UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ALOI: um Agente para a Localização e Organização de Informações

por

MARCELO FAORO DE ABREU

Dissertação submetida à avaliação,
como requisito parcial, para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, janeiro de 2003.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Abreu, Marcelo Faoro de
ALOI: um Agente para a Localização e Organização de Informações / por Marcelo Faoro de Abreu – Porto Alegre : PPGC da UFRGS, 2003.
82 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2003. Orientador: Geyer, Cláudio Fernando Resin.

1. Agentes Web. 2. Classificação de textos. 3. Perfil de usuário. 4. Extração de informações. 5. Internet. I. Geyer, Cláudio Fernando Resin. II. Título

Agradecimentos

Vencida mais esta etapa do constante processo de aprendizado é de suma importância agradecer e homenagear àqueles que, talvez sem mesmo saber, muito colaboraram para a realização deste trabalho.

Ao meu orientador, professor Cláudio Geyer, que com muita paciência e sabedoria me guiou pelo caminho correto. Além de orientador foi um grande amigo, ajudando a superar as dificuldades impostas pela falta de experiência e a própria distância.

À Universidade de Caxias do Sul, pela viabilização do curso e pelo apoio, sem o qual talvez não fosse possível a sua realização.

Aos meus pais, pelo apoio e incentivo que sempre me foi dado, além dos exemplos de determinação, dignidade, esperança e justiça pelos quais sempre me guiei durante toda a vida.

À minha namorada, Roberta, pelo companheirismo e compreensão dos momentos em que estive ausente, tanto para a realização das disciplinas quanto da própria dissertação.

A Deus, pela existência e pela oportunidade de realizar tanto esta quanto outras realizações e trabalhos.

Sumário

Lista de Abreviaturas

HTML	Hipertext Manager Language
HTTP	Hipertext Tansfer Protocol
IA	Inteligência Artificial
KBS	Knowledge Base System
TFIDF	Term Frequency Times Inverse Document Frequency
URL	Uniform Resouce Location
WWW	World Wide Web

Lista de Figuras

Lista de Tabelas

Resumo

Este trabalho é um estudo sobre agentes inteligentes e suas aplicações na Internet. São apresentados e comparados alguns exemplos de *software* com funcionalidades para extrair, selecionar e auxiliar no consumo de informações da Internet, com base no perfil de interesse de cada usuário. O objetivo principal deste trabalho é a proposição de um modelo geral e amplo de agente para a obtenção e manutenção de um repositório de *links* para documentos que satisfaçam o interesse de um ou mais usuários. O modelo proposto baseia-se na obtenção do perfil do usuário a partir de documentos indicados como modelos positivos ou negativos. O ponto forte do modelo são os módulos responsáveis pela extração de informações da Internet, seleção quanto a importância e armazenamento em banco de dados das URLs obtidas, classificadas quanto a usuário, categoria de assunto e assunto. Além disso, o modelo prevê a realização de freqüentes verificações de integridade e pertinência dos *links* armazenados no repositório. Com base no modelo proposto foi implementado um protótipo parcial. Tal protótipo contempla os módulos responsáveis pela obtenção de informações, seleção das informações pertinentes e classificação e armazenamento dos *links* de acordo com o assunto. Finalmente, o protótipo implementado permaneceu em execução por um determinado período, gerando alguns resultados preliminares que viabilizaram uma avaliação do modelo.

Palavras-chaves: agentes Web, classificação de textos, perfil de usuário, extração de informações, Internet.

TITLE: "ALOI – AN INFORMATION SEARCHER AND ORGANIZATOR AGENT".

Abstract

This work is a study about the intelligent agents and their applications in the Internet. It is presented and compared some examples of software with fuctions to extract, select and help the consume of the information from the Internet, based on the interest profile of each user. The main objective of this paper is the proposition of a general and wide model of the agent to the obtaining and maintenance of a repository of links to documents that satisfy the interest of one or more users. The proposed model is based on the obtaining of the user's profile from the indicated documents as positive or negative models. The strong points of this model are the responsible modules of the information extraction from the internet, selection of the importance and storage in database of the obtained URLs, classification of the user, subject category and subject. Furthermore, the model is in charge of frequently checking the integrity and relevance of the links stored in the repository. Based on the porposed model, it was implemented a parcial prototype. Such prototype considers the responsible module by the obtaining of the information, selection of the relevant information and classification and storage of the links according to the subject. Finally, the implemented model remained in execution for a determined period of time, generating some preliminary results that make viable an evaluation of the model.

Keywords: WEB agents – Texts classification, users profile, information extraction, Internet.

1 Introdução

O tema do presente trabalho é um estudo sobre agentes inteligentes e suas aplicações na Internet e a comparação entre alguns softwares existentes neste gênero. A partir deste estudo definiu-se a proposição de um novo modelo de agente capaz de obter o perfil de interesse do usuário, buscar informações que satisfaçam a este perfil e manter atualizado um repositório de *links* para cada usuário, sendo que o enfoque principal da proposta está obtenção e classificação de documentos. Com base nesse modelo foi implementação um protótipo parcial para avaliação e validação do mesmo.

1.1 Motivação

Nos últimos anos observou-se que, devido à grande evolução da tecnologia e os fortes investimentos realizados no setor de telecomunicações, a Internet passou a ter uma crescente utilização e importância para disponibilização e divulgação de dados e, principalmente, conhecimentos, o que tornou a Web uma fonte inesgotável de informações para as mais diversas e variadas áreas do conhecimento humano [THE 98].

A grande velocidade e dinamismo com que as informações estão sendo disponibilizadas na rede criaram uma certa dificuldade para os usuários que mantêm páginas de *links* sobre assuntos específicos. A manutenção destas páginas tem se tornado um processo que cada vez mais consome tempo de seu responsável, além de abrir margem para que as páginas permaneçam desatualizadas em relação ao conteúdo que está disponível[ABR 2000].

Atualmente, existem inúmeras aplicações e protótipos de modelos[KRU 97], baseados em agentes inteligentes, que têm como objetivo auxiliar o consumo das informações disponibilizadas na Web [PAZ 99]. Estas aplicações possuem as mais variadas finalidades, desde realizar simples buscas a partir de termos informados pelos usuários, até realizar o aprendizado das preferências pessoais [AVG 97] de cada usuário e, baseadas nisto, efetuarem buscas de informações que atendam às necessidades do usuário.

Entretanto, existe a carência de uma aplicação que tenha como objetivo a obtenção de conhecimentos sobre o perfil de um usuário em específico e, a partir destes conhecimentos, comunicar-se com outros agentes disponíveis na Web a fim de obter novos *links* e adicioná-los em repositórios, além de manter a consistência dos *links* já existentes. A necessidade da proposição de um agente com tais características motiva a realização de uma pesquisa, mais aprofundada, sobre as técnicas e os parâmetros que podem ser utilizados para a criação de um modelo de agente, capaz de manter os repositórios de *links* atualizados, bem como a própria concepção de um novo modelo de agente que supra tais carências.

1.2 Proposta

O objetivo principal deste trabalho é a apresentação de um modelo genérico e amplo de agente, capaz de adquirir conhecimentos sobre as preferências de um ou mais usuários, que mantêm páginas de *links*. A partir dos conhecimentos adquiridos, o agente deverá ser capaz de monitorar a navegação de cada usuário na Web, para enriquecer o conhecimento sobre seu perfil, e comunicar-se com outros agentes disponíveis, a fim de obter *links* que satisfaçam ao perfil de interesses do usuário e possam ser adicionados ao repositório. Além disto, o agente deverá monitorar os *links* já existentes no repositório com o objetivo de manter a integridade de referência destes *links*.

A especificação deste modelo apresenta algumas particularidades, pois leva em consideração o fato de que uma porção do agente seria implementada para fins de validação e observação de comportamento. Porém o foco principal é dado ao modelo e não à implementação prevista.

Destacam-se como principais contribuições deste trabalho:

- A proposta de um modelo de extração de conhecimentos a partir de documentos extraídos na Internet, utilizando mecanismos que tenham como base a quantidade de termos e sua formatação;
- A proposta de uma estrutura de armazenamento para modelos de usuários e repositório de *links*;
- A definição de critérios de seleção e classificação de assuntos, bem como as metodologias para a obtenção dos valores para cada critério;
- A definição de um mecanismo de constante verificação de integridade dos *links* armazenados em repositórios;
- A integração de diferentes metodologias, apresentadas na bibliografia, em um modelo único e com a possibilidade de substituição de qualquer uma destas metodologias por outras, de acordo com as necessidades específicas.

1.3 Organização do Texto

O restante do presente texto está dividido em cinco capítulos. No capítulo 2, são apresentados os principais conceitos e características de Agentes e Modelos de Usuários, dando-se um enfoque maior aos Agentes Web, suas classificações e aplicações. No capítulo 3, é apresentado um estudo sobre as características, funcionalidades e metodologias utilizadas por alguns agentes Web comerciais ou acadêmicos.

No capítulo 4, encontram-se as principais contribuições deste trabalho. Neste capítulo está descrita a especificação do modelo proposto para a busca, seleção, classificação e validação de *links*. No capítulo 5 são apresentados o protótipo implementado, suas características e interfaces de utilização, e ainda uma avaliação preliminar dos resultados obtidos. Finalmente, no capítulo 6, são sintetizadas as conclusões deste trabalho e apresentados alguns apontamentos para possíveis trabalhos futuros.

2 Agentes e Modelo de Usuário

Neste capítulo é apresentada uma revisão bibliográfica sobre agentes e modelos de usuários. Inicialmente, são descritos os conceitos básicos de agentes inteligentes de uma forma geral, suas características e propriedades, em seguida são apresentadas as diversas categorias de agentes inteligentes que atuam sobre a Internet. Finalmente, são descritos os conceitos de Modelo de Usuário, sua importância, seus requisitos e formas de aprendizagem.

2.1 Considerações Iniciais

A grande quantidade de informações disponíveis na Internet fez surgir um problema para os usuários de computadores pessoais, que, passaram a ter uma certa dificuldade em localizar suas informações com precisão e rapidez. Observando este problema, pesquisadores da área de Inteligência Artificial começaram a trabalhar no sentido de desenvolver aplicações que utilizam agentes inteligentes capazes de obter o modelo do usuário, ou seja, seu perfil de interesse, e, a partir deste modelo, realizar buscas de informações na rede de acordo com as preferências e interesses de cada usuário.

2.2 Definição de Agentes

O termo agente, em computação recebe por parte dos autores definições diferenciadas[FRA 97]. Entre muitas definições citamos algumas como:

Um agente é algo que pode ser visto e perceber seu ambiente através de sensores e agir sobre este ambiente através de atuadores.

Stuart Russell & Peter Norving

.

Agentes autônomos são sistemas computacionais que habitam algum ambiente dinâmico e complexo, percebem e agem autonomamente neste ambiente, realizando um conjunto de objetivos ou tarefas pelos quais foram projetados.

Pattie Maes

Definimos um agente como uma entidade de software persistente dedicada a um propósito específico. Persistência distingue agentes de subrotinas; agentes possuem suas próprias idéias sobre como realizar tarefas, agendadas por eles próprios. Propósito especial distingue-os de aplicações multifuncionais; agentes são, tipicamente, muito menores.

D. Smith, ª Cypher & J. Spohrer

Agentes inteligentes executam, continuamente, três funções: percepção das condições dinâmicas do ambiente; ações que afetam as condições do ambiente; e raciocínio para interpretar as percepções, resolver os problemas, extrair inferências e determinar ações.

Barbara Hayes-Roth

Analisando as diversas definições de agentes e formalizando uma definição que constitui a essência do agente, segundo os autores, um agente autônomo é um sistema situado nos limites de um ambiente, do qual faz parte, que percebe e age sobre este ambiente, procurando executar sua agenda e causando efeito que pode ser percebido futuramente[FRA 97].

Segundo esta definição, um software para ser um agente deve: agir pela própria agenda, perceber e causar efeito sobre o seu ambiente e agir, de forma autônoma, durante um período de tempo. Essas características diferenciam-no de um programa de software que, normalmente, age somente quando é ativado, produz saídas segundo as entradas fornecidas.

2.3 Características dos Agentes Inteligentes

Existe um consenso entre a comunidade de IA em relação às características que um agente deve possuir. Brenner [BRE 98] classifica as características em duas grandes categorias: propriedades internas e propriedades externas (Fig. 2.1).

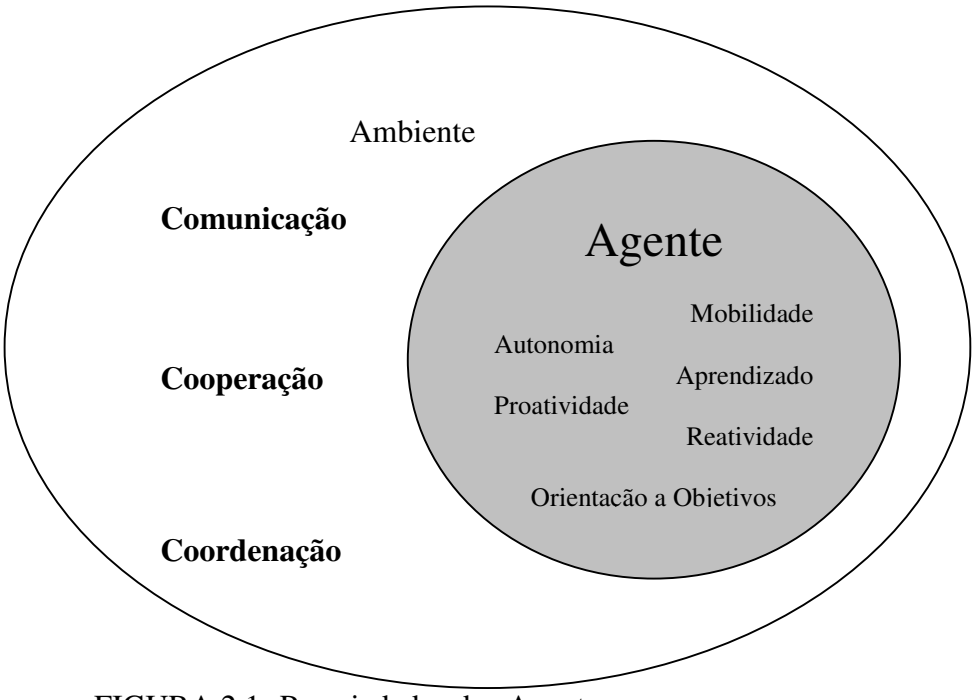


FIGURA 2.1- Propriedades dos Agentes

Propriedades internas são aquelas que compõem o agente, isto é, as propriedades que determinam as ações a serem executadas pelo agente. Propriedades internas incluem a habilidade de aprendizado, reatividade, autonomia e orientação ao objetivo. Propriedades externas são aquelas que pertencem ao meio onde o agente está inserido. Elas incluem todas aquelas que dizem respeito à comunicação entre os agentes com os usuários e entre os agentes com outros agentes. Dentre as principais propriedades, [BRE 98] cita reatividade, orientação a objetivos, aprendizado, autonomia, mobilidade, cooperação e comunicação:

- **Reatividade:** A propriedade de Reatividade possibilita ao agente a capacidade de reagir apropriadamente às influências ou informações do seu ambiente. Este ambiente pode consistir de outros agentes, usuários humanos, fontes externas de informações ou até mesmo de objetos físicos.

- **Orientação ao Objetivo :** A propriedade de Orientação ao Objetivo faz com que o agente trabalhe em função de perseguir objetivos. Normalmente um objetivo, ou meta, é subdividido em vários sub-objetivos a fim de tornar a resolução do problema mais simples.

- **Aprendizado:** Cada agente deve ter um mínimo de inteligência. A inteligência dos agentes possui uma grande variação de tipos, ou seja, podemos ter agentes bastante simples e com inteligência bem limitada até agentes bastante complexos, altamente inteligentes. A inteligência dos agentes é formada por três principais componentes: base de conhecimentos interna, capacidade de raciocínio a partir da base de conhecimentos existente e habilidade de aprender ou adaptar-se às alterações do ambiente.

- **Autonomia:** Uma das mais importantes diferenças entre os agentes e os softwares tradicionais é a capacidade que o agente tem de perseguir um objetivo autonomamente, isto é, sem a necessidade de interações ou comandos do ambiente externo. O agente não necessita de autorização do usuário ou de outros agentes para executar cada passo, ele tem capacidade de ação própria. Para que um agente tenha autonomia é fundamental que ele também possua as habilidades de aprender, perseguir um objetivo, movimentar-se pela rede, comunicar-se com outros agentes, dentre outras.

- **Mobilidade:** Mobilidade é a habilidade que os agentes têm de navegar pelas redes. Agentes móveis são capazes de deslocar-se de um computador para outro através das redes, ao contrário dos agentes estacionários que ficam estáticos em um computador específico. Outra habilidade que os agentes móveis possuem é o envio e recebimento de mensagens de outros agentes existentes na rede.

- **Comunicação:** Os agentes necessitam interagir com o ambiente e com os outros agentes, a propriedade de comunicação proporciona esta interação. A comunicação entre os agentes é feita através de uma linguagem própria, isto é, existem protocolos específicos para que os agentes enviem e recebam mensagens de outros agentes e do ambiente. Nestes protocolos são definidos domínios de perguntas e respostas que podem ser enviadas e recebidas.

- **Cooperação:** Quando o nível de complexidade do problema a ser resolvido é alto podem ser utilizados diversos agentes trabalhando de forma cooperativa, ou seja, em um ambiente são distribuídos alguns agentes que trabalharão de forma concorrente mas com um objetivo comum. Estes agentes compartilham conhecimento utilizando a propriedade de comunicação.

2.4 Agentes Web

A grande evolução da Internet, nos últimos anos, abriu um leque muito grande de aplicações para os agentes inteligentes, bem como transformou-se em um ambiente vasto para a utilização destes agentes.

Com o surgimento de inúmeros agentes, com as mais diversas finalidades na Web, existe a possibilidade de classificá-los em grande categorias, tais como: agentes de recuperação e filtragem de informações, agentes para filtragem de mensagens, agentes assistentes de navegação e outros.

2.4.1 Agentes de Recuperação e Filtragem de Informações

Os agentes de recuperação e filtragem têm como objetivo principal auxiliar os usuários na busca de informações no WWW. Os principais representantes desta categoria são as máquinas de busca (*search engines),* que são programas, baseados em agentes inteligentes, que buscam automaticamente na Web novas informações[BRE 98]. AltaVista[1] e MetaCrawler[2] são dois exemplos de máquinas de busca.

A maioria dos agentes desta categoria operam de forma estacionária com inteligência limitada. Uma possível evolução para estas aplicações poderia ser o uso de agentes inteligentes móveis. As máquinas de busca existentes são agrupadas em diferentes níveis, de acordo com seu estágio de desenvolvimento. A figura 2.2 representa estes níveis.

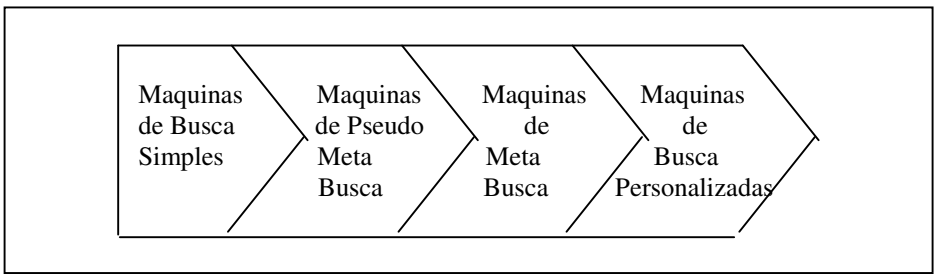


FIGURA2.2-Níveis de desenvolvimento das máquinas de busca[BRE98]

Uma máquina de busca simples (*simple search engines*) representa o nível mais baixo das ferramentas de busca na Internet, mas já utiliza tecnologia de agentes

[1] www.altavista.com
[2] www.metacrawler.com

inteligentes. Como característica central destas ferramentas temos o armazenamento de todas informações encontradas em um banco de dados, que pode ser centralizado ou distribuído. AltaVista ou Cade[3] são representantes dos *simple search engines.* O tempo de atualização dos bancos de dados dificilmente acompanha a velocidade com que novas informações são inseridas na Web, por isto, o resultado das buscas requisitadas pelos usuários pode ser insuficiente, ou seja, poderão faltar documentos importantes que estão disponíveis mas ainda não foram inseridos no banco de dados da ferramenta utilizada[BRE 98].

Para realizar buscas mais eficientes e precisas os usuários mais experientes utilizam a combinação de diversas *simple search engines.* O nível *pseudo meta search engines* suporta esta técnica, cujo principio é a utilização de um conjunto de sistemas de busca como ponto de partida para consultas especificadas pelo usuário. O usuário deverá selecionar qual sistema ele prefere usar, a partir de uma lista fornecida pela aplicação. CUSI (Configurable Unified Search Engine)[4] é um exemplo deste nível [BRE 98].

O nível *meta search engines* proporciona a consulta simultânea a diversos sistemas de busca simples, ou seja, executa automaticamente buscas utilizando *simple search engines* de forma paralela fornecendo ao usuário o resultado das diversas buscas de forma resumida e simplificada em um único formulário, tal como nas ferramentas mais simples. Este tipo de agente não possui banco de dados porque faz uso dos bancos de dados das diversas *simple search engines.* SavvySearch (guaraldi.cs.colostate.edu:2000/form) e MetaCrawler são exemplo deste tipo de ferramenta[BRE 98].

