

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC**  
**CURSO DE PÓS-GRADUAÇÃO ESPECIALIZAÇÃO EM MBA E**  
**BANCO DE DADOS**

**FABRIZIO COLOMBO MACHADO**

**INTEGRANDO *CHATTERBOT* E AGENTE ANIMADO DE INTERFACE**  
**EM UM AMBIENTE VIRTUAL DE APRENDIZAGEM**

**CRICIÚMA, JUNHO DE 2005**

**FABRIZIO COLOMBO MACHADO**

**INTEGRANDO *CHATTERBOT* E AGENTE ANIMADO DE INTERFACE  
EM UM AMBIENTE VIRTUAL DE APRENDIZAGEM**

Monografia apresentada à Diretoria de Pós-graduação da Universidade do Extremo Sul Catarinense- UNESC, para a obtenção do título de especialista em MBA em Banco de Dados.

Orientadora: Prof<sup>a</sup>.(MSc). Leila Lais Gonçalves

**CRICIÚMA, JUNHO DE 2005**

## **AGRADECIMENTO**

*Não poderia deixar de agradecer a minha professora, orientadora e amiga Leila, minha família e especialmente minha namorada Aline pela compreensão e estímulo para a conclusão de mais um projeto de vida. Apesar de todas dificuldades, estas pessoas queridas, não deixaram de me apoiar num momento tão importante. Galera, obrigado pela força!*

**“A única coisa que se consegue ao resolver um problema é voltar a situação anterior. Portanto o verdadeiro progresso não é obtido solucionando-se problemas, mas sabendo-se usar as oportunidades criadas pela necessidade de resolvê-los.”**

**Drucker, 1995.**

## RESUMO

O uso da *Internet* na Educação vem sendo acompanhado de muitos estudos e pesquisas sobre as possibilidades, recursos e ferramentas *Web* para educação a distância. O número de instituições de ensino que, de alguma forma, vêm adotando Ambientes Virtuais em sua atuação pedagógica é expressivo. Com essa medida, surgem dificuldades inerentes à inexperiência, ou ainda, pouca prática de grande parte dos usuários em relação às funcionalidades presentes nas ferramentas e do uso da própria *Internet*. A presente pesquisa apresenta uma solução para minimizar estas dificuldades, apoiada na comunicação usuário e aplicação. Esta alternativa propõe uma integração entre agentes animados de interface e agentes de conversação (*chatbots*) no ambiente virtual *Learnloop*. O papel do *chatbot* é esclarecer dúvidas do usuário quanto à utilização das ferramentas do ambiente por meio de diálogos em linguagem natural. Já o agente animado de interface possibilita a aplicação de ações humanas no processo de comunicação como gestos, fala, feições e movimentos. Dessa forma, busca-se fornecer um canal intuitivo e, ao mesmo tempo, motivador no processo de interação entre o usuário e o ambiente.

**Palavras-chave:** *chatbot*; agentes animados de interface; *aiml*; ambiente virtual de aprendizagem

## LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura básica de um <i>chatbot</i> .....	18
Figura 2 – Assistente do <i>Office</i> – Janela de configuração.....	30
Figura 3 – Personagens padrão dos agentes da <i>Microsoft</i> .....	30
Figura 4 – Personagens personalizados de agentes animados.....	31
Figura 5 – Relação entre qualidade e custo de suporte.....	35
Figura 6 – Funil de Serviços.....	36
Figura 7 – Arquitetura do protótipo.....	38
Figura 8 – Sentenças em linguagem natural.....	42
Figura 9 – Sentenças formatadas para as categorias da linguagem <i>AIML</i> .....	42
Figura 10 – Arquivo de configuração do contêiner <i>web</i> .....	44
Figura 11 – Arquivo de configuração dos <i>bots</i> .....	45
Figura 12 – Arquivos dos <i>logs</i> de conversão.....	46
Figura 13 – Arquivo de configuração da persistência dos <i>logs</i> de conversação...46	
Figura 14 – Interface de interação do usuário.....	48
Figura 15 – <i>Msagent Peddy</i> dando boas vindas.....	49
Figura 16 – Funcionamento do agente da <i>Microsoft</i> .....	50
Figura 17 – Formulário de configuração do <i>bot</i> .....	55
Figura 18 – Arquivos de <i>log</i> de conversação.....	56
Figura 19 – Tela de administração das mensagens de ajuda.....	57
Figura 20 – Administração da apresentação da ajuda <i>online</i> (Área Pessoal).....	58
Figura 21 – Fluxo de interação: Usuário com Classe de Comunicação.....	59
Figura 22 – Parte do Arquivo <i>AIML</i> .....	59
Figura 23 – Fluxo de interação: Classe de Comunicação com o <i>Chatbot</i> .....	60
Figura 24 – Fluxo de interação: Classe de Comunicação com o Agente Animado.....	60
Figura 25 – Agente atuando como tutor do ambiente virtual.....	61
Figura 26 – Atalho para ocultar o agente de interface.....	62
Figura 27 – Agente de interface em modo de espera.....	62

## LISTA DE TABELAS

Tabela 1 – Propriedades e requisitos comuns a agentes.....	26
--	----

## LISTA DE ABREVIATURAS E SIGLAS

**CGI:** *Common Gateway Interface*

**EAD:** Educação a Distância

**HTML:** *HyperText Markup Language*

**HTTP:** *HyperText Transfer Protocol*

**XML:** *Extensible Markup Language*

**IA:** Inteligência Artificial

**IA-ED:** Inteligência Artificial na Educação

**COM:** *Component Object Model*

**SEAD:** Setor de Educação a Distância

**PHP:** *Personal Home Page tools*

**UNESC:** Universidade do Extremo Sul Catarinense

**SMA:** Sistemas Multiagentes

**IRC:** *Internet Relay Chat*

**AOL:** América On-Line

**NLP:** *Natural Language Processing*

**FAQ's:** *Frequents Asqued Questions*

**A.L.I.C.E.:** *Artificial Language Intelligence*



## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
1.1 Objetivo .....	13
1.2 Estrutura do Trabalho .....	13
<b>2 CHATTERBOT.....</b>	<b>15</b>
2.1 Classificação dos <i>chatterbots</i> quanto a seu uso .....	16
2.2 Arquitetura geral de um <i>Chatterbot</i> .....	17
2.3 <i>Chatterbots</i> e Educação a Distância.....	19
<b>3 LINGUAGEM NATURAL e AIML (Artificial Intelligence Markup Language) .....</b>	<b>21</b>
3.1 Processamento da Linguagem Natural.....	21
3.2 Estado da Arte e Modelo de Comunicação .....	23
3.3 AIML (Artificial Intelligence Markup Language) .....	23
<b>4 AGENTES E SISTEMAS MULTIAGENTES (SMA) .....</b>	<b>25</b>
4.1 Sistemas Multiagentes (SMA).....	25
4.2 Agentes .....	25
4.2.1 Agentes de Interface .....	27
4.2.2 Agentes animados da <i>Microsoft</i> – ( <i>Microsoft Agents</i> ) .....	29
<b>5 DEFINIÇÕES DO PROTÓTIPO.....</b>	<b>32</b>
5.1 Metodologia .....	32
5.2 Motivação (expectativas e benefícios da solução) .....	33
5.3 O Protótipo .....	36
5.3.1 Arquitetura do protótipo.....	37
5.3.1.1 Ambiente Virtual de Aprendizagem – <i>Learnloop</i> .....	38
5.3.1.2 Base de Conhecimento.....	39
5.3.1.3 BOT – Módulo de interpretação .....	42
5.3.1.4 Agente de Interface.....	47
5.3.1.5 Classe de Comunicação .....	51
5.3.1.6 Usuários .....	51
5.4 Funcionamento do Protótipo .....	52
5.4.1 Ambiente do Protótipo.....	52
5.4.2 Recursos disponíveis: .....	54
5.5 Fluxo de uma interação do usuário com o <i>chatterbot</i> .....	58
<b>6 CONCLUSÃO .....</b>	<b>63</b>
<b>REFERÊNCIAS.....</b>	<b>65</b>

## 1 INTRODUÇÃO

O uso da Internet no campo pedagógico introduz novas possibilidades, recursos e ferramentas para Educação a Distância (EaD). Dentre as inovações, tem-se os Ambientes Virtuais de Aprendizagem (AVA) que são, segundo Krüger (1993), locais que propiciam interação, cooperação, análise, interpretação, observação, teste de hipóteses, elaboração e a construção de relações que constituem a construção de novos conhecimentos.

O número de instituições de ensino que, de alguma forma, vêm adotando estes ambientes virtuais em sua atuação pedagógica é expressivo. Estes ambientes, em sua estrutura geral, buscam propiciar diferentes espaços para interação e programação de situações de aprendizagem, onde, na maioria das vezes, o responsável pela elaboração e estruturação destes espaços é o próprio professor. Deste modo, surgem dificuldades inerentes à inexperiência, ou ainda, pouca prática de grande parte dos usuários – professores e alunos, em relação às funcionalidades presentes nestes ambientes.

Outra questão, quanto ao uso dos AVAs, é a relação entre professor e aluno. Na sala de aula presencial o contato físico e o relacionamento baseado na troca de conhecimento e na comunicação direta entre professores e alunos é um fator relevante que estimula e propicia a interação. Já em um ambiente virtual essa interação se dá por meio de tecnologias de cooperação e comunicação. A eficiência, neste contexto, encontra-se na rapidez e agilidade com que a interação é efetuada. No tocante à eficácia, a tecnologia além de disponibilizar situações de interação,

deve disponibilizar no mundo virtual formas de estimular e motivar a comunicação entre os participantes de EaD.

A utilização de elementos característicos da comunicação humana aliada a um conjunto de instruções que facilitem sanar as dúvidas referentes a utilização de um ambiente virtual é uma provável solução para as questões levantadas. Para estes fins é necessário que se utilizem tecnologias que busquem emular movimentos, feições e principalmente o raciocínio humano.

Segundo Sganderla (2003), a inteligência Artificial (IA), neste cenário, serve de suporte a estes sistemas provendo meios para simular um relacionamento mais racional entre o homem e a máquina. A utilização de arquiteturas baseadas em sociedades de aprendizagem onde os princípios dos sistemas multiagentes apresentam um potencial adequado para o ensino cooperativo (WILGES, 2004).

Wilges (2004) afirma ainda, que a utilização de agentes em ambientes virtuais de aprendizagem traz benefícios substanciais no que tange os aspectos pedagógicos, possibilitando interações mais naturais e intensas no relacionamento do usuário com o sistema. Baseado nisso, é necessário à inclusão de agentes inteligentes que simulem a racionalidade, movimento e feições humanas numa interação para tornar as aulas à distância mais atrativas, estimular o aluno e diminuir o impacto do contato pessoal.

Tendo em vista as necessidades e as áreas de conhecimento que estão inseridas, nesta pesquisa é realizado um levantamento bibliográfico sobre a tecnologia de agentes inteligentes de conversação (*chatterbots*) e de agentes animados de interface e linguagem natural. Este estudo embasa a proposta da solução-protótipo apresentada.

A pesquisa busca enfatizar a integração das tecnologias de agente aplicadas na educação a distância, trazendo formas mais intuitivas de interação com o ambiente virtual, bem como, maneiras distintas de armazenamento, busca e recuperação de dados e transformação destes em informação.

## 1.1 Objetivo

Esta pesquisa tem como objetivo principal propor uma aplicação que integre agentes animados de interface e de conversação no processo de comunicação e interação de um ambiente virtual de aprendizagem. Esta ferramenta deve ser capaz de auxiliar os usuários na utilização das ferramentas e funcionalidades do ambiente virtual de maneira mais simples, natural e, por decorrência mais agradável.

Desta forma a pesquisa busca atingir especificamente os objetivos de: investigar as aplicações de agentes inteligentes e suas formas de utilização em ambientes virtuais de aprendizagem; estudar maneiras de persistência e busca em dados de uma base de conhecimento; possibilitar comunicação por meio de linguagem natural entre o ambiente e o usuário; e modelar a aplicação e as configurações de integração ao ambiente virtual - *Learnloop*<sup>1</sup>.

## 1.2 Estrutura do Trabalho

No capítulo dois é apresentado o conceito, arquitetura, classificações e aplicações dos *chatterbots* e a relação deste com EaD. Em seguida no terceiro

---

<sup>1</sup> O *LearnLoop* é um Ambiente Virtual de Aprendizagem *Open Source* distribuído sob licença GNU (GPL), utilizado pela Unesc nos cursos na modalidade de Educação a Distância e na Educação Presencial - nas disciplinas da graduação, pós-graduação e no Colégio de Aplicação (SEAD, 2004).

capítulo são descritos conceitos do processamento de linguagem natural, modelos de comunicação e a linguagem de marcação AIML utilizada pelos *chatebots*. O capítulo quatro expõe conceitos e características de sistemas multiagentes e agentes, principalmente dos agentes animados de interface. No capítulo cinco encontramos a descrição da metodologia, arquitetura, tecnologias, recursos e técnicas utilizadas para a construção do protótipo desta pesquisa.

## 2 CHATTERBOT

Este capítulo conceitua *Chatterbot*, apresentando sua arquitetura, suas classificações e aplicação destes no educação a distância.

De acordo Alice (2004), *chatterbots* são essencialmente sistemas inteligentes que utilizam uma linguagem de marcação para compor uma base de conhecimento e interagir com seres humanos respondendo questões referentes a esta base de conhecimento em linguagem natural.

