

Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Carlos Alberto Valente Andrade

**ARCABOUÇO PARA O DESENVOLVIMENTO DE
PORTAIS COLABORATIVOS**

São Paulo

2004

Carlos A. Valente Andrade

ARCABOUÇO PARA O DESENVOLVIMENTO DE
PORTAIS COLABORATIVOS

Carlos Alberto Valente Andrade

**ARCABOUÇO PARA O DESENVOLVIMENTO DE
PORTAIS COLABORATIVOS**

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, para obtenção do título de Mestre em Engenharia de Computação.

Área de concentração: Engenharia de Software

Orientador: Dr. Mário Yoshikazu Miyake

São Paulo

2004

Andrade, Carlos Alberto Valente

Arcabouço para o desenvolvimento de portais colaborativos. / Carlos Alberto Valente Andrade. São Paulo, 2004.
98p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Engenharia de Software.

Orientador: Prof. Dr. Mário Yoshikazu Miyake

1. Portal colaborativo 2. Agentes de informação 3. Ontologia 4. WIKI
5. PERL (Linguagem de programação) 6. Mecanismo de busca 7.
Groupware 8. Tese I. Instituto de Pesquisas Tecnológicas do Estado de
São Paulo. Centro de Aperfeiçoamento Tecnológico II. Título

CDU 004.822:004.41(043)
A553a

AGRADECIMENTOS

Agradeço, em primeiro lugar, ao Grande Arquiteto do Universo em possibilitar desafiar os meus limites físicos e mentais, mesmo não sendo um privilegiado em nenhum desses pontos. Aos meus pais, que mesmo com toda a simplicidade cultural que receberam ao longo de seus dias, conseguiram transmitir os princípios básicos morais para me considerar um cidadão útil à sociedade.

Meus agradecimentos especiais ao meu orientador, pela paciência e incentivo a participação nesta jornada de conhecimentos, compartilhando suas idéias e reflexões e possibilitando um melhor aperfeiçoamento técnico.

Preciso também agradecer aqueles que, direta e indiretamente, me estimularam na concretização deste trabalho, tanto personagens mais íntimos como o meu próprio irmão, como àqueles mais distantes na figura de alunos, técnicos, colegas que me foram muito em conta. Lembro-me de quantas vezes a minha Coordenadora Micky da Universidade Anhembi-Morumbi me perguntou: “Já terminou o Mestrado?”.

E finalmente não posso esquecer de agradecer a minha filha e esposa por estarem sempre comigo, me cobrando, me “empurrando” para nunca esmorecer nesta minha grande meta.

Para vocês todos, o meu mais profundo e sincero obrigado!

RESUMO

Este trabalho apresenta proposição de um arcabouço para desenvolvimento de portais colaborativos, constituído de elementos como agentes de informação, ontologias, ambiente Wiki, CGI com a linguagem PERL, mecanismos de busca e infraestrutura de *Groupware*, aplicado a qualquer área do conhecimento humano. Para mostrar a aplicabilidade desse arcabouço foi desenvolvido uma proposta de Portal Colaborativo especificamente voltado para a área de Engenharia de Software.

Palavras-chave: portais colaborativos, ontologias, agentes de informação, wiki, groupware, perl, mecanismos de busca.

ABSTRACT

In this essay is proposed a framework for the creation, implementation and maintenance of a collaborative portal including information agents, ontologies, Wiki, CGI, PERL programming language, search engines, and Groupware infrastructure. The proposed framework may be applied in any knowledge area. The feasibility of the proposed framework was tested with an example in the domain of Software Engineering.

Key-words: collaborative portal, ontology, information agents, wiki, groupware, perl, search engines.

Lista de ilustrações

Figura 1	Pirâmide do Conhecimento	5
Figura 2	Os três conceitos básicos do <i>Groupware</i>	12
Figura 3	Divisão em camadas de um Portal Colaborativo	13
Figura 4	Estrutura Básica do Portal Colaborativo	17
Figura 5	Áreas principais das páginas do Portal	18
Figura 6	Mascote adotado pela linguagem PERL	19
Figura 7	Funcionamento esquematizado do CGI	24
Figura 8	Arquitetura geral de um Mecanismo de Busca	29
Figura 9	Combinação das características de autonomia, cooperação e aprendizagem	39
Figura 10	Sistemas Multi-agentes cognitivos para a recuperação, classificação e extração integradas de informação da Web	47
Figura 11	Estrutura básica dos agentes	72
Figura 12	Detalhes básicos do agente	73
Figura 13	Visão detalhada dos principais programas usados no Portal Colaborativo	75
Figura 14	Comparativo com uma aplicação CGI típica	75
Gráfico 1	Bilhões de documentos textuais indexados	37

Lista de abreviaturas

ACM	<i>Association of Computer Machinery</i>
CCS	<i>Computing Classification System</i>
CGI	<i>Common Gateway Interface</i> - padrão de comunicação entre servidores HTTP, bancos de dados e outras fontes de informação
ES	Engenharia de Software
HTML	<i>HyperText Markup Language</i> -O principal formato das páginas na internet
HTTP	<i>Hypertexto Transfer Protocol</i> - Protocolo de Transferência de Hipertexto. Protocolo utilizado para a transferência de páginas HTML, ou seja, utilizado na WWW
IA	Inteligência Artificial
IEEE	<i>The Institute of Electrical and Electronics Engineers</i>
IME	Instituto de Matemática e Estatística
IP	<i>Internet Protocol</i> , ou, Protocolo Internet - é parte do protocolo TCP/IP para transmissão de pacotes pela internet
IRC	<i>Internet Relay Chat</i> – software de comunicação (bate-papo)
KA	<i>Knowledge Areas</i> – Áreas de Conhecimento
KM	<i>Knowledge Management</i> – Gestão do Conhecimento
MB	Mecanismos de Busca
RI	Recuperação de Informação
SWEBOK	<i>Software Engineering Body of Knowledge</i>
TI	Tecnologia de Informação
URL	<i>Uniform Resource Locator</i> - localizador uniforme de recursos
WEB	<i>World Wide Web</i> (WWW), rede de computadores mundial

Sumário

Resumo	
Abstract	
Lista de ilustrações	
Lista de abreviaturas	
Capítulo 1	
1 INTRODUÇÃO	1
1.1 Motivação	2
1.2 Metodologia de Pesquisa	3
1.3 Organização do Trabalho	4
Capítulo 2	
2 PROPOSTA DO TRABALHO	5
2.1 Caracterização do Portal Colaborativo	7
2.1.1 Portais Horizontais e Verticais	8
2.1.2 Portal do Conhecimento	9
2.1.3 Portal Colaborativo	10
2.2 Ferramentas Computacionais do Portal Colaborativo	13
2.3 Modelo da Proposta	16
Capítulo 3	
3 INFRAESTRUTURA INTERNA E EXTERNA	19
3.1 Infraestrutura Interna	19
3.1.1 Linguagem PERL	19
3.1.1.1 Expressões Regulares	21
3.1.1.2 CGI com PERL	23
3.1.2 Mecanismos de Busca	25
3.1.2.1 Dissecando os Mecanismos de Busca	26
3.1.2.2 Técnicas de Recuperação	31
3.1.2.3 Tipologia dos Mecanismos de Busca	33
3.1.2.3.1 Busca Indireta – Diretórios	33
3.1.2.3.2 Busca Direta - Motores de Busca	35

