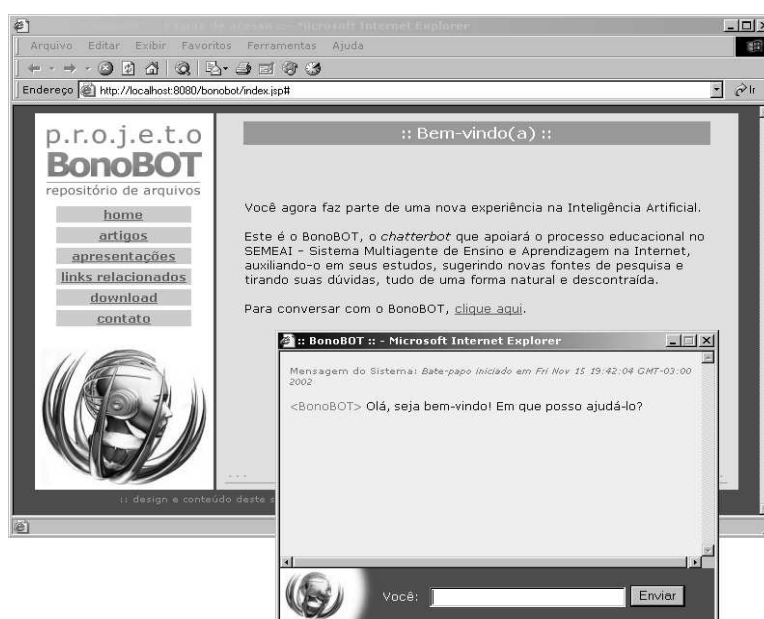


Figura 3. Interface do BonoBOT

4.3. Módulo de Aquisição de Conhecimento

Conforme o modelo, o sistema possui duas formas de aquisição de conhecimento: a primeira, natural, pela qual o conhecimento seria adquirido através do bate-papo com o professor, e a segunda, impositiva, na qual os dados seriam inseridos diretamente na base de conhecimento (BC) do robô. O protótipo contempla somente a segunda opção. Desta forma, sugere-se o desenvolvimento deste recurso adicional como trabalho futuro. A aquisição impositiva de conhecimento é provida pelas classes *Area*, *Noticia*, *Ajuda* e *TipoRelacionamento*, na qual são implementados os métodos para armazenamento de dados na base de conhecimento do sistema. Os métodos destas classes são acionados em dois momentos: (1) durante a interação do professor com o ambiente do Semeai, no momento em que este cadastra conhecimento no STI, podendo o usuário optar para que estas informações sejam também utilizadas pelo robô; (2) Através de um cadastro desenvolvido especialmente para inclusão de conteúdo na base de conhecimento.

Com relação a interface, além das citadas, o sistema possui duas interfaces diferenciadas: a de interação em linguagem natural (bate-papo), aberta a todos os usuários do Semeai, e o cadastramento de conhecimento, restrito apenas a professores cadastrados. Nesta última, os professores cadastrados no Semeai tem acesso também ao cadastro de conhecimento do BonoBOT. Assim, a base de conhecimento do *chatterbot* pode ser consultada, alterada e ampliada, dando ao robô a flexibilidade de utilizar conceitos e informações não existentes no STI.