Os *chatterbots* alcançaram maior destaque, no cenário atual, em função das mudanças que a internet trouxe nas formas de comunicação. Sendo assim, sua utilização também migrou para diversas áreas, entre elas na educação a distância - tudo para facilitar a comunicação entre o homem e a máquina neste tipo de aprendizagem.

Além disso, nas ferramentas de educação a distância o *chatterbot* está sendo utilizado para tornar a comunicação mais natural. Ele pode assumir diversas funções, mas fundamentalmente sua principal característica é a do diálogo com o usuário, ou seja, nas circunstâncias de interações usuário x máquina. Neste caso, em uma interação, sobre um determinado domínio de conhecimento, esses sistemas inteligentes buscam simular o raciocínio humano de uma conversação.

Para Primo (2005), o *chatterbot* pode ser definido como um *bot* (simplificação da palavra inglesa *Robot*) ou um agente, pois simula uma atividade humana para um usuário ou outro programa. Os *chatterbots* se diferenciam dos demais agentes pelo seu uso em conversações com o propósito de simular uma racionalidade humana na conversa com a máquina.

Esta categoria de agente encontra-se com denominações diferentes na literatura, como, por exemplo, Simulador de Conversação, *Verbot*, *ChatBot*, Agente de Conversação. Mas todas identificam um tipo específico de agente que é capaz de “entender”, dentro de certas limitações, a linguagem natural (PRIMO, 2005).

A partir deste capítulo se descreverá algumas características referentes às tecnologias empregadas para o desenvolvimento destes *chatterbots*.

## 2.1 Classificação dos *chatterbots* quanto a seu uso

Os *chatterbots* podem ser classificados conforme o propósito para qual são destinados (LAUREANO, 1999):

- ***Chatterbot de Entretenimento***: tem o propósito de instigar o usuário de forma dinâmica e divertida, demonstrando interesse e bom humor numa interação com o usuário;

- ***Chatterbot de FAQ's (Frequent Asqued Questions)***: seu objetivo é responder a questões sobre um determinado domínio de conhecimento. Uma característica que diferencia dos *chatterbots* de entretenimento é o fato de não estar preocupado em se passar por humano, admitindo ignorância quando não encontra resposta para uma pergunta;

- ***Chatterbot de Suporte***: o que difere este *chatterbot* dos anteriores é a necessidade de uma maior interação com usuário, buscando questionar as necessidades e dificuldades do mesmo para determinada situação;

- ***Chatterbot de Marketing***: como o próprio nome sugere é uma categoria de *chatterbot* que tem a função de demonstrar algum produto. Deve possuir as

características de entretenimento aliadas as características dos *chatterbots* de *FAQ's* para desempenhar bem o seu papel;

- ***Chatterbot de propósito Geral***: segundo Laureano (1999), existe outros *chatterbots* que não se ajustam a nenhuma das categorias anteriores, porque possuem uma filosofia de funcionamento muito peculiar.

Um dos usos mais recentes para *chatterbots* é a inclusão destes em ambientes virtuais para o ensino e entretenimento (DEBRA, 2002). Outros autores também dão força a esta afirmação.

Segundo Coelho (2004), Liane Tarouco (coordenadora do PGIE/UFRGS) em sua palestra apresentada no II *Workshop* em Informática na Educação levantou grandes perspectivas para o futuro da educação a distância, destacando o uso de robôs de conversação em ambientes de ensino. Coelho (2004) ainda acrescenta que os *chatterbots* para este uso podem ser um bom apelo motivacional para incentivar a adoção desta modalidade de educação.

## 2.2 Arquitetura geral de um *Chatterbot*

Conforme Dyck (2005), grande parte dos *chatterbots* desenvolvidos atualmente compartilham de uma arquitetura geral, constituída basicamente por quatro componentes principais: a interface de interação do usuário, os modelos de usuário e de domínio, e o elemento responsável pelo processamento das sentenças.

Com base nestas afirmações, que também são compartilhadas por outros autores, como Laureano (2004), pode-se agrupar alguns componentes em módulos para simplificar o entendimento da arquitetura de um *chatterbot*, de acordo modelo



proposto na Figura 1. A arquitetura a seguir, a qual se baseia esta pesquisa, foi adaptada a partir da representação proposta pelos autores citados anteriormente.

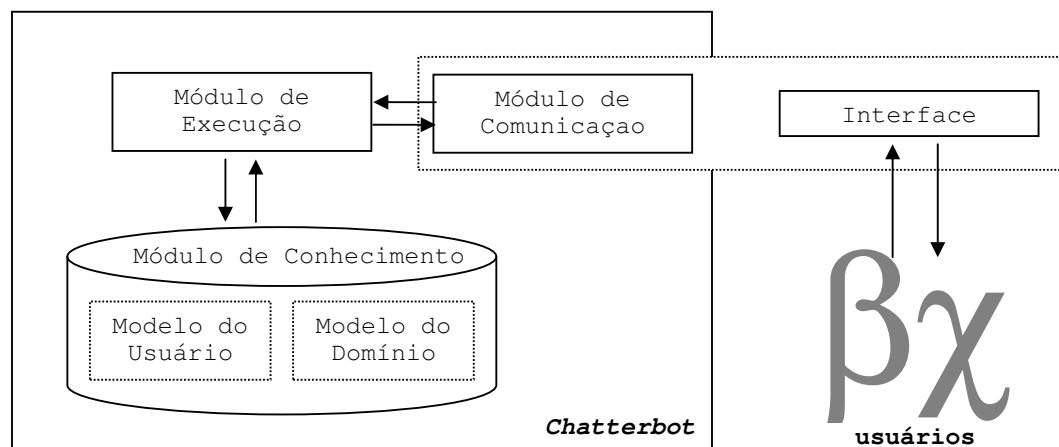


Figura 1 – Arquitetura básica de um *chatterbot* (adaptado de Laureano, 2004).

**Módulo de Comunicação:** módulo responsável em prover a comunicação entre o usuário e o sistema. Por este canal de entrada são recebidas as sentenças em linguagem natural enviadas pelo usuário para serem processadas pelo *software*.

**Módulo de Conhecimento:** agrupa os modelos de usuário responsáveis em persistir as informações sobre o usuário e o contexto da conversação e o modelo de domínio que, por sua vez, transmite as informações sobre um ou até diversos assuntos que o *chatterbot* será apto a conversar.

**Módulo de Execução:** esse é o componente central da arquitetura de um *chatterbot*. Deve ser capaz de processar as sentenças recebidas do usuário utilizando estratégias adequadas de seleção de respostas disponíveis na base de conhecimento do *chatterbot*. Para tanto, *softwares* são implementados com alguma técnica de seleção de respostas como o casamento de padrão, rede de ativação de

respostas, raciocínio baseado em casos (RBC) ou a lógica *fuzzy* (LAUREANO, 2005).

### **2.3 Chatterbots e Educação a Distância**

De acordo Belloni (1999 *apud* Sganderla, 2003), as novas tecnologias de informação e comunicação possibilitam a interação de professor/aluno e a interatividade com materiais e recursos multimídia de forma excelente.

Atualmente a educação a distância busca apropriar, com a utilização de tecnologias e ferramentas interativas, o seu ambiente àquele que há em uma sala de aula presencial. Para alcançar tal cenário é preciso adotar uma metodologia adequada para o desenvolvimento e utilização destas ferramentas interativas (SGANDERLA, 2004).

Um dos recursos que vem despontando atualmente são os agentes inteligentes desenvolvidos para simular uma conversa entre homem-máquina através das trocas de mensagens (*chatterbots*). Primo (1988 *apud* Sganderla, 2004) relata que no processo de aprendizagem um dos elementos principais é a comunicação, transferindo assim todos os problemas relacionados às questões de lingüísticas para os *chatterbots*.

Desde o surgimento dos primeiros *chatterbots* diversas versões e variantes, inclusive novas técnicas de conversação, surgiram possibilitando o avanço nos estudos e implementação. A técnica de Inteligência Artificial (IA) chamada Processamento de Linguagem Natural (*Natural Language Processing*) é uma das mais utilizadas para simular uma conversação no mínimo convincente (*ibidem*).

Como um *chatterbot* pode simular uma conversa  o mais amig  vel e racional, eles creditam grande potencial para auxiliarem os usu  rios de ambientes virtuais de aprendizagem. Isso demonstra que o uso de *chatterbots* em ambientes de Educa  o a Dist  ncia ainda apresenta muito potencial para o estudo e a pesquisa.

### 3 LINGUAGEM NATURAL e AIML (*Artificial Intelligence Markup Language*)

Este capítulo apresenta alguns conceitos para o processamento da linguagem natural que nortearam as técnicas para as construções dos *chatbots*. Traz também um panorama uma das linguagens mais difundida para a construção da base de conhecimento utilizada pelos *chatbots* atuais, o AIML (*Artificial Intelligence Markup Language*).

#### 3.1 Processamento da Linguagem Natural

Conforme Lage (2004), inicialmente, segundo a lingüística estrutural, as tentativas de um diálogo homem-máquina partiram do conceito de que o processo de comunicação consistiria na codificação, pelo emissor, de uma mensagem e em sua decodificação, pelo receptor, que restabeleceria o sentido original (modelo proposto por Sannon & Weaver). Isso colocaria o emissor como dominante da conversação. Este modelo não leva em consideração as escolhas léxicas e gramaticais, lapsos, inferências, vagueza, imprecisão, ambigüidade e outras características que a comunicação humana possui.

Paul Grice, na década de 1970, tentou esclarecer algumas dependências que os participantes envolvidos na conversação atribuíram um ao outro. Ambos admitem que nenhum será: lacônico, nem revelará mais do que lhe foi perguntado (máxima quantidade); deliberadamente falso (Máxima qualidade); Excessivamente minucioso (máxima da relevância); vago, ambíguo, ou construirá de maneira desordenada seu discurso (Máxima da clareza). (*ibidem*)

Uma outra teoria cibernética sobre a comunicação humana foi proposta por Gordon Pask nesta mesma década. Ele enfatizava a natureza pessoal da realidade. Em sua teoria, vida e inteligência conflitam entre construção que se constroem materialmente e individualmente (peculiar e fechada) e interação aberta e partilhada construída a partir de experiências sociais e conceitos.

As teorias sugerem que se utilize o conceito do aprendizado dinâmico e interno de auto-organização do conhecimento, determinado pela interação com o meio e com outros indivíduos. Desta maneira, procurando estabelecer uma cooperação mútua para o entendimento entre homens e dos homens com as máquinas.

Em 1988 Sperber e Wilson aprofundaram a proposta de Grice alterando-a significativamente. Eles pregavam que a informação relevante para o receptor é aquela que, combinada com seu contexto, produz informação nova (*ibidem*).

Segundo Laje (2004):

O conceito de contexto compreende, aí, a percepção do que precede e o que sucede imediatamente a informação (contexto do enunciado), das circunstâncias em que ela é dita (contexto da enunciação) e mais o acervo de memória - experiências, valores, esquemas de pensamento. É o contexto que determina o significado de proposições comuns como "o café me dá sono", "feche a porta", "não se esqueça de mim", "ele é alto para a idade", etc.

Esta proposta tenta substituir o modelo proposto por Grice mediante o conceito de implicação textual, permitindo que os participantes busquem maior relevância para o enunciado.

Ainda de acordo Laje (2004), os autores observaram que o cérebro humano trabalha buscando o maior número de informações utilizando o mínimo de esforço possível. O processo de compreensão utiliza mecanismos similares ao da

lógica, mas partem de suposição ditas verdadeiras para resultar em outras que serão tidas como verdadeiras.

### 3.2 Estado da Arte e Modelo de Comunicação

O modelo de comunicação que reflete o estado da arte mais atual da pesquisa sobre comunicação humana é o de Gerbner (*ibidem*). Neste modelo o emissor percebe o evento (seleciona, avalia e contextualiza) e em seguida acessando um canal mediático, codifica-o em forma e conteúdo.

Lages (2004) explana que reproduzir e simular este tipo de comunicação por meio de ferramentas tecnológicas não é ainda uma tarefa simples. Uma forma de ultrapassar essas barreiras é se limitar a uma percepção de uma situação controlada e regulamentada com a emissão da mensagem, com vocabulário e sintaxes restritas a um domínio.

### 3.3 AIML (Artificial Intelligence Markup Language)

Segundo ALICE (2004), o AIML foi desenvolvido no período de 1995 a 2000 pela comunidade ALICE – *Artificial Intelligence Foundation* ([www.alicebot.org](http://www.alicebot.org)) com uma gramática proprietária que formava base para o primeiro *Alicebot*. Alguns anos, após atualizações visando a padronização da gramática, foi adotado o XML(*eXtensible Markup Language*).

Atualmente baseada em XML, a linguagem AIML, assim como sua derivada, é descrita por marcações denominadas de *tags* que compõem pares de entrada-saída. Quanto mais bem planejada a base AIML, maior será o desempenho

de coerência na conversação. Em contra partida, uma base de conhecimento grande torna difícil a manutenção do conhecimento do *chatterbot*.

Devido a característica de padronização da linguagem *AIML*, proporcionada pelo *XML*, existem hoje diversas tecnologias de código aberto responsáveis pelo processamento das bases *AIML*. Isto incentiva a comunidade a desenvolver mais e mais aplicações e ferramentas que se beneficiem da linguagem *AIML*, tornando o padrão cada vez mais sedimentado e confiável.