3.1.2.3.3 Mecanismos de Busca Múltiplo – Metamotores	37
3.1.3 Agentes de Software	38
3.1.3.1 Arquitetura de Agentes	38
3.1.3.1.1 Agentes Intencionais, Reativos e Híbridos	40
3.1.3.1.2 Agentes de Interface	41
3.1.3.1.3 Agentes Colaborativos	41
3.1.3.1.4 Agentes Móveis	42
3.1.3.1.5 Agentes de Informação	42
3.1.4 Ontologias	43
3.1.5 Ambiente Wiki Wiki	47
3.1.5.1 Estrutura de um Wiki	50
3.2 Infra-estrutura Externa (Síncronas e Assíncronas)	51
3.2.1 Wiki como ferramenta externa de comunicação	53
3.2.2 Diferenças entre Blog e Wiki	55
Capítulo 4	
4 EXPERIMENTOS NO PORTAL COLABORATIVO	57
4.1 Experimentos com Wiki	59
4.2 O Ambiente Wiki: UseModWiki	61
4.3 A Linguagem PERL, Expressões Regulares e CGI	64
4.4 Construção do Agente para o Portal	65
4.5 Ontologia SWEBOK	67
4.5.1 Ontologia ACM	68
4.5.2 Outras Ontologias	70
4.6 Visão Geral	71
4.6.1 Detalhes do Funcionamento dos Agentes	72
4.6.2 Processos do Modelo	76
4.7 Decisões Técnicas Tomadas e Justificativas	78
Capítulo 5	
5 CONCLUSÕES FINAIS	80
5.1 Desdobramentos Futuros	81
Referências	83
Referências consultadas	87

Anexos	93
A. Hacking Google URLs	93
B. Relação da Ontologia de Engenharia de Software da ACM	95
C. Listagem dos Programas em PERL	98

ARCABOUÇO PARA O DESENVOLVIMENTO DE PORTAIS COLABORATIVOS

1 INTRODUÇÃO

“A internet é simplesmente uma biblioteca, um acesso instantâneo a tudo, seja isso bom ou ruim. A internet é o telefone, a internet é um círculo de amigos. A internet é um quadro de avisos, a internet é o rádio, a internet é a televisão, a internet são os livros, a internet são os diários, a internet é a discussão em um bar, a internet é um bazar de igreja. A internet é qualquer coisa. Ela não é mais do que um meio”.

LUNDBERG, George - editor-chefe da Medscape

Em um artigo bastante divulgado na internet intitulado *The Cathedral and the Bazaar* (A Catedral e o Bazar), (RAYMOND, 2000) discute, através de analogias, dois estilos fundamentais e diferentes de desenvolvimento de software: o modelo **catedral**, da maior parte do mundo comercial, e o modelo **bazar** do mundo do Linux.

Sinteticamente, chega-se à idéia de que o modelo de desenvolvimento de programas através da Web, utilizando a colaboração on-line, é extremamente poderoso, com estratégias distintas, e quebrando velhos paradigmas do desenvolvimento tradicional.

Análogo a esse estilo de programação, utilizando o potencial colaborativo da internet, pretende-se como objetivo, descobrir, identificar, e criar um modelo básico dos fundamentos técnicos para criação e desenvolvimento de um Portal Colaborativo, e para mostrar a sua aplicabilidade, focar na área do conhecimento de Engenharia de Software (ES).

Dessa forma, utiliza-se de todo ferramental, síncrono e assíncrono, de *Groupware* para permitir o compartilhamento de informações e apoio entre

os participantes da comunidade do Portal. E, complementando o modelo experimental de Portal Colaborativo, utiliza-se do recurso computacional denominado agente de informação, para captar de forma específica, na área de ES, material passível de pesquisa e geração de conhecimento no Portal.

Para tanto, utiliza-se, como ponto de partida, a ontologia do SWEBOK (Software Engineering Body of Knowledge) desenvolvido pelo IEEE (SOFTWARE ENGINEERING COORDINATING COMMITTEE, 2001). Tal documento foi deveras importante para este trabalho pois, para demonstrar a proposta, precisava-se de uma área de conhecimento que já possuísse uma ontologia previamente discutida e validada por especialistas. Além do mais, havia a necessidade de filtrar informações específicas para pesquisadores (professores, alunos, cientistas), testando efetivamente o modelo proposto.

Durante a pesquisa, fez-se contato com especialistas, do IME-USP, na ferramenta colaborativa Wiki, o que contribuiu para a utilização desse versátil ambiente como base fundamental no desenvolvimento do arcabouço do Portal Colaborativo.

1.1 Motivação

A construção de um Portal Colaborativo sempre foi um sonho pessoal, desde o início do Mestrado Profissional em Engenharia de Computação. A Gestão do Conhecimento é um tema extremamente apaixonante, e a possibilidade de identificar e descobrir quais seriam os elementos técnicos básicos, na criação de um Portal Colaborativo, mais motivante ainda.

Este trabalho permitiu “navegar” por mares revoltos, de densidades diferentes, dentro de uma emocionante viagem, por várias áreas do

conhecimento. Estudou-se Mecanismos de Busca, Agentes Inteligentes, Mapas Conceituais, Ontologias e outros temas que dificilmente teríamos a oportunidade de aprofundar num curso regular. Ora concentrando-se no tema, ora dele desviando, pela curiosidade natural que qualquer pesquisador tem, por um tema que lhe agrada. Portanto, foi uma oportunidade fantástica para explorar mais profundamente essas áreas de TI, e como elas estão inseridas em outras disciplinas.

No processo de organizar essas informações de forma coerente e com uma metodologia adequada, conseguiu-se testar proposta do Portal Colaborativo, na área de Engenharia de Software.

1.2 Metodologia de Pesquisa

Visando o gerenciamento das metas e qualidade desta dissertação, conforme Severino (2000), adotou-se a metodologia estruturalista da seguinte forma:

Pesquisa: engloba a busca, seleção e a leitura de artigos e publicações referentes aos assuntos relacionados, objetivando o levantamento do estado da arte na área em questão;

Desenvolvimento do Modelo Proposto: consiste na elaboração de um arcabouço de desenvolvimento de um Portal Colaborativo, com aplicação ao tema Engenharia de Software;

Aplicação do Arcabouço: representa a fase de construção do Portal Colaborativo conforme o modelo apresentado anteriormente.