Experiências mostram que a busca de informações converge para áreas de especial interesse dos usuários, ou seja, os resultados das buscas possuem características semelhantes. Isto permite a criação de bancos de dados personalizados com informações específicas sobre preferências de cada usuário, este seria o nível de *personalized search engines.*

2.4.2 NewsWatcher

Os agentes da categoria *NewsWatcher* (monitoração de notícias) são utilizados por grupos que possuem interesses comuns em determinados tipos de mensagens e notícias disponibilizadas na Internet. A definição do perfil de cada usuário permite que sejam filtradas as informações e notícias extraídas por ele, de acordo com seus interesses. A tabela 2.1 apresenta alguns exemplos de aplicações *NewsWatcher.*

TABELA 2.1 - Exemplos de aplicações *NewsWatcher* [BRE 98]

Nome da Aplicação	**Endereço WWW**
After Dark Online	www.afterdark.com
Marimba – Castanet Tuner	www.marimba.com
PointCast Network	www.pointcast.com

[3] www.cade.com.br
[4] www.nexor.co.uk/public/cusi

Os agentes *NewsWatcher* são baseados em dois principais conceitos, a personalização dos canais de notícias e a propagação automática das informações.

a) Personalização de canais de notícias (*news channels)*:

A contínua e rápida disponibilização de informações na Internet é o principal fator para o grande desenvolvimento das aplicações da categoria NewsWatcher. A tarefa central destas aplicações é filtrar as informações de interesse pessoal a partir de uma grande quantidade disponível. A definição do interesse pessoal de cada usuário é um pré-requisito básico para a realização destas tarefas.

Existem disponíveis na rede inúmeros canais de notícias, estes canais contém os mais diversos conteúdos e podem ser configurados, a partir dos perfis dos usuários, para retornarem somente as informações de interesse destes usuários.

Estas configurações produzem subconjuntos de notícias e informações que serão destinadas aos grupos de usuários com interesses semelhantes.

b) Propagação automática das informações:

A propagação automática das informações é outra função dos agentes, que devem enviar aos demais participantes do grupo as últimas notícias e informações obtidas por cada integrante. Para que a propagação das notícias seja possível é necessário que sejam especificados parâmetros como tipo de conexão, tempo máximo de transmissão e intervalo de tempo entre as atualizações. Baseados nos parâmetros pré definidos o agente inicia a conexão, filtra as informações a serem transmitidas e transmite aos participantes do grupo com os mesmos interesses.

2.4.3 Agentes de Navegação Antecipada (*Advising and Focusing)*

Esta categoria de agentes têm a característica de auxiliar os usuários durante a sua navegação pela Web, em outras palavras, são agentes pessoais que auxiliam os usuários a buscarem suas informações de forma mais fácil e precisa.

Os agentes desta categoria observam as ações dos usuários durante suas navegações pela Web e procuram aprender quais são as preferências de cada usuário, traçando assim um perfil do usuário. O conhecimento adquirido pelo agente será utilizado em buscas futuras no sentido de filtrar somente as informações que são pertinentes[BRE 98].

Agentes *advising and focusing* procuram antecipar-se à navegação que o usuário vai fazendo, ou seja, enquanto o usuário lê uma determinada página o agente continua a navegação procurando assuntos que estejam dentro do perfil do usuário e quando os encontra sugere ao usuário. A tabela 2.2 apresenta alguns exemplos de agentes desta categoria e seus respectivos endereços na Web.

TABELA 2.2 - Exemplos de Aplicações advising and focusing [BRE 98]

Nome da Aplicação	Endereço WWW
Web Browser Intelligence	www.networking.ibm.com/wbi/wbisoft.htm
Letizia	www.media.mit.edu/people/lieberarcy/Letizia
Webdoggie	rg.media.mit:80/projects

2.4.4 Agentes de Processamento de e-mails

Esta categoria de agentes têm como objetivo principal o processamento das mensagens recebidas pelos usuários. Este processamento baseia-se , principalmente, na filtragem dos e-mails recebidos baseando-se em critérios pré estabelecidos pelo usuário.

Os agentes de processamento de e-mail ficam monitorando o recebimento de mensagens e, baseados no perfil de interesse do usuário, aplicam mecanismos de filtragem principalmente sobre os campos autor e assunto.

Algumas ferramentas baseadas em agentes também executam, além da filtragem de mensagens, a distribuição destas mensagens de forma automática aos demais participantes dos grupos, que possuam perfis de interesse semelhantes[BRE 98].

2.5 Modelo de Usuário

Aplicações que efetuam buscas inteligentes na Web, utilizado agentes, necessitam ter conhecimento sobre o modelo do usuário, isto é, para que as buscas realizadas Web resultem em informações relevantes aos interesses dos usuários é necessário que o agente de busca tenha conhecimento sobre o perfil e as preferências pessoais de cada usuário, em diversas áreas[MIN 96]. Os mecanismos de aquisição de conhecimentos sobre modelos de usuários serão abordados neste capítulo.

2.5.1 Importância do Modelo de Usuário

Existem hoje inúmeras ferramentas baseadas em agentes inteligentes que realizam buscas de informações na Web, filtram e-mails e realizam outras operações a partir do perfil de usuário, como foi descrito na secão anterior. Estas ferramentas utilizam fortemente o conceito de modelo de usuário como ponto de partida para suas tarefas, bem como devem realimentar este modelo com os resultados obtidos para que as tarefas futuras possam ser mais eficientes e mais focadas aos reais interesses do usuário.

A inserção de um modelo de usuário em uma aplicação de busca na internet e o seu relacionamento com os demais componentes desta aplicação é apresentada na figura 2.3, que representa a arquitetura da aplicação “News Dude”.

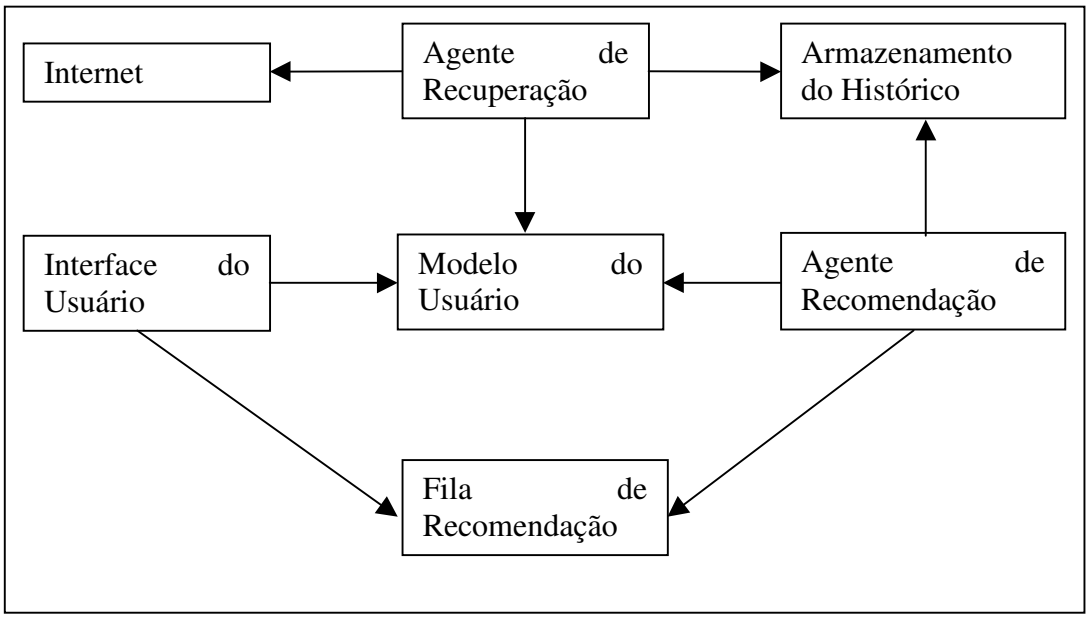


FIGURA 2.3 - Inserção do modelo de usuário na aplicação “News Dude”

A estrutura apresentada na figura 2.3 procura mostrar a ligação do modelo do usuário com os demais agentes do sistema. A figura apresenta o modelo do usuário sendo relacionado com os demais componentes da aplicação, neste caso, a cada recomendação proposta pela aplicação é feita uma verificação se a informação recuperada é compatível com as especificações do modelo, casa seja interessante ao usuário esta recomendação é armazenada em um banco de históricos e o modelo de usuário é atualizado. O usuário, a partir de suas interações com a aplicação, também auxilia na atualização do modelo.

2.5.2 Requisitos de um Modelo de Usuário

A especificação do modelo do usuário é um assunto que vem sendo abordado por diversos autores, tais como [BIL 96] e [KRU 97], entretanto boa parte deles concorda que um modelo de usuário para ser eficiente deve observar alguns pré-requisitos:

- O modelo deve ser capaz de representar um usuário com múltiplos interesses em diferentes tópicos, pois muitas vezes uma mesma pessoa possui interesse em mais de uma área, sendo estas totalmente independentes uma da outra.

- O modelo deve ser flexível ao ponto de adaptar-se às alterações de interesse do usuário rapidamente, mesmo depois de um longo período de treinamento.

- O modelo deve levar em conta a interação com o usuário para adicionar os conhecimentos obtidos à base que armazena o perfil de cada usuário em cada tema.

2.5.3 Aprendizado do Modelo de Usuário

O processo de aprendizado do modelo do perfil de interesse de cada usuário em diferentes áreas é um dos itens mais importantes na modelagem de usuário. Existe uma grande diversidade de mecanismos que têm como objetivo proceder este aprendizado.

2.5.3.1 Aprendizado a Partir de Amostras

Para a implementação do processo de aprendizado do modelo de usuário, normalmente as aplicações utilizam algoritmos que requerem conjuntos de exemplos positivos sobre os interesses dos usuários ("o usuário está interessado em ...") e conjuntos de exemplos negativos, ou seja, assuntos em que o usuário não tem interesse ("o usuário não está interessado em ...").

A partir dos documentos submetidos como exemplos positivos e negativos as aplicações procedem o aprendizado em dois passos, o primeiro é a avaliação das características de todas as páginas submetidas e o segundo é a aplicação de algoritmos de aprendizado sobre as características extraídas.

A maioria dos algoritmos de aprendizado necessitam que os conjuntos de exemplos sejam representados na forma de vetores boleanos [BIL 96], assim sendo, as ferramentas que implementam o aprendizado do modelo de usuário devem converter os textos fornecidos para estes vetores. Porém, não é possível converter todo o texto em vetores, devendo a aplicação selecionar algumas palavras que serão adicionadas ao vetor.

Cada aplicação possui o seu critério próprio para determinar quais palavras serão submetidas ao algoritmo de aprendizado. Dentre os critérios utilizados, o mais simples e também muito usado é o que classifica as palavras pela freqüência com que elas ocorrem no texto.

Os conhecimentos obtido pelo algoritmo de aprendizado, a partir dos exemplos submetidos, serão armazenados e servirão de base ao modelo de usuário e passarão a ser utilizados durante as buscas realizadas. Durante as buscas o modelo pode ser incrementado aplicando-se os mesmos mecanismos citados acima sobre os documentos recuperados e classificados como positivos ou negativos através de interação com o usuário.

2.5.3.2 Aprendizado a Partir do Histórico

A utilização do histórico do consumo de informações pelos usuários é outra fonte de referência muito utilizada para a aquisição dos conhecimentos sobre as preferências pessoais e conseqüente formação do modelo de usuário. Este mecanismo utiliza como base para aquisição do conhecimento as páginas, os caminhos, os *links* e outros locais por onde o usuário já passou.

Existem inúmeras estratégias de aprendizado sobre modelos de usuário, a partir de históricos de navegação. [BIL 98] apresenta uma estratégia onde o aprendizado é feito a partir de dois modelos: um representa os interesses do usuário a curto prazo, e o outro representa os interesses do usuário a longo prazo.

A diferença básica entre os modelos de curto prazo e de longo prazo está no domínio do tempo. O modelo de aprendizado de curto prazo utiliza as observações feitas mais recentemente sobre a navegação do usuário como subsídios, desta forma os assuntos de interesse do usuário serão ajustados mais rapidamente no modelo. Já o modelo de longo prazo armazena as observações mais gerais, ou seja, é um refinamento aplicado sobre o modelo de curto prazo. Esta divisão é feita porque usuários podem ter interesse em determinadas informações por um tempo limitado. Estes seriam os interesses a curto prazo, porém existem assuntos em que o usuário sempre terá interesse, estes seriam interesses a longo prazo.

2.6 Considerações Finais

Os agentes inteligentes, com a evolução da Internet, passaram a ser fortemente utilizados, principalmente como ferramentas de auxilio na busca de informações, uma vez que a quantidade de informações disponíveis na rede hoje em dia é muito grande, dificultando a localização de informações de forma precisa e rápida.

Neste capítulo, foram apresentados os principais conceitos de modelagem de usuário, modelagem esta, que torna-se fundamental no desenvolvimento de qualquer aplicação que tenha como objetivo recuperar informações da Internet baseada no perfil de interesse do seu usuário.

3 Agentes de Busca na Internet

Neste capítulo é apresentado um estudo sobre algumas aplicações de busca de informações na Web baseados em agentes, foram estudadas as aplicações InfoFinder, Letizia, Search Advisor e WebWatcher. Ao final do capítulo é apresentada uma comparação entre as aplicações citadas.

3.1 InfoFinder

O agente InfoFinder é uma aplicação que procura aprender o perfil de interesse do usuário sobre alguns documentos a partir de amostras submetidas a ele durante sua navegação. Uma característica forte do InfoFinder é a possibilidade de aprendizado a partir da mínima interação com o usuário, ao qual é perguntando sobre a pertinência ou não de cada documento, informação esta que será utilizada durante o processo aprendizado como amostra para buscas futuras[KRU 97].

O InfoFinder aprende os perfis gerais dos documentos por heurística, extraindo frases que são prováveis representantes dos tópicos principais de cada documento. Esta heurística contrasta totalmente com os métodos mais comuns, e caros, que usam todas as palavras do documento para aprender. O algorítmo de aprendizado do InfoFinder gera uma árvore de procura, que o sistema converte em uma *string* boleana e submete a uma máquina de busca genérica[KRU 97]. O InfoFinder executa estas buscas durante a noite e envia para os usuários atualizações regulares de novos e interessantes documentos, sem que o usuário tome a iniciativa de acessar o agente.

3.1.1 Aprendizado do Interesse do Usuário

Ao ler um documento o usuário pode decidir que este documento é de seu interesse ou não, em uma determinada área. Uma vez analisado o documento, o usuário informa, através de um ícone, ao InfoFinder se este documento é pertinente ou não, o agente então pede para que o usuário informe a categoria em que o documento se enquadra. O nome das categorias não irá influenciar no processo de busca, somente será utilizado para agrupar os documentos e interagir com o usuário. O InfoFinder então armazena o documento em uma coleção de amostras para utilizar futuramente como subsídio ao processo de busca[KRU 97].

Após o usuário determinar um conjunto de documentos com relevantes ou não relevantes a uma determinada categoria, o InfoFinder usa este conjunto de amostras para montar uma *string* de busca para cada categoria. Para realizar o aprendizado sobre o interesse do usuário o agente executa, basicamente, três passos:

- Extrair semanticamente as frases significantes de cada documento.
- Criar árvores de decisão para cada categoria baseado nas frases extraidas.

- Transformar cada árvore de decisão em uma *string* de procura.

O primeiro passo é o processamento dos documentos do conjunto de amostras, usando heurística para extrair as frases significantes de cada documento. Esta heurística procura observar a tendência que os autores têm em utilizar métodos de destaque, com itálico e negrito, às idéias principais do texto. De cada documento o algoritmo extrai diversas frases, porém, algumas podem não representar efetivamente o assunto do texto. Estas frases podem ser excluídas da amostra através de interação como usuário[KRU 97].

Após o usuário selecionar alguns documento para servirem como amostra o InfoFinder utilizará estes documentos para determinar o perfil de interesse de cada usuário. O sistema realiza este processo de aprendizado para cada categoria de usuário, permitindo assim que usuários com interesses diferentes tenham visões diferenciadas sobre os mesmos documentos. A figura 4.1 mostra o exemplo de um conjunto de amostras da categoria "Java", observe que cada entrada na tabela possui a indicação se as frases são pertinentes ou não, indicadas pelos sinais "+" ou "-" respectivamente.

User	Indicators	Processed
John Doe		
Java		
	- 2-D, 3-D, Accelaretor, Art Brieva, ASI, CD-ROM, Christmas, CMP, Computer	01/03/96 11:04:20 am
	- 3-D, Chromatic Research Inc, Cirrud Logic Inc, Conference, Douglas Bartek	01/03/96 11:04:15 am
	- BEIJING, Beijing Software Engineering Center, CHINA, Chinese, Comission,	01/03/96 11:04:11 am
	- Bernd Pischetsrieder, BMW, DONN, E-mail, Germany, IG Metall, INDIVIDUAL	01/03/96 11:04:07 am
	+ApplixWare, C++, C++ like, Client/Server, Goulde, IBM, Internet, Java, JavaSc	01/03/96 11:04:02 am
	- ATM, Directory Services/Mail, EDI, e-mail, Internet, IXC, Java, LEC, LEC and I	01/03/96 11:03:55 am
	- Java, Server Inetaddress	01/03/96 11:03:40 am
	+3GL, C++, Client/Server, CORBA, In Hong Kong, Interact, Internet, Jara, Java,	01/03/96 11:03:35 am

FIGURA 3.1 - Exemplo de um conjunto de amostras da categoria "Java"

Cada documento do conjunto de amostras é representado pelas frases que foram extraídas do documento original. Após cada amostra estar efetivamente reduzida a um conjunto de frases significantes, o InfoFinder utiliza um algoritmo para a criação de uma árvore de decisão composta das frases positivas e negativas, esta árvore será futuramente usada na criação de *strings* boleanas[KRU 97].

A figura 3.2 apresenta um exemplo de árvore de decisão, criada a partir de uma categoria chamada "Java". A árvore mostra claramente que o InfoFinder enfocou principalmente nas frases significantes dos documentos, sem desviar-se para frases irrelevantes ao contexto do documento.

É importante salientar que a aprendizagem feita até chegar na árvore de decisão final fundamentou-se em todo o conjunto de documentos armazenados como amostragem, isto é, a árvore é resultante da combinação de todas as frases selecionadas de todos os documentos, limitando-se à categoria escolhida. Desta forma temos uma

única árvore para cada categoria de assunto, englobando todos os documentos disponíveis daquela categoria.

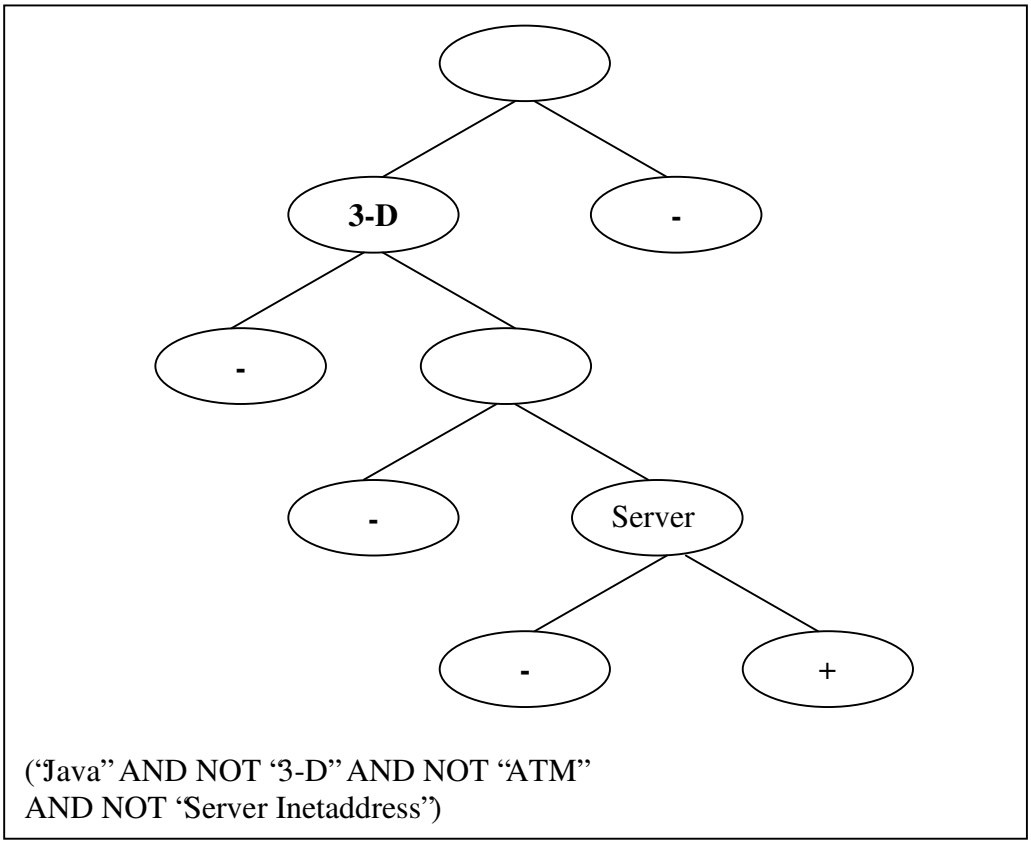


FIGURA 3.2 - Exemplo de árvore de decisão da categoria "Java" [KRU 97]

A figura 3.2 mostra a transformação realizada pelo InfoFinder da árvore de decisão para uma *string* boleana. Esta *string* gerada é enviada pelo *browser* do usuário para máquinas de busca tradicionáis para que sejam localizados novos documentos de interesse do usuário. Os documentos resultantes da busca são mostrados ao usuário que, por sua vez, pode analisá-los e classifica-los como pertinentes ou não, desta forma o banco de conhecimentos sobre as preferências do usuário em determinada categoria é atualizado, ficando cada vez mais refinado.

As operações vistas acima deixam claro que o processo de aprendizado do InfoFinder é interativo, pois as amostras de documentos interessantes ou não são enviados pelo usuário, que também é responsável por determinar se os documentos obtidos pelo agente são positivos ou negativos, a fim de atualizar o banco de amostras e enriquecer a filtragem das próximas buscas.

3.1.2 Busca de Documentos na Web

Como vimos na seção anterior a busca de documentos na Web, atendendo aos interesses do usuário não é de responsabilidade do InfoFinder. Para este trabalho o InfoFinder

lança mão de máquinas de busca tradicionais, porém fica com a responsabilidade de montar as *strings* boleanas que serão enviadas a estas máquinas a fim de obter os resultados esperados pelo usuário.

3.2 Letizia

Letizia é um agente autônomo que foi desenvolvido no Instituto de Tecnologia de Massachusetts, Cambridge nos Estados Unidos, como um protótipo para auxiliar durante a navegação na Internet sobre os *browsers Netscape Navigator e Mosaic*.