## 4 AGENTES E SISTEMAS MULTIAGENTES (SMA)

A seguir descreve-se um pouco sobre o conceito de agentes inteligentes e a classificação que estes recebem quanto a seu uso e o meio onde estão inseridos.

### 4.1 Sistemas Multiagentes (SMA)

A metáfora de inteligência utilizada em SMA é a Comunidade Inteligente, isto é, o comportamento social que é a base para a inteligência do sistema. Os membros de uma comunidade inteligente podem ser desde extremamente simples até extremamente complexos (BITTENCOURT, 1996 apud SEEWALD, 2004, p. 14).

Sistemas Multiagentes são sistemas compostos por um conjunto de agentes autônomos inseridos num contexto social único que trabalham para a resolução de uma meta global simples (SEEWALD, 2004).

### 4.2 Agentes

A definição para o termo agente se diverge entre muitos autores. A sua concepção está relacionada aos diferentes pontos de vista associados à funcionalidade fornecida pelo agente.

Segundo Maes (1994 *apud* Vargas, 2003), que conceitua agente sob o ponto de vista de seu uso como assistentes pessoais, agentes são componentes de *software* que atuam autonomamente para atender as necessidades do usuário.



Já Franklin e Grasse (1996 *apud* Costa, 2005) consideram agente como um sistema que faz parte de um ambiente, onde ele percebe e age para atender suas próprias necessidades.

Enquanto que Woldge e Jennings (1994 *apud* Vargas, 2003) relatam agentes como sistemas que possuem um comportamento baseado na representação de suas crenças, comportamento e desejos.

Para esta pesquisa se seguirá como referência a especificação de agente definida por Shomam (1997 *apud* Costa, 2004), que segundo Costa é uma definição mais específica e aceitável por grande parte dos pesquisadores. Ele define assim agente como uma entidade computacional funcionando continuamente e de forma autônoma em um ambiente particular, freqüentemente habitado por outros agentes e processos.

Mesmo com todas estas definições existentes para conceituar o termo agente, existe um conjunto de características em comum para um *software* ser considerado um agente inteligente (NWANA, 1996 *apud* Vargas, 2003). Algumas destas características são apresentadas na tabela abaixo.

Tabela 1 – Propriedades e requisitos comuns a agentes.

Propriedade	Sinônimo	Significado
<b>Reativo</b>	Sentido e agindo	Responde de forma temporalizada a mudanças no ambiente
<b>Autônomo</b>		Exerce controle sobre suas ações
<b>Orientado a objetos</b>	Pró-ativo e intencional	Não age apenas em resposta ao ambiente
<b>Temporalmente contínuo</b>		É um processo em continua operação
<b>Comunicativo</b>	Socialmente hábil	Comunica-se com os outros agentes ou até com pessoas
<b>Aprendiz</b>	Adaptável	Muda de comportamento e virtude de experiências anteriores
<b>Móvel</b>		Capaz de transportar-se de uma máquina para outra
<b>Flexível</b>		Ações não são programadas em scripts
<b>Personagem</b>		De “personalidade” e estado emocional verossímil

Fonte: Primo, 2005.

Os agentes utilizados nesta pesquisa levam em consideração as características apresentadas, e outras também citadas a seguir:

**Credibilidade:** a impressão passada de estar interagindo com um ser vivo é conseguida com os movimentos e comportamentos humanos do agente animado da *Microsoft*;

**Reatividade:** o agente toma decisões e atitudes a determinadas ações no ambiente virtual apresentando determinadas ações para determinadas mudanças;

**Continuidade:** fica ativo, continuamente monitorando o comportamento do ambiente e do usuário.

Os agentes se distinguem também em categorias dependendo do seu propósito, por isso, a partir de agora se apresentarão aquelas estudadas para o desenvolvimento desta pesquisa.

#### 4.2.1 Agentes de Interface

Costa (2005) denomina **agente de interface** aquele agente com a capacidade de aprender ou simplesmente aquele que serve como assistente pessoal. A principal característica é sua autonomia para executar tarefas para seus usuários. Basicamente, estes estão em execução continua em um determinado ambiente monitorando as atividades do usuário e reagindo a determinadas situações do ambiente ou ações do usuário.

### **Agentes Animados de Interface**

Os **agentes animados de interface** combinam as propriedades dos agentes de interface tradicionais aos elementos de som e movimento (PARRA, 2005).

Para Parra (2005), os agentes animados de interface são personagens autônomos, animados com movimentos, expressões e falas que possuem características e formas de seres vivos. Capazes de reagir a determinadas situações do ambiente ou de determinadas interações do usuário.

Os agentes animados de interface são agentes que, por meio de gestos corporais específicos dos seres humanos, conseguem uma interação mais natural que meios puramente computacionais desprezam. Eles trazem algumas características intrínsecas, muito importantes para o desenvolvimento do protótipo desta pesquisa, como a motivação da curiosidade do usuário.

Por isso, com a utilização deste tipo de agente possibilita-se proporcionar uma interface mais amigável para o usuário interagir com o sistema. Isso, através do uso de um paradigma mais intuitivo sem a necessidade de aquisição de novos conhecimentos para interpretar as respostas do agente de conversação (*chatbot*).

Os agentes animados de interface podem também ser utilizados por outros agentes autônomos para servir de canal de comunicação mais natural com o usuário final.

Há outras denominações para os agentes de interface dependendo do contexto onde estão inseridos. Segundo Giraffa (1999 *apud* Bocca, 2004), classifica-se de **agente pedagógico** aqueles agentes que estejam imersos numa sociedade de agentes desenvolvidos para fins educacionais, podendo atuar como:

**Tutores:** destinados ao ensino dirigido ao aluno;

**Assistentes:** colaborando com a aprendizagem do aluno;

**Facilitador:** auxiliando somente quando solicitado.

Desta forma, acrescentando as características dos recursos multimídias aos agentes pedagógicos temos a definição dos agentes classificados como **agentes pedagógicos animados** (BOCCA, 2004).

#### **4.2.2 Agentes animados da *Microsoft* – (*Microsoft Agents*)**

Segundo Parra (2005), os agentes animados da *Microsoft* são uma boa opção para soluções de interfaces baseadas em personagens animados. É uma tecnologia que disponibiliza um conjunto de ferramentas e *softwares* programáveis que permitem a criação de interfaces mais ricas e intuitivas. Desta maneira, desenvolve-se agentes animados de interface com recursos de reconhecimento de voz, sintetização de textos e movimentos animados.

A adoção desta tecnologia agrega muito valor atrativo que pode ser utilizada tanto em uma página *Web* quanto em uma aplicação *desktop*.

Um exemplo de sucesso deste tipo de tecnologia é o agentes disponível nas ferramentas da suíte *Microsoft Office*, da própria *Microsoft* (figura 2) que trabalha como um assistente pessoal.

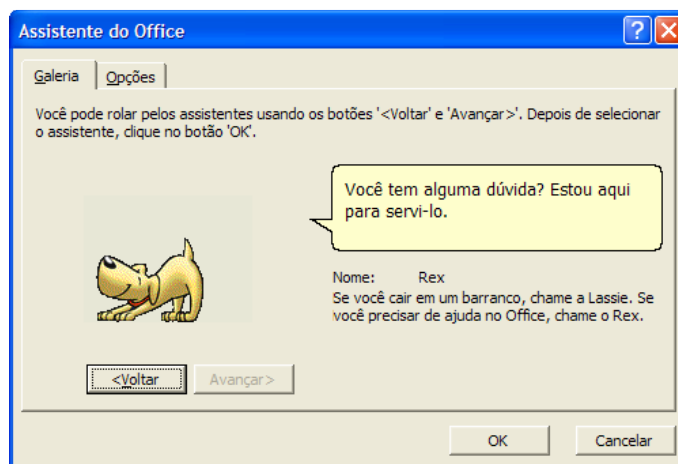


Figura 2 – Assistente do Office – Janela de configuração.

No *site* oficial da *Microsoft* é possível localizar todo o material e ferramentas necessárias para implementação de agentes animados. São disponibilizados (<http://www.microsoft.com/msagent/>) quatro personagens básicos, conforme apresentados na Figura 3, os quais podem ser utilizados por qualquer aplicação.

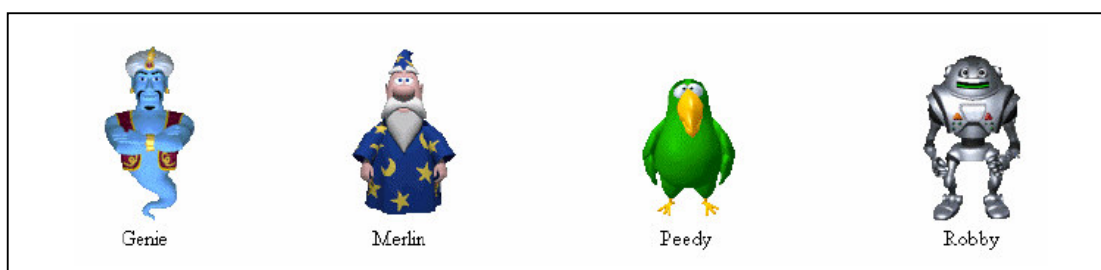


Figura 3 – Personagens padrão dos agentes da Microsoft.

Além destes personagens há a possibilidade de se criar novos personagens específicos às necessidades de cada aplicação. Com gestos, movimentos e características próprias. Um exemplo pode ser observado na Figura 4, que apresenta personagens próprios disponíveis no *site* [www.bellcraft.com](http://www.bellcraft.com).



Figura 4 – Personagens personalizados de agentes animados.

Os agentes animados da *Microsoft* apresentam algumas características relevantes e decisivas na escolha da tecnologia para a utilização no protótipo em questão (Parra, 2005), como segue:

- Integração fácil ao sistema como um componente com estrutura orientada a objetos;
- Pode ser codificado com qualquer linguagem com suporte a COM (*Component Object Model*), como *C++*, *Visual Basic*, inclusive como um controle *ActiveX* que facilita a programação com linguagens de script como *VBscript* e *Jscript*;
- Possibilidade de conversão de textos em voz e de reproduzir arquivos de áudio;
- Reconhecimento de voz, que pode ser usado como entrada de dados para enviar comandos ao agente,
- Ferramentas disponíveis para a criação de personagens animados e para integração com qualquer aplicação que suporte a tecnologia.

Enfim, os agentes da *Microsoft* podem ser utilizados para os mais variados propósitos, desde simples anfitriões que saúdam usuários de um sistema até assistentes pessoais ou tutores de um ambiente, como no caso deste no protótipo aqui estudado.

## 5 DEFINIÇÕES DO PROTÓTIPO

Para tornar um protótipo hábil algumas informações precedentes e muito relevantes são necessárias, portanto, a seguir, apresenta-se a metodologia, as definições e principalmente a arquitetura para o seu desenvolvimento. Num segundo momento, as características e utilidades do conseqüente experimento prático também serão apresentadas.

### 5.1 Metodologia

Devido a complexidade do tema e por se tratar da integração de tecnologias (agentes de interface / agentes de conversação (*chatbots*) / Ambiente Virtual de Aprendizagem), a qual o domínio total ainda não é possível, optou-se previamente pela familiarização das ferramentas, características e criação das mesmas.

Após esse estudo, definir estratégias e uma arquitetura para integrar e interagir com a tecnologia dos agentes juntamente com o Ambiente Virtual de Aprendizagem (AVA) é outro ponto do alicerce para concretizar o protótipo.

Por conseguinte, analisar as dificuldades encontradas na utilização do Ambiente Virtual *Learnloop* também são necessárias para definir e criar a base de conhecimento para que o *chatbot* responda satisfatoriamente as “interrogações” relacionadas a utilização das ferramentas do ambiente virtual de aprendizagem.

E por fim, implementar o protótipo com essa arquitetura em uma interface que possibilite o usuário final interagir com os agentes (*chatbot* e o agente da

*Microsoft*) para buscar informações que sanem seus questionamentos quanto ao uso do ambiente virtual.

## **5.2 Motivação (expectativas e benefícios da solução)**

Segundo Gonçalves (2004), a utilização da *Web* como plataforma para a educação a distância vem sendo estudada e difundida em proporções crescentes a cada dia. A Universidade do Extremo Sul Catarinense (Unesc) com a intenção de se incluir também através de sua comunidade acadêmica no mundo dos cursos a distância, conta com um Setor de Educação a Distância (Sead) próprio. Esse fomenta o ensino por meios de comunicação e colaboração pela *internet*, facilitando e incentivando o acesso dos professores e alunos ao ambiente virtual de aprendizagem *Learnloop*.

Todavia, este novo cenário de educação traz consigo também a necessidade de aquisição de novos conhecimentos para se utilizar ferramentas de ensino não dominadas até então por muitos professores e alunos da instituição.

A deficiência neste conhecimento acaba inibindo e contribuindo muito para desmotivação do usuário e abandono do ambiente virtual. Por isso, a proposta deste protótipo é baseada nas perspectivas de Coelho (2004), referentes a educação a distância. Ele prevê a forte tendência em utilização de formas mais racionais para a interação com ferramentas de colaboração à distância.

Robôs de Conversação, *Softbots* ou simplesmente *Chatterbots*, como, aliás, se denominará neste projeto, possuem características motivacionais fortes, podendo estabelecer uma conversa mais individualizada e racional sobre determinado assunto.