1.3 Organização do Trabalho

Este trabalho foi estruturado em 5 partes, com cada parte num capítulo específico.

O Capítulo 1 contém uma apresentação geral do trabalho, a motivação para a sua elaboração e a metodologia.

O Capítulo 2 apresenta o desenvolvimento do trabalho propriamente dito. Nessa parte enfoca-se, conceitualmente, o Portal Colaborativo e diferenças com relação a outros tipos de portais. Em seguida, detalham-se as ferramentas computacionais necessárias para a construção de um Portal Colaborativo. Finalizando, é introduzida a proposta de arcabouço para Portal Colaborativo.

No Capítulo 3, o foco é a Infra-estrutura, interna e externa, necessária para comportar tecnicamente um Portal Colaborativo. Nessa parte, aprofundam-se os principais recursos técnicos que contribuem para a criação do Portal Colaborativo.

O Capítulo 4, foca os experimentos realizados durante a construção do Portal Colaborativo em Engenharia de Software, e como foram consolidadas as ferramentas computacionais mencionadas no Capítulo 3, formando um único bloco lógico. Finalmente, no Capítulo 5 estão a conclusão do trabalho e desdobramentos futuros.

2. PROPOSTA DO TRABALHO

Canhos (sd), menciona em seu trabalho de biodiversidade: “Sabe-se que oceanos de **dados** geram rios de **informação**, que geram córregos de **conhecimento** e que, por sua vez, geram gotas de **sabedoria**.” (grifos do autor). Pode-se visualizar todos esses elementos, graças a Tuthil (1990) numa forma hierárquica, na Pirâmide do Conhecimento (Figura 01).

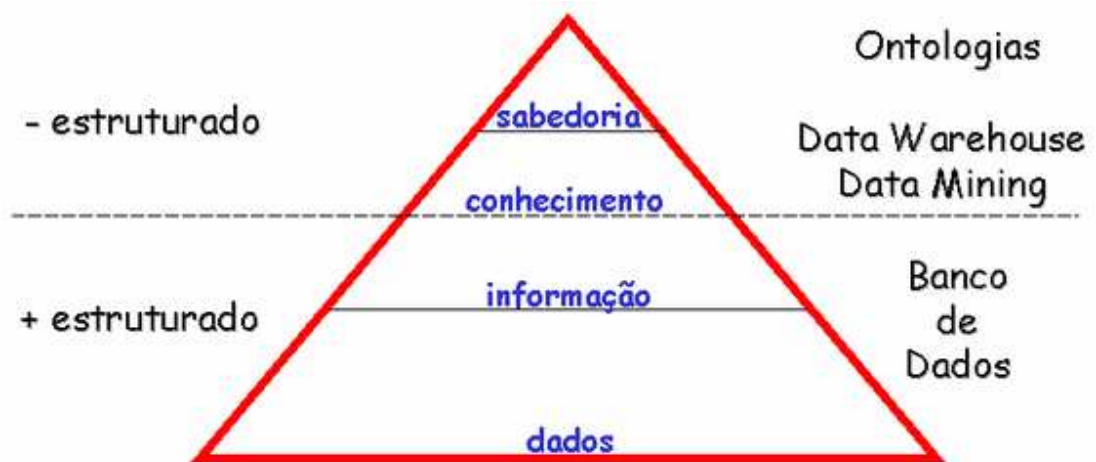


Figura 1: Pirâmide do Conhecimento (adaptado de TUTHIL, 1990)

Não raro, confundem-se dados com informação, informação com conhecimento, conhecimento com sabedoria. Pode-se lembrar das palavras do poeta e filósofo americano Eliot (1934): “Que é feito da sabedoria que perdemos no conhecimento? E do conhecimento que perdemos na informação?” (*Where is the life we have lost in living? Where is the wisdom we have lost in knowledge? Where is the knowledge we have lost in information?*), somente para referenciar a complexidade destes conceitos.

Conforme Navega (2002):

Conhecimento é a gerência de cabeças de pessoas, pois somente essas cabeças é que possuem conhecimento, pelo menos com as limitações tecnológicas dos computadores atuais. Computadores, arquivos, relatórios, livros, documentos, tudo isso são apenas repositórios de informação. O conhecimento só aparece no instante em que

essas informações estão no interior de um agente capaz de utilizá-las para alguma atitude, ou seja, para alguma ação.

A decorrência dessa afirmação é proveniente do encadeamento de conceitos fundamentais que Navega, autor citado anteriormente, relaciona:

- Informação: É toda coleção de descrições simbólicas de mudanças de estado de um sistema qualquer. O conteúdo informacional de uma mensagem qualquer é dado pela avaliação da probabilidade de ocorrência dos símbolos que compõe a mensagem, nos termos definidos por Shannon (1948).
- Agente: É um sistema (organismo ou máquina) que pode trocar informação com o meio ambiente que o cerca e que tem estados internos que se alteram com o tempo. Algumas dessas alterações internas de estado devem ser função da troca de informações que o agente executa com o ambiente que o circunda. Exemplos típicos seriam os camundongos, gatos, elefantes, seres humanos, e, surpreendentemente, CD Players, computadores, *web browsers*, etc.
- Conhecimento: É uma coleção de informações situadas no interior de um agente e que o habilita a atuar no meio ambiente com eficácia maior do que se esse agente não dispusesse dessa informação. Podemos dizer que um camundongo pode ter "conhecimento" sobre ratoeiras na medida em que ele disponha de uma série de informações experiênciais que o permita evitar a armadilha.
- Inteligência: Pura e simplesmente é a habilidade (ou uma medida da habilidade) de um agente gerar (criar) conhecimento. Se não há elaboração de conhecimento novo, não se está falando de inteligência.

Portanto, para conceituar-se um Portal Colaborativo, ou mesmo de Conhecimento, deve-se afirmar (adaptando as palavras de Navega) que os portais clássicos se transformam em conhecimento, no instante em que um ser humano incorpore essas informações em sua cognição, e passe a usá-las em alguma atividade específica.

Para tanto, existe a necessidade de implementar um portal clássico com ferramentas computacionais adequadas, permitindo a interação e a cognição entre os pesquisadores, e possibilitando efetivamente a geração do conhecimento numa atividade específica. Em outras palavras, aumentar o nível de colaboração e confiança entre estes personagens.

Conforme os trabalhos dos economistas Landes (1998) e Fukuyama (1996), chega-se à conclusão de que o conhecimento e a riqueza surgem inúmeras vezes em sociedades com altos níveis de colaboração e confiança.