O Letizia fica observando as ações do usuário e procura imitá-lo a fim de fornecer sugestões e informações para futura navegação, ou seja, o Letiza antecipa-se à navegação buscando informações que sejam do interesse do usuário. A decisão sobre a pertinência ou não de um determinado documento é tomada a partir das observações feitas sobre as ações realizadas pelo usuário anteriormente [LIE 2001].

3.2.1 Interface do Letizia

A interface do Letizia é composta por três janelas abertas sobre o navegador. A primeira é a janela original do *browser*, onde o usuário irá executar suas ações de forma normal, podendo ignorar a presença do agente. As outras duas janelas ficam localizadas no lado direito do vídeo e são utilizadas para controle do agente. A janela superior mostra as páginas candidatas, isto é, as páginas que estão sendo analisadas pelo Letizia para possível indicação ao usuário. A janela inferior mostra as páginas que o Letizia está recomendando ao usuário como sendo de seu interesse. O usuário pode continuar navegando normalmente sem considerar o agente ou aceitar as sua sugestões a qualquer momento[LIE 95]. A figura 4.3 apresenta a interface básica do Letizia.

O Letizia não é limitado à configuração das três janelas, na prática o número mínimo de janelas para o funcionamento são duas, a janela de navegação do usuário e a janela de recomendações do Letizia. A janela de páginas candidatas não necessita estar sempre visível, pois sua finalidade é demonstração do trabalho realizado e *debug*. É possível também a exibição de mais de uma janela de recomendações, neste caso é útil quando necessita-se realizar buscas sobre mais de um perfil de usuário[LIE 2001].

Aceitar uma sugestão do Letizia consiste em simplesmente desviar-se para a janela onde a sugestão está sendo mostrada e continuar a navegação. Outra opção pode ser a inclusão da página na lista de preferências do navegador ou salvá-la no disco[LIE 2001].

A interface do Letizia é destacada como sendo um de seus pontos fortes, pois, prove buscas simultâneas à navegação do usuário. Esta característica é citada pelos autores [LIE 95] como sendo um grande diferencial sobre os agentes de busca tradicionais onde a interação ocorre em dois tempos, ou seja, o usuário envia uma solicitação ao agente e aguarda enquanto este realiza as buscas, somente permitindo navegação nas páginas sugeridas no final do processo.

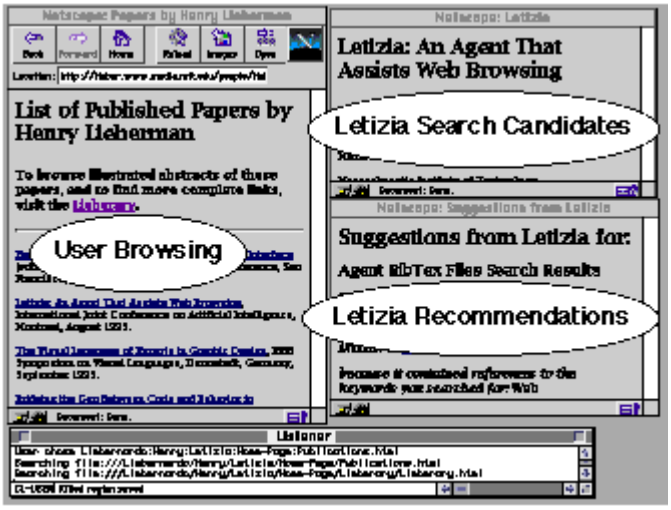


FIGURA 3.3 - Interface do Letizia [LIE 95]

O Letizia, para realizar suas recomendações, baseia-se no contexto da página em que o usuário está navegando no momento, desta forma é possível que o usuário veja a sua página de trabalho e mais uma página que está sendo recomendada, a partir do interesse do usuário naquele instante.

3.2.2 O Processo de Busca e Recomendação do Letizia

O processo de busca do Letizia é baseado na observação da estrutura de controle do navegador. A estrutura básica á apresentação de páginas compostas de conjuntos de textos, figuras e *links***,** estes *links* conduzem a outras páginas também compostas de textos, figuras e *links*, que conduzem novamente para outras páginas e assim por diante. Desta forma, o usuário vai descendo cada vez mais na hierarquia da rede. O navegador armazena, em uma pilha, todas as páginas percorridas recentemente pelo usuário, permitindo que este volte às páginas anteriores.

Quando um usuário está buscando uma determinada informação utilizando simplesmente o navegador, normalmente ele acessa diversos *links* e volta, até encontrar a informação procurada ou, muitas vezes, não aprofunda-se aos níveis mais baixos podendo correr o risco de não localizar a página desejada. Esta forma de busca incentiva a no sentido vertical, ou seja, cada vez mais o usuário aprofundando-se na hierarquia das páginas. A figura 3.4 representa a busca tradicional, realizada verticalmente.

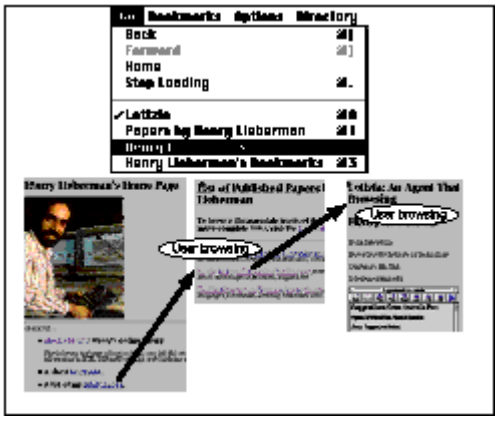


FIGURA 3.4 - Busca tradicional na Web [LIE 95]

O Letizia procura compensar esta tendência, implementando buscas no sentido horizontal. Estas buscas são realizadas a partir da página onde o usuário está navegando, ou seja, o Letízia entra em cada *link* da página atual procurando informações que sejam de interesse do usuário nos níveis inferiores, mas procurando não distanciar-se muito do usuário, pois normalmente as informações encontram-se próximas e não em níveis muito profundos. Em outras palavras, o Letizia antecipa a navegação, entrando nos *links* a procura de informações de interessantes ao usuário[LIE 95]. A figura 3.5 procura representar o mecanismo de busca do Letizia.

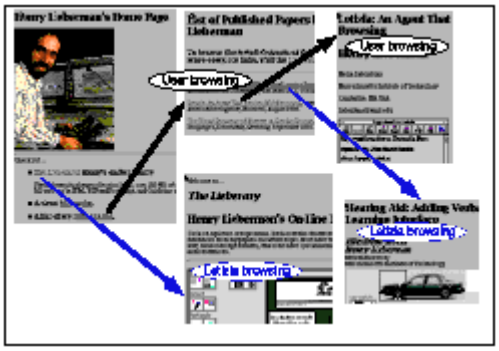


FIGURA 3.5 - Busca Horizontal da Web [LIE 95]

Para analisar o conteúdo dos documentos, o Letizia utiliza uma simples medida de freqüência de palavras denominada TFIDF (*term frequency times inverse document frequency*). Esta técnica [SAL 89] diz que palavras que são relativamente comuns no documento, mas relativamente raras em geral são boas indicações sobre o conteúdo.

O Letizia também acumula um banco de informações sobre o perfil de interesse do usuário, estas informações são coletadas durante a sua navegação e serão consideradas durante o processo de busca.

3.3 Search Advisor

O Search Advisor (Aconselhador de Procura), desenvolvido no Departamento de Informática de Universidade da Macedônia, é um agente inteligente que tem como objetivo principal automatizar a construção de estratégias de busca, em domínios específicos, para ajudar o usuário a localizar e recuperar informações utilizando as mais diversas máquinas de procura e fontes de informações.

O Search Advisor também auxilia no treinamento dos usuários "novatos" na tarefa de buscar informações, fornecendo dados adicionais que são armazenadas em uma árvore de decisão que o sistema constrói durante o processo de procura[AVG 97]. O agente utiliza a combinação dos meta-conhecimentos adquiridos durante buscas anteriores e os argumentos de pesquisa fornecidos pelo usuário para treinar os novos usuários.

A estrutura básica do sistema é composta de quatro níveis, que estão representados na figura 3.6.

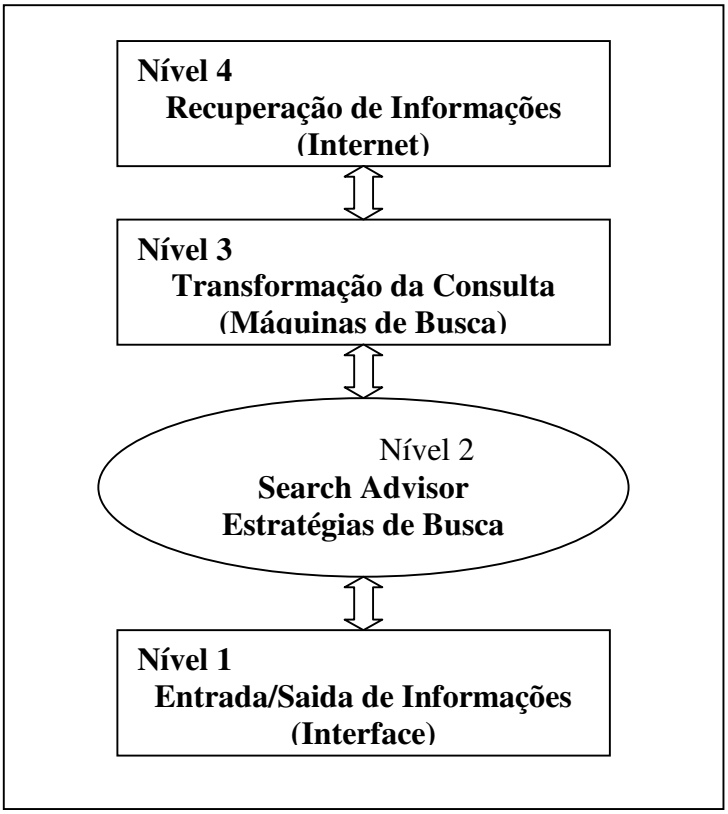


FIGURA 3.6 - Níveis da estrutura do Search Advisor

O primeiro nível do Search Advisor (Nível 1) é o responsável pela interação com o usuário, receber as entradas e fornecer os resultados. O usuário acessa o sistema e através de uma caixa de diálogo fornece os argumentos de pesquisa, e em seguida recebe a estratégia de procura proposta e o resultado da busca na Internet[AVG 97].

O segundo nível (Nível 2), chamado de SEARCH ADVISOR, é o nível de aconselhamento. É neste nível que são implementadas as combinações entre as entradas fornecidas pelo usuário e os conhecimentos acumulados pelo agente durante as buscas anteriores, objetivando a definição de uma estratégia de busca.

O terceiro nível (Nível 3) é o responsável pelo transformação das estratégias de busca definidas no nível 2 para *strings* de procura que serão submetidas às diferentes máquinas de busca disponíveis na Internet.

O quarto nível (Nível 4) é o responsável pela efetiva recuperação das informações, nesta etapa o agente envia parâmetros definidos no nível 3 para as máquinas de busca tradicionais que serão utilizadas no processo de localização dos documentos[AVG 97].

3.3.1 O Fluxo de Operações do Search Advisor

O fluxo de operações do Search Advisor é composto, basicamente de cinco passos, que são representados na figura 3.7.

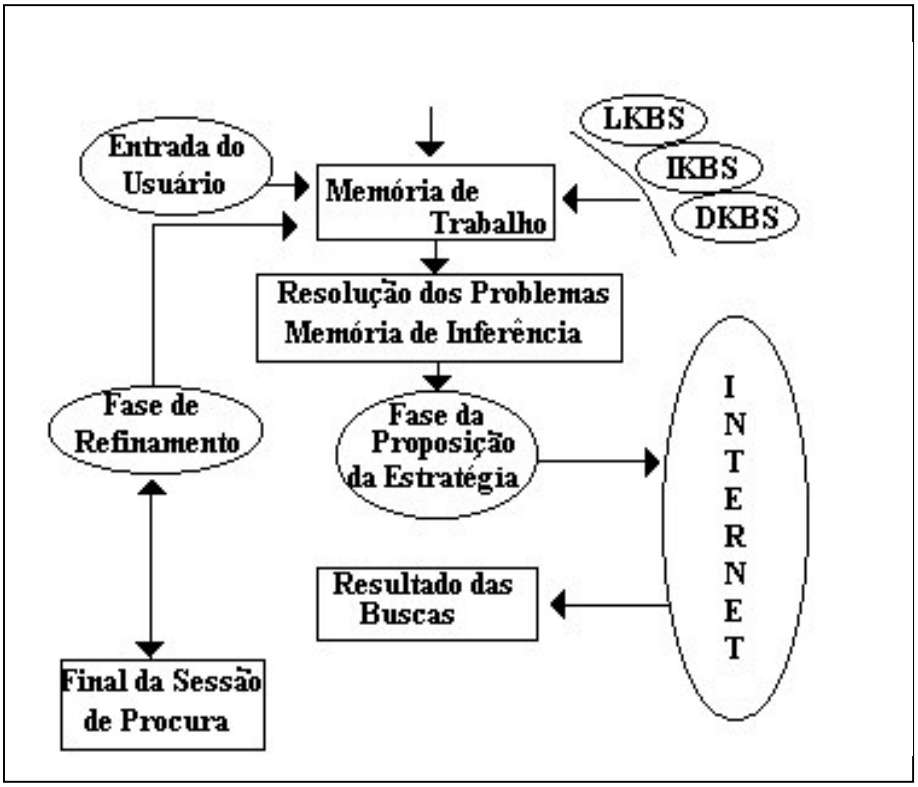


FIGURA 3.7 - Fluxo de Operações do Search Advisor [AVG 97]

Passo 1: O sistema é inicializado utilizando as entradas do usuário e os meta-conhecimentos armazenados em três diferentes KBS (Bases de Conhecimento). O usuário acessa o sistema via WWW e define os termos da sua busca.

Passo 2: As entradas do usuário são transferidas ao componente de "aconselhamento", onde serão determinadas as estratégias de busca.

Passo 3: Os resultados do componente de "aconselhamento" são reportadas ao usuário e enviadas ao componente responsável pela construção das *strings* de procura.

Passo 4: A consulta, já montada, é aplicada sobre repositório de documentos da Internet e os resultados enviados de volta ao usuário.

Passo 5: De acordo com os resultados obtidos o usuário pode aceitar como válido e concluir a busca ou entrar numa sessão de refinamento da busca.

3.3.2 O Componente de Recomendação do Search Advisor

O componente de recomendação, ou aconselhamento, do Search Advisor é composto basicamente de três partes que são apresentadas na figura 3.8:

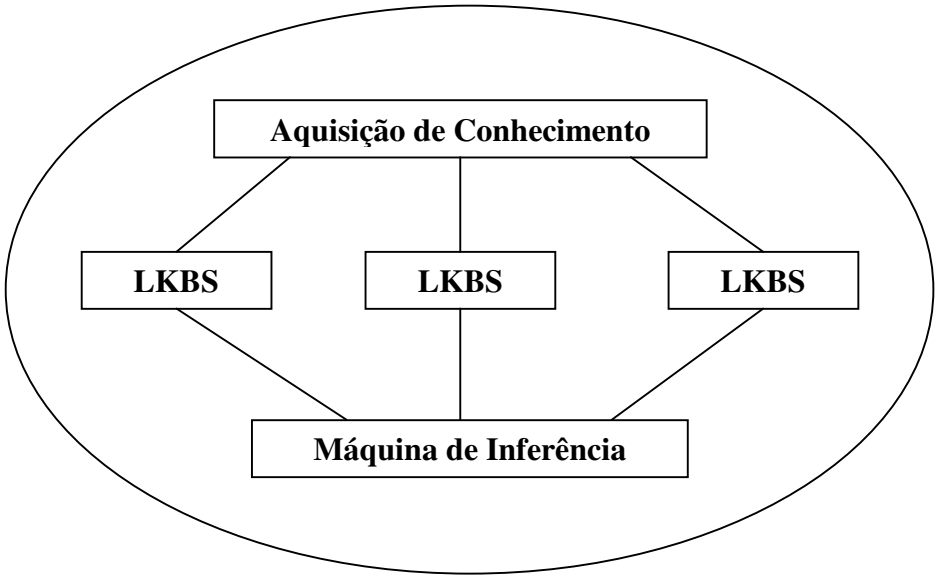


FIGURA 3.8 - Estrutura do Componente de Recomendação

a) Componente de aquisição de conhecimento, que é responsável por extrair conhecimentos a partir das buscas que são realizadas através da ferramenta, estes conhecimentos são classificados por domínio e armazenadas em uma base de conhecimentos chamada KBS (Knowledge Base System) [AVG 97].

b) Três diferentes bases de conhecimento (LKBS, IKBS, DKBS): O primeiro tipo de base de conhecimento é chamado LKBS (*Librarian Knowledge Base System*), nesta base serão armazenados os conhecimentos obtidos a partir de pessoas especializadas no ramo de bibliotecas, esta base armazenará as melhores regras que os bibliotecários utilizam para localizar obras em um biblioteca tradicional. O segundo tipo de base de conhecimento é chamado IKBS (*Internet Knowledge Base System*), esta base armazena as melhores técnicas utilizadas por especialistas em buscas na Internet. O terceiro tipo de é chamado de DKBS (Domain *Knowledge Base System),* esta base armazena os domínios de assuntos obtidos através das diversas consultas, o perfil de cada usuário é melhorado a cada consulta.

c) Uma máquina de inferência, que quando acionada pelo usuário, combina os conhecimentos armazenados em LKBS, IKBS e DKBS a fim de elaborar a melhor estratégia de busca possível. A máquina de inferência também é responsável pelo refinamento da consulta, caso o usuário deseje [AVG 97].

3.4. WebWatcher

WebWatcher é um agente inteligente que tem como objetivo auxiliar o usuário durante o processo de navegação na Web. O agente acompanha o usuário, página a página, sugerindo *links* que possam ser de interesse do usuário e aprendendo com este.

Os autores [JOA 98] comparam o WebWatcher a um guia de turismo, que ao entrar em um museu pede aos turistas quais são suas obras de interesse e, a partir daí, busca obras com o perfil indicado e sugere ao turista. Por outro lado o guia também adquire conhecimentos a partir de conversas com os turistas e novos museus visitados. Este é o principio básico de funcionamento do WebWatcher, interativamente ele comunica-se com o usuário indicando por onde ele deve ir, e o usuário comunica-se com o agente fornecendo-lhe subsídios para seu aprendizado. Assim como o guia de turismo, com o passar do tempo o WebWatcher acumula conhecimentos sobre os locais por onde já passou.

O WebWatcher, ao contrário das máquinas de busca tradicionais que realizam pesquisa através da presença ou não de determinadas palavras, pode aprender que termos ou palavras diferentes podem fazer parte do mesmo domínio de assunto, por exemplo “learning machine” e “neural networks” pertencem ao mesmo grupo de interesse. Também é levando em conta a sucessão das páginas e como estas se relacionam, assim a busca não fica limitada a simples palavras, que muitas vezes podem não representar o assunto procurado [JOA 98].

O WebWatcher é iniciado a partir do navegador WWW, neste momento ele solicita que o usuário informe uma curta descrição do assunto de seu interesse, a partir daí o agente passará a acompanha-lo durante a navegação.

3.4.1 Interface e Serviços do WebWatcher

Além da lista de comandos, adicionada na página, o WebWatcher adiciona *ícones* ao redor dos *links* que são considerados interessantes, a partir dos conhecimentos anteriores do agente.

Caso o usuário aceite uma sugestão de *link* o WebWatcher insere sua interface na página atual e busca nesta página *links* que possam ser interessantes, sugerindo-os. Em páginas que possuem muitos *links* de assuntos interessantes ao usuário o sistema seleciona os três melhores e os sugere.

Além de indicações de *links* o WebWatcher fornece outros tipos de auxílio, como buscas através de palavras chaves, utilizando uma variação da máquina de busca Lycos aplicada sobre um conjunto de páginas previamente visitadas pelo agente. Isto acontece quando o usuário utiliza o comando "*Show me Similar Pages*" (Mostre-me as páginas similares), neste caso o WebWatcher mostra uma lista de páginas com conteúdo semelhante ao da página atual [JOA 98].

A interface do WebWatcher á adicionada na parte superior da página onde o usuário encontra-se, esta interface é composta de uma lista de comandos que podem ser utilizados pelo usuário para comunicar-se com o agente. A figura 3.9 apresenta uma página com a interface do WebWatcher adicionada.

FIGURA 3.9 - Interface do WebWatcher

O WebWatcher possui uma opção onde o usuário marca uma ou mais páginas para serem monitoradas pelo agente e quando ocorre alguma alteração no conteúdo da página um e-mail é enviado ao usuário informando-o [JOA 98].

O WebWatcher também possui um mecanismo para verificar se os objetivos do usuário foram atingidos, isto ocorre no momento de encerrar a sessão, quando o usuário deve optar por clicar em "Sair com objetivos alcançados" ou "Sair com objetivos não alcançados". Esta ação fornece subsídios ao agente para futuras buscas.

3.4.2 A Forma de Implementação do WebWatcher

O WebWatcher é implementado como um servidor e uma estação de trabalho, separando as ações da rede, simulando um *proxy*[JOA 98]. O agente WebWatcher forma uma camada entre o usuário e a WWW, como mostra a figura 3.10 Antes de devolver a página ao usuário são feitas três modificações:

a) A lista de comandos do WebWatcher é adicionada ao topo da página;

b) Para cada *link* da página original, seu URL é substituído por um novo URL que aponta para o servidor do WebWatcher;

c) Caso o WebWatcher encontre algum *link* que seja de interesse do usuário, este *link* recebe uma marcação que indica sua recomendação.

Enquanto aguarda que o usuário leia a página o WebWatcher avança, entrando nos *links* e selecionando assuntos interessantes de forma que o tempo de procura seja minimizado. Quando o usuário passa para outra página o foco do agente é atualizado para esta outra página e são realizadas as modificações citadas a cima. Este processo é executado enquanto o agente estiver ativo [JOA 98].