Somado as características dos agentes de interface que interagem de forma mais pessoal com o usuário e de serem capazes de simular gestos e feições humanas nesta interação, pode-se assim encorajar mais professores e alunos oferecendo uma ajuda mais próxima da racional nas questões que tangem a utilização das ferramentas do *Learnloop*.

A união das duas tecnologias de interação agregadas ao ambiente virtual (*Learnloop*) da instituição possibilita, enfim, gerar qualidade no atendimento e suporte aos usuários desse ambiente.

Outro ponto respeitável no emprego deste protótipo diz respeito a diminuição de custos referentes a pessoal envolvido no atendimento e suporte aos usuários do *Learnloop*. Conforme CONVERSAGENT (2004), pesquisas demonstram que as pessoas buscam qualidade no atendimento as suas dúvidas a determinada questão. Esta qualidade implica em custos para criar uma estrutura que atenda de forma individualizada cada pessoa.

Esta escala (custo x atendimento pessoal) é proporcional, quanto mais impessoal for o atendimento menor o custo para implantação, mas em contra partida menor a qualidade no atendimento, isto pode ser observado no gráfico da Figura 5.

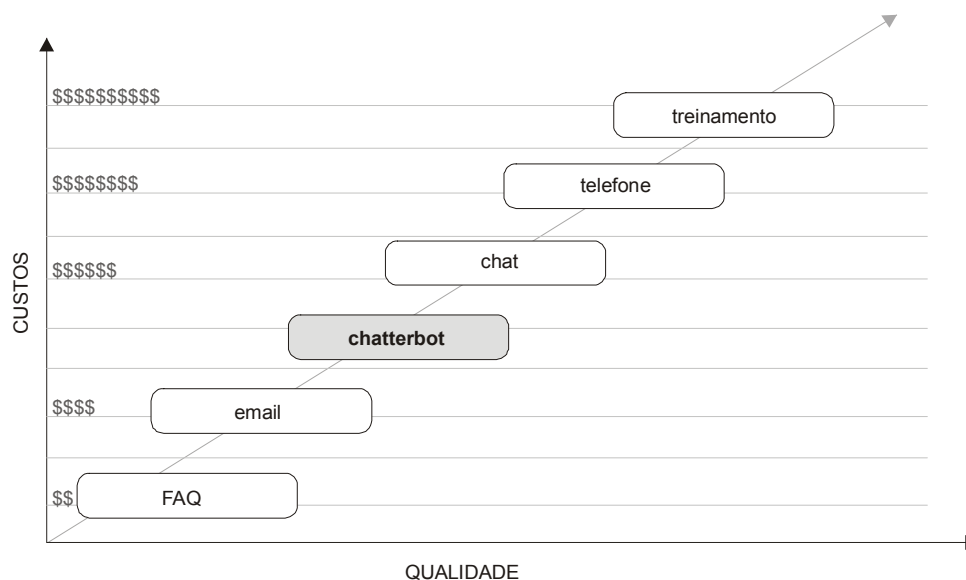


Figura 5 – Relação entre qualidade e custo de suporte (adaptado de *CONVERSAGENT*, 2004).

Um suporte com uma interação de qualidade e automatizado que evite a necessidade de pessoas para dirimir dúvidas freqüentes a diversas pessoas seria o ideal para as instituições baixarem os custos.

Atualmente o setor de educação a distância da instituição oferece algumas alternativas para o atendimento dos usuários do *Learnloop* relacionadas a dúvidas de utilização das ferramentas do ambiente.

As principais foram apresentadas na Figura 5, mas há algumas como *FAQ* e manuais com qualidade e custo reduzido devido a impessoalidade no atendimento. E há outras com qualidade superior com um atendimento mais personalizado como as oficinas de treinamento, mas que exigem também uma estrutura (humano e física) com um custo mais elevado.

A procura pelo uso destes meios passa por uma espécie de funil. As possibilidades de o usuário sanar suas dúvidas vão sendo descartadas através de sua busca que caminha por um nível mais impessoal. Não encontrando uma

resposta ele busca suprir sua necessidade em um nível superior que possui maior qualidade e maior custo para instituição, de acordo a Figura 6.

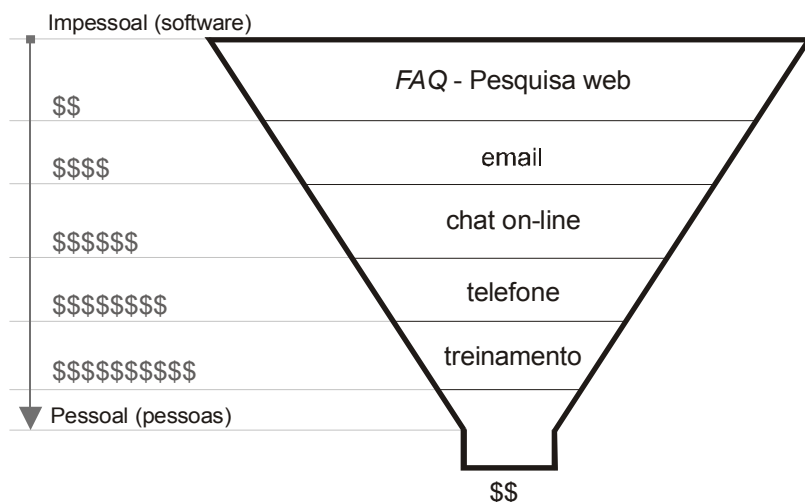


Figura 6 – Funil de Serviços (adaptado de *Conversagent*, 2004).

A solução proposta através do protótipo encontra-se em um meio de aliar a qualidade no atendimento, ou seja, personalizado e baixar os custos na automatização do mesmo. Isso é possível, pois se diminui a necessidade de envolver pessoas no processo, sem perder a individualidade e personalidade no atendimento, e também devido às características de racionalidade e personalidade implícitas nas tecnologias utilizadas no protótipo.

### 5.3 O Protótipo

O seu desenvolvimento tem como objetivo maior definir e aplicar os aspectos referentes à comunicação de agentes. Para tanto, a criação de uma base de conhecimento utilizando a linguagem de marcação *AIML*; a utilização de *chatbot* para buscas nesta base e interação com um agente de interface para

interagir com o usuário e, por fim, a integração desses com o ambiente virtual de aprendizagem da Instituição, possibilitará uma futura implantação e adequação do protótipo ao projeto global do ambiente virtual da Universidade.

O protótipo proposto para este trabalho segue as características de um sistema multiagente inserindo um agente animado de interface em colaboração com outro agente de conversação (*chatbot*) num único contexto de um ambiente virtual de aprendizagem.

### 5.3.1 Arquitetura do protótipo

O papel principal do prototipo é cumprir sua função de auxiliar o usuário, no manejo das ferramentas do *Learnloop*, de maneira a sugerir um raciocínio mais próximo possível do raciocínio humano. Para este propósito foi necessário criar uma base de conhecimento que é utilizada pelo *chatbot* e uma interface de comunicação que faz a ponte entre este e o agente de interface desenvolvido com a utilização dos agentes de interface da *Microsoft* o qual agi reciprocamente com o usuário final.

Forma-se assim um conjunto de tecnologias que interagem entre si e são partes integrantes do ambiente virtual *Learnloop* promovendo a motivação e auxiliando os usuários na administração de suas salas virtuais. A Figura 7 mostra esta estrutura de comunicação entre os componentes envolvidos.

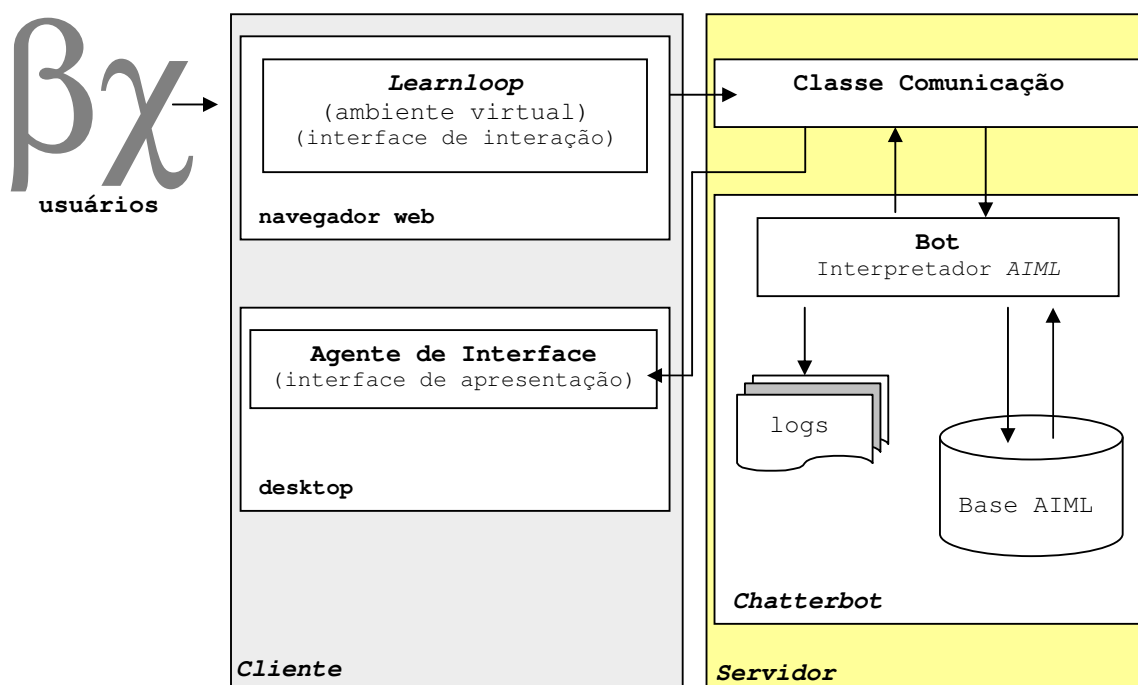


Figura 7 – Arquitetura do protótipo.

#### 5.3.1.1 Ambiente Virtual de Aprendizagem – *Learnloop*

A ferramenta *open source* desenvolvida para o aprendizado colaborativo foi produzida por Dane através do Instituto *Vicktoria* na *Colhenburg Busuness School* em *Golhenburg*, Suécia (GONÇALVES, 2004). Trata-se basicamente de um ambiente de comunicação e colaboração em grupo fundamentada na *Web*.

No caso da Unesc, o ambiente de educação à distância adotado é o *Learnloop* que no decorrer de seu uso vem sendo avaliado pela comunidade acadêmica e também adaptado as necessidades da instituição tanto pelo setor de educação a distância como, em parceria, com a diretoria de informática. Dessa

forma, suas características *open source* facilitam a adoção e a integração também do protótipo ao ambiente.

Segundo seus desenvolvedores ([www.Learnloop.org](http://www.Learnloop.org)) o *Learnloop* foi criado para ser usado na plataforma *linux/Unix-hosted*, contudo como a sua tecnologia utilizou o *php* (4.1 ou superior) e o banco de dados *MySql* (3.3 ou superior) o ambiente herdou assim as características de independência de plataforma, ou seja, pode ser utilizado em diversos sistemas operacionais distintos.

Um outro ponto essencial para a construção do protótipo diz respeito ao acesso. O ambiente, desde de que se tenha um navegador *Web*, pode ser acessado de qualquer lugar pelo usuário.

#### **5.3.1.2 Base de Conhecimento**

O domínio do conhecimento necessário para se utilizar todas as ferramentas de comunicação e colaboração do *Learnloop* é muito amplo. Por isso, para este protótipo será compreendida apenas uma funcionalidade específica do ambiente.

Contudo, fica-se claro que é perfeitamente possível estender a funcionalidade do protótipo para facilmente interagir com diversos domínios distintos, simplesmente adicionando novas bases de conhecimento. Essas bases em analogia com o mundo real podem ser consideradas como o cérebro do *chatterbot*.

Criou-se para o protótipo uma base de conhecimento com conteúdo suficiente para contemplar respostas a questões relacionadas à tarefa de incluir alunos nas salas virtuais.

A opção por esta particularidade deve-se pelo fato de ser “a dúvida” a demanda maior de dificuldade por parte dos usuários. De acordo com informações obtidas pelos e-mails de suporte do SEAD percebe-se que esse “tema” é freqüentemente citado.

Desta maneira, o escopo do conhecimento deste protótipo limita-se a informações necessárias para auxiliar os professores no procedimento de inclusão dos alunos nas suas salas virtuais.

### **Aquisição de conhecimento**

A base de conhecimento do *chatterbot* pode ser incrementada pela simples edição dos arquivos de marcação que constituem o conjunto e relacionamento das respostas referentes ao tema.

Num primeiro momento esta base foi criada a partir dos manuais de utilização do *Learnloop* disponíveis no próprio ambiente virtual ([www.ead.unesc.net](http://www.ead.unesc.net)) e da análise dos *e-mails* de suporte recebidos e respondidos pelo setor de educação a distância da instituição.

No decorrer do projeto é possível agregar mais conhecimento e melhorar esta base por meio da análise dos logs <sup>2</sup> de conversação que o *software* do *bot* gera a cada conversa do usuário com o *chatterbot*.

### **AIML – Artificial Intelligence Markup Language**

É por meio da comparação realizada entre as perguntas enviadas pelo usuário e os dados disponíveis na base de conhecimento que o *chatterbot* consegue compreender e emitir respostas cabíveis ao contexto da conversa.