E, somente para citar a visão de Lévy (1999), que se intitula a si mesmo como “engenheiro do conhecimento”, afirma que o conhecimento tem uma conotação ampliada: abrange muito mais do que informação, do que aprendizagens conscientes, racionais e individuais. Esta postura se relaciona com uma atitude científica que começa a tomar fôlego nos dias atuais: a da cognição como correspondendo à vida, como invenção do mundo e de si mesmo. Neste sentido, a cognição somente é passível na interação.

2.1 Caracterização do Portal Colaborativo

Inicialmente, tem-se como objetivo a conceituação genérica de um portal e suas clássicas divisões em Horizontais e Verticais, para, em seguida, caracterizar-se um Portal Colaborativo.

Trata-se o Portal de um *site* na internet, ou área dentro de um *site*, que atrai usuários pelo seu conteúdo (SCHEER, 2002). O objetivo, ao construir e disponibilizar um portal clássico, é permitir que o *site*, baseado nas informações que armazena, atraia o maior tráfego possível.

Em função do tráfego que converge, canaliza receitas de publicidade, gera cobertura de imprensa, tornando-o o ponto central de uma indústria ou de um grupo de interesse, atraindo usuários e clientes futuros para o *site*.

Entre os portais mais usados, os mecanismos de busca são os mais destacados. Qualquer internauta praticamente os utiliza como ponto de partida em suas pesquisas e navegações. Evoluíram eles, de tal forma, que, hoje, encontram-se portais específicos para certas áreas do conhecimento humano, ou genéricos, ricos de conteúdo, abrangendo desde informações corriqueiras, dados jornalísticos, até comércio de produtos e serviços.

2.1.1 Portais Horizontais e Verticais

Pode-se definir, de forma compacta, que Portais Horizontais são agregadores de conteúdo e audiência, com diferentes canais temáticos, destinados aos diversos tipos de internautas. Em oposição, os Verticais (ou Vortais) são centrados em um tema específico, com conteúdos aprofundados.

Do ponto de vista comercial, pode-se considerar que os Portais Horizontais atenderiam às demandas genéricas de largas audiências - UOL, por exemplo - enquanto os Verticais focariam numa indústria específica, ou numa única Cadeia de Valor (SAAB, 2000). Os portais podem ser, inclusive, desenvolvidos em diversos idiomas, para facilitar o acesso em outros países.

Detalhando os Portais Verticais ou Vortais, pode-se dizer que têm como objetivo principal o fato de serem o ponto de partida para pessoas interessadas num assunto específico, produto, indústria ou serviço. Estes portais podem, também, catalogar outros *sites* de interesse inseridos da indústria em que se especializam, assim como disponibilizam atividades de *e-Learning*, artigos especializados, e acesso a especialistas em alguns tópicos específicos.

2.1.2 Portal do Conhecimento

Como o conceito de Portal Colaborativo está muito ligado ao contexto dos Portais do Conhecimento, pretende-se inicialmente defini-lo.

Para entender o que vem a ser um Portal do Conhecimento, pode-se ver o artigo de Murray (1999), do IDC. Este relatório descreve quatro estágios da evolução dos portais, baseado no tipo de conteúdo e de ferramentas que estes disponibilizam para o usuário:

- 1.º Estágio: Os Portais de Informação conectam pessoas com a informação;
- 2.º Estágio: Os Portais de Comunicação fornecem potencialidades computacionais para trocas de informação de todos os tipos. Exemplo: Fóruns, Chats;
- 3.º Estágio: Os Portais Verticais conectam pessoas, com outras pessoas, baseadas em suas habilidades, perícias e interesses;
- 4.º Estágio: Os Portais de Conhecimento combinam todos os estágios acima, para entregar conteúdos personalizados, baseados no que cada usuário está buscando.

Neste caso, o Portal do Conhecimento é uma aplicação ou um dispositivo que fornece a relação personalizada e adaptável para quem quer descobrir e interagir com outras pessoas, aplicações e conteúdos relevantes.

Portal é muito diferente de um *site* simples com *frames*, porque possui características adicionais, tais como as citadas por Murray (1999):

Portais não devem ser voltados apenas para o conteúdo, pois ficam sendo inadequados, mas devem nos conectar a tudo e a todos que necessitamos, e proporcionar todas as ferramentas necessárias para que possamos trabalhar juntos.

2.1.3 Portal Colaborativo

Portais Colaborativos, ou Cooperativos, são definidos por Fox (2001), de forma genérica, como ambientes de construção compartilhada de conhecimento, utilizando basicamente ferramentas síncronas e assíncronas de *Groupware*.

O recurso tecnológico, conhecido por *Groupware*, visa interligar as diferentes áreas dentro do portal, para promover a agilização da comunicação, execução de tarefas e construção do conhecimento. Esta categoria de *software* surgiu no início da década de 90 (com o destaque para o Lotus Notes), despontando como elemento para apoiar a evolução das estruturas organizacionais, no sentido de ampliar as possibilidades de trabalho em grupos.

Conceituar *Groupware* não é tarefa fácil, uma vez que não há consenso geral entre os especialistas. No contexto do estudo apresentado, considerou-se a seguinte definição: “Tecnologia de informação usada para ajudar pessoas a trabalharem juntas de maneira mais eficiente” (AL-OMAIM, 1997).

As ferramentas de *Groupware*, ou aplicações de trabalho em grupo, possibilitam aos indivíduos realizar tarefas em conjunto, através de uma variedade de aplicações, incluindo correio eletrônico, agendamento em grupo, acesso remoto à rede da empresa, acompanhamento de tarefas, compartilhamento de informações, discussão em grupo, conferência eletrônica, ambiente integrado de colaboração, gerência de fluxo de trabalho (*workflow*), e gerência de documentos.

Para Candotti (1999), as ferramentas de *Groupware* são embasadas em três conceitos (Comunicação, Colaboração e Coordenação – 3C's), associados à forma como as pessoas podem trabalhar em grupo:

Comunicação: Dar suporte a integração fácil e rápida dos grupos, através do envio de informações, solicitações, arquivos, documentos e instruções. Exemplos de software, seguindo este conceito:

- Chat - possibilita a comunicação das pessoas, através de mensagens eletrônicas em tempo real, utilizando a internet;
- Correio Eletrônico (e-mail) - permite a troca de mensagens eletrônicas entre indivíduos ou grupos deles; esta comunicação ocorre de maneira assíncrona;
- Videoconferência - para a comunicação através da transmissão de imagem e som, via internet.

Colaboração: Faculta que pessoas trabalhem juntas, em projetos ou processos comuns, permitindo combinar experiência (conhecimento tácito) e compartilhamento de informações. As tecnologias de colaboração vem sendo influenciadas por duas grandes tendências: tecnologia de redes e globalização.

Coordenação: Facilita a automação e gerenciamento de seqüência de ações ou tarefas que visem alcançar um objetivo. Para tal, são definidas a ordem em que as tarefas são realizadas e as pessoas envolvidas na

realização das mesmas; os processos são disparados por comandos enviados pelos envolvidos ou por ações automáticas previamente programadas.