5. Conclusões Finais

Este trabalho apresentou o BonoBOT, um robô para conversação na Internet com características de agente pedagógico, cujo principal objetivo é auxiliar os usuários de um STI – o Semeai. O sistema está completamente implementado, inserido ao Semeai e em fase de avaliação e validação. Para dar suporte à linguagem natural, o Processamento da Linguagem Natural foi substituído por uma técnica de tratamento da linguagem utilizando-se definições em XML que, embora de forma limitada, dá suporte adequado às necessidades do robô. A aquisição de conhecimento impositiva foi priorizada nesta implementação, através do desenvolvimento de uma classe que dá suporte para inclusão, alteração e exclusão na base de dados do *chatterbot*. Percebe-se como principal contribuição deste trabalho a proposta de um *chatterbot* como ferramenta de interação em um STI – o Semeai. Desta forma, acredita-se que o uso destas ferramentas em ambientes virtuais possam se transformar em poderosos recursos de educação. Para isto, é importante que o *bot* seja usado de forma adequada, sob uma técnica pedagógica correta, com elementos internos motivadores e recursos didáticos capazes de fazer despertar nos alunos e nos professores a aceitação desse novo paradigma, além da capacidade técnica inteligente de estabelecer uma relação de incentivo à participação do aluno e auxílio ao professor. Trabalhos como o de [Heredia 2002] confirmam esta premissa.

Ainda há muito a realizar neste trabalho, tais como: (1) implementação de um mecanismo de aquisição natural de conhecimento, tendo-se como ponto de partida as idéias propostas na modelagem do robô; (2) um estudo estrutural aprofundado do diálogo na língua portuguesa para a ampliação e otimização do arquivo de definições de comportamento do *chatterbot*; (3) a implementação do Processamento de Linguagem Natural em todos os níveis de interpretação; (4) desenvolvimento de um mecanismo de tradução para regionalismos e linguagem informal utilizada pelos internautas no bate-papo; (5) ampliação do escopo de conhecimentos gerais do *chatterbot*, o que lhe dará maior desenvoltura no diálogo com interlocutores de interesses diferentes. Dentro deste contexto, observa-se a importância deste trabalho, que além de propor um *chatterbot* para ambientes virtuais de aprendizagem, explorando as vantagens da comunicação através da linguagem natural, também contribui na produção de material científico sobre o assunto, abrindo um campo de pesquisa bastante importante academicamente entre as universidades que participam deste trabalho.

Agradecimentos

Este trabalho foi parcialmente financiado pela FAPERGS e pelo PAEP - UNILASALLE. Os autores agradecem também a Wagner Nobres Soares, que colaborou com vários aspectos relacionados à concepção e implementação do sistema.