---

<sup>2</sup> Registro que armazena toda interação de uma sessão entre o usuário e o *chatterbot*.

A independência de arquitetura e o padrão aberto, resultados das características *XML3 (Extended Markup Language)*, adotadas pela linguagem *AIML (Artificial Intelligence Markup Language)* tornaram-se decisivos para a adoção desta linguagem de marcação para a persistência e criação da base de conhecimento do protótipo. Por sua vez, ele deve ser inserido em um ambiente virtual de aprendizagem *open source* como é o *Learnloop*.

Este projeto *open source* basea-se em Alice (2005), que explica que a linguagem *AIML* foi inventada para o projeto de inteligência artificial *A.L.I.C.E. (Artificial Language Intelligence)*, mas também caracteriza a origem de todos os outros projetos de *chatterbots*.

A escolha desta linguagem torna o protótipo independente da tecnologia que utiliza para implementação do motor (*bot*) que interpreta o arquivo *AIML* e faz os relacionamentos entre as sentenças enviadas pelo usuário e as categorias presentes no arquivo *AIML*. Além disso, a base de conhecimento do projeto criada com a linguagem *AIML* pode ser utilizada por qualquer *bot* que sigam as características do *chatterbot* ALICE (uma lista destes *bot* está disponível em <http://www.alice.org/downloads>).

Já para criação inicial do arquivo *AIML* da base de conhecimento do protótipo foi utilizado o *software Pandorawrite* (disponível on-line no endereço <http://www.pandorabots.com>). Um programa eficiente que traduz sentenças em linguagem natural simples, como apresentadas na Figura 8, para o formato de marcação *AIML* como se observa na Figura 9.

Com a finalização desse arquivo se usufruiu assim de todas as possibilidades de interação que as *tags* de marcação da linguagem *AIML* dispõe

---

<sup>3</sup> Meta-linguagem voltada para especificação e marcação.



para a criação das categorias. Isso tem o intuito de dar maior credibilidade ao contexto do diálogo entre o usuário e o *chatterbot*.

```
Oi, como vc pode me ajudar?

    Oi, basta vc me perguntar o que deseja saber.

Como obtenho minha senha?

    É simples, basta solicitar nova senha no link na página de login.
```

Figura 8 – Sentenças em linguagem natural.

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0">
  <category>
    <pattern> Oi como vc pode me ajudar </pattern>
    <template> Oi, basta vc me perguntar o que deseja saber. </template>
  </category>
  <category>
    <pattern> Como obtenho minha senha </pattern>
    <template>
      É simples, basta solicitar nova senha no link na página de login.
    </template>
  </category>
</aiml>
```

Figura 9 – Sentenças formatadas para as categorias da linguagem AIML.

### 5.3.1.3 BOT – Módulo de interpretação

O *bot* (robo) ou motor do *chatterbot* é o *software* responsável por interpretar as relações contidas na base de conhecimento por meio das *tags* de marcações AIML e retornar a sentença de resposta ao usuário.

A escolha do *bot* para o protótipo levou em consideração o cenário *open source* adotado pela instituição no ambiente virtual de aprendizagem, assim como a independência de plataforma e a facilidade para integração no *Learnloop*. Sob estas diretivas, o *bot* adotado no protótipo foi o do *chatterbot Anna*

(<http://annabot.sourceforge.net/>) desenvolvido com a linguagem de programação *Java* da *Sun*, de código livre e que segue também os padrões da linguagem *AIML*. Anna foi um dos *chatterbots* finalistas do *Loebner Prize Competition*<sup>4</sup> de 2002.

### ***Chatterbot Anna e Program D***

ANNABOT (2005) descreve que Anna é um *chatterbot* projetado para passar no teste de Turing. Teste o qual uma pessoa interagindo numa conversação com a maquina deve ser incapaz de identificar que está conversando com um robô. Anna é originária do projeto ALICE que utiliza toda a tecnologia proveniente dos *chatterbots* que seguem o projeto ALICE.

E por se basear num interpretador *AIML* implementado em *Java* e amplamente utilizado por *chatterbots*, o *software Program D* (pode ser encontrado gratuitamente no endereço <http://aitools.org/downloads/>), foi determinado para este protótipo também, como explicado adiante.

O *bot* tem o papel de analisar a sentença passada pelo usuário e tentar generalizar esta entrada por meio de substituições básicas que são executadas a partir dos arquivos de substituição e das categorias *AIML*. Portanto, sempre existirá uma resposta para uma pergunta por mais generalizada que ela seja, e dependendo da qualidade da base de conhecimento formatada esta resposta pode ser exatamente a esperada pelo usuário.

A escolha do *Program D* para a implementação do *bot* do protótipo possibilita algumas vantagens, dentre as quais a de se obter diversas maneiras de disponibilizar o acesso a este *bot*. Ele pode ser feito por ferramentas de troca de

---

<sup>4</sup> Campeonato de inteligência artificial de robôs de conversação (*chatterbots*).

mensagens instantâneas como *IRC*, *AOL*, *MSN*, ou localmente por console ou ainda, como, a necessária para o protótipo, pelo protocolo *HTTP*.

Este *software* utiliza ainda um contêiner *Web open source (jetty)*, embutido para interpretar o *servlet*, capaz de fazer a tradução dos arquivos *AIML*

A customização deste serviço é facilmente executada alterando-se o arquivo de configuração *conf/jetty.XML*. Para o protótipo foi apenas alterada a porta padrão onde o *bot* estudado fica escutando requisições na porta padrão 80, conforme destacado em um trecho do arquivo *jetty.XML* na Figura 10 a seguir.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure
1.0//EN" "http://jetty.mortbay.org/configure_1_1.dtd">
<Configure class="org.mortbay.jetty.Server">
  <Call name="addListener">
    <Arg>
      <New class="org.mortbay.http.SocketListener">
        <Set name="Port">80</Set>
        <Set name="MinThreads">5</Set>
        <Set name="MaxThreads">255</Set>
        <Set name="MaxIdleTimeMs">60000</Set>
        <Set name="MaxReadTimeMs">60000</Set>
        <Set name="LowResourcePersistTimeMs">5000</Set>
      </New>
    </Arg>
  </Call>
</Configure>
```

Figura 10 – Arquivo de configuração do contêiner *Web*.

Ele também possibilita a configuração de um número ilimitado de *bots* que estarão somente a espera de requisições. Para isso, elaborou-se um novo *bot* para o protótipo editando o arquivo de configuração *conf/startup.XML* (Figura 11). Nessa listagem podemos observar que cada *bot* é definido entre as *tags* de marcação *<bots></bots>* e entre as *tags* *<properties></properties>*. Desse maneira, defini-se também as propriedades que o *bot* vai ter como nome e a quantidade máxima de usuários que ele pode atender, como por exemplo, para 30.

Uma parte muito importante de configuração do nosso *bot* se encontra nas *tags* *<predicates href="predicates.XML"/>*, *<substitutions href="substitutions.XML"/>* e

`<sentence-splitters href="sentence-splitters.XML"/>` que configuram respectivamente os nomes dos arquivos de predicados, substituição e *splitter* que o *bot* utilizara para generalização das sentenças recebidas.

```
<?xml version="1.0"?>
<!--This is an example startup.xml file for Program D.-->
<!--programd-startup is the root element and is required.-->
<programd-startup>
  <bots>
    <bot id="Unesc" enabled="true">
      <!--Bot predicates are set using the property tag.
      These are just examples.
      Be sure to set properties BEFORE loading AIML files.-->
      <properties>
        <property name="name" value="Unesc"/>
        <property name="maxclients" value="30"/>
      </properties>
      <!--Listener types must correspond to the appropriate
      identifiers for listeners that are available in your
      classpath.-->
      <listeners>
      </listeners>
      <!--You can include such things as predicates, substitutions, and
      sentence-splitters directly here, but it is likely
      that multiple bots will use the same ones, so it's more
      convenient to use the href attribute on such elements,
      as shown.-->
      <predicates href="predicates.xml"/>
      <substitutions href="substitutions.xml"/>
      <sentence-splitters href="sentence-splitters.xml"/>
      <!--You may enumerate each file you want the bot to load, or use
      simple glob-like expressions with "*".
      The path is relative to the location of this file.-->
      <learn>../unesc_brain/*.aiml</learn>
    </bot>
  </bots>
</programd-startup>
```

Figura 11 – Arquivo de configuração dos *bots*.

Por padrão toda conversação de uma sessão de usuário é gravada em arquivos de *log* no padrão *XML* para posterior análise destes *logs* e melhoria da base de conhecimento. Um exemplo deste arquivo é apresentado na Figura 12 abaixo.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="../../resources/logs/view-
chat.xsl"?>
<!DOCTYPE ALLOW_HTML_ENTITIES [ <!ENTITY % HTMLlat1 PUBLIC "-//W3C//ENTITIES
Latin1//EN//HTML"    "../resources/DTD/xhtml-lat1.ent"> %HTMLlat1; <!ENTITY %
HTMLsymbol    PUBLIC    "-//W3C//ENTITIES    Symbols//EN//HTML"
"../resources/DTD/xhtml-symbol.ent"> %HTMLsymbol; <!ENTITY % HTMLspecial
PUBLIC    "-//W3C//ENTITIES Special//EN//HTML"    "../resources/DTD/xhtml-
special.ent"> %HTMLspecial; ]>
<exchanges>
  <exchange>
    <timestamp>2005-04-09 22:56:52</timestamp>
    <userid>pavilion</userid>
    <clientname>user</clientname>
    <botid>Unesc</botid>
    <input>firsthello</input>
    <response></response>
  </exchange>
  <exchange>
    <timestamp>2005-04-09 22:57:26</timestamp>
    <userid>pavilion</userid>
    <clientname>user</clientname>
    <botid>Unesc</botid>
    <input>oi</input>
    <response>Oi, no que posso ajudar?</response>
  </exchange>

```

Figura 12 – Arquivos dos logs de conversão.

Com algumas alterações simples no arquivo de configuração *servers.properties* do *Program D* é possível ainda utilizar o banco de dados *Mysql* para armazenar estes *logs*, como se destaca a Figura 13.