Como exemplo desta categoria de software, tem-se o *Workflow*, que permite automatizar e coordenar os processos de negócios de uma empresa, aplicando-se em processos que exigem a preparação de informações estruturadas e ordenadas. Serve para determinar o fluxo do processo, mostrando as etapas corretas para concretização do mesmo, permitindo o acompanhamento de todas as atividades que o constituem (CARVALHO; FERREIRA, 2000).

A Figura 2, mostra como os três conceitos básicos se completam, subsidiando o conceito de *Groupware*.

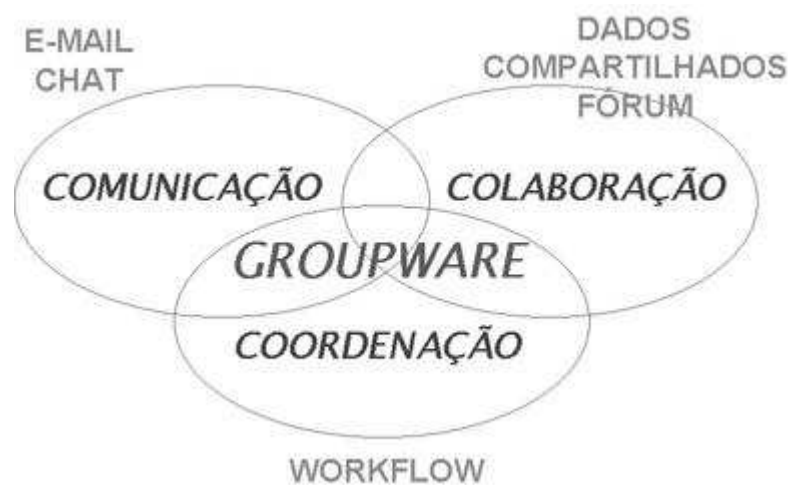


Figura 2: os três conceitos básicos do *Groupware* (adaptado de CARVALHO; FERREIRA 2000).

De acordo com Al-Omair (1997), para que um sistema de *Groupware* seja considerado eficiente, deve congrega três componentes:

- Interface amigável para o usuário;
- Ambiente de desenvolvimento para programar aplicações que atendam às necessidades específicas;

- Infra-estrutura - arcabouço do sistema que suporta os outros dois componentes.

2.2 Ferramentas Computacionais do Portal Colaborativo

Este trabalho envolveu, além da revisão bibliográfica, a participação em várias Listas de Discussão, Grupos de Estudos, troca de E-mails e visitas presenciais com especialistas da área, que resultaram na proposição de uma infra-estrutura adequada para a criação do Portal Colaborativo.

Baseando-se no conjunto de funcionalidades em camadas, pregada por Terra;Gordon (2002), na arquitetura do Portal Colaborativo proposto, têm-se as seguintes camadas:



Figura 3: Divisão em camadas de um Portal Colaborativo, conforme proposto por Terra; Gordon (2002)

- **Camada de Apresentação e Personalização:** define como os usuários enxergam e customizam a informação que é disponibilizada ou acessada pelo Portal. Com soluções mais intuitivas, sendo

oferecidas aos usuários, permite aos mesmos acessar a informação em ambientes mais relevantes e contextualizados;

- **Taxonomia e Mecanismo de Busca:** determina quão fácil será para os usuários encontrarem informação relevante, com base em um conjunto de critérios de busca. Essa é uma funcionalidade essencial que tem melhorado muito, ao longo dos últimos anos, graças a um conjunto de iniciativas com foco em Gestão do Conhecimento. Ao mesmo tempo, os mecanismos de busca têm-se desenvolvido integrando variadas fontes de informação, tornando-se, assim, mais sofisticadas, intuitivas e adaptáveis às diferentes necessidades do usuário;
- **Aplicações Web:** enquanto muitas aplicações foram elaboradas baseadas nas ferramentas *Web*, a maioria dos sistemas legados adquiriu uma interface internet e manteve esta arquitetura. Soluções sofisticadas de plataformas de portais fornecem uma série de recursos que tornam muito fácil integrar ambos os tipos de aplicações sem maiores necessidades de ajustes. No caso de sistemas mais tradicionais, a integração, na camada de apresentação, é facilitada pelo uso de interfaces padronizadas de aplicações (API's) desenvolvidas pelos fornecedores do Portal e/ou pelas comunidades de usuários. Essas API's permitem acesso à informação estruturada, e também à não estruturada. Neste caso, como se utilizou uma interface internet, isso é realizado pelo ambiente Wiki.

Para tanto, verifica-se a necessidade de se construir tanto uma Estrutura Interna, como Externa. Ou seja, uma infra-estrutura para o Portal Colaborativo, moldando-o com ferramentas adequadas, como também uma infra-estrutura colaborativa e de comunicação para acesso externo ao portal, principalmente pelos usuários.

Um dos critérios básicos que foi adotado na escolha dessas ferramentas foi optar-se sempre por softwares Open Source, tanto por princípio, quanto pelos custos envolvidos.

Para montar-se a Infra-estrutura Interna do Portal foi preciso contar com um domínio próprio intitulado “novainter.net”, um servidor APACHE, linguagem PERL (com uso das Expressões Regulares e o ambiente CGI), a instalação do ambiente colaborativo Wiki, Mecanismos de Busca, Agentes Inteligentes de Informação, e Ontologias.

Sob essa ótica, os principais e mais significativos ferramentais externos do Portal Colaborativo são:

FERRAMENTAS SÍNCRONAS

- Chats;

FERRAMENTAS ASSÍNCRONAS

- Fóruns Eletrônicos, ou Listas de Discussão;
- Ambiente Wiki;
- Correio Eletrônico; e
- Blogs (WebLog).

Cada um desses itens serão detalhados a seguir, no intuito de mostrar a função dos ferramentais computacionais, no contexto da proposta.

2.3 Modelo da Proposta

Para ter-se uma estratégia, ou seja, uma metodologia para a resolução dos desafios apresentados anteriormente, desenvolveu-se um arcabouço específico. Para testá-lo de forma efetiva, embora sendo um Portal experimental, este foi deixado disponível, na Web, para qualquer pesquisador interessado (liberando também para um Grupo de Discussão específico de alunos e pesquisadores).

Basicamente, a estrutura dessa proposta é baseada em uma arquitetura formada por componentes passivos (que guardam dados, informações e conhecimentos sobre o domínio estudado) e componentes ativos (agentes de informação).

Na Figura 4, pode-se visualizar elementos que permitem construir a estrutura básica do Portal Colaborativo, e os principais processos de toda a proposta. Serão detalhados, a seguir, seus principais componentes.