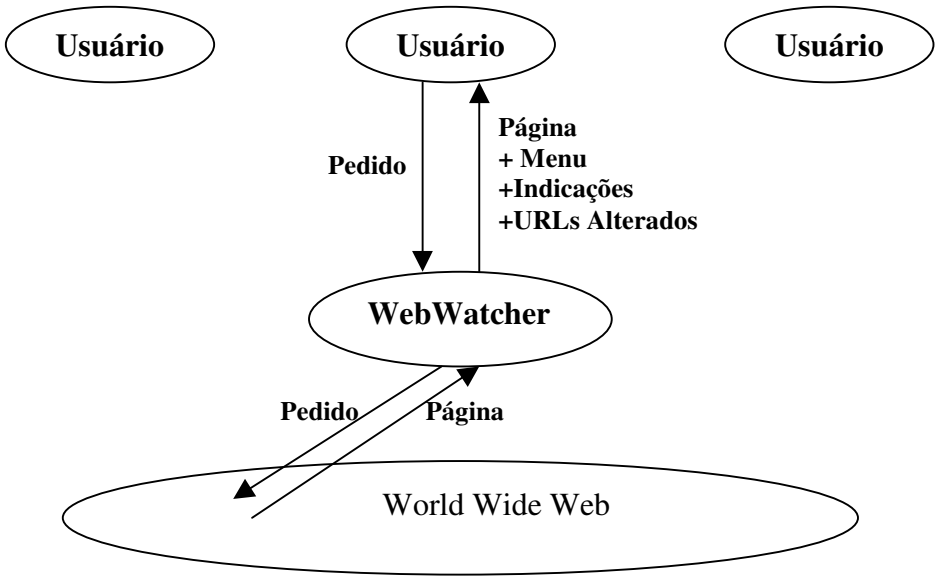


FIGURA 3.10 - Interação usuário, WebWatcher e WWW

3.4.3 O Processo de Aprendizagem do WebWatcher

O processo de aprendizado no WebWatcher é baseado no assunto inicial informado pelo usuário em combinação com os *links* acessados durante a navegação. Em outras palavras, durante a navegação o agente armazena o texto dos *links* que foram acessados juntamente com o assunto inicial informado, desta forma, a base de conhecimentos sobre os relacionamentos entre páginas é enriquecido [MLA 99].

Para que o agente possa fornecer sugestões de boa qualidade é necessário que haja um certo tempo de aprendizado, tempo este necessário para que sejam feitas combinações entre diversos assuntos e para que o agente possua um bom conhecimento sobre a rede e os interesses do usuário [MLA 99].

3.5 Comparação entre os Agentes

Observando os agentes apresentados nas sessões anteriores podemos fazer um comparativo entre eles. Os critérios para comparação escolhidos foram o objetivo proposto por cada um, o momento em que as buscas são realizadas, o nível de interação com o usuário, a fonte para obtenção de conhecimento, a máquina de busca utilizada, a realimentação a partir dos resultados e a forma de apresentação dos resultados. A tabela 3.1 apresenta uma comparação baseada nestes critérios.

TABELA 3.1 - Comparação entre os agentes

Critério	InfoFinder	Letízia	Search Advisor	WebWatcher
Objetivo	Buscar páginas de interesse do usuário.	Antecipar a navegação, sugerindo *links*.	Automatizar a construção de consultas e treinar usuários novatos.	Antecipar a navegação, sugerindo *links* e monitorar a atualização de páginas.
Momento da Busca	Quando o equipamento está ocioso.	Durante a navegação do usuário.	Quando solicitado pelo usuário.	Durante a navegação do usuário.
Nível de Interação com o usuário	Baixo	Alto	Alto	Alto
Máquina de Busca Utilizada	Diversas máquinas simples	Própria	Diversas máquinas simples	Própria + Lycos
Realimentação Baseada em Resultados	Sim	Sim	Sim	Sim
Apresentação dos Resultados	Através de lista de sugestões.	Através de Janela própria	Através de interface própria na WWW.	Na própria página através de ícones.

Observando a comparação feita na tabela a cima é possível notar que os agentes WebWatcher e Letizia possuem características bem semelhantes, diferenciando-se pela interface e pelo mecanismo de monitoração de atualização de páginas que o WebWatcher possui e o Letizia não. Os agentes InfoFinder e Search Advisor também possuem algumas características em comum, como a utilização de outras máquinas de busca para coleta informações na rede e a aquisição de conhecimento. Porém o InfoFinder diferencia-se porque realiza suas buscas quando o usuário não está utilizando o equipamento, ao contrário do Search Advisor que entra em atividade somente quando solicitada consulta por parte do usuário.

3.6. Comentários Finais

A partir das análises feitas sobre os quatro agentes, observamos que ferramentas deste tipo podem ser muito úteis aos usuários da Internet, durante suas buscas de informações, pois a medida que o tempo passa os agentes vão adquirindo maior conhecimento sobre as preferências pessoais de cada usuário e tornando o acesso a informações específicas mais fácil e eficiente. Por outro lado, observamos que nenhum dos agentes estudados aqui possui algum mecanismo de atualização de páginas de *links,* mantidas por muitas pessoas e que exigem dessas pessoas um bom trabalho para serem mantidas atualizadas.

4 A Modelagem do Agente ALOI

Neste capítulo, é apresentada uma proposta de modelo para a construção de um agente capaz de reconhecer o perfil de interesse do usuário ao buscar informações na Web, localizar informações que satisfaçam a este interesse e atualizar repositórios de informações de forma organizada quanto a assunto.

Inicialmente, explicitam-se as motivações que levaram ao desenvolvimento deste trabalho e seus objetivos. Em seguida é apresentada a estrutura global do modelo e suas interações com os diferentes módulos e com o ambiente, e finalmente é detalhada a proposta através de uma especificação de *software.*

4.1 Considerações Iniciais

Atualmente, existem diversos modelos e protótipos de aplicações baseados em agentes inteligentes, alguns dos quais citamos no capítulo anterior, que têm como objetivo auxiliar o consumo e a organização das informações disponibilizadas na Web [PAZ 96]. Estas aplicações possuem as mais variadas finalidades, desde realizar simples buscas a partir de termos informados pelos usuários, até realizar o aprendizado das preferências pessoais de cada usuário e baseadas nisto efetuar buscas de informações que atendam às necessidades do usuário.

Entretanto, poucos agentes são capazes de obter conhecimentos sobre o perfil de interesse de um usuário em específico, e, a partir destes conhecimentos comunicar-se com outros agentes disponíveis na Web a fim de adicionar *links* em repositórios e manter a consistência dos links já existentes. A concepção de uma aplicação com esta finalidade deve observar uma série de fatores importantes, tais como, a aquisição do conhecimento sobre os assuntos que são interessantes ao repositório de links, a forma de busca das informações na Internet, o conteúdo das informações localizadas, os critérios para a inclusão da referência em um repositório ou não, dentre outros.

A necessidade do desenvolvimento de um aplicativo com as características citadas acima motiva a proposição de um modelo geral de agente capaz de obter conhecimentos sobre o perfil de interesse dos usuários, buscar informações na Internet que atendam ao perfil, selecionar as informações pertinentes, manter um banco de dados com links de cada usuário organizado por assunto e realizar verificações periódicas sobre a consistência e a importância dos links armazenados.

4.2 Estrutura Geral do Modelo

O objetivo principal deste trabalho é propor um modelo geral para um agente capaz de obter o perfil de interesse do usuário na Web, utilizar este perfil para localizar informações que satisfaçam ao usuário e manter um repositório de informações organizadas por assunto. O projeto proposto considera que todo o agente será modelado, porém durante a fase de implementação será dado enfoque ao módulo responsável pela atualização do repositório de informações.

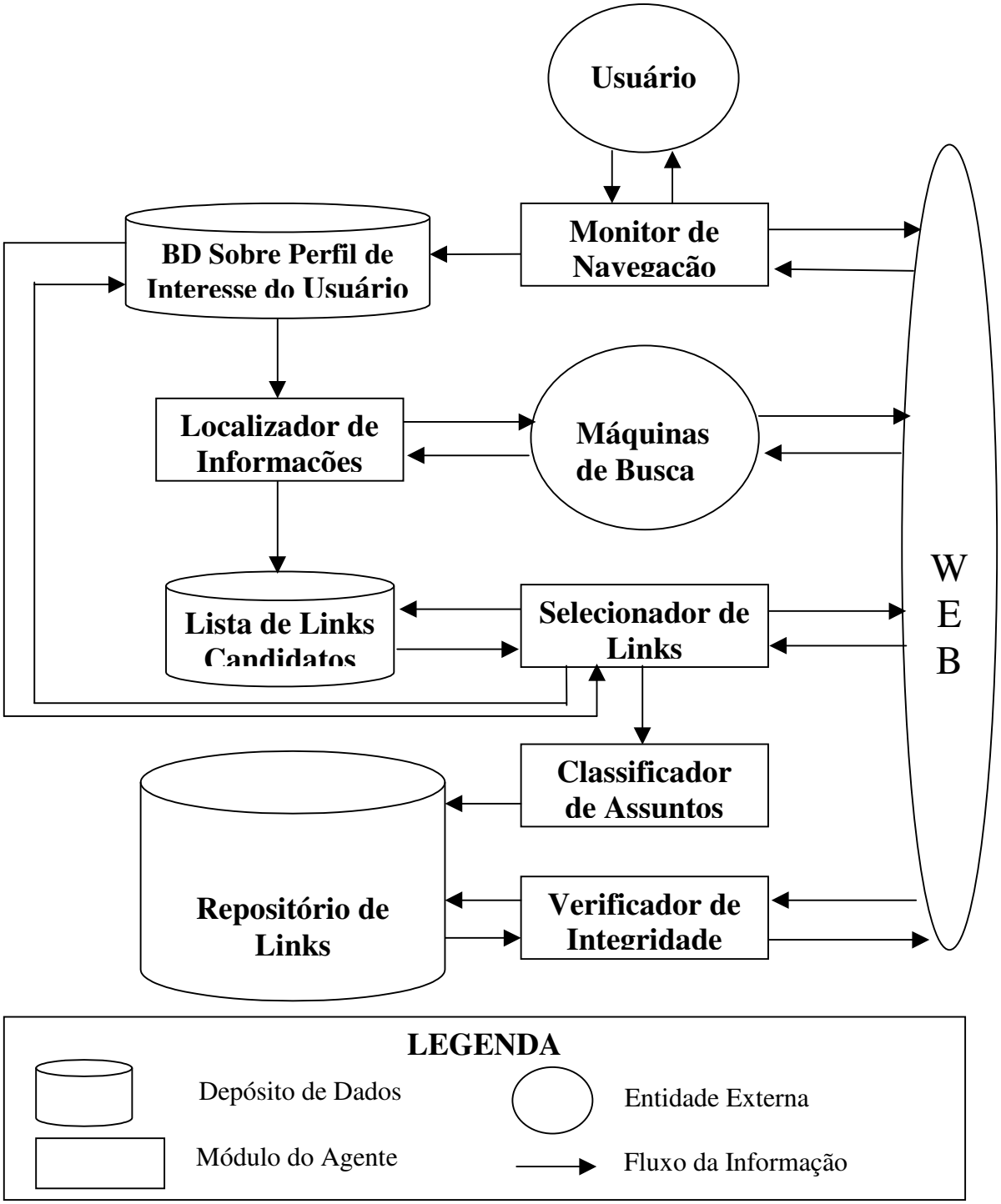


FIGURA 4.1. Estrutura Global do Modelo

De uma forma geral o modelo pode ser dividido em cinco grandes módulos, apresentados na Figura 4.1 :

- **Monitor de Navegação**: Este módulo terá a função de acompanhar o usuário durante sua navegação e, a partir daí, procurar atualizar uma base de conhecimentos sobre o perfil de interesse do usuário, tanto de forma automática quanto interativa;

- **Localizador de Informações**: Este módulo será responsável por buscar na Web informações que satisfaçam ao perfil de interesse do usuário e adicioná-las em uma lista, com a finalidade de serem selecionadas futuramente para o repositório, ou não;

- **Selecionador de Links:** O módulo selecionador deverá escolher, através de diversos critérios, quais *links* existentes na lista criada pelo localizador de informações serão efetivamente adicionados ao repositório definitivo;

- **Classificador de Assuntos:** Este módulo será o responsável pela classificação por assunto das informações selecionadas, pelo módulo anterior, e inserí-las de forma organizada no repositório;

- **Verificador de Integridade**: Este módulo deverá, periodicamente , realizar verificações em todos os *links* armazenados no repositório com a finalidade de retirar eventuais *links* desatualizados, inconsistentes ou ainda, que não correspondam mais ao perfil do usuário.

A figura 4.1 ilustra o relacionamento entre os diversos módulos do modelo e o fluxo de informações entre eles. Durante a navegação do usuário, o módulo monitor de navegação busca obter o perfil de interesse do usuário e atualizar o banco de dados sobre interesse do usuário. O módulo localizador de informações utiliza as informações armazenadas no banco de dados sobre perfil do usuário para gerar consultas que serão submetidas a diferentes máquinas de busca tradicionais. Os resultados obtidos nas buscas geradas anteriormente são armazenados em um banco de dados de endereços candidatos, e serão selecionados pelo módulo selecionador de *links* e submetidos ao classificador de assuntos a fim de serem armazenados no repositório de *links* de forma organizada quanto ao assunto. O módulo verificador de integridade tem como objetivo acessar cada um dos endereços armazenados no repositório para verificar se ainda estão ativos e consistentes e se é necessário removê-los.

4.3. Perfil de Interesse do Usuário

O perfil de interesse do usuário será utilizado na localização de informações a serem selecionadas e posteriormente adicionadas ao repositório de *links*. Para que seja possível estabelecer um perfil de usuário serão utilizados exemplos de páginas

desejáveis e páginas não desejáveis[KRU 97]. As páginas indicadas como exemplos também serão classificadas quanto ao assunto de interesse.

No modelo proposto, cada usuário poderá possuir um perfil específico e este perfil poderá ser dividido quanto a assunto. O modelo será definido utilizando, em partes, conceitos apresentados por [KRU 97] em seu agente InfoFinder.

4.3.1 Obtenção de Dados para o Perfil do Usuário

Para que se inicie a criação de um perfil de usuário será necessário que o próprio usuário, ao navegar na Web, indique alguns exemplos de páginas positivas e negativas e classifique-as quanto ao assunto através de uma interface do módulo monitor de navegação definido na seção 4.4.

Uma vez indicadas as páginas de exemplo será feita uma análise heurística (ver seção 4.3.2) sobre cada página com a finalidade de extrair os termos mais relevantes que possam representar o assunto tratado em cada página e futuramente servirem como argumento de busca para o módulo localizador de informações.

O perfil do usuário será, também, atualizado durante a utilização do agente através de análises heurísticas feitas sobre os documentos presentes no repositório de *links* e os novos adicionados a ele.

Os termos obtidos dos modelos serão armazenados em uma estrutura específica de um banco de dados relacional, a fim de proporcionar um acesso rápido e organizado aos dados.

4.3.2 Heurística

A extração dos termos que possam identificar o assunto que o texto trata é tarefa complicada, uma vez que existem muitos termos que possuem mais de um significado e podem induzir ao erro ao classificar um texto. Além disto os documentos disponíveis na Web, em sua grande parte HTML, não possuem um padrão de metadados, o que facilitaria muito sua extração.

Durante a análise do conteúdo dos documentos submetidos como exemplo e extração dos termos relevantes serão considerados todos os termos que fujam ao padrão do texto como um todo, como por exemplo, negritos, itálicos e fontes de tamanho diferenciado. Além disto, serão considerados os termos que ocorrem com grande freqüência no corpo do texto e os que estão entre *tags* que identificam título, autor e outros, quando estes existirem [BOL 98].

Para que seja possível identificar os termos que fogem ao padrão do texto é necessário descobrir qual é este padrão. Para isto será feita uma varredura sobre todo o

documento e será criada uma estrutura auxiliar (ver Figura 4.2), baseada em banco de dados relacional, para contar quantas palavras ocorrem com cada formatação de texto.

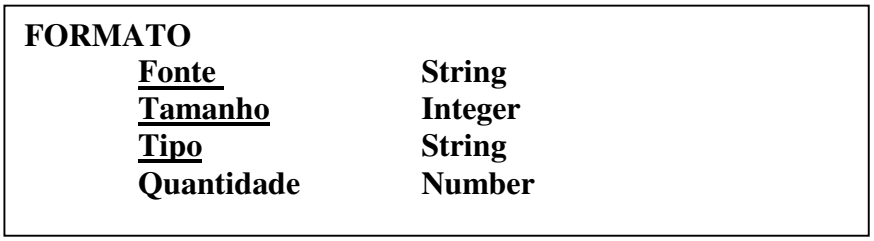


FIGURA 4.2 - Estrutura auxiliar de armazenamento para formatos.

A figura 4.2 apresenta a estrutura da tabela utilizada como auxiliar para a contagem da quantidade de palavras em cada padrão de formatação de texto. Ao varrer o documento, para cada termo será incrementado o campo q*uantidade*. Os campos *fonte, tamanho e tipo* são os campos chave da tabela. Desta forma a *tupla* que possuir o campo quantidade com maior valor representa o padrão de formatação do texto.

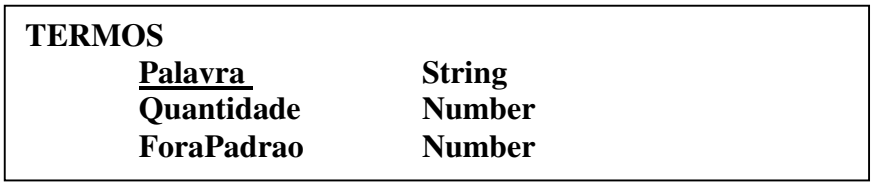


FIGURA4.3-Estrutura auxiliar de armazenamento para contagem de termos.

Uma vez descoberto o padrão de formatação do texto é realizada uma segunda varredura no texto com o objetivo de identificar os termos que mais ocorrem e quais estão fora do padrão de formatação. A figura 4.3. apresenta uma estrutura de armazenamento a ser utilizada para a realização da contagem de ocorrência de cada palavra no texto. Durante a varredura cada termo será adicionado à tabela TERMOS, que possui como campo chave o próprio termo, desta forma, quando o termo já existir na tabela o campo *quantidade* será incrementado e, caso o termo esteja fora do padrão de formatação o campo *ForaPadrao* é incrementado. Certamente as palavras que repetem-se muito no texto têm boa possibilidade de representarem o assunto que está sendo tratado, principalmente, se estas palavras coincidem com as que fogem do padrão de formatação do texto [LAW 99].

Porém nos textos, normalmente, existem muitos termos conectivos e adjuntos adnominais que, certamente serão os que mais ocorrerão. Para que não haja distorção nos resultados em função destes termos é proposta a criação de uma estrutura própria para armazenar estes termos, nas diferentes línguas, e, quando ocorrer a segunda varredura, citada anteriormente, estes termos devem ser desconsiderados.

FIGURA 4.4 - Estrutura de armazenamento para termos desconsiderados

A figura 4.4 apresenta a estrutura auxiliar utilizada para armazenar os termos conectivos e adjuntos adnominais, estes termos devem ser incluídos no momento da implementação e poderão ser atualizados pelo próprio usuário. A chave desta tabela é composta pelo termo e pela língua, porque o agente deverá considerar documentos em diferentes línguas.

Após a obtenção dos termos que mais ocorrem no texto deve-se buscar *tags* que indicam títulos, autores ou outros dados que possam identificar o assunto do documento. Para a composição da base de dados sobre o perfil de interesse do usuário serão considerados os termos mais significativos de cada documento, os termos obtidos dos títulos, autor e instituição, além do assunto informado pelo usuário e a classificação do documento como exemplo positivo ou negativo[BOL 98].

Para que seja possível detectar os vinte termos mais significativos que ocorrem no texto a tabela TERMOS (ver figura 4.3) será ordenada pelos campos *ForaPadrao* e *Quantidade* em ordem decrescente, assim buscando as vinte primeiras tuplas temos os termos que mais ocorrem com formatação fora do padrão.

4.3.3 Armazenamento do Perfil do Usuário

O armazenamento dos dados sobre o perfil de interesse do usuário se dará sobre banco de dados relacional, organizado de forma que possa ser dividido por usuário e, para cada usuário, por assunto de interesse.

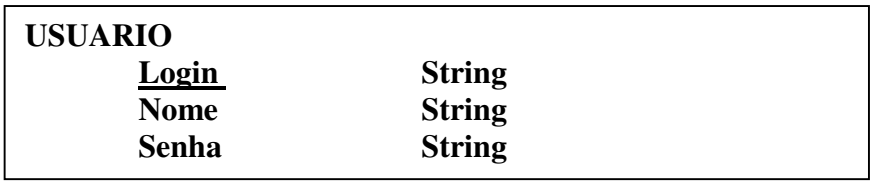


FIGURA 4.5 - Entidade Usuário

A figura 4.5 apresenta a estrutura de armazenamento da entidade responsável por manter os cadastros dos usuários. Esta entidade será utilizada para identificar cada usuário que utiliza o agente e, a partir daí, criar e manter o seu perfil de interesse.

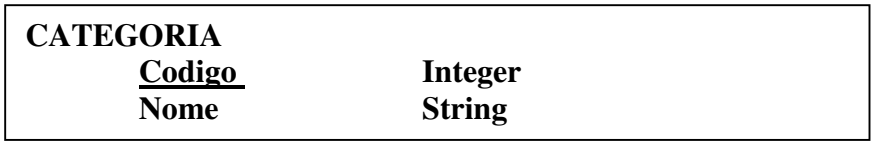


FIGURA 4.6 - Entidade Categoria de Assunto

Para que possamos criar alguma organização quanto a assunto em nosso perfil de usuário definimos uma entidade CATEGORIA, apresentada na figura 4.6, com o objetivo de classificar os documentos de forma geral. Para detalharmos a divisão dos assuntos utilizamos, também, uma entidade chamada ASSUNTO, onde o usuário poderá cadastrar os assuntos de uma determinada categoria de interesse e futuramente relacioná-los aos documentos exemplo. A estrutura da entidade ASSUNTO pode ser visto na figura 4.7. Os dados cadastrados em CATEGORIAS e ASSUNTOS serão utilizados também como forma de organização do repositório de *links*, especificado na seção 4.8.