```

# -----
# Database logs
# -----

# Enable chat logging to the database?
# * Be sure that the database configuration (later in this file) is valid.
programd.logging.to-database.chat=true

```

Figura 13 – Arquivo de configuração da persistência dos *logs* de conversação.

Por isso, a arquitetura utilizada para o protótipo não poderia deixar de ser uma arquitetura *Web*, sendo que toda a solução serve de apoio para a educação à distância, onde o ambiente virtual de aprendizagem, no qual o protótipo será integrado, é disponível em uma interface *Web* por meio da *internet*.

Assim, com toda a configuração feita e após iniciar o servidor, o *chatterbot* deste protótipo fica a espera de requisições na porta 80 e pode começar a interagir com usuário por meio de uma interface *Web* qualquer. A integração do *chatterbot* com o *Learnloop*, desta forma, se torna mais simples a partir do momento que utilizam arquiteturas semelhantes.

#### 5.3.1.4 Agente de Interface

Até aqui quase todos os componentes que compõem o *chatterbot* do protótipo em questão estão definidos e implementados - um módulo desenvolvido em *Java* pra interpretar e relacionar as categorias *AIML* e uma base de conhecimento criada com as *tags* de marcação *AIML*. Para completar então o *chatterbot* é necessário uma interface para interação do usuário com o robô.

Para este propósito foram implementadas duas soluções: uma para o protótipo com interface de interação onde o usuário entra com as sentenças que são enviadas ao *bot* e a segunda, com outra interface de apresentação que é responsável por retornar as respostas que o *bot* devolve para o usuário.

##### Interface de Interação

Cria-se, por sua vez, uma interface simples (em *php*, *html* e *javascript*) que é incorporada ao próprio *Learnloop*. E por meio de um formulário o usuário inicia uma sessão de conversação com o *chatterbot*. Por um campo de texto o usuário interage com suas perguntas conforme destacado na Figura 14. Este campo de texto pode estar acessível em qualquer lugar do ambiente virtual e a qualquer momento que o usuário sinta a necessidade de ajuda.

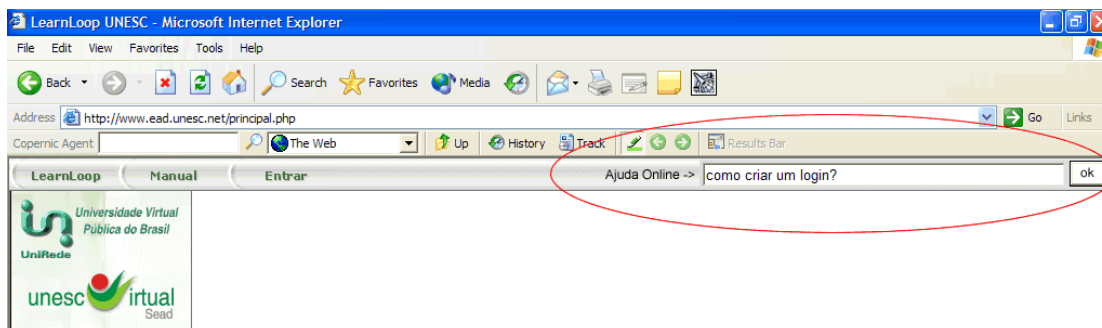


Figura 14 – Interface de interação do usuário.

### Interface de Apresentação

As interações funcionais, como ressalta Sarmento (2004), com a cognição que a emoção humana estabelece são fundamentais para melhor aproveitar a computação inteligente que ferramentas em tempo real necessitam.

Para simular as feições humanas na interação do usuário com o *chatterbot* foi implementado um agente de interface da *Microsoft* representado pelo personagem da Figura 15.

Ele serve de interface de apresentação para o protótipo e é responsável em exibir as respostas visualmente em balões e por voz. Assim ele interpreta e converte as sentenças em áudio e ainda executa as ações recebidas que dão os movimentos ao personagem.

Deve ficar claro que é possível, simplesmente modificando o parâmetro na configuração do ambiente, definir o personagem que será utilizado pelo protótipo.

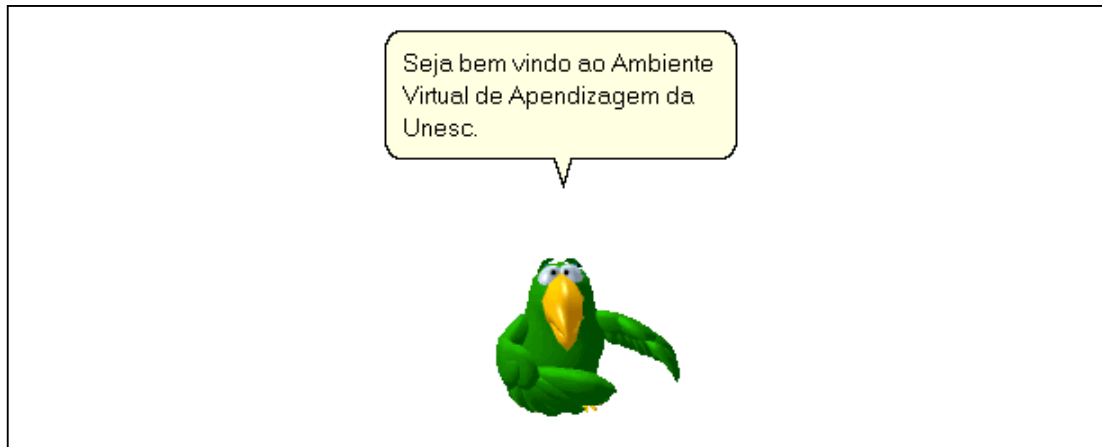


Figura 15 – Agente animado da Microsoft Peddy dando boas vindas.

Optou-se por este personagem no protótipo por estar disponível gratuitamente pela *Microsoft* em seu site oficial (<http://www.microsoft.co/msagent>). Porém, é possível e recomendável um futuro estudo para adoção de um personagem que dê maior credibilidade no ensino e se identifique ao ambiente virtual e a missão da instituição.

Os personagens dos agentes de interface da *Microsoft* são criados com a simples sobreposição de imagens distintas para cada ação por ele executada. E no próprio site da *Microsoft* estão disponíveis ferramentas para este propósito.

Algumas pesquisas e trabalhos sobre a modelagem e implementação de agentes animados para educação a distância já podem ser encontrados na literatura acadêmica (JAQUES, 2003).

A facilidade de se utilizar *javascript* para implementar o controle do agente permite a independência de navegador para a utilização do agente e permite a integração dos controles no ambiente virtual de forma mais natural.

Assim, o agente executa suas ações e falas a partir de *scripts* de execução criados dinamicamente em uma página em *php*. As instruções recebidas



pelo *script* são compostas por uma ação física de movimento e uma ação verbal enviadas pelo *chatterbot* da busca efetuada na própria base de conhecimento.

A Figura 16 mostra o processo do *script* do agente *Microsoft* identificando o fluxo dos dados recebidos do *chatterbot* e manipulados pelo *script*.

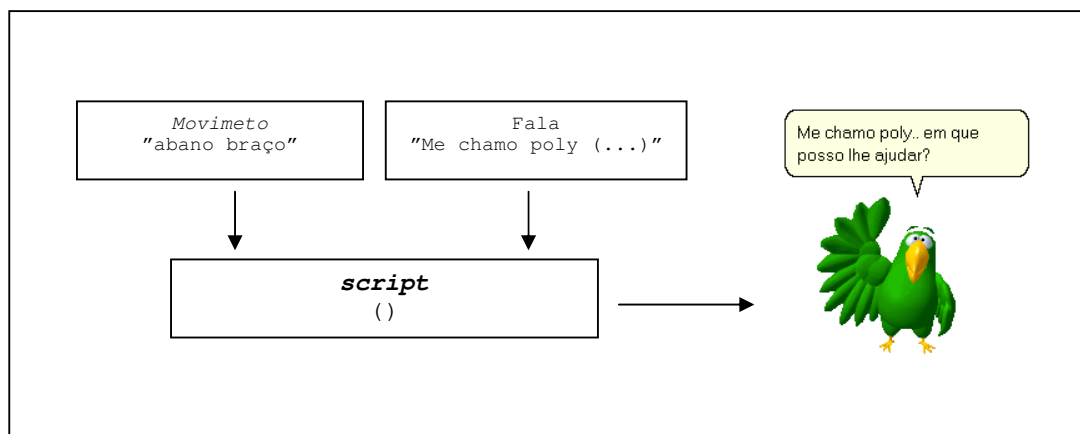


Figura 16 – Funcionamento do agente da *Microsoft*.

O agente do protótipo não só pode ser utilizado para apresentar as respostas referentes a utilização do *Learnloop* que são obtidas da interação do usuário com o *chatterbot*, mas ele também pode trabalhar como um tutor do ambiente repassando informações necessárias ao usuário assim que ele acesse determinada ferramenta ou local do ambiente.

Além do que, o agente pode ser facilmente desabilitado também pelo usuário para que não se torne inconveniente àqueles mais experientes e que não necessitam de ajuda para utilizar o *Learnloop*.

#### 5.3.1.5 Classe de Comunicação

Para que as partes envolvidas no protótipo se comuniquem foi adaptada uma classe *open source*, desenvolvida em *php*, para desempenhar este papel de integração. A classe é responsável por transmitir as requisições (perguntas) do usuário para o *chatbot* que faz o relacionamento da sentença com as sentenças disponíveis nos arquivos *AIML* e devolve para a classe a resposta que será exibida pelo agente de interface.

Esta classe foi empregada para servir de canal de comunicação e também de facilitar o envio de comandos de movimento ao agente da *Microsoft* sem a necessidade de alterações mais profundas no *bot* utilizado no protótipo.

As ações de movimento são incluídas no arquivo *AIML* juntamente com a resposta separada por um curinga especial (#) que é reconhecido pela classe de comunicação que faz o tratamento necessário e envia o texto com a resposta e o comando de ação - formatados para o padrão que o agente de interface exige.

#### 5.3.1.6 Usuários

Os usuários definidos neste protótipo são os mesmos identificados no ambiente virtual de aprendizagem da Unesc (*Learnloop*): o usuário Administrador, o usuário Professor e o usuário Aluno.

O **Administrador** no *Learnloop* tem o papel de gerenciar o ambiente administrando usuários, salas e ferramentas. Neste protótipo o seu papel continua o mesmo da administração do ambiente virtual, mas ainda tornando-o responsável dos seguintes pontos:

- Habilitar ou desabilitar de forma global para o ambiente o agente de interface;
- incremento e definição da base de conhecimento do *chatterbot*;
- administrar os *logs* de conversação que auxiliam no *tunning* da base de conhecimento;
- definição das mensagens padrões para o agente de interface,
- gerenciar o agente de interface, como tutor do ambiente, definindo quando e em quais páginas ele deve aparecer automaticamente e também quais mensagens e movimentos apresentará.

Os **Professores** e **Alunos** interagem com o protótipo apenas controlando a presença da ajuda do agente de interface; habilitando ou desabilitando sua aparição no ambiente e enviando questões por meio do *chat* de conversação.

Na próxima sessão, onde se apresentarão funcionalidades do protótipo em uma aplicação usual, será possível observar mais claramente o papel de cada usuário na interação com o protótipo e deste com o ambiente virtual.

## 5.4 Funcionamento do Protótipo

Esta sessão descreve o funcionamento aplicado do protótipo e suas características funcionais, na interação com seus usuários finais e com o ambiente virtual onde está inserido, como segue.

### 5.4.1 Ambiente do Protótipo

O protótipo se integra ao Ambiente Virtual – *Learnloop*, e cumpre o papel principal de auxiliar os usuários do ambiente na utilização das ferramentas e

funcionalidades tanto de forma automática ou como tutor, ou ainda na solicitação direta do usuário.

Algumas de suas características notórias foram imprescindíveis para o seu prosseguimento, como descritos abaixo:

**Para os usuários do ambiente virtual da instituição:**

- O usuário não necessita de conhecimentos maiores para utilizar o protótipo na busca de informações e ajuda. Isso torna a utilização do protótipo muito simples.

**Para os desenvolvedores e o suporte tecnológico:**

- Independência de plataforma e modularidade suficiente para possibilitar que os agentes envolvidos possam ser implantados em máquinas separadas geograficamente.
- Código fonte aberto facilita qualquer customização necessária para futuras implementações e melhorias necessárias para funcionalidades específicas da instituição.

**Para a Instituição:**

As tecnologias empregadas abrem uma área para estudo da comunidade acadêmica e produzem resultados concretos que podem ser utilizados dentro da própria instituição, já que a área de educação a distância e o ambiente virtual da Instituição estão em aperfeiçoamento e estudo pela comunidade.

### 5.4.2 Recursos disponíveis:

O protótipo implementado possui alguns recursos particulares e principais, a saber, como:

**Administração Geral:** o *bot* pode ser configurado através de parâmetros disponíveis no formulário de administração do *bot*, integrado ao *Learnloop* (Figura 17), que configuram as características gerais de funcionamento do protótipo;

**Entrada das perguntas:** por meio das sentenças enviadas pelo formulário disponível no topo do ambiente virtual (Figura 14) o usuário interage diretamente com o *bot*;

**Exibição das respostas e mensagens:** o agente de interface da *Microsoft* exibe as respostas enviadas pelo *bot* em balões de explicação conforme a exibido anteriormente na Figura 15.

O **Administrador** do ambiente virtual configura o protótipo acessando a área de administração do ambiente no formulário disponível na tela de configuração do protótipo onde pode ser indicado de forma visual o caminho do arquivo *AIML* responsável pela base de conhecimento como destaca a Figura 17.

Neste mesmo local ele deve selecionar um personagem, disponível na lista, para apresentação das mensagens e respostas ao usuário. Outra possibilidade deste formulário é a de poder habilitar ou desabilitar a apresentação do agente de interface globalmente no ambiente virtual.

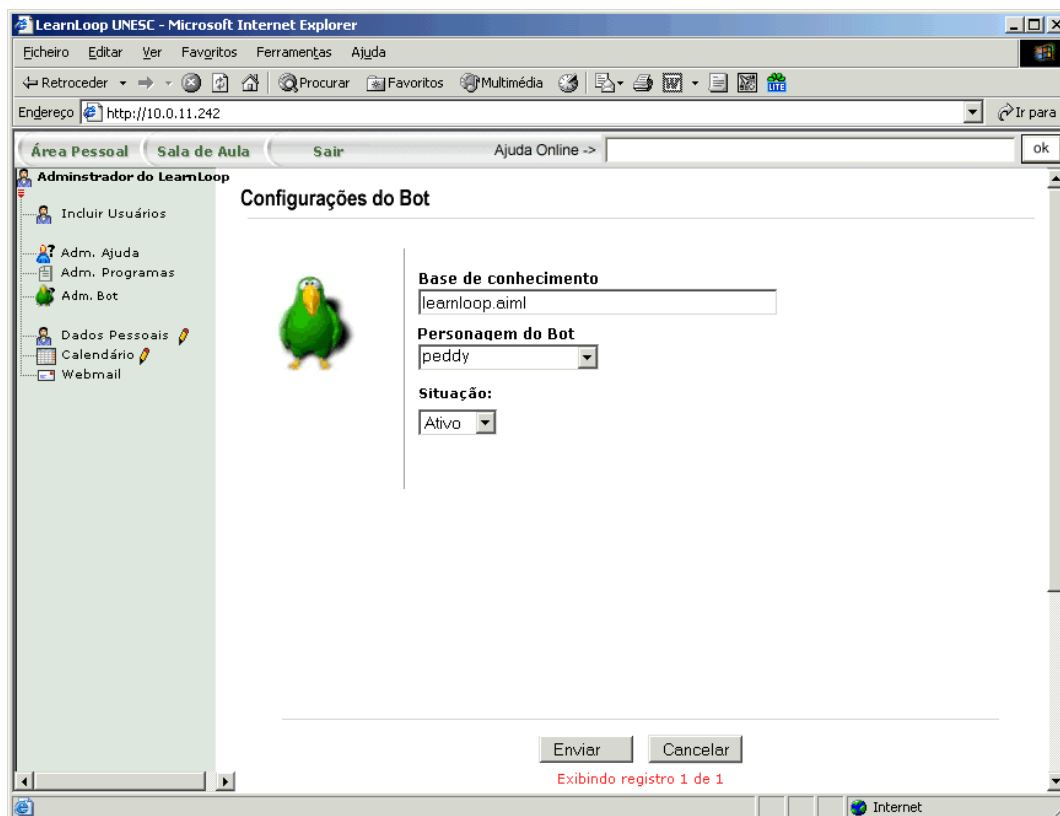


Figura 17 – Formulário de configuração do *bot*.

Outras configurações do *chatbot* não possuem uma interface visual para administração. A edição dos arquivos *AIML* que compõem a base de conhecimento devem ser editados com outras ferramentas de edição independente do ambiente virtual.

Da mesma forma, os *logs* de conversação, que são gerados a cada sessão de interação do usuário com o *chatbot*, devem ser administrados.

Conforme observado na Figura 18, os *logs* de conversação são basicamente arquivos de textos que registram o par de sentenças: questão do usuário e resposta do *chatbot* para esta questão. Por tanto, analisando estes arquivos o administrador tem também a possibilidade de melhorar a base de conhecimento do *chatbot*, dando maior credibilidade na ajuda ao usuário.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="../../resources/logs/view-
chat.xsl"?>
<!DOCTYPE ALLOW_HTML_ENTITIES [ <!ENTITY % HTMLlat1 PUBLIC "-
//W3C//ENTITIES Latin1//EN//HTML"    "../resources/DTD/xhtml-lat1.ent">
%HTMLlat1; <!ENTITY % HTMLsymbol PUBLIC    "-//W3C//ENTITIES
Symbols//EN//HTML"    "../resources/DTD/xhtml-symbol.ent"> %HTMLsymbol;
<!ENTITY % HTMLspecial PUBLIC    "-//W3C//ENTITIES Special//EN//HTML"
"../resources/DTD/xhtml-special.ent"> %HTMLspecial; ]>
<exchanges>
  <exchange>
    <timestamp>2005-04-09 15:56:52</timestamp>
    <userid>localhost</userid>
    <clientname>user</clientname>
    <botid>Learnloop</botid>
    <input>boasvindas</input>
    <response></response>
  </exchange>
  (...)
  <exchange>
    <timestamp>2005-04-09 16:29:06</timestamp>
    <userid>localhost</userid>
    <clientname>user</clientname>
    <botid>Learnloop</botid>
    <input>como criar um login?</input>
    <response>Você deve estar matriculado em algum curso da Unesc. Qual
seu curso?</response>
  </exchange>
</exchanges>