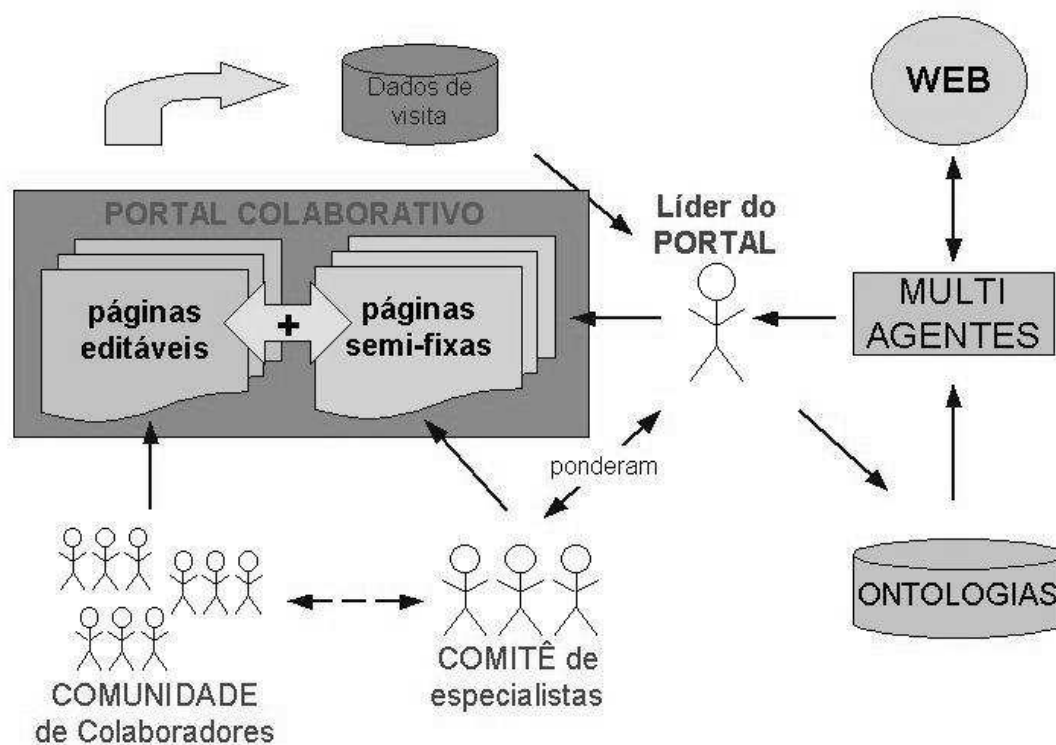


Figura 4: Estrutura Básica do Portal Colaborativo

Percebe-se, inicialmente, de imediato, a importância do personagem Líder do PORTAL; o mesmo, além de definir a estrutura inicial do Portal, poderá, com seu conhecimento específico, da área em estudo, dar a carga inicial das principais páginas semifixas. Compete-lhe, também, graças ao recurso do Agente de Informação, encontrar material adequado para o Portal. É, portanto, o Líder uma das principais figuras neste Modelo.

Em seguida, destaca-se a importância do COMITÊ de Especialistas. Inserido no princípio colaborativo, o Líder do PORTAL pondera e interage com esse grupo de *experts*, na área de Engenharia de Software. Os mesmos atuam numa região nobre do Portal intitulada de páginas semifixas (Núcleo do Portal, na Figura 5). Ou seja, num espaço mais reservado, onde se pretende que poucos tenham a permissão de alterar esse conteúdo, preservando o Núcleo do Portal.

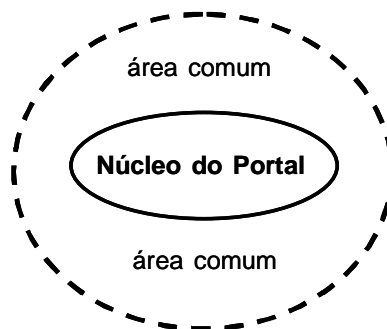


Figura 5: Áreas principais das páginas do Portal

Outro elemento em destaque no modelo é a COMUNIDADE de Colaboradores. Sinteticamente, seriam usuários, cadastrados ou não no Portal, com intuito de extrair e trocar informações, assim como participar ativamente na construção do conhecimento, na comunidade. Os colaboradores têm acesso livre à área comum, tanto de leitura como de gravação, mostrada na figura anterior. No entanto, a integração ao Núcleo do Portal somente ocorrerá depois que, por seus méritos e por sua participação ativa, venha a se transformar num elemento do COMITÊ.

3 INFRAESTRUTURA INTERNA E EXTERNA

A ênfase do trabalho está no uso das ferramentas colaborativas que vão possibilitar a criação do conhecimento. Para tanto, divide-se este capítulo em duas partes: Infra-estrutura Interna e Infra-estrutura Externa, detalhando-se melhor as ferramentas adotadas e suas características técnicas.

3.1 Infra-estrutura Interna

3.1.1 Linguagem PERL

O significado da sigla PERL, conforme seu criador Larry Wall (ainda em plena atividade em seu *site* <<http://www.wall.org/~larry/>>), define como *Practical Extraction and Report Language*, ou seja, uma linguagem prática para extração e apresentação de dados.



Figura 6: Mascote adotado pela linguagem PERL

Como Larry Wall apresenta humor todo atípico, chegou a declarar que, na verdade, PERL poderia significar *Pathologically Eclectic Rubbish Lister* (Listagem de Futilidades Patologicamente Ecléticas) ao contrário da definição oficial *Practical Extraction and Report Language*, conforme citado no *site* oficial da linguagem (<<http://www.perl.com>>). Ainda hoje, é a linguagem mais utilizada, no mundo, para a criação de páginas interativas na Web.

PERL, desenvolvido na década de 80, é um ambiente de programação de uso geral, com suporte à programação orientada a objeto, de alto nível, de fácil aprendizagem, e implementável nas principais plataformas como o Unix, Linux, Windows, Mac (GUELICH, 2001).

Trata-se de uma linguagem interpretada, com característica de *Software Livre*, embora com pequena restrição quanto às regras ditadas pelo GPL. Os códigos, por definição, sendo uma linguagem de *scripts*, ficam abertos. Possui recursos poderosos para processamento de *strings*, através das Expressões Regulares, interação com o sistema operacional e com a rede, além de facilidades como gerenciamento automático de memória e *casting* inteligente e automático de tipos (FILIPO, 2000).

Desenvolvido para implementar um ambiente de catalogação de erros, assim como o Linux, os códigos foram logo disponibilizados na internet por Wall. Assim sendo, criou-se, rapidamente, uma comunidade em torno desse projeto, de extrema dedicação, chamada de *Perl Porters*, citada por toda a internet onde o tema seja PERL. Seus membros incorporaram várias melhorias à linguagem, tornando-a bastante completa.