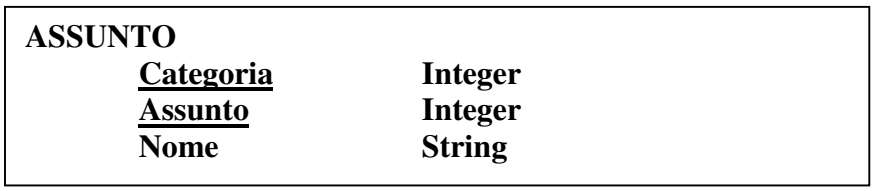


FIGURA 4.7 - Entidade Assunto

O perfil de interesse do usuário, basicamente, se dará pelos exemplos de documentos armazenados e classificados como positivos ou negativos armazenados na entidade DOCUMENTO, que é apresentada na figura 4.8. Além da ordenação natural da entidade, dada pela chave primária, também será adicionada uma ordenação pelo campo *Ocorrencias,* pois este campo será de suma importância durante a montagem das consultas para busca de informações, o que será detalhado na seção 4.5.

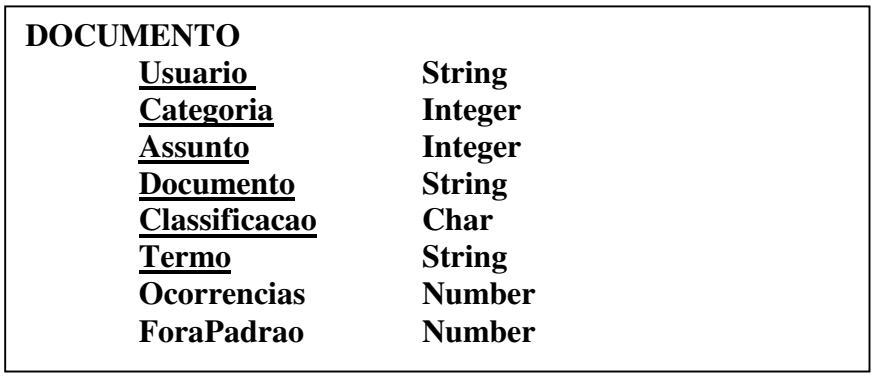


FIGURA 4.8 - Entidade Documento

A estrutura aqui proposta, para armazenar o perfil do usuário em bancos de dados relacionais, visa fornecer tanto uma recuperação rápida aos dados, quanto realizar consultas de diferentes maneiras, com isto, permitindo que vários documentos exemplo sejam combinados no sentido de fornecer maior eficiência para buscas na Web a partir de cada perfil.

Os campos *usuário, categoria* e *assunto* tem como objetivo identificar de quem é o perfil armazenado e qual assunto está tratando. O campo *documento* identifica o nome do documento, o campo *classificação* identifica se o exemplo é positivo ou negativo, o campo *termo* armazena cada palavra considerada como exemplo, o campo *ocorrência* armazena a quantidade de vezes que aquele termo ocorre no documento e o campo *forapadrao* armazena a quantidade de vezes que o termo ocorre fora do padrão de formatação do documento.

4.4 Monitor de Navegação

O módulo monitor de navegação tem como principal objetivo fornecer uma interface que possibilite ao usuário interagir com o agente para influenciar na atualização do seu perfil, indicando páginas como exemplos positivos ou negativos e classificá-las quanto a categoria e assunto[ASN 97]. Também o monitor deve permitir que o usuário adicione alguma página ao repositório de *links*, crie, altere e exclua categorias e assuntos.

Além das funções citadas acima o monitor de navegação tem como função acompanhar o usuário ao navegar na Web e, a partir daí, procurar obter e atualizar de forma automática e constante o seu perfil de interesse. O perfil é um componente fundamental para o êxito na localização de informações pelos demais módulos do agente, por isto sua manutenção deve ser freqüente.

A interface do monitor de navegação deve ser simples e de fácil utilização, ficando ativa, em retaguarda, e podendo ser chamado pelo usuário quando for necessário. Para que possa obter informações sobre quais páginas o usuário navega, o monitor de navegação trabalha como um *proxy*, ou seja, intercepta as comunicações HTTP recebidas pelo *browser* e, sobre elas aplica os procedimentos de extração de termos vistos na seção 4.3.2.

Após extraídos e contados os termos de cada documento eles são comparados com o perfil de interesse do usuário já armazenado. Para esta comparação é utilizada a técnica TFIDF de [SAL 89], melhor detalhada na seção 4.6.2. Caso o resultado da comparação caracterize que o documento é compatível com o perfil de interesse do usuário, este documento será inserido no banco de dados que armazena os modelos de interesse. O nível de semelhança entre cada documento analisado e o perfil, para que seja selecionado, é configurado pelo usuário. Outra possibilidade de configuração deste módulo, por parte do usuário, é a atualização ou não do perfil do usuário. Esta opção é importante pois, após algum tempo atualizando o perfil este pode ficar muito grande, comprometendo a performance do sistema.

4.5 Localizador de Informações

O módulo localizador de informações tem como principal função localizar informações que satisfaçam ao perfil de interesse do usuário, utilizando, para isto, sites

de busca tradicionais como Altavista, Lycos [5] , e outras. As informações localizadas também deverão ser filtradas, validadas e seus endereços adicionados na lista de *links candidatos*, a fim de futuramente serem selecionadas para o repositório de informações.

4.5.1 Montagem de Consultas para Sites de Busca

A primeira tarefa do módulo localizador de informações será obter, no perfil do usuário, termos que possam ser utilizados na montagem de consultas para serem submetidas aos diversos sites de busca.

Para a obtenção de termos que sejam relevantes para a realização de uma busca serão considerados os termos que mais ocorrem dentro de cada documento, e fora do padrão de formatação, combinando-se os diversos documentos de cada categoria e assunto, pois de uma forma geral podemos considerar que estes termos deverão existir nos documentos que são de interesse do usuário.

A consulta à base de dados que armazena o perfil do usuário se dará através da leitura da entidade DOCUMENTO, ordenada de forma decrescente pelos campos *Ocorrências e ForaPadrão,* para que seja possível recuperar os termos que mais ocorrem e com formatação fora do padrão em cada documento. Serão considerados úteis para a geração de consultas todos os termos armazenados em cada documento, porém para cada consulta serão utilizados no máximo três termos de cada documento, e estes combinados com termos de outros documentos do mesmo assunto e categoria. Desta forma poderemos ter uma grande diversidade de consultas, porém a redundância de termos na mesma consulta deve ser evitada, assim como a redundância de consultas inteiras, para isto será utilizada uma estrutura auxiliar de armazenamento que possuirá o termo como parte da chave primária.

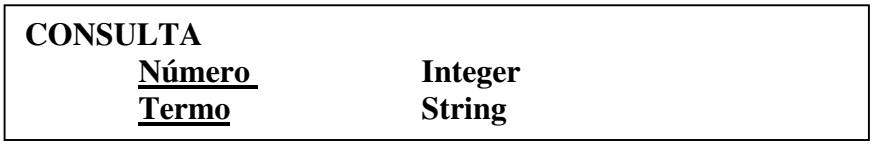


FIGURA 4.9 - Entidade Consulta

A figura 4.9 apresenta a estrutura auxiliar utilizada para evitar a utilização redundante de termos na mesma consulta. O campo *Número* será utilizado para identificar cada consulta gerada.

A efetiva montagem das consultas se dará através da combinação de todos os termos de cada documento. O número de termos utilizado em cada consulta poderá ser configurado pelo usuário no módulo de parametrização, que será visto mais adiante, entretanto, é conveniente lembrar que quanto menor o número de termos, mais ampla será a pesquisa e a possibilidade das páginas resultantes representarem o interesse do usuário será menor. Por outro lado, se a quantidade de termos for excessiva, a busca será muito específica, e, provavelmente resultará em um poucas páginas, ou mesmo em nenhuma.

[5] www.lycos.com

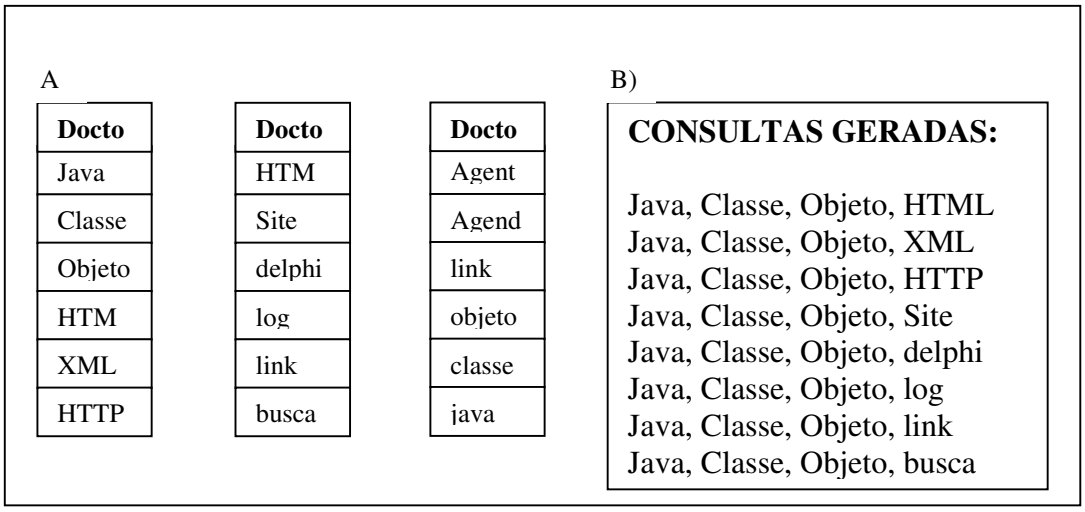


FIGURA 4.10: Exemplo da montagem de Consultas a partir do perfil do usuário.

A figura 4.10 representa um exemplo de montagem de algumas consultas, a partir de três documentos considerados como modelos positivos de um usuário, categoria e assunto. Observe que cada termo é combinado com todos os outros, do mesmo documento e dos outros documentos, gerando assim uma grande quantidade de consultas. No exemplo apresentado, estamos considerando que usuário configurou em quatro a quantidade de termos para cada consulta.

4.5.2 Envio das Consultas para os Sites de Busca

Uma vez selecionados os termos para a realização das buscas, estes devem ser submetidos aos principais sites de busca existentes. Para que uma busca seja enviada a um site de busca, é necessário montar uma URL específica para cada site de busca, uma vez que cada um deles possui um padrão próprio, apesar de todos terem alguma semelhança.

Observando alguns mecanismos de busca pudemos notar que a maioria das URLs montadas são compostas, basicamente, de três partes principais, a primeira parte identifica o site e o comando a ser executado, a segunda parte passa como parâmetros os termos a serem buscados, separados por símbolos próprios, e a terceira parte passa outros parâmetros particulares de cada site de busca.

a) Exemplo Google
http://www.google.com/search?q=ufrgs+agentes+internet&btnG=Pesquisa+Google&hl=pt&lr=

b) Exemplo AltaVista
http://www.av.com/cgibin/query?q=%2Bagentes+%2Bufrgs+%2Binternet&kl=XX&pg=q&Translate=on&search.x=21&search.y=14

c) Exemplo Cadê
http://busca.cade.com.br/scripts/engine.exe?p1=ufrgs+agentes+internet&p2=1&p3=1&p5=10&submit=Busca

d) Exemplo Yahoo
http://search.yahoo.com/bin/search?p=ufrgs+agentes+internet

e) Exemplo Excite
http://search.excite.com/search.gw?search=ufrgs+agentes+internet

f) Exemplo Lycos
http://www.lycos.com/srch/index.html?query=ufrgs+agentes+internet&lpv=1&loc=fromlycosmain

FIGURA 4.11- Exemplos de URLs de montadas por sites de busca.

A figura 4.11 apresenta exemplos de URLs montados por alguns sites de busca como Google, Altavista, Yahoo, Excite, Lycos, etc. Para a obtenção dos exemplos apresentados submetemos a cada um dos sites buscas para os termos "UFRGS", "agentes" e "Internet". Além destes termos vários outros foram testados e observamos que o padrão do URL não se alterou, exceto na segunda parte onde são passados os termos para busca. Entretanto, alguns sites de busca podem apropriar algumas particularidades, o que inviabiliza a construção de um padrão único.

Visto que não pode-se criar um padrão único para a submissão de URLs a diferentes mecanismos de busca e, é importante utilizar a maior quantidade possível destes mecanismos, torna-se interessante a definição dos padrões de URL de cada site de busca como parâmetros do sistema, assim o próprio usuário poderá configurar quais sites serão utilizados e com quais recursos.

Para o armazenamento dos parâmetros dos sites de busca é utilizada uma estrutura auxiliar com atributos para identificar o mecanismo de busca, os parâmetros iniciais, os caracteres separadores dos termos e os parâmetros finais. A figura 4.12 representa a estrutura utilizada para armazenar os parâmetros.

SITESDEBUSCA	
Nome	**String**
Inicio	**String**
Final	**String**
Separador	**String**
URLInicial	**String**
URLFinal	**String**
TagResInicial	**String**
TagResFinal	**String**

FIGURA 4.12 - Armazenamento de Parâmetros para Sites de Busca.

Cada combinação de termos, descrita na seção anterior, deve ser enviada a todos os sites de busca cadastrados pelo usuário, desta forma, diversas buscas são executadas em paralelo, procurando aproveitar ao máximo os recursos de comunicação disponíveis.

A figura 4.12 apresenta, também, os parâmetros URLInicial, URLFinal, TagResInicial e TagResFinal que serão utilizados durante o tratamento do retorno das buscas e seleção dos links, o que será visto nas próximas seções.

4.5.3 Tratamento do Retorno das Buscas

O resultado de cada consulta submetida a um site de busca é, normalmente, um documento HTML. Este documento HTML é composto de publicidade, links diversos, funcionalidades do site e uma lista de páginas encontradas, quando existirem, e seus respectivos URLs.

Como o nosso objetivo é extrair os URLs encontrados, devemos utilizar algum padrão de *tags* que represente exclusivamente o *link* de cada página encontrada. Através da análise de diversos retornos de sites de busca, observamos que cada um deles possui o seu padrão próprio.

```
<p><A HREF=http://www.mec.gov.br/seed/paped/projetos.shtm>MEC -
Educação a Distância - Projetos Selecionados</A><font size=-1><br> <b>...</b>
de Física Via <b>Internet</b>. <b>...</b> Produção ). Um Ambiente
<b>Inteligente</b> para Aprendizado <b>...</b> Pereira<br>
Rodrigues (<b>UFRGS</b> - Computação), <b>Agente</b> Avaliação de Ensino
<b>...</b>
<br><font color=green>www.mec.gov.br/seed/paped/projetos.shtm - 44k - <A
HREF=/search?q=cache:Lt5aIqVKH1M:www.mec.gov.br/seed/paped/projetos.shtm
+ufrgs+agente+inteligente+internet&hl=pt class=l>Em cache</A> - <A
HREF=/search?hl=pt&lr=&safe=off&num=10&q=related:www.mec.gov.br/seed/pa
ped/projetos.shtm class=l>Páginas Semelhantes</A></font></font><br>
```

FIGURA 4.13 - Trecho HTML de retorno da busca do Google.

Analisando o documento HTML de retorno de busca submetida ao Google é possível observar que os resultados são apresentados na forma de parágrafos identificados pela tag <p>. A figura 4.13 apresenta um trecho de documento retornado pelo Google.

```
<span class=s>
URL: http://www.inf.ufrgs.br/~fontes/seminario.html
</span>
<br>
<a
href="http://jump.altavista.com/trans.go?urltext=http://www.inf.ufrgs.br/~fontes/se
minario.html&language=pt">Translate</a> 
```

FIGURA 4.14 - Trecho HTML de retorno da busca do AltaVista.

Analisando o documento retornado pelo AltaVista é possível observar que os URLs localizados, quando existem, são localizados entre as tags e . Observa-se, também, que o endereço é sempre precedido da string URL:. A figura 4.14 apresenta um trecho do documento retornado pelo AltaVista.

```
<li><a
href="http://click.hotbot.com/director.asp?id=1&amp;target=http%3A%2F%2Fj
ms%2Ew%2Ecl%2Fabe%2Fafi%2Ehtml&amp;query=agentes+inteligentes+internet
&amp;rsource=LCOSWF"><B>Agentes</B> físicos <B>inteligentes</B>:
¿bendición o maldición para periodistas?</a>

    - Uno de los primeros investigadores que habló públicamente en nuestro país de
la existencia de los "<B>agentes</B> físicos <B>inteligentes</B>" fue el profesor
norteamericano Thomas Cooper del Emerson College de Bos<br><i><font
face="verdana" size="-1"
color="#666666">http://jms.w.cl/abe/afi.html</font></i><br>

    [<a
href="http://translation.lycos.com/?p=http%3A%2F%2Fjms%2Ew%2Ecl%2Fabe%
2Fafi%2Ehtml">Translate</a>]<p></p>
</li>
```

FIGURA 4.15 - Trecho HTML de retorno da busca do Lycos.

Analisando o documento retornado pelo Lycos pode-se observar que os resultados estão sempre entre as tags <li> e e representados na forma de links, como pode ser visto na figura 4.15.

Como foi observado, nos sites de busca analisados, cada um possui o seu padrão de retorno, porém todos eles seguem um formato em comum, de apresentar seus URLs entre tags específicos e que identificam unicamente o resultado de cada busca.

Isto permite criar uma generalização para que o agente utilize qualquer site de busca através da configuração da entidade SITESDEBUSCA, apresentada na figura 4.12.

O efetivo tratamento do retorno das buscas se dará pela localização de expressões regulares que satisfaçam ao padrão especificado na entidade SITESDEBUSCA de cada ferramenta, e a extração da URL contida entre as tags especificadas.

Quando são localizados muitos documentos, os sites de busca apresentam o resultado em várias páginas e, os principais sites, possuem mecanismos de *ranking*[GOO 02] com o objetivo de mostrar nas primeiras páginas os documentos considerados mais importantes. Diante disto e, procurando tirar proveito do *ranking* dos próprios sites de busca , neste modelo é considerada somente a primeira página de resultados de cada site de busca, mesmo porque, se fossem utilizados todos os resultados seria gerado um volume muito grande links, o que poderia causar uma sobrecarga no sistema.

Uma vez extraídos os links, estes serão adicionados a uma lista de links candidatos, para futuramente serem selecionados para inclusão no repositório definitivo ou excluídos. A próxima seção trata da lista de *links* candidatos.

4.5.4 Lista de Links Candidatos

Para que sejam selecionados os *links* de maior relevância e que realmente atendam ao perfil de interesse do usuário, todos os *links* localizados pelos sites de busca serão inseridos em uma lista temporária, onde ficarão até serem selecionados, ou descartados, pelo Módulo Selecionador.

LINKSCANDIDATOS	
URL	**String**
Data_Criação	**Date**
Data_Atualização	**Date**
Referências	**Number**
Categoria	**Integer**
Assunto	**Integer**
Verificada	**Char**
Pontos	**Integer**

FIGURA 4.16 - Estrutura de Armazenamento Lista de *Links* Candidatos.

A figura 4.16 apresenta a estrutura utilizada para armazenar, temporariamente, os *links* durante o processo de seleção. O campo *URL* é a chave primária da tabela para que o mesmo link não seja adicionado mais de uma vez na lista. Os campos *Data_Criação* e *Data_Atualização* armazenam a data de criação e de última atualização de cada página.

O campo *Referências* armazena a quantidade de referências encontradas em outras páginas à URL armazenada. Esta informação será obtida e utilizada durante o processo de seleção, que será visto na próxima seção. Os campos *Verificada*, *Pontos, Categoria e Assunto* também serão utilizados na seleção e serão descritos mais adiante.

4.6 Selecionador de Links

O módulo *Selecionador de Links* tem com função selecionar, da lista de candidatos, quais links serão adicionados ou não no repositório definitivo. O processo de seleção deverá ser baseado em diversos critérios de seleção, tais como quantidade de referências ao link, idade da página, qualidade de conteúdo e outros.

4.6.1 Critérios de Seleção

Para a realização da seleção dos *links* a serem adicionados no repositório definitivo foram estabelecidos alguns critérios baseados em qualidade, pertinência e popularidade de cada documento obtido pelos sites de busca.

- **Popularidade Global:** O critério de Popularidade Global baseia-se na quantidade total de refêrencias de cada documento avaliado e tem como objetivo selecionar os documentos mais referenciados por outros documentos. Porém, este critério não pode ser usado isoladamente, pois documentos novos, obviamente, terão uma quantidade menor de referências, mas isto não significa que o documento seja menos importante. Para que este critério tenha validade é fundamental que seja combinado com os demais critérios avaliados;

- **Popularidade Média:** O critério de Popularidade Média procura suprir as deficiências da Popularidade Global. Este critério é, basicamente, a Popularidade Global dividido pela quantidade de dias que a página está disponível;

- **Tempo Mínimo de Publicação:** O critério Tempo Mínimo de Publicação procura evitar que páginas muito recentes sejam adicionadas ao repositório definitivo. O tempo mínimo de publicação é configurado pelo usuário através do módulo de parametrização.

- **Qualidade do Conteúdo:** O critério Qualidade do Conteúdo de um documento é baseado na semelhança entre o seu conteúdo e o perfil de interesse do usuário. Quanto maior a semelhança, maior é a qualidade do conteúdo. Este critério é representado na forma de pontos, obtidos pela comparação entre a freqüência de termos no documento e a freqüência de termos do perfil de interesse do usuário.

Os critérios acima citados são considerados, para seleção, na seguinte ordem: Qualidade do Conteúdo, em ordem crescente e Popularidade Média e Popularidade Global em ordem decrescente. A organização dos critérios nestas ordens faz com que os documentos com maior qualidade de conteúdo e maior popularidade

localizem-se no início da lista. Desta forma serão selecionados os *n* primeiros *links*, onde *n* é a quantidade de *links* configurados pelo usuário para serem adicionados por período.