```

Figura 18 – Arquivos de log de conversação.

O administrador ainda pode configurar o agente para atuar como um tutor fazendo com que ele apresente de forma automática mensagens em determinados locais de acesso (telas) do usuário.

Na tela de administração de Mensagem de Ajuda (Figura 19) é possível informar em qual local, qual mensagem e movimento o agente de interface deve apresentar ao usuário. Esta mensagem será apresentada apenas aos usuários que não tenham desabilitado o suporte de ajuda *online* na tela de configuração pessoal (Figura 20).

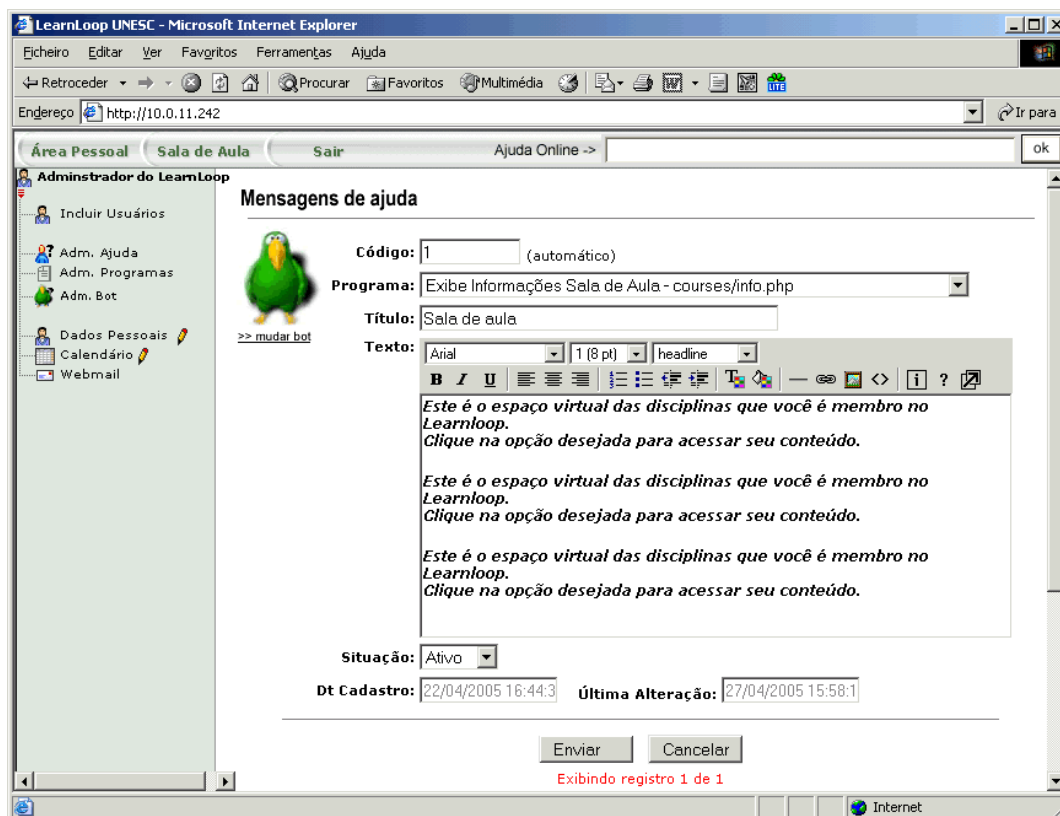


Figura 19 – Tela de administração das mensagens de ajuda.

Qualquer usuário, mesmo não cadastrado ou autenticado no ambiente virtual, pode interagir com o *chatbot*, simplesmente enviando sua pergunta pelo formulário de ajuda *online* disponível no topo do ambiente virtual (figura 14).

Por sua vez, a opção de disponibilizar a ajuda já na página de *login* do ambiente, mesmo a usuários não autenticados, deve-se à necessidade levantada na análise dos *emails* de suporte recebidos pela equipe do Sead. O número de *emails* solicitando auxílio para acessar o ambiente virtual é muito significativo. Muitos usuários não sabem como encontrar os dados de *login* no ambiente, onde solicitar uma nova senha ou mesmo como proceder para se cadastrar no ambiente.



Já aos usuários autenticados no ambiente tem disponível (Figura 20) um *link* que possa habilitar ou desabilitar a apresentação do agente de interface como tutor.

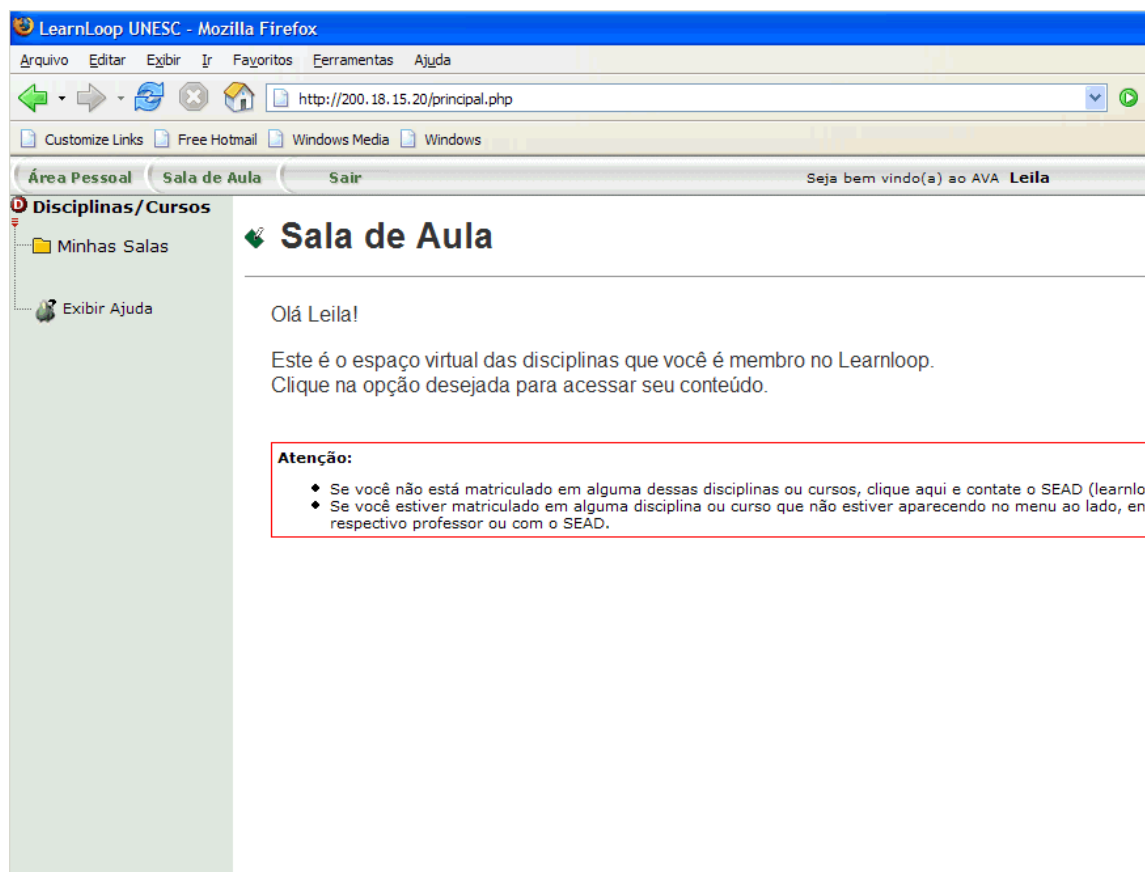


Figura 20 – Administração da apresentação da ajuda *on-line*.

Por isso, esta configuração foi incluída ao protótipo para que não se torne incômodo à utilização do agente como tutor aos usuários mais experientes.

### 5.5 Fluxo de uma interação do usuário com o *chatbot*

Todo o processo de interação entre usuário e *chatbot* inicia no momento que o usuário faz a solicitação por meio da interface de interação

apresentada anteriormente na Figura 14. Ao entrar com a sentença “Como criar um login?”, e enviar o formulário, o usuário dá início ao processo de interação com o protótipo. A sentença é então enviada para classe de comunicação que referencia o *bot* do *chatbot* a encaminhar a sentença do usuário como entrada.

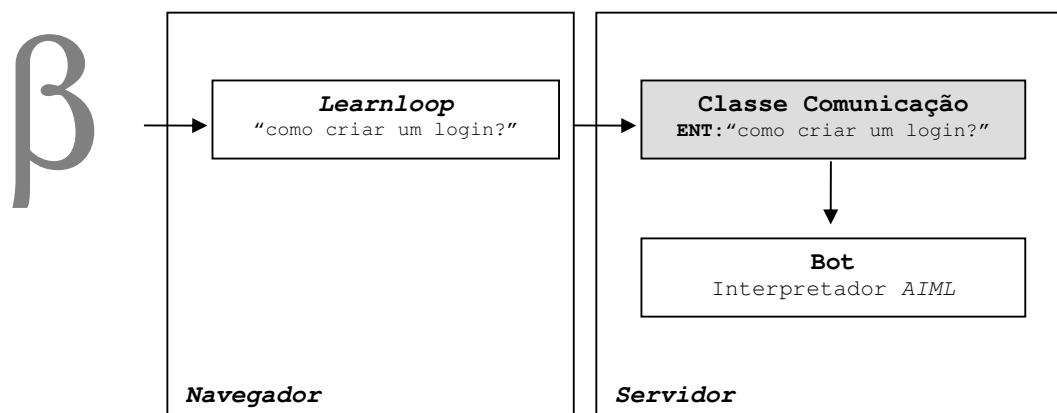


Figura 21 – Fluxo de interação: Usuário com Classe de Comunicação.

Assim o *bot* recebe a sentença e faz as reduções, traduções e referências necessárias com base nas categorias especificadas no arquivo *AIML* (Figura 22) da base de conhecimento do *chatbot* e retorna a sentença com a resposta e comando de movimento para a classe de comunicação conforme a Figura 23.

```
<category>
<pattern>* CRIAR UM LOGIN *</pattern>
  <template>
    Para ser cadastrado você deve solicitar
    a um professor que inclua você em sua sala de aula.
    #GestureRight
  </template>
</category>
```

Figura 22 – Parte do Arquivo *AIML*.

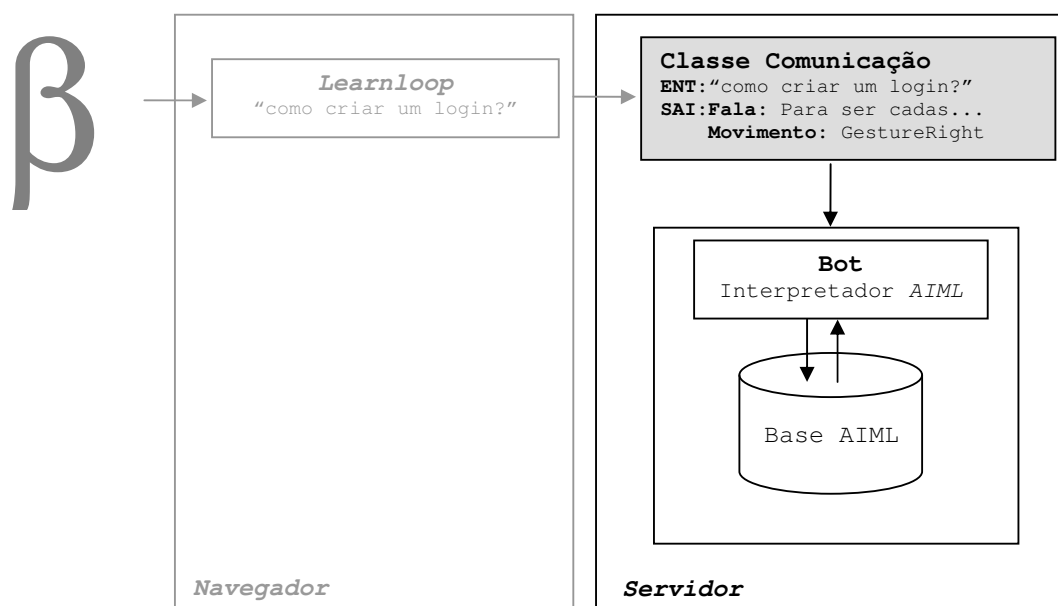


Figura 23 – Fluxo de interação: Classe de Comunicação com o *Chatterbot*.