Embora se apresente como uma linguagem de *scripts*, seus executáveis funcionam compilados, sendo rápidos e poderosos. Esta característica facilita em muito o trabalho do analista, pois os programas podem ser desenvolvidos, modificados e testados facilmente, sem a perda de tempo inerente aos procedimentos de compilação de executáveis. Outra grande vantagem do PERL é possuir, embutido no compilador, um depurador de excelente funcionamento.

Por causa dos seus recursos de formatação de texto e para o desenvolvimento de programas pequenos, porém sofisticados, PERL tornou-se a linguagem preferencial dos desenvolvedores CGI, que é o caso, no

desenvolvimento do Portal Colaborativo. A versão que foi adotada foi a 5, embora esteja em desenvolvimento a versão 6.

3.1.1.1 Expressões Regulares

As Expressões Regulares (freqüentemente abreviada como “regex” - *Regular Expression*) tiveram origem em 1942, quando dois neurologistas (Warren McCulloch e Walter Pitts) publicaram um estudo que tinha como teoria o funcionamento dos neurônios. Anos depois, o matemático Stephen Kleene descreveu algebricamente os modelos desse estudo, utilizando símbolos para representar seus recém-criados conjuntos regulares (do inglês *regular sets*). Com a criação dessa notação simbólica, nasceram as Expressões Regulares que, durante cerca de 20 anos, foram bastante estudadas pelos matemáticos (VILLAS BÔAS, 2001).

Baseado na documentação on-line de Jargas (2003) pode-se afirmar que, em 1968, um algoritmo de busca utilizado no editor de textos *qed* (embutido por Ken Thompson), que depois virou o *ed*, editor padrão dos primeiros sistemas Unix, veio a se transformar em diversas variantes, no aplicativo *egrep*. Esta ferramenta era capaz de pesquisar através de inúmeros arquivos e pastas, usando uma Expressão Regular, listando os seus resultados.

Henry Spencer, em 1986, por fim implementou as expressões a seu modo, num pacote pioneiro, criado em C, intitulado *regex*, que tratava das Expressões Regulares de forma original, permitindo que qualquer um pudesse incluí-lo, gratuitamente, em seu próprio programa.

Uma máquina de Expressões Regulares - máquina *regex* - é um software capaz de "casar" (o que é dito popularmente "bater") uma *regex* fornecida

por alguém, com algum texto. Na verdade, recorre-se ao uso de uma série de fórmulas matemáticas. Ao se fazer uso das *regex*, no entanto, não é preciso se preocupar com isto.

O termo "casar", mencionado anteriormente, equiparar-se-ia ao termo em inglês *match*, no sentido de bater, conferir, combinar, igualar, encontrar, encaixar. É como ocorre em um terminal de caixa automático, em que o usuário só retirará o dinheiro se sua senha digitada "casar" com aquela já cadastrada no banco.

Outro marco importante na história das Expressões Regulares foi o advento da linguagem de programação PERL. Esta linguagem utiliza a mesma sintaxe de Expressões Regulares usada pela velha *grep* do UNIX. Entretanto, o PERL 5 introduziu diversos melhoramentos e poderosos recursos à sua sintaxe *regex*. Expressões Regulares são parte integrante da linguagem PERL, e são apontadas como razão principal de tantos programadores usarem essa linguagem (História das Expressões Regulares. [online]. Disponível em: <<http://www.helpscribble.com.br/webhelp/navbar/hs30010.htm>>).

Existem algumas linguagens atuais que se utilizam do suporte a Expressões Regulares tais como: Python, PHP, C++ e Delphi (essas últimas como bibliotecas de código).

Um programador pode economizar imenso esforço de programação com o simples uso de Expressões Regulares. É muito comum escrever programas para analisar o conteúdo de uma cadeia de caracteres (*strings*). Com o uso de Expressões Regulares, consegue-se, com poucas linhas, o que não se conseguiria com dúzias de linhas de código puro, economizando, também, tempo de depuração.

Apesar dos princípios básicos das Expressões Regulares terem se originado há mais de 35 anos, é um conceito que persiste e que não tem sofrido alterações significativas com o passar do tempo.

Para possibilitar o treinamento e os diversos testes com as Expressões Regulares, utilizadas nos programas PERL, escritos para o Portal Colaborativo, utilizou-se a ferramenta *free* Visual REGEXP (<<http://laurent.riesterer.free.fr/regexp/>>). Foi utilizado este poderoso recurso da linguagem PERL para a realização de filtros mais extensos, nos agentes inteligentes.

3.1.1.2 CGI com PERL

O CGI (*Common Gateway Interface*) estabelece um padrão de comunicação entre servidores HTTP, bancos de dados e outras fontes de informação (NAGAMACHI, 2000). É uma porta de interface comum a muitas linguagens como: C++, PERL, Visual Basic. O CGI permite tornar uma página da Web mais interativa para seus visitantes. Algumas das possibilidades que podem ser incluídas numa página são: contadores de acesso, livros de visitas, chats, fóruns, listas de atualização, pesquisas internas, restrição do acesso a páginas e/ou diretórios, votações.

PERL, como já visto, é uma linguagem de programação e, graças a sua popularidade, erroneamente virou sinônimo de CGI, que, no entanto, trata-se de um protocolo de comunicação (um código). É através dele que o servidor HTTP (ou servidor Web de páginas) intermedia a transferência de informações entre um programa (no mesmo computador que o servidor) e um cliente HTTP (o *browser* de um usuário) (URRUTIA, 2002).

Estes programas realizam processamento específico de informação e usualmente produzem código HTML ou texto puro, que será enviado ao servidor HTTP, e este, por sua vez, enviará para o cliente *www* que solicitou o programa CGI (CINTRA, 2001).