4.6.2 Processo de Seleção

O processo de seleção de *links* é iniciado a partir da *Lista de Links Candidatos*, apresentada na seção 4.5.4. A primeira etapa é a realização de uma verificação dos links candidatos com o objetivo de eliminar da lista os *links* que não atendam aos critérios de *Tempo Mínimo de Publicação.* Esta tarefa é bastante simples, uma vez que consiste numa leitura seqüencial da lista, comparação das datas armazenadas com os parâmetros estabelecidos pelo usuário e exclusão dos links que estiverem fora do padrão especificado.

¬ **Obtenção do Critério de Qualidade do Conteúdo:**

A qualidade do conteúdo de um *link*, neste trabalho, é considerada maior quando o seu conteúdo se aproxima mais do perfil de interesse do usuário.

Para a comparação, entre o perfil de interesse do usuário e o documento recuperado, convertemos o banco de dados que contém os exemplos positivos de interesse e o documento recuperado em vetores, onde cada elemento representa o peso dos termos no documento, calculados pelo método "*Term Frequency X Inverse Document Frequency*" (TFIDF) de [SAL 89], encontramos o ângulo entre um vetor e outro. O menor ângulo encontrado é o nosso critério de qualidade, pois, quanto menor o ângulo, maior é a proximidade do documento com o perfil e assunto esperado pelo usuário[CHE 98].

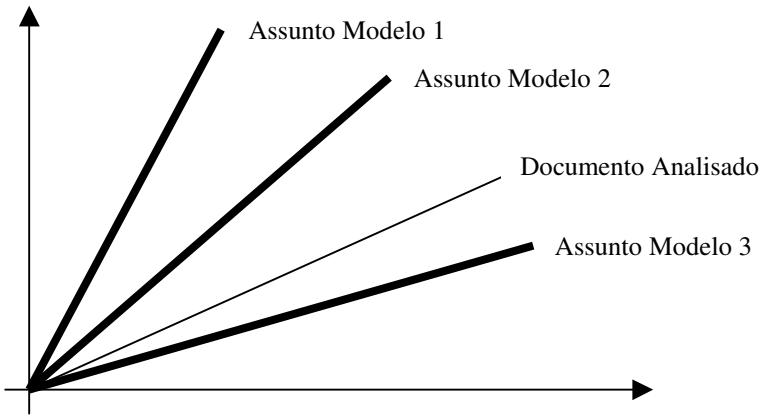


FIGURA 4.17 Comparação entre Vetores

Como um usuário pode ter inúmeros assuntos de interesse no mesmo banco de dados, para cada assunto é criado um vetor específico e, o ângulo entre cada vetor de documento localizado é comparado com todos os vetores sendo considerado o menor ângulo como critério de seleção, vinculado ao assunto mais próximo. A figura 4.17 apresenta uma representação gráfica da comparação de um documento com três assuntos de interesse. É possível observar que o documento analisado aproxima-se mais do "Assunto Modelo 3".

O método TFIDF, de Salton, é utilizado para calcular a medida de freqüência de uma palavra no documento. Segundo [SAL 89], o peso W_i de uma palavra d_i num dado documento é dado por:

$$Wi = \left(0{,}5 + 0{,}5\frac{tf(i)}{tfmax}\right)\left(\log\frac{n}{df(i)}\right)$$

Onde:

tf(i) $\rightarrow$ é a freqüência do termo *i* no documento *t*, isto é, o número de vezes que a palavra d_i aparece no documento;

df(i) $\rightarrow$ é a freqüência do documento, isto é, o número de documentos da coleção que contém a palavra d_i.

n $\rightarrow$ é o número total de documentos da coleção;

tf_{max} $\rightarrow$ é a freqüência máxima de uma palavra, entre todas as palavras do documento.

Uma vez obtido o peso de cada palavra do documento analisado, cada peso é colocado no vetor, na posição correspondente à palavra. O mesmo ocorre para cada conjunto de modelos positivos de interesse, atrelados ao seu assunto. Assim temos um vetor de pesos para o documento analisado e um vetor para cada assunto de interesse do usuário[CHE 98].

O valor do critério de qualidade utilizado é o menor ângulo obtido entre o vetor que representa o documento analisado e cada um dos vetores que representam os assuntos de interesse do usuário[BAU 2002]. A obtenção do ângulo entre os vetores é dada pela seguinte fórmula:

$$angulo = a\cos\left(\frac{\sum(ai * bi)}{\sqrt{\sum_{vetora}(ai)^2} * \sqrt{\sum_{vetorb}(bi)^2}}\right)$$

onde, *a* representa o vetor analisado e *b* representa o vetor de um assunto de interesse. A fórmula acima é aplicada para cada um dos vetores de interesse, sendo que o menor ângulo é aproveitado, pois representa maior proximidade entre um assunto e o documento analisado.

¬ **Obtenção do Critério de Popularidade Global:**

O critério de Popularidade Global é baseado na quantidade de referências detectadas em outros documentos para o documento analisado. O mecanismo utilizado para obter o número de referências a um documento utiliza os mesmos sites de busca utilizados para obter os links candidatos.

Como a maioria dos sites de busca apresentam a quantidade de páginas encontradas após cada busca, podemos submeter a URL de cada link candidato como argumento de busca para os sites disponíveis, assim obtemos a quantidade de páginas que fazem referência ao *link* em cada ferramenta.

Para que possamos capturar a quantidade de páginas encontradas, em cada pesquisa, é necessário conhecer o padrão de retorno cada site de busca, para isto analisamos alguns documentos HTML retornados.

```
<b class="results">Localizar Resultados de
"+ufrgs"</b>      <b class="pagesFound"
NOWRAP>
 
Páginas da Web: 62,386 
páginas encontradas.</b>
```

FIGURA 4.18 Trecho em HTML Retorno de Pesquisa no AltaVista

A figura 4.18 apresenta um trecho do retorno em HTML de uma busca submetida ao AltaVista. Neste trecho é observado que a quantidade de páginas encontradas localiza-se após o texto ‘Páginas da Web: ‘; que por sua vez está entre as tags <b class="pagesFound" NOWRAP> e .

Assim como o AltaVista, os demais sites de busca possuem padrões próprios para exibir seus resultados. Estes padrões são armazenados nos campos *TagResInicial* e *TagResFinal* da entidade SITESDEBUSCA, detalhada na seção 4.5.2. Quando retorna cada busca, os padrões de cada site são utilizados para obter-se a quantidade de ocorrências da URL consultada.

Como utilizamos *n* sites de busca, cada um retorna uma quantidade diferente de documentos encontrados. Para que tenhamos um número único utilizamos a média aritmética entre todos os resultados como sendo o nosso critério de Popularidade do documento. Portanto, o valor da Popularidade é dado por:

$$p = \frac{\sum_{sites} pe}{n}$$ onde,

$p \rightarrow$ é a Popularidade Global;

$pe \rightarrow$ é a quantidade de páginas encontradas em cada busca e,

$n \rightarrow$ é a quantidade de sites de busca utilizados.

¬ **Obtenção do Critério de Popularidade Média:**

O Critério de Popularidade Média tem como objetivo corrigir algumas distorções que podem ser geradas quando utilizamos o Critério de Popularidade Global, pois documentos novos, obviamente, possuem Popularidade Global Baixa, mas podem não ser menos importantes que alguns documentos com Popularidade Global Alta.

A obtenção do Critério de Popularidade Média é relativamente simples, uma vez que seu valor é o resultado da divisão da Popularidade Global pela quantidade de dias em que o documento está disponível na Internet. Para obtermos a quantidade de dias em que o documento está disponível subtraímos da data atual a data em que o documento foi criado.

¬ **O Processo de Seleção Baseado nos Critérios**

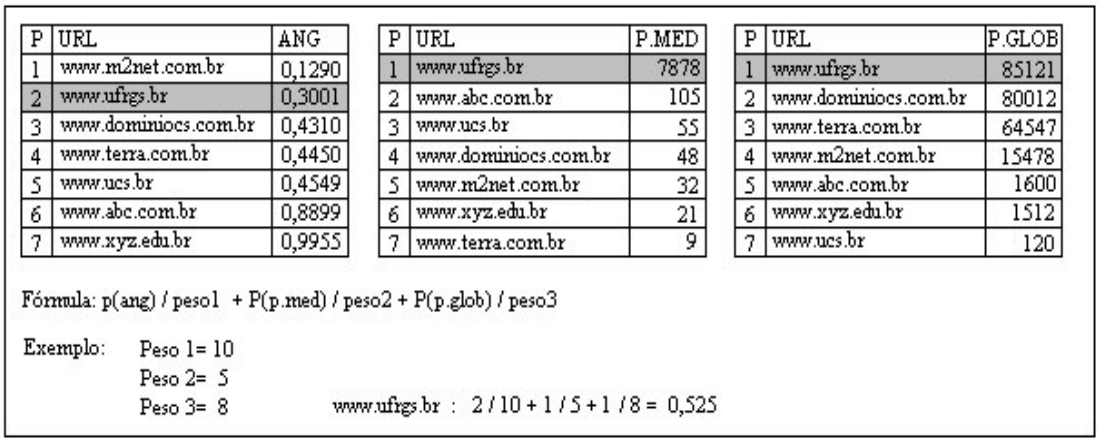
Após a obtenção dos critérios citados acima, estes são utilizados para a seleção de quais *links* serão adicionados ao repositório definitivo. Para a realização desta tarefa são considerados os critérios Qualidade do Conteúdo, Popularidade Global e Popularidade Média, de forma que os três tenham contribuição, mais ou menos equivalente, durante o processo de seleção.

A simples ordenação da lista de *links* candidatos, pelos valores dos critérios, não permite uma seleção coerente, pois privilegia um critério em detrimento dos outros. Por exemplo, se um *link* tem critério de qualidade excelente e popularidade baixa, talvez este *link* não seja tão importante quanto outro de qualidade mais baixa e popularidade alta. Portanto, é fundamental que a seleção seja feita considerando todos os critérios.

Para considerar todos os critérios de forma consistente, foi criada uma ordenação para cada critério, sendo que para o critério de qualidade a lista é ordenada de forma crescente, pois o valor armazenado é o ângulo do vetor em relação ao modelo, assim sendo, quanto menor o ângulo, maior é a qualidade do documento. Os critérios de popularidade (Global e Média) são ordenados de forma decrescente, de forma que os links com maior popularidade fiquem no início.

Com o objetivo de dar maior flexibilidade ao usuário, no processo de seleção, o peso de cada critério pode ser configurado através de parâmetros. Os parâmetros consistem de valores numéricos, informados pelo usuário no intervalo de 0 (onde o critério é desconsiderado) a 10. Quanto maior o valor informado para o critério maior será sua importância na seleção.

A seleção, propriamente dita, usa a posição de cada link avaliado em cada uma das ordenações, dividido pelo peso de cada critério. A soma dos valores obtidos resulta no valor pelo qual os links são selecionados. Estes valores são ordenados em ordem crescente e os *n* primeiros são selecionados, onde *n* é quantidade de links configurados pelo usuário para serem adicionados num determinado intervalo de tempo.



P	URL	ANG
1	www.m2net.com.br	0,1290
2	www.ufrgs.br	0,3001
3	www.dominiocs.com.br	0,4310
4	www.terra.com.br	0,4450
5	www.ucs.br	0,4549
6	www.abc.com.br	0,8899
7	www.xyz.edu.br	0,9955

P	URL	P.MED
1	www.ufrgs.br	7878
2	www.abc.com.br	105
3	www.ucs.br	55
4	www.dominiocs.com.br	48
5	www.m2net.com.br	32
6	www.xyz.edu.br	21
7	www.terra.com.br	9

P	URL	P.GLOB
1	www.ufrgs.br	85121
2	www.dominiocs.com.br	80012
3	www.terra.com.br	64547
4	www.m2net.com.br	15478
5	www.abc.com.br	1600
6	www.xyz.edu.br	1512
7	www.ucs.br	120

FIGURA 4.19 - Exemplo do Calculo de Peso Para Seleção.

A figura 4.19 ilustra o processo de calculo utilizado para obter-se o valor que é utilizado para ordenar a lista de candidatos para seleção. Observe, que no exemplo acima foi calculado o valor para a URL www.ufrgs.br, este mesmo processo repete-se para todas as URL armazenadas. Os pesos utilizados são configurados pelo usuários e são utilizados para todas as URL avaliadas.

Após a obtenção do valor final de cada *link*, a lista é ordenada de forma crescente por este valor. Os menores valores são considerados os melhores, e serão os selecionados para inserção no repositório definitivo.

4.7 Classificador de Assuntos

O módulo Classificador de Assuntos é o responsável pelo armazenamento dos *links* selecionados, no repositório definitivo, de forma organizada quanto à categoria e assunto.

O processo de classificação dos *links* quanto à categoria/assunto é relativamente simples, uma vez que durante o processo de seleção (ver seção 4.6), quando é obtido um valor para o critério de qualidade, cada *link* é vinculado à categoria/assunto que contem o conteúdo mais semelhante.

Como os *links* selecionados já são pré-classificados, o Classificador de Assuntos tem com tarefa mais importante armazená-los no Repositório. Esta tarefa consiste, apenas em gravar os *links* no banco de dados, especificado na próxima seção. Outra importante tarefa deste módulo é a de atualizar o perfil de interesse do usuário. Para isso, são inseridos no banco de dados do perfil os termos que mais ocorrem em cada documento selecionado.

4.8 Repositório de Links

Para que se tenha maior flexibilidade e facilidade de manutenção, foi optado por organizar o armazenamento dos *links* selecionados em banco de dados relacional. Tal organização permite que os *links* sejam adicionados, excluídos, alterados e consultados, também por outras ferramentas, possibilitando a integração com outros sistemas e agentes.

REPOSITORIO	
Usuário	**String**
Categoria	**Integer**
Assunto	**Integer**
URL	**String**
Data_Incl	**Date**

FIGURA 4.20 - Estrutura de Armazenamento do Repositório de Links

A figura 4.20 apresenta a estrutura de armazenamento adotada para o repositório de *links*. Pela estrutura adotada, cada *link* é classificado quanto a assunto, cada assunto quanto a categoria e cada categoria, quanto a usuário. Desta forma é possível manter, na mesma entidade, *links* de inúmeros usuários divididos quanto a Categoria e Assunto.

A recuperação das informações armazenadas pode ser realizada por qualquer ferramenta que utilize banco de dados relacional, tais como programas em Java, Delphi, PHP, ASP, e outras. Assim, cada usuário pode construir sua interface própria para visualização dos *links*.

4.9 Verificador de Integridade

O módulo Verificador de Integridade tem como objetivo manter o Repositório consistente, isto é, eliminar *links* que não estejam respondendo, definitiva ou temporariamente.

A verificação da integridade é realizada a cada fatia de tempo, conforme configuração feita pelo usuário. O processo de verificação consiste em ler todo o repositório e acessar cada um dos links armazenados. Após acessar os links o seu código de retorno é analisado, sendo que os que estão indisponíveis são armazenados, temporariamente, em uma tabela de avaliação, pois em muitos casos, o link está indisponível temporariamente e seria precipitado excluí-lo definitivamente.

AVALIACAO	
Usuário	**String**
Categoria	**Integer**
Assunto	**Integer**
URL	**String**
Retorno	**Integer**
Verificações	**Integer**

FIGURA 4.21- Estrutura de Armazenamento Tabela de Avaliação

A figura 4.21 apresenta a estrutura de armazenamento utilizada para manter os links que possuem algum tipo de problema. Nesta estrutura, além do usuário, categoria, assunto e URL, são armazenados o código do retorno e a quantidade de verificações já realizadas. Quando o código de retorno não é o mesmo armazenado, e o código refere-se a um problema, a quantidade de verificações é zerada, ou seja, inicia-se um processo de verificação para cada novo código.

A quantidade de verificações já realizadas é utilizada para determinar quando o *link* será definitivamente excluído do repositório, pois, os links armazenados na entidade AVALIACAO são verificados com maior intensidade, conforme intervalo de tempo configurado pelo usuário, até que o link esteja no estado normal, quando é eliminado da entidade AVALIACAO ou a quantidade de verificações atinja o limite configurado pelo usuário como parâmetro. Neste caso o link é eliminado do repositório e também da entidade AVALIAÇÃO.

4.10 Parâmetros Configuráveis

Como foi descrito nas seções anteriores, este modelo de agente procura ser bastante flexível, tendo vários parâmetros configuráveis pelo usuário. Estes parâmetros são armazenados em uma tabela do banco de dados relacional. A figura 4.22 apresenta a estrutura de armazenamento utilizada.

PARAMETROS	
Usuário	**String**
Termos_Busca	**Integer**
Tmp_Min_Pub	**Integer**
Int_Ins_Link	**Integer**
Qtd_Links_Ins	**Integer**
Peso_Qualidade	**Float**
Peso_Pop_Global	**Float**
Peso_Pop_Média	**Float**
Int_Ver_Integr	**Integer**
Qtd_Ver_Integr	**Integer**
Int_Tent_Integr	**Integer**
Nivel_Sem_Mn	**Float**
Atual_Perfil	**Char**

FIGURA 4.22. Estrutura de Armazenamento dos Parâmetros do Usuário

Cada usuário pode configurar o agente conforme suas preferências, assim, temos uma ocorrência na entidade PARAMETROS para cada usuário. Os parâmetros configuráveis são os seguintes:

- **Termo_Busca:** Determina a quantidade de termos usados para cada busca submetida a sites de busca;
- **Tmp_Min_Pub:** Determina o tempo mínimo, em dias, de publicação para que um link seja considerado durante o processo de seleção;
- **Int_Ins_Link:** Determina o intervalo de tempo para inserção de novos links no repositório, o tempo é expresso em dias;
- **Qtd_Links_Ins:** Determina a quantidade de links a serem inseridos no repositório a cada inserção;
- **Peso_Qualidade:** Determina o peso do critério de qualidade durante o processo de seleção; o valor pode variar de 0 a 10;
- **Peso_Pop_Global:** Determina o peso do critério de Popularidade Global durante o processo de seleção; o valor pode variar de 0 a 10;
- **Peso_Pop_Media:** Determina o peso do critério de Popularidade Média durante o processo de seleção, o valor pode variar de 0 a 10;
- **Int_Ver_Integr:** Determina o intervalo de tempo para a execução das verificações de integridade; o valor é informado em dias;
- **Qtd_Ver_Integr:** Determina a quantidade de verificações realizadas para eliminar um link inconsistente;
- **Int_Tent_Integr:** Determina o intervalo de tempo entre cada verificação de consistência na tabela AVALIACAO; valor informado em horas;
- **Nivel_Sem_Mn:** Determina o nível de semelhança entre um documento e o perfil de interesse do usuário, para que o documento seja inserido como modelo pelo Monitor de Navegação; o valor pode variar de 0 a 1, sendo que quanto menor, maior será a semelhança;
- **Atual_Perfil:** Determina se o Monitor de Navegação deve atualizar ou não o perfil de interesse do usuário automaticamente.

Os parâmetros, aqui apresentados, não são os únicos configuráveis pelo usuário. Na seção 4.5 foram apresentas possibilidades de configurar quais sites de busca serão utilizados. Desta forma o agente pode ser ajustado aos interesses e preferências de cada usuário.

4.11 Considerações Finais

Neste capítulo, foi apresentada a proposta de um modelo de agente capaz de obter o conhecimento sobre o perfil de interesse de usuários que mantenham páginas de *links*, buscar documentos com base no perfil do usuário, selecionar documentos e inserir suas referências num repositório de links, organizados quanto ao assunto, e ainda, verificar freqüentemente a consistência do repositório armazenado.

A proposta apresentada baseia-se na combinação de algumas técnicas citadas nos agentes estudados nos capítulos anteriores com outras técnicas que são introduzidas como contribuição deste trabalho. Portanto, a principal contribuição deste

trabalho é a apresentação de um modelo global fruto da combinação de algumas técnicas e a criação de outras.

Por se tratar de um modelo geral e amplo, alguns módulos foram descritos com uma profundidade menor, tais como o Monitor de Navegação, o Localizador de Informações e o Verificador de Integridade, entretanto, foi dado um enfoque especial aos módulos Selecionador e Classificador de Assuntos.

Com o objetivo de validar o modelo proposto, foi implementado um protótipo que contempla, principalmente, os módulos de Localização de Informações, Selecionador de Links e Classificador de Assuntos.

5 O Protótipo Implementado

Uma vez realizada a especificação do modelo para um agente capaz de manter repositórios de links, com base no perfil de interesse do usuário, foi implementado um protótipo que contempla alguns módulos deste modelo, com o objetivo de validá-lo. O presente capítulo descreve o protótipo implementado. Inicialmente são apresentadas as características pretendidas para o mesmo, bem como o ambiente e as ferramentas utilizadas para a implementação. Em seguida apresenta-se a estrutura da implementação e a interface de utilização. Finalmente é apresentada uma descrição sobre a avaliação realizada e alguns resultados preliminares.

5.1 Considerações Iniciais

O protótipo foi implementado a partir do modelo proposto e tem como principal objetivo validar o modelo, através da realização de simulações dos principais módulos. Por se tratar de um modelo geral e amplo, optou-se pela implementação parcial, ou seja, somente alguns módulos foram implementados.

Os módulos escolhidos para a implementação foram os seguintes:

- Localizador de Informações;
- Selecionador de Links;
- Classificador de Assuntos;
- Verificador de Integridade;
- Monitor de Navegação (Parcialmente).

Do módulo Monitor de Navegação foram implementadas as funcionalidades responsáveis pela interface do usuário, deixando-se de lado o acompanhamento à navegação do usuário e a respectiva atualização do perfil de interesse.

Todos os depósitos de dados especificados no modelo foram implementados, entretanto, os dados necessários para o funcionamento do agente que dependem de módulos não implementados foram adicionados manualmente durante a realização dos testes.

5.2 Ambiente e Ferramentas Utilizadas

Considerando o fato de que a especificação do modelo tenha sido feita para que o agente seja executado na máquina do cliente, e, que os dados de configuração, estruturação e resultados obtidos pela sua execução também sejam armazenados localmente, foram selecionados o ambiente de execução, a linguagem de programação e o banco de dados para a implementação do protótipo.