Neste momento a classe de comunicação separa o comando que irá definir o movimento do agente de interface da sentença que se apresentará no balão de texto. Isto pode ser visto representado na figura a seguir.

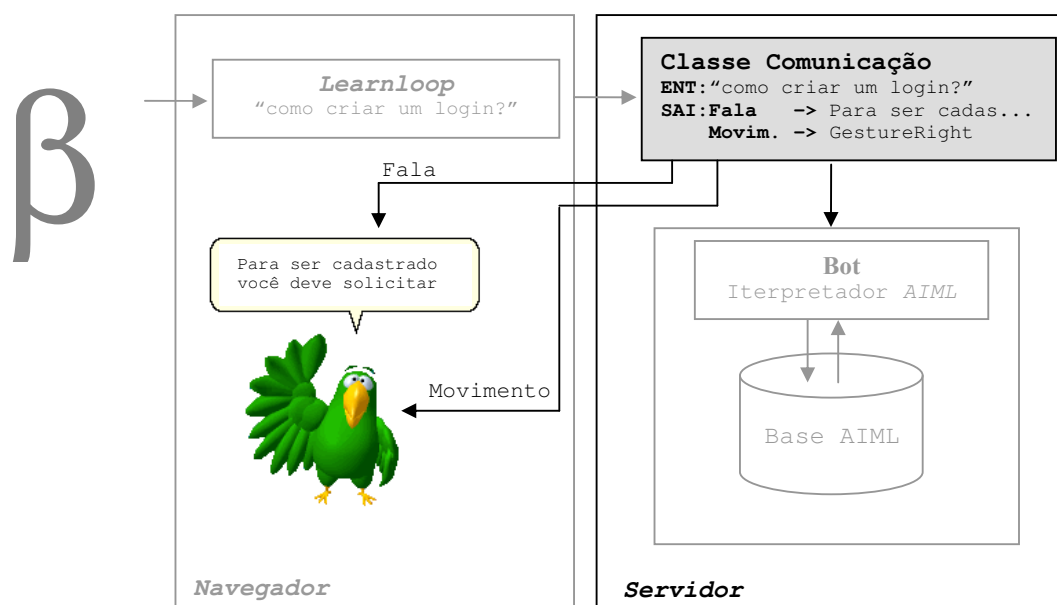


Figura 24 – Fluxo de interação: Classe de Comunicação com o Agente Animado.

Conforme comentado antecipadamente o agente também pode ser utilizado como tutor no auxílio automático e aparecer sem a solicitação do usuário. Ele assim pode apresentar dicas, como por exemplo, de como utilizar melhor uma determinada ferramenta do ambiente. Isto acontece a partir do momento que o usuário acessa uma página do ambiente que possua uma entrada de mensagem automática. Estas mensagens e relacionamento com determinada página devem ser cadastradas anteriormente pelo administrador do ambiente, como pode ser observado anteriormente na Figura 19.

Uma destas intervenções pode ser observada na próxima figura, quando dá dicas de utilização da área pessoal para o usuário assim que ele a acessa.



Figura 25 – Agente atuando como tutor do ambiente virtual.

Na figura anterior pode ser observado o agente de interface atuando como tutor do ambiente virtual, dando dicas de utilização da área pessoal para o usuário assim que ele acessa esta área.

Da mesma maneira, o agente ficará visível ao usuário na tela enquanto não solicitar que ele seja oculto. Esta solicitação pode ser feita com o *botão* direito do mouse sobre o próprio agente (Figura 26).



Figura 26 – Atalho para ocultar o agente de interface.

Quando oculto o agente fica disponível para o usuário em um ícone na barra de tarefas do sistema para que o usuário possa solicitar a sua apresentação novamente. Mas enquanto o agente aguarda solicitações do usuário ele também executa gestos aleatórios que passam a impressão de que está aguardando alguma interação do usuário. Um exemplo é (Figura 27) onde o personagem aparece comendo uma bolacha enquanto aguarda interagir com o usuário.



Figura 27 – Agente de interface em modo de espera.

Desta maneira, apresentou-se o funcionamento e o fluxo de uma interação do usuário com o *chatterbot* do protótipo, creditando assim a todos os mecanismos e funcionalidades do projeto a real efetivação por qualquer usuário do ambiente.

## 6 CONCLUSÃO

Com a disseminação da educação a distância, é fundamental buscar soluções que melhorem a qualidade dos serviços e funcionalidades oferecidos nesta modalidade de ensino. Uma maneira de contribuir neste processo é tornar mais transparente e acessível o uso da tecnologia aos usuários bem como, humanizar as trocas de informação nas interações dentro do ambiente, utilizando formas de comunicação essencialmente humanas.

Esta pesquisa apresentou um estudo, baseado no levantamento bibliográfico, e um protótipo que aplicou Inteligência Artificial, em específico a tecnologia de agentes, para tornar mais eficiente e eficaz a utilização de ambientes virtuais. Duas problemáticas foram abordadas: o auxílio na utilização das funcionalidades do AVA e a introdução de elementos humanos no diálogo entre usuário e ambiente.

A proposta apresentada nesta pesquisa para humanização dos diálogos integra duas tecnologias – Linguagem Natural e agentes animados de interface. Esta integração visa tornar mais amigável e prazerosa a experiência de interação com o ambiente virtual de aprendizagem inserindo elementos humanos no processo de comunicação como gestos, fala, feições e movimentos. Dessa forma, busca-se fornecer um canal intuitivo e, ao mesmo tempo, motivador no processo de interação entre o usuário e o ambiente.

Para auxiliar o usuário na utilização das funcionalidades do AVA é necessário prever as possíveis dúvidas e prover instruções de uso para solucioná-las. As dificuldades podem ser inerentes à inexperiência, ou ainda, pouca prática de

grande parte dos usuários em relação aos recursos do ambiente. Porém, elas podem se multiplicar e se diferenciar de acordo com cada necessidade do utilizador. Neste caso, é necessário uma base de conhecimento que possa ser ampliada a partir de novos questionamentos e dúvidas. No protótipo desenvolvido foi criada uma base de conhecimento com a linguagem de marcação AIML, que engloba basicamente a funcionalidade de inclusão de usuários no AVA *Learnloop* mas que pode ser facilmente ampliada.

Integrando *chatterbots* e agentes animados de interface no ambiente se conseguiu agregar qualidade e humanizar a interação com o usuário, tornando mais natural e facilmente compreensível as respostas enviadas pelo sistema.

Enfim, deve ficar claro que toda essa tecnologia de interpretação da linguagem natural, característica dos *chatterbots*, empregada em conjunto com os recursos multimídia dos agentes animados, não garante o aprendizado. Estes devem estar inseridos em um ambiente maior de colaboração e com técnicas pedagógicas específicas para tornarem ainda mais produtivo o processo educacional.

De acordo os estudos explanados, aponta-se para que nos próximos anos estas tecnologias possam se transformar em recursos importantes no processo de educação a distância. Desta forma, podendo assim ser utilizados em ambientes virtuais como elemento de motivação despertando o interesse para este modelo de ensino e oferecendo alternativas mais simples, naturais e transparentes de relacionamento entre os usuários e o sistema.

## REFERÊNCIAS

ALICE. **The A.L.I.C.E. Foundation**. Disponível em: <<http://www.alicebot.org>>. Acesso em out. 2004.

ANNABOT. **Site Oficial**. Disponível em: <<http://www.annabot.org>>. Acesso em jan. 2005.

BOCCA, Everton; JAQUES, Patrícia; VICARI, Rosa. Modelagem e Implementação da Interface para Apresentação de Comportamentos Animados e Emotivos de um Agente Pedagógico Animado. **Revista Novas Tecnologias na Educação**. Porto Alegre, v. 1, n.2. Disponível em: <<http://www.cinted.ufrgs.br/renote/set2003/artigos/modelagemeimplementacao.pdf>>. Acessado em nov 2004.

COELHO, Luciano Roth *et al.* **Junior, um chatterbot para educação a distância**. Universidade Federal do Rio Grande do Sul. Disponível em: <<http://www.niee.ufrgs.br/ribie2000/papers/255/>>. Acesso em jan. 2004.

CONVERSAGENT. **Optimizing Customer Service**. Disponível em: <<http://www.conversagent.com>>. Acesso em nov. 2004.

COSTA, Marcello Thiry Comicholi da. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino a Distância**. Florianópolis: PPGEF da UFSC, 1999. Tese de Doutorado. Disponível em: <<http://www.eps.ufsc.br/teses99/thiry/index.htm>>. Acesso em dez. 2004.

DEBRA, 2002. Disponível em: <http://debra.dgibt.doc.ia/>

DRUCKER. **Administrando em tempos de grandes mudanças**. SP, Brasil, 1995.

DYCK, André Fabiano *et al.* **Chatterbot – Robôs que Conversam**. Universidade Federal de Santa Catarina. Disponível em: <<http://www.ufsc.br/>>. Acesso em jan. 2004.



LAGE, Nilson. **Processamento de Linguagem Natural**. [on-line]. UFSC. Disponível em: <<http://www.jornalismo.ufsc.br/bancodedados/lage-natural.html>>. Acesso em 10 out. 2004.

LAUREANO, E. A. G. C. **ConsultBot - Um Chatterbot Consultor para Ambientes Virtuais de Estudo na Internet**. [on-line]. Recife, PE, Brasil, Agosto de 1999. Disponível em: <<http://www.di.ufpe.br/~tg/1999-1/eagcl.doc>>. Acesso em: out. 2004.

LEARNLOOP. **Ambiente Virtual de Educação a Distância**. Disponível em: <<http://www.Learnloop.org>>. Acesso em jan. 2005.

MICROSOFT. **Microsoft Windows Technologies – Microsof Agent**. Disponível em: <<http://www.microsoft.com/msagent>>. Acesso em set. 2004.

PARRA, Jesús Heras. **Utilización de agentes animados para interfaces avanzadas de ayuda**. Madrid. Grupo de Tecnología del Habla da Universidad Politécnica de Madrid. Disponível em: <<http://lorien.die.upm.es/~juancho/pfcs/jhp/Informaci%A2n%20del%20proyecto.doc>>. Acesso em fev 2005.

PRIMO, Alex Fernando Teixeira. **Chatterbots: Robôs de Conversa**. Disponível em: <<http://cybelle.cjb.net>>. Acesso em mar. 2005.

SEEWALD, Henrique Smidt Seewald. **Estudo e definição dos critérios para Construção de um Agente formador de Grupos Colaborativos - Projeto AMIA**. 2004. 50 f. Monografia (Tecnologia em Informática) - Universidade Luterana do Brasil, Canoas.

SGANDERLA, Rachele et al. **BonoBOT: Um Chatterbot para Interação com Usuários em um Sistema Tutor Inteligente**. Disponível em: <<http://www.nce.ufrrj.br/sbie2003/publicacoes/paper46.pdf>>. Acesso em out. 2004.

SILVA, Adriana Barbosa da. **Um Chatterbot em AIML Plus que conversa sobre horóscopo**. PE, Brasil, Setembro de 2003.

TECNOCIÊNCIA. Site Oficial. [on-line]. **Chatterbots, robôs do diálogo**. Disponível em: <<http://www.tecnociencia.com.br/comunidade>>. Acesso em out. 2004.

TOMABECHI, WALLACE. Hideto, Richard. **Chatterbots Go Native**: Considerations for an eco-system fostering the development of artificial life forms in a human world. Disponível em: <  
<http://www.pandorabots.com/pandora/pics/chatterbotsgonative.doc>>. Acesso em out. 2004.

VARGAS, Ariel; RAABE, André Luís Alice. Desenvolvimento, aplicação e Análise de Agentes Animados de Interface em um *Software* Educacional. In: **III CONGRESSO BRASILEIRO DE COMPUTAÇÃO** - CBCOMP2003, 2003, Itajaí-SC. Anis. 2003. v. 1, p. 1370-1381.

WALLACE. Richard. **AIML Overview**. Disponível em: <  
<http://www.pandorabots.com/pandora/pics/wallaceAIMLtutorial.html>>. Acesso em nov. 2004.

\_\_\_\_\_. **Don't Read Me - A.L.I.C.E. and AIML Documentation**. Disponível em: <[www.alicebot.org/articles/wallace/dont.html](http://www.alicebot.org/articles/wallace/dont.html)>. Acesso em out. 2004.

W3C, World Wide *Web* Consortium. **eXtensible Markup Language**. Disponível em: <<http://w3c.org/XML/>>. Acesso em nov. 2004.