Referências

- Agustini, Alexandre. (1995) Estudo Inicial sobre o Processamento da Linguagem Natural. Porto Alegre: PUCRS. Trabalho Individual I.
- Belloni, Maria Luiza (1999) Educação a distância. Campinas: Autores Associados.
- Geyer, C. F. R., Ferrari, D. N., et al. (2001) SEMEAI - SistEma Multiagente de Ensino e Aprendizagem na Internet. In: XII SBIE 2001 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. Vitória, ES, Brasil.
- Giraffa, Lúcia M. M.(1999) Uma Arquitetura de Tutor Utilizando Estados Mentais. Porto Alegre: CPGCC da UFRGS. Tese de Doutorado.
- Heredia, E., Omar, N., Mendonça, J. C. (2002) um agente de comunicação em linguagem natural no contexto de um sistema multiagentes orientado à tutoria inteligente na WWW. In: XIII SBIE 2002 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. São Leopoldo, RS, Brasil.
- Horowitz, E. (1998) Migrating Software to the World Wide Web. In: IEEE Software, 15(3). pp.18-21.
- Laven, Simon. (2002a) Classic Chatterbots – Eliza. The Simon Laven Page. Disponível em <<http://www.spaceports.com/~sjlaven/eliza.html>>.
- Laven, Simon. (2002b) AIML Bots. The Simon Laven Page. Disponível em <<http://www.spaceports.com/~sjlaven/aiml.html>>.
- Lévy, Pierre.(1999a) Cibercultura. São Paulo: Ed. 34. 264p.
- Lévy, Pierre.(1999b) O que é virtual. São Paulo: Ed. 34.
- Lippmann, Andrew. (1998) O arquiteto do futuro. Revista Meio & Mensagem, São Paulo, n. 792, janeiro.
- Lucena, C. J. P.; et al. (1999) AulaNet: Ajudando Professores a Fazerem seu Dever de Casa. In: EDUCAÇÃO E APRENDIZAGEM NA SOCIEDADE DA INFORMAÇÃO, v. 1, p. 105-117, 1999, Rio de Janeiro, RJ. Anais... Rio de Janeiro: SBC.
- Maçada, Débora Laurino; et al. (1999) Aprendizagem Cooperativa em Ambientes Telemáticos. Revista Informática na Educação: teoria e prática. v.2, n 1, p. 19 – 28, maio, 1999.
- Moran, José M. (2000) Ensino e Aprendizagem Inovadores com Tecnologias. Revista Informática na Educação: Teoria & Prática – PGIE – UFRGS, pp.137 – 144.
- Oliveira, Alex Francisco Gonçalves.(2001) Agente Tutor Inteligente em um Ambiente Multiagente para Ensino a Distância. Canoas: Centro Universitário La Salle.
- Pixelbot: Virtus – Laboratório de Hiperídia (2002) Disponível em: <<http://150.161.189.220/pixel/index.htm>>.
- Primo, Alex Fernando Teixeira. (1998a) Interação Mútua e Interação Reativa: uma proposta de estudo. Disponível em <<http://usr.psyco.ufrgs.br/~aprimo/pb/esprialpb.htm>>..
- Primo, Alex F. T. (1998b) O Uso de Chatterbots em Educação a Distância. In: IV Workshop de Informática na Educação. Porto Alegre: UFRGS, 12p.
- Primo, Alex F. T., Coelho, Luciano Roth. (2000) Júnior, um chatterbot para educação a distância. In: RIBIE 2000 – Congresso Ibero-Americano de Informática na Educação. Viña del Mar, Chile.
- Primo, Alex F. T., Coelho, Luciano Roth. (2002). Comunicação e inteligência artificial: interagindo com a robô de conversação Cybelle. In: Estratégias e Culturas da Comunicação. Brasília: UnB pp. 83-106.
- REL - Rede Escolar Livre RS (2001). Explorando a educação a distância. Disponível em <http://www.redeescolarlivre.rs.gov.br/>
- Rocha, E. V. (2002) O ambiente TelEduc para educação a distância baseada na web: Princípios, funcionalidades e perspectivas de desenvolvimento.", em MORAES, M. C. (Org.) Educação a distância: Fundamentos e práticas. Campinas, SP: UNICAMP/NIED. cap. 11, p. 197-212
- Russel, Stuart J., Norwig, Peter. (1995) Artificial Intelligence: a modern approach. Rio de Janeiro: Prentice-Hall do Brasil,932p.
- Steuer, Jonathan.(1992) Defining virtual reality: dimensions determining telepresence. Journal of Communication. n.42, p.72-93.
- Sun Microsystems, Inc. (2002a) The Java Tutorial. Disponível em <http://java.sun.com>.
- Sun Microsystems, Inc. (2002b) Tomcat@Jakarta. Disponível em <http://java.sun.com/products/jsp/tomcat>.
- Sun Microsystems, Inc. (2002c) Core servlets and java server pages. Disponível em <http://java.sun.com>.
- Sun Microsystems, Inc. (2002d) Java Server Pages. <<http://java.sun.com/products/jsp>>.
- Turing, A.M.(1950) Computer Machinery and Intelligence. Mind. Vol. 9, No. 236, pp. 433-460.
- Wallace (2002), Richard S. How it all started. Disponível em: <http://alicebot.org/articles/wallace/start.html>>.
- WebCT. (2002) <http://www.webct.com>.
- W3C - World Wide Web Consortium.(2002) <<http://www.w3.org>>.
- Wooldridge, Michael, Jennings , Nick. (1994) Agents Theories, Architecture and Language: A Survey In: ECAI - Workshop onAgent Theories, Architectures and Languages, Amsterdam. pp.1-32.