O ambiente operacional utilizado foi o *Microsoft Windows 98*, tendo-se optado por este ambiente por ser ele fortemente utilizado pelos usuários finais, e, sendo o agente para uso local sua interface e utilização torna-se mais facilitada.

A linguagem de programação selecionada para a implementação do protótipo foi a *Borland Dephi 4.* Tal linguagem foi escolhida pelas sua características de orientação a objetos, suporte a bancos de dados e suporte à programação para Internet, além da vasta disponibilidade de componentes para as mais variadas funções.

O banco de dados utilizado foi o *Paradox 7*. Tal banco foi escolhido pelas suas características de simplicidade e pouco consumo de espaço em disco, características desejáveis a um sistema de execução local.

Cabe salientar que o protótipo implementado tem como objetivo principal a validação do modelo, portanto outras implementações feitas no futuro poderão utilizar outros ambientes e ferramentas, de acordo com os objetivos pretendidos.

5.3 Estrutura de Implementação

A implementação do protótipo foi estruturada em um conjunto de programas, reunidos num projeto de forma integrada. Relacionado ao projeto e aos seus programas, também estão as tabelas do banco de dados.

A figura 5.1 apresenta a estrutura geral do projeto e suas interações com o banco de dados e o ambiente. Pela figura é possível perceber a utilização e a atualização das tabelas do banco de dados por cada programa, bem como a comunicação dos módulos do agente com a Internet.

A implementação das rotinas de busca, seleção de links e verificação de integridade foram implementadas utilizando-se a classe *Tthread* que viabiliza a execução de vários processos de forma concorrente, característica necessária, neste projeto, para o máximo aproveitamento dos recursos de comunicação. Com a utilização de *threads* é possível ter buscas submetidas a diferentes *sites* ao mesmo tempo.

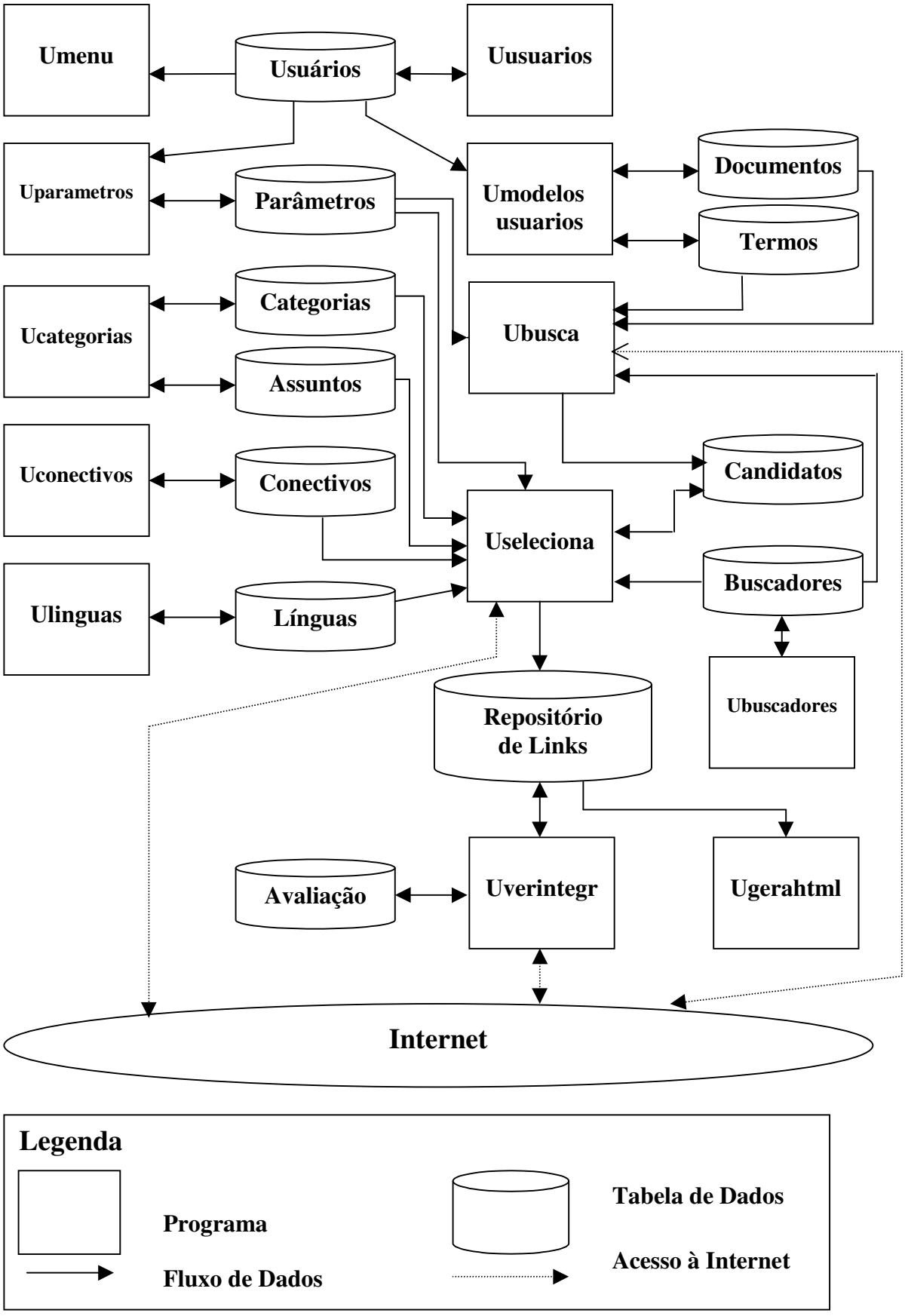


FIGURA 5.1. Estrutura geral da implementação.

A tabela 5.1 apresenta as tabelas do banco de dados utilizadas pelos programas implementados.

TABELA 5.1 – Tabelas do banco de dados utilizadas.

TABELA	FUNÇÃO
Tassuntos	Armazenar os assuntos de interesse do usuário. Estas informações são necessárias para a estruturação do repositório final de links.
Tavaliacao	Armazenar, temporariamente, os links que possuem algum tipo de inconsistência.
Tbuscadores	Armazenar dados sobre os sites de busca, seus parâmetros e strings utilizadas para a extração das URLs.
Tcandidatos	Armazenar, temporariamente, os links extraídos durante as buscas. Os dados permanecem nesta tabela até a sua seleção para o repositório ou exclusão.
Tcategorias	Armazenar as categorias de assunto de interesse do usuário, informações necessárias para estruturação do repositório.
Tconectivos	Armazenar os termos conectivos ou irrelevantes à interpretação dos documentos analisados.
Tdocumentos	Armazenar documentos amostra de interesse do usuário.
Tlinguas	Armazenar as línguas utilizadas durante as buscas.
Tparametros	Armazenar os parâmetros configurados por cada usuário para o funcionamento geral do agente.
Trepositorio	Armazenar os links selecionados de forma organizada quanto a usuário, categoria de assunto e assunto.
Ttermos	Armazenar os termos de interesse de cada usuário por categorias e assunto.
Tusuarios	Armazenar os dados para login do usuário.

A tabela 5.2, apresentada a seguir, relaciona os programas implementados no projeto, uma breve descrição das suas funções e as tabelas utilizadas ou atualizadas por cada módulo.

TABELA 5.2 - Descrição dos programas implementados.

PROGRAMA	FUNÇÃO	TABELAS
Ubusca	Buscar na Internet páginas que atendam ao perfil de interesse do usuário e adicioná-las na lista de candidatos.	Tbuscadores Tcandidatos Tdocumentos Tparametros Ttermos
Ubuscadores	Realizar a manutenção dos parâmetros dos sites de busca utilizados, bem como, cadastrar e excluir sites.	Tbuscadores
Ucategorias	Realizar a manutenção da estrutura de organização dos assuntos em categorias e assuntos.	Tassuntos Tcategorias
Uconectivos	Realizar a manutenção do cadastro de termos conectivos, ou outros, que devam ser desconsiderados durante as análises dos textos.	Tconectivos Tlinguas
Ugerahtml	Realizar a geração de arquivo HTML com base no repositório de links criado.	Tassuntos Tcategorias Trepositorio Tusuarios
Ulinguas	Realizar a manutenção do cadastro das línguas que serão utilizadas nas buscas.	Tlinguas
Umenu	Fornecer a interface principal do agente, permitir que o usuário ative outros programas e, principalmente, ativar de forma autônoma os programas de busca, seleção, verificação e outros.	Tparametros Tusuarios
Umodelosusuarios	Realizar o cadastramento de documentos para serem utilizados como modelo de perfil de interesse durante os testes do agente.	Tdocumentos Ttermos Tusuarios
Uparametros	Realizar a configuração do agente conforme suas preferências pessoais de cada usuário.	Tparametros Tusuarios
Useleciona	Selecionar quais links, dos armazenados na lista de candidatos, serão adicionados ao repositório definitivo.	Tassuntos Tbuscadores Tcandidatos Tcategorias Tconectivos Tlinguas Tparametros Trepositorio
Uusuarios	Realizar a manutenção do cadastro de usuários e suas respectivas senhas, este cadastro é utilizado para o login do usuário.	Tusuarios
Uverintegr	Realizar as verificações de integridade dos links armazenados no repositório.	Tavaliacao Trepositorio

O resultado da compilação do projeto é um único programa executável, sendo que este programa, quando em produção fica ativo na memória trabalhando em segundo plano. O programa *Umenu* é o responsável pela ativação dos processos de busca, verificações de integridade, seleções, e outros que devem ficar em constante execução. Para a implementação das características de execução contínua foi utilizado o componente *timer* do *Delphi*. Este componente foi programado para a cada fatia de tempo ativar os processos que estiverem agendados.

Outra situação possível para a aplicação é a sua operação em primeiro plano, neste caso o programa poderá ser utilizado pelo usuário para a realização de manutenções de cadastros e configuração dos parâmetros.

5.4 Interface de Utilização

A interface implementada levou em consideração a utilização do agente em ambiente local e a possibilidade de várias pessoas utilizarem o mesmo computador. Para que o usuário utilize a aplicação, com base no seu perfil pessoal, é necessário que ele esteja *logado* na mesma. O processo de *login* se dá através de um formulário próprio, que exige a identificação e a senha de cada usuário. O formulário de *login* é apresentado na figura 5.2.

FIGURA 5.2. Formulário de Login do Usuário.

A interação do usuário com a aplicação se dá através de um formulário principal, o qual conta com um menu suspenso com as opções: Arquivo, Ações, Consultas e Sair. Além do menu, a interface também conta com atalhos para os principais módulos. A figura 5.3 apresenta a interface principal do protótipo.

A opção "Arquivo" do menu principal permite o acesso aos cadastros de termos conectivos, línguas, categorias de assuntos, usuários, modelos de usuários, parâmetros e buscadores.

A opção “Ações” permite o acesso à visualização dos processos de busca de links, seleção de links e verificação de integridade. Além disso, possibilita a geração de arquivos HTML a partir do repositório atualizado.

A opção “Consulta” permite o acesso às consultas dos cadastros, do repositório de links e da lista de links candidatos.

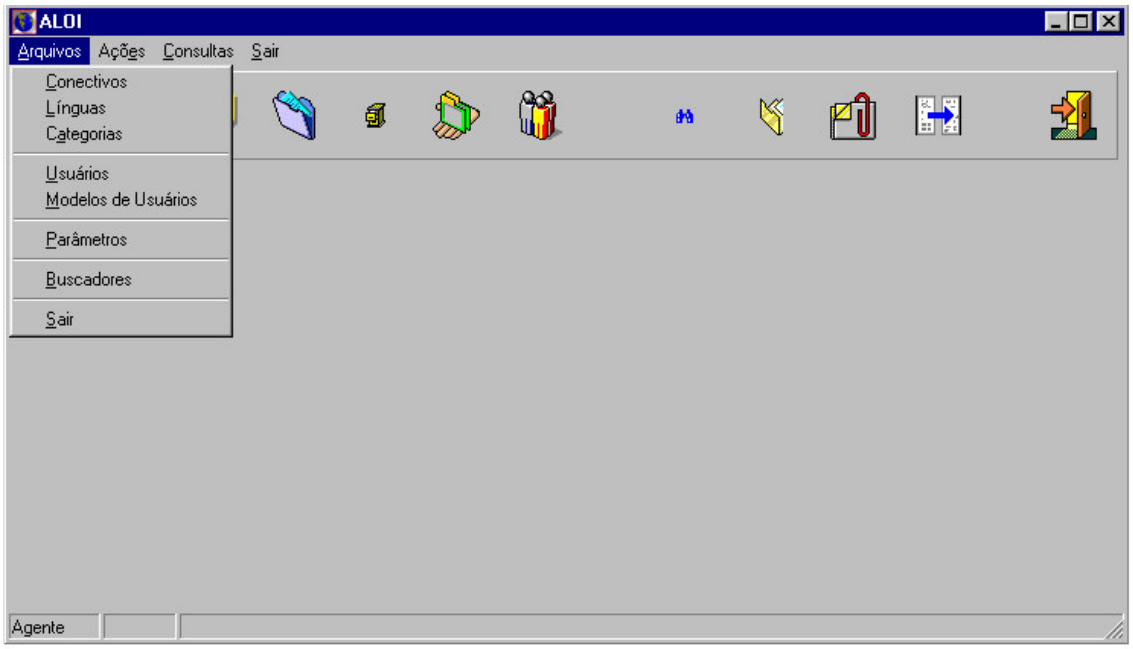


FIGURA 5.3. Interface Principal da Aplicação.

Os programas da opção “Arquivo”, do menu principal, têm como objetivo a realização de manutenções nos cadastros e parâmetros necessários para o funcionamento do agente. Assim sendo, todos adotam um padrão de operação único que possibilita ao usuário a inclusão, alteração e exclusão de qualquer registro armazenado, bem como a navegação entre eles. A figura 5.4 apresenta a interface do programa responsável pela manutenção dos parâmetros, como um exemplo do padrão adotado.

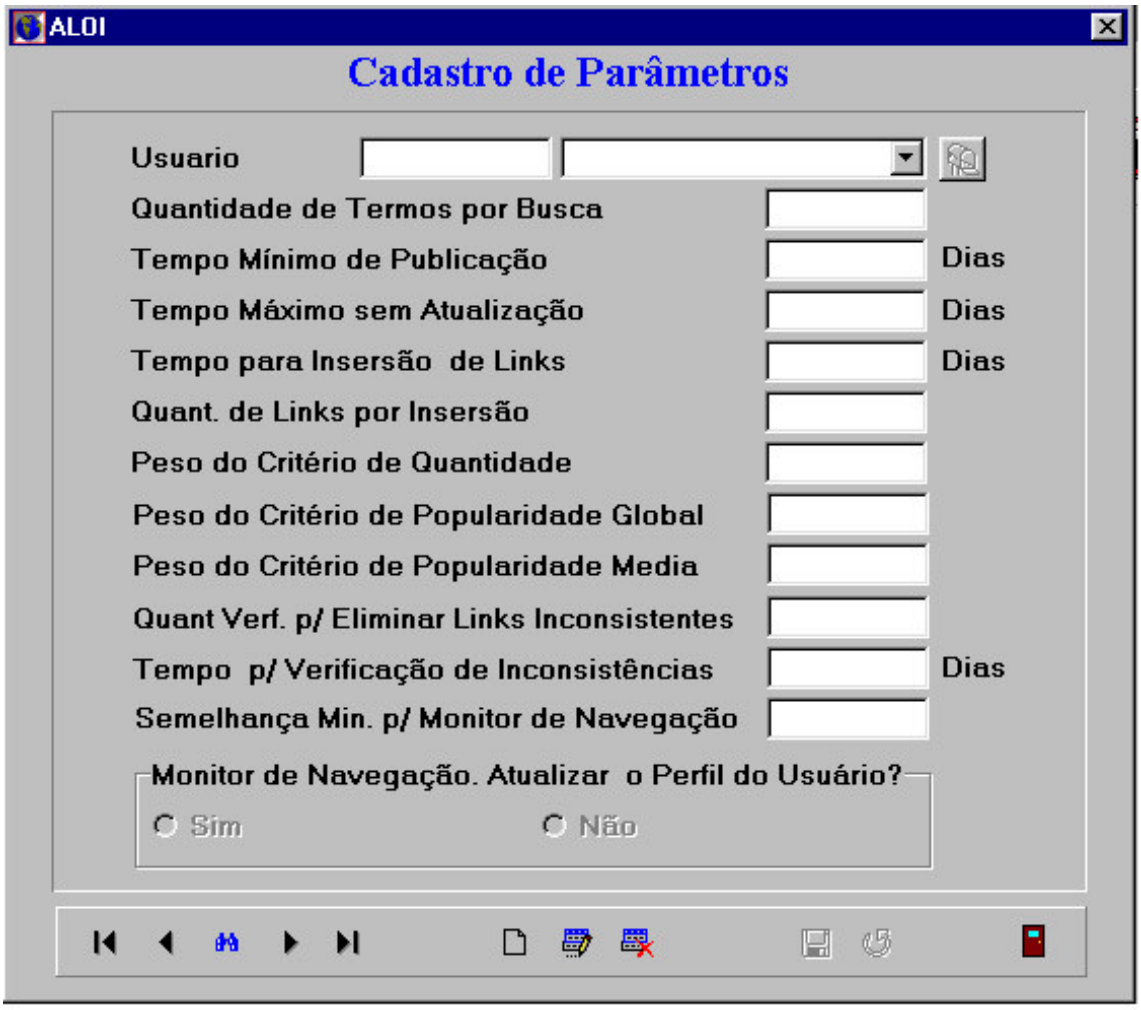


FIGURA 5.4. Exemplo de um Programa do Menu Arquivo

Sob a opção "Ações", do menu principal, estão as opções "Busca", "Verificação de Integridade", "Processo de Seleção", e "Gerar HTML". As três primeiras têm em comum a característica de execução em retaguarda, portanto, estas opções permitem que o usuário apenas visualize o andamento dos processos. A opção "Gerar HTML", que não foi especificada no modelo, somente foi implementada com o objetivo de gerar arquivos HTML a partir do repositório de links e sua estrutura, para possibilitar uma melhor visualização.

A opção "Busca" exibe o processo corrente de buscas dos links a partir dos termos armazenados no modelo do usuário. O formulário de visualização deste módulo exibe: os termos que estão sendo utilizados como argumento de busca, os sites de busca aos quais estão sendo submetidas as consultas, a URL submetida e uma lista dos links extraídos a cada submissão de consulta. A figura 5.5 apresenta a interface do módulo "Busca".

FIGURA 5.5 - Interface do módulo "Busca"

FIGURA 5.6 - Interface do módulo "Processo de Seleção"

A figura 5.6 apresenta a interface do módulo responsável pela seleção dos links que serão inseridos no repositório definitivo. Assim com no módulo "Busca", este formulário tem como função exclusiva permitir a visualização do estado do processo.

O processo de seleção é executado em intervalos de tempo, conforme configurado pelo usuário, portanto, sua interface apresenta dados referente à situação atual do módulo, tais como: data e hora da próxima execução, quando em estado de espera, e, URL analisada, valores obtidos para os critérios de qualidade, popularidade global e popularidade média. Além disso, é apresentada a situação do *ranking*, atualizada a cada verificação.

A opção "Verificação de Integridade", do menu principal, viabiliza a visualização do processo de teste de integridade dos links armazenados no repositório. Assim como o processo de seleção, a verificação de integridade é executada em intervalos de tempo estipulados pelo usuário, assim sendo, a interface apresentada na figura 5.7 mostra a data e hora da próxima execução ou o seu estado, cada URL analisada e seu respectivo código de retorno , bem como a lista dos links inconsistentes, que sofrem novas verificações.

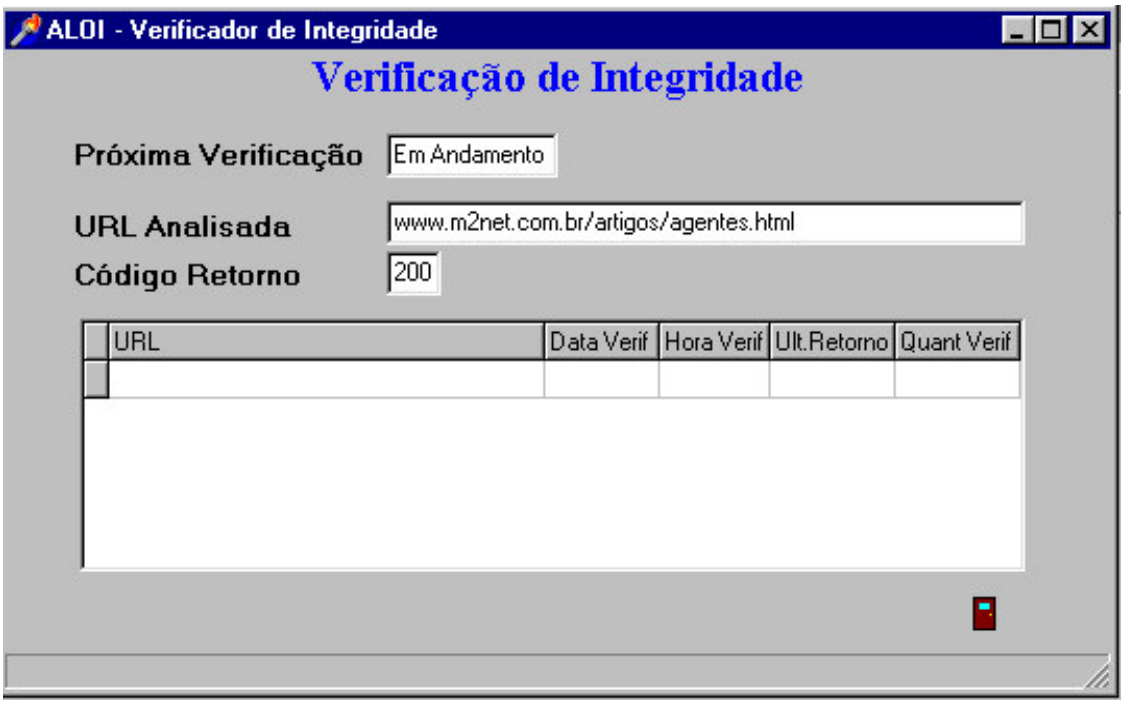


FIGURA 5.7. Interface do módulo "Verificador de Integridade"

A interface implementada para a opção "Gera HTML" é bastante simples, sendo formada por um formulário no qual o usuário pode selecionar categoria, assunto, nome do arquivo destino e caminho onde ele será gravado.

A opção "Consultas" do menu principal tem como objetivo viabilizar consultas aos diversos cadastros e ao repositório de links. A estruturação da interface e a forma de utilização dos formulários de consulta são padronizadas para que sejam amigáveis ao usuário.

As consultas podem ser realizadas baseadas em diferentes ordenações e, ainda, utilizando argumentos de busca. A figura 5.8 ilustra um exemplo de interface de consulta. O exemplo refere-se a consulta de Categorias e Assuntos.

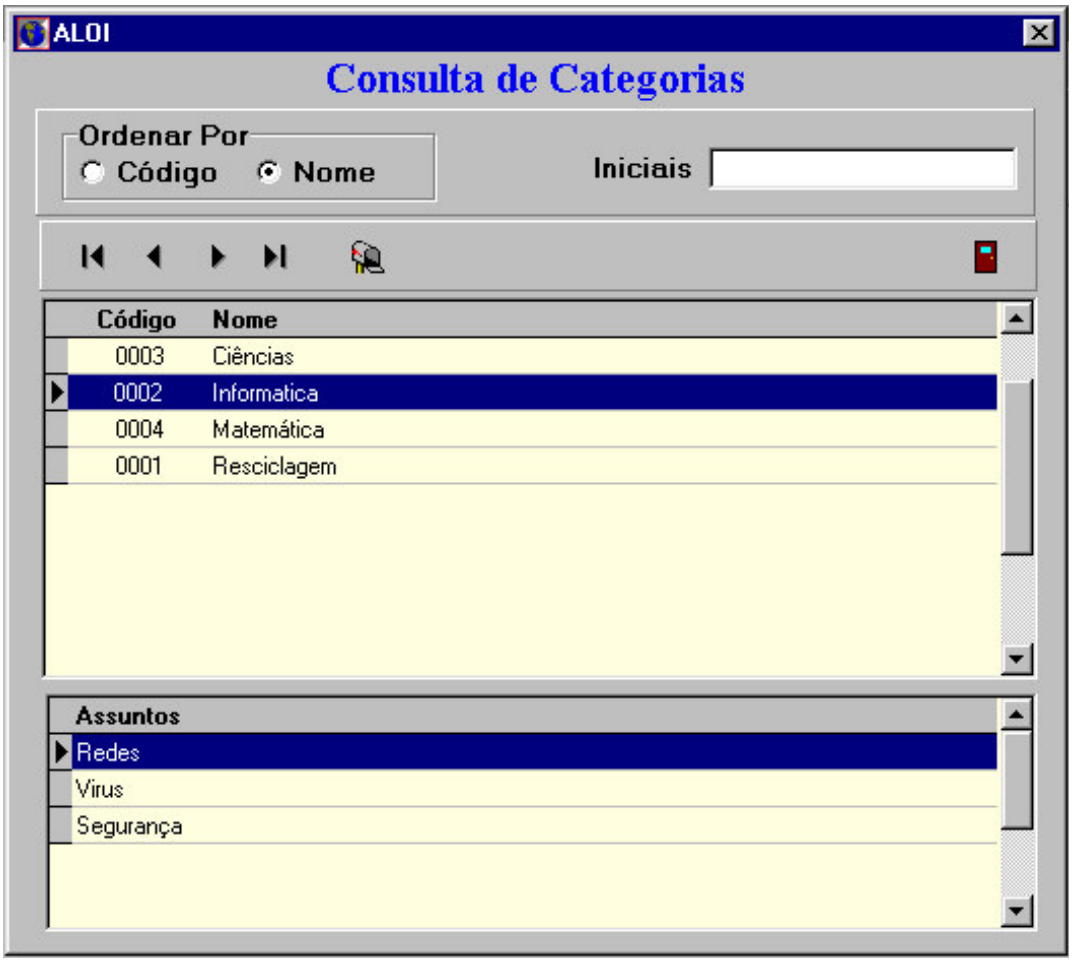


FIGURA 5.8 - Exemplo da Interface das Consultas

5.5 Avaliação do Protótipo

Com o objetivo de avaliar o protótipo implementado e validar o modelo proposto, foram inseridos alguns modelos de documentos de interesse de um usuário, foram especificadas configurações para este usuário e o agente foi submetido a um determinado tempo de execução em caráter experimental.

5.5.1 Ambiente Utilizado nos Testes

Para a realização dos testes o protótipo foi instalado sobre o seguinte ambiente de hardware, software e comunicação:

Hardware: Microcomputador com processador K7 850 Mhz com 128 Mb RAM e HD 20 Gb.

Sistema Operacional: MS Windows 98

Banco de Dados: Paradox 7

Comunicação: Acesso à Internet via canal dedicado Vant 256 Kbps.

O ambiente utilizado foi selecionado em função da disponibilidade de meios e recursos no período em que se realizaram os testes.

5.5.2 Dados Considerados nos Testes

O comportamento do protótipo é significativamente alterado em função da configuração definida por cada usuário. Para a realização dos testes preliminares, aqui descritos, utilizamos os seguintes valores para cada parâmetro:

- **Termo_Busca:** Foram utilizados três termos para busca submetida aos sites cadastrados;
- **Tmp_Min_Pub:** O tempo mínimo de publicação, para seleção, considerado foi de 30 dias;
- **Tmp_Max_S_Atu:** O tempo máximo sem atualização para seleção considerado foi de 365 dias;
- **Int_Ins_Link:** O intervalo de tempo para inserção de links no repositório foi configurado para 3 dias;
- **Qtd_Links_Ins:** A quantidade de links estipulada para inserção, no intervalo, é 10 links, ou seja, 10 links a cada 3 dias;
- **Peso_Qualidade:** O peso do critério de qualidade durante o processo de seleção foi fixado em 6;
- **Peso_Pop_Global:** O peso do critério de Popularidade Global durante o processo de seleção foi fixado em 4;
- **Peso_Pop_Media:** O peso do critério de Popularidade Média durante o processo de seleção foi fixado em 5;
- **Int_Ver_Integr:** O intervalo de tempo para a execução das verificações de integridade foi configurado em 2 dias;
- **Qtd_Ver_Integr:** A quantidade de verificações realizadas para eliminar um link inconsistente foi configurado em 5;
- **Int_Tent_Integr:** O intervalo de tempo entre cada verificação de consistência na tabela AVALIACAO, foi configurado em 4 horas;
- **Nivel_Sem_Mn:** O nível de semelhança entre um documento e o perfil de interesse do usuário, para que o documento seja inserido como modelo pelo Monitor de Navegação, foi considerado 0,5;
- **Atual_Perfil:** Como neste protótipo não estamos considerando a atualização do perfil automaticamente, definimos este ítem com "Não".

Os sites de busca cadastrados nos parâmetros de buscadores e utilizados para a avaliação do protótipo foram os seguintes:

- Lycos (www.lycos.com)
- Google (www.google.com)
- Yahoo (www.yahoo.com)

Os dados do modelo de interesse do usuário foram adicionados manualmente, uma vez que a implementação realizada, por ser parcial, não contempla a sua obtenção de forma automática.

O interesse do usuário, nos testes, foi dividido em três categorias distintas: Informática, Administração e Fruticultura. Cada categoria foi dividida em três assuntos. A tabela 5.3 apresenta a estrutura de categorias e assuntos usada.

TABELA 5.3 – Estrutura de categorias e assuntos usados nos testes.

CATEGORIA	ASSUNTO
Informática	Inteligência Artificial
	Banco de Dados
	Sistemas Operacionais
Administração	Marketing
	Recursos Humanos
	Financeiro
Fruticultura	Pomares
	Classificação
	Mudas

A realização dos testes se deu num período de seis dias, totalizando 144 horas de execução ininterruptas. O protótipo ficou em execução durante este período em situações diversas, ou seja, em alguns momentos o computador foi utilizado normalmente para trabalho e em outros momentos permaneceu somente como o agente em execução.

5.5.3 Resultados Preliminares

Apesar de limitados, os testes aplicados sobre o protótipo puderam apresentar alguns resultados sobre a validade do modelo proposto. Os principais critérios observados durante e após a execução foram os seguintes:

- A quantidade total de links encontrados;
- A quantidade de documentos selecionados para os repositório;
- A relevância dos documentos selecionados.

A tabela 5.4 apresenta a quantidade de links encontrados para cada categoria e assunto, no período em que o protótipo permaneceu em avaliação.

TABELA 5.4 – Quantidade de links encontrados no período.

CATEGORIA	ASSUNTO	QUANTIDADE
Informática	Inteligencia Artificial	35210
	Banco de Dados	85321
	Sistemas Operacionais	48429
	Total Categoria	**168960**
Administração	Marketing	15842
	Recursos Humanos	9521
	Financeiro	22354
	Total Categoria	**47717**
Fruticultura	Pomares	5842
	Classificação	8341
	Mudas	15421
	Total Categoria	**29604**
	Total Geral	**246281**

A tabela 5.5 apresenta a quantidade de links selecionados e inseridos no repositório para cada assunto e categoria, durante a avaliação. Além disso, é apresentada uma coluna com a quantidade e o percentual de links considerados realmente relevantes. O número de links relevantes foi obtido através de uma análise visual do conteúdo de cada documento.

TABELA 5.5 – Quantidade de links inseridos no repositório e relevantes.

CATEGORIA	ASSUNTO	SELECIONADOS	RELEVANTES
Informática	Inteligência Artificial	20	16 = 80%
	Banco de Dados	20	12 = 60%
	Sistemas Operacionais	20	14 = 70%
	Total Categoria	**60**	**42 = 70%**
Administração	Marketing	20	8 = 40 %
	Recursos Humanos	20	11 = 55 %
	Financeiro	20	13 = 65 %
	Total Categoria	**60**	**32 = 53 %**
Fruticultura	Pomares	20	18 = 90 %
	Classificação	20	7 = 35 %
	Mudas	20	9 = 45 %
	Total Categoria	**60**	**34 = 57 %**
	Total Geral	**180**	**108 = 60 %**

Como é possível observar, através da tabela 5.5, o índice médio de aproveitamento dos links selecionados ficou em torno de 60 %, este índice pode ser melhorado com o aumento do intervalo de tempo entre as seleções, o aumento da quantidade de termos em cada modelo de documento e, principalmente, a utilização de técnicas que combinem a presença de termos juntos e não somente de forma isolada.

Outro item que foi observado foi a impossibilidade da análise de documentos do tipo PDF, PS e outros. Mas esta não é uma deficiência do modelo e sim do protótipo implementado.

5.6 Considerações Finais

Neste capítulo, foi descrita a implementação de um protótipo de agente para a obtenção, organização e manutenção de links em repositórios, a partir de um perfil de interesse de usuário, com base na especificação do modelo proposto no capítulo anterior.

O protótipo implementado não cobre toda a especificação do modelo, pois foi proposto um modelo geral. Foi dado um enfoque especial aos módulos responsáveis pela busca de links, seleção, classificação quanto a assunto e verificação de integridade. Os dados referentes ao perfil de interesse do usuário, utilizados para a realização de testes, foram informados manualmente através do módulo "Modelos de Usuários". O modelo prevê um mecanismo para a obtenção automática do interesse do usuário, porém neste protótipo não foi contemplada.

6 Conclusões

O visível crescimento no volume de informações disponibilizadas na Internet, proporcionada pelos grandes avanços tecnológicos no setor de telecomunicações nos últimos tempos, vem fomentando as pesquisas e o desenvolvimento de inúmeros agentes inteligentes destinados ao auxílio no consumo e publicação de informações na Web. Neste trabalho foram estudadas as principais categorias de agentes para Internet e alguns exemplos de aplicações existentes, tais como InfoFinder, Letizia, Search Advisor e WebWatcher que têm como característica comum a função de obter informações que atendam aos interesses de seus usuários.

Atualmente, boa parte dos agentes que atuam na Internet levam em consideração o perfil de interesse dos usuários, tanto de forma individual quanto coletiva. Tendo em vista esta tendência, foram realizados alguns estudos sobre modelos de usuários, suas características, formas de obtenção e utilização.

A partir dos estudos e comparações realizadas entre os modelos já existentes de agentes, apresentou-se a proposta de um modelo de agente para a localização e organização de informações que atendam ao perfil de interesse dos usuários, bem como possibilite a constante atualização deste perfil. O modelo proposto têm como base a busca das informações na Internet, utilizando-se de sites de busca tradicionais, e visa o armazenamento e a manutenção dos links para as informações obtidas em um repositório organizado quanto a usuário, categoria de informação e assunto.

No modelo apresentado procura-se especificar todos os módulos que formam o agente, desde a obtenção do perfil do usuário até a constante verificação de integridade dos links armazenados no repositório, passando pela localização e, principalmente, seleção e classificação das informações obtidas.

Os estudos realizados sobre agentes direcionados para Internet, modelos de usuário e algumas aplicações já existentes forneceram subsídios para a proposição de um novo modelo que procura explorar as principais virtudes de cada aplicação e conceito estudado, combinando-os em uma proposta ampla e abrangente, com vistas a suprir as necessidades de localização e organização de links descobertas pelos demais modelos.

Uma das principais contribuições deste trabalho é a apresentação de um modelo genérico, amplo e flexível. Estas características permitem a substituição de qualquer um dos métodos utilizados por outros que possam ser mais adequados a cada aplicação ou necessidade específica. Outra possibilidade é a integração do modelo com outros já existentes, de forma que módulos possam ser utilizados em cooperação. Um exemplo seria o aproveitamento de um perfil de interesse de outra aplicação como base de conhecimento para a realização das buscas.

A possibilidade de personalização do modelo, por parte do usuário, é outro ponto que deve ser destacado, pois cada usuário pode manter o seu próprio conjunto de parâmetros e perfil de interesse, sem a ocorrência de conflitos e misturas de

assuntos. Assim sendo, várias pessoas podem utilizar o mesmo computador e sistema, cada uma definindo suas prioridades, critérios, intervalos de execução, e outras características pessoais.

Um protótipo parcial foi implementado com o objetivo de validar os principais módulos do modelo. Foram implementados os módulos responsáveis pela busca de links, seleção, classificação quanto a assunto e verificação de integridade. A execução do protótipo por um determinado tempo demonstrou um aproveitamento de aproximadamente 60% dos documentos obtidos. Tal resultado foi obtido a partir de uma configuração e ambiente de execução específicos. Provavelmente a alteração dos parâmetros e características de execução modificará os resultados.

Durante a avaliação do protótipo foram observadas algumas deficiências do modelo, dentre elas destaca-se o tratamento dos termos de forma individual, ou seja, muitos termos que varias vezes ocorrem conjuntamente a outros podem afetar significativamente o resultado das buscas. Portanto é fundamental a inserção de um mecanismo que possibilite o tratamento de termos de forma conjunta.

Outro ponto observado foi o tratamento individual das páginas, ou seja, cada página localizada é tratada de forma individual, sem ligação nenhuma com o *site* no qual está inserida. Seria interessante adicionar ao modelo um mecanismo que possibilitasse o tratamento de *sites* completos.

O protótipo implementado, possibilitou uma avaliação prática das virtudes e deficiências da proposta. A realização dos testes sobre este protótipo mostrou, preliminarmente, que o modelo atende aos objetivos propostos, porém são necessárias algumas evoluções, principalmente no processo de seleção, como por exemplo a avaliação de termos que normalmente ocorrem juntos, para que seja possível a obtenção de um índice de acerto maior.

6.1 Trabalhos Futuros

Na realização dos testes sobre o protótipo ficaram evidentes algumas deficiências do modelo, tais como a carência de uma regra, ou um conjunto de regras, que considerem a ocorrência de múltiplos termos conjuntamente, tais regras dariam ao modelo maior eficácia durante os processos de busca e seleção.

Um tópico a ser estudado futuramente, seria então, o conjunto de regras disponíveis e a possibilidade da proposição de novas regras para a análise de termos que ocorrem conjuntamente.

O modelo aqui apresentado propõe um agente que opera no modo cliente, limitando-se ao escopo de cada usuário. Outro ponto a ser explorado futuramente é a criação de um modelo que possa operar em um servidor atendendo grupos de usuários que tenham interesses comuns.

Outra deficiência observada foi o tratamento de cada página como um documento isolado, ou seja, não é considerado o *site* e sim cada página. Pretende-se então, propor um mecanismo que vincule os documentos extraídos ao seu *site*.

Bibliografia

[ABR 2000] ABREU, M.F. de. **Um Estudo Sobre Agentes Inteligentes na Web**. 2000. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

[ASN 97] ASNICAR, F. A.; TASSO, C. ifWeb: a Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. In: INTERNATIONAL CONFERENCE ON USER MODELING, 6.,1997. **Adaptative Systems and User Modeling on the World Wide Web.** Disponível em: <http://fit.gmd.de/UM97/Tasso/Tasso.html>. Acesso em: 20 jun.2002.

[AVG 97] AVGOUSTOS, A. T.; KONSTANDINOS, G. M. **Search Advisor:** Training Internet uses towards search sessions. Thessaloniki: Department of Informatics, University of Macedonia, 1997.

[BAU 2002] BAUER, T.; LEAKE, D. B. **WordSieve:** a method for real-time context extration. Bloomington: Computer Science Department, Indiana University, 2002.

[BIL 96] BILLSUS, D.; PAZZANI, M. **Revising User Profiles:** the search for interesting web sites**.** Irvine: Department of Information and Computer Science, University of California, 1996.Disponível em: <http://www.ics.edu/mlearn/MLPapers.html>. Acesso em: 12 maio 2002.

[BIL 98] BILLSUS, D.; PAZZANI, M. **A Personal News Agent that Talks, Learns and Explains.** Irvine: Department of Information and Computer Science, University of California, 1998. Disponível em: <http://www.ics.edu/~mlearn/MLPapers.html>. Acesso em: 12 maio 2002.

[BOL 98] BOLLACKER, K. D.; LAWRENCE, S.; GILES, C. L. CiteSeer: an Autonous Web Agent for automatic retrieval and identification of interesting publications. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 2., 1998. **Proceedings...** New York: ACM Press, 1998. p. 116-123.

[BRE 98] BRENNER, W.; ZARNEKOW, R.; WITTIG, H. **Intelligent Software Agents.** Berlin: Springer-Verlag, 1998.

[CHE 98] CHEN, L.; SYCARA, K. WebMate: A Personal Agent for Browsing and Searching. In: INTERNATIONAL CONFERENCE ON ACM. AUTONOMOUS AGENTS, 2. , 1998. **Proceedings...** New York: ACM Press, 1998.

[FRA 97] FRANKLIN, S.; GRAESSER, A. Is it an Agent, or a Program? A taxonomy for autonomous agent. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, ATAL, 3., 1997. **Intelligent Agents III**: agent, theories, architectures, and languages: proceedings. Berlin: Springer-Verlag, 1998.

[GOO 2002] GOOGLE TECHNOLOGY. **Google searches more sites more quickly, delivering the most relevant results:** PageRank Explained. USA 2002. Disponível em: <http://www.google.com/technology/>. Acesso em: 10 abr. 2002.

[JOA 98] JOACHIMS, T.; FREITAG, D.; MITCHELL, T. **WebWatcher:** a tour guide for the World Wide Web. Pittsburgh: School of Computer Science, Carnegie Mellon University, 1998.

[KRU 97] KRULWICH, B.; BURKEY, C. The InfoFinder Agent: learning user interest through heuristic phrase extration. **IEEE Expert,** New York, v.12, n.6, p. 22-27, Sept./Oct. 1997.

[LAW 99] LAWRENCE, S.; GILES, C.L.; BOLLACKER, K. **Digital Libraries and Autonomous Citation Indexing**. Princeton: NEC Research Institute, IEEE, 1999.

[LIE 95] LIEBERMAN, H. **Autonomous Interface Agents.** Cambridge: Media Laboratory, Massachusetts Institute of Technology, 1995.

[LIE 2001] LIEBERMAN, H.; FRY, C.; WEITZMAN, L. Exploring the Web with Reconnaissance Agents. **Communications of the ACM**, New York, p. 69-75, 2001.

[MIN 96] MINIO, M.; TASSO, C. User Modeling for Information Filtering on INTERNET Services: exploiting an extended vertion of the UMT shell. In: INTERNATIONAL CONFERENCE ON USER MODELING, 15., 1996. **Proceedings...** Hawaii, 1996. Disponível em: <http://www.cs.ju.oz.au/bob/um96-workshop.html>. Acesso em: 05 dez.2001.

[MLA 99] MLADENIC, D. Machine Learning Used by Personal WebWatcher. In: WORKSHOP ON MACHINE LEARNING AND INTELLIGENT AGENTS, ACAI, 1999, Chania. **Proceedings...** [S.l.: s.n.], 1999.

[PAZ 96] PAZZANI, M.; MURAMATSU, J.; BILLSUS, D. **Syskill & Webert: Identifying web sites**. Irvine: Department of Information and Computer Science. University of California, 1996. Disponível em: <http://www.ics.edu/~mlearn/MLPapers.html>. Acesso em: 01 maio 2001.

[PAZ 99] PAZZANI, M.; BILLSUS, D. **Adaptative Web Site Agents**. Irvine: Department of Information and Computer Science, University of California, 1999. Disponível em: <http://www.ics.uci.edu/~mlearn/MLPapers.html>. Acesso em: 01 maio 2001.

[SAL 89] SALTON, G; BUCKLEY, C. Term-Weighting Approaches in Automatic Text Retrieval. **Information Processing and Management: an International Journal**, [S.I.], v.24, n.5, p.513-523, 1989.

[THE 98] THEILMANN, W.; ROTHERMEL, K. Domain Experts for Information Retrieval in the World Wide Web**.** In: INTERNATIONAL WORKSHOP ON COOPERATIVE INFOR MATION AGENTS, CIA, 2., 1998. **Cooperative Information Agents II**: learning, mobility and electronic commerce for information discovery on the internet: proceedings. Berlin: Springer-Verlag, 1998.