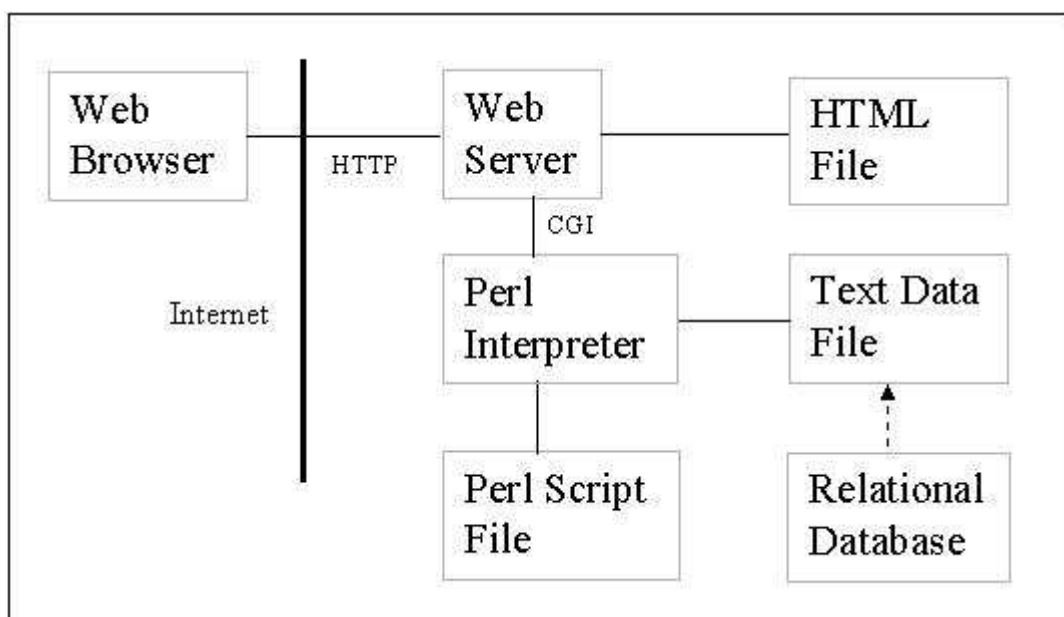


Figura 7: Funcionamento esquematizado do CGI. Retirado do artigo de LAZAR, Z. Peter. CGI vs. server-side javascript for database applications. [online]. Disponível em: <http://developer.netscape.com/viewsource/lazar_cgi.html>. (26out.2003).

Conforme a Figura 7, pode-se ver que, enquanto à esquerda se apresenta um típico *browser*, sendo utilizado por algum usuário, tem-se à direita do *backbone* toda a estrutura que existe num servidor HTTP padrão, com seus arquivos de páginas HTML. Observa-se um destaque para a área CGI e, neste caso retratando bem a realidade do Portal, a linguagem PERL com seu respectivo interpretador. Além do mais, PERL pode acessar tanto uma simples base de dados, como até mesmo um complexo gerenciador de banco de dados (Oracle, SQL-Server, MySQL).

Detalhando melhor o funcionamento técnico da interface CGI, permite que se executem programas, como se fossem verdadeiras portas de acesso *gateways* ao servidor HTTP. Estes *gateways* são programas ou *scripts* (também chamados "CGI-BIN", criando-se diretório com o mesmo nome)

que ao receberem requisições de informação, retorna(m) documento(s) conforme os resultados que lhe(s) é (são) pedido(s). Tal documento pode já existir no servidor HTTP, ou ainda pode ser gerado *on the fly* naquele momento pelo *script*, para atender alguma requisição especial do usuário.

O padrão de um *script* é ler variáveis específicas, processar dados e fornecer o devido retorno; portanto, ele pode ser desenvolvido, em princípio, em qualquer linguagem.

O processo aplicado para passar as variáveis especiais do Google, na hora da primeira filtragem, foi a técnica utilizada conforme prega Cintra (2001):

Uma das formas de passar dados para um programa CGI é usando o método GET. No método GET uma URL é seguida de uma *string* contendo as informações a serem passadas para o programa CGI. Como os dados digitados nos formulários serão enviados para o programa CGI como parte do URL, ao usar o método GET não é possível ocultar os dados através de criptografia. Sempre que for usado o método GET, o servidor HTTP colocará uma *string* contendo os dados do formulário, no formato *urlencoded*, na variável de ambiente QUERY_STRING.

O formato mais utilizado pelos programas CGI para enviar sua saída é chamado de *parsed header output*. De fato, só é requerido nos servidores HTTP entender este formato. Este tipo de saída consiste de um cabeçalho e um corpo de saída, separados por uma linha em branco. O cabeçalho deve, obrigatoriamente, conter uma linha iniciada com "Content-type:". O restante desta linha deve conter a especificação do tipo e subtipo de mídia contidos no corpo da saída. Por exemplo: Content-type: text/html.

Esta técnica poderá ser percebida na parte documentada, sob o título PARSED, no código-fonte do programa do Portal em PERL (vide CD anexado, na contra-capa).

3.1.2 Mecanismos de Busca

Terra; Gordon (2002), consideram ser o Mecanismo de Busca uma das ferramentas necessárias ao funcionamento de um Portal.

Na verdade esta Dissertação começou, inicialmente, com a curiosidade sobre o funcionamento básico dos Mecanismos de Busca. Somente com o avançar da revisão bibliográfica, e após longas pesquisas, é que ficou clara a enorme possibilidade e abrangência que estes recursos computacionais poderiam ter num Portal Colaborativo.

Não existe, ainda, na Língua Portuguesa uma padronização quanto à forma de expressar tecnicamente os Mecanismos de Busca. Em toda revisão bibliográfica constataram-se as mais diversas e possíveis formas de expressar este conceito. Algumas das possibilidades estão destacadas abaixo:

Engenho de Busca

Ferramenta de Busca

Ferramenta de Pesquisa

Máquina de Busca

Mecanismo de Busca

Motor de Pesquisa

Motor de Busca

Search Engines (termo mais utilizado em inglês)

Neste trabalho utilizou-se, preferencialmente, o termo Mecanismo de Busca, e a correspondente sigla MB, detalhando-se, a seguir, este recurso computacional.

3.1.2.1 Dissecando os Mecanismos de Busca

Neste momento, é apresentada uma visão da estrutura e arquitetura dos principais Mecanismos de Busca existentes na Web, como são construídos, e as técnicas envolvidas na criação dos mesmos.

A internet, a mais vasta rede que conecta mais de 170 países no mundo, tem em sua estrutura básica computadores interligados, que podem ser de diferentes tipos, portes, e sistemas, devendo apenas possuir o mesmo protocolo *Transmission Control Protocol/Internet Protocol* (TCP/IP).

Por sua importância e características próprias, os Mecanismos de Busca são, notadamente, objetos de estudo nas áreas de conhecimento da Informática, da Comunicação, da Linguística, da Psicologia, da Engenharia de Redes, e da Ciência da Informação.

A *World Wide Web* trouxe a multimídia para computadores conectados a servidores (*hosts*), permitindo a navegação por *sites* informacionais, interligados por *links* presentes nas páginas iniciais (*homepages*) de outros *sites*, realizando a parte cliente, do Sistema Cliente/Servidor que compõe a internet.

As características marcantes da Web são a utilização: da hipermídia (som, imagem, hipertexto); do *Hypertext Markup Language* (html); do *Hypertext Transfer Protocol* (http); e dos programas gráficos (*browsers*) para a navegação das páginas Web (Mosaic, Netscape, Internet Explorer, Opera).

Em face do inúmero conteúdo na Web, cresce a importância, para milhões de usuários, de como encontrar informações relevantes, separadas da imensa massa de dados sem importância. O primeiro passo para encontrar algo válido na internet é através dos Mecanismos de Busca. E isso funciona praticamente como uma função Matemática. O usuário insere palavras-chave de sua pesquisa, e o MB devolve uma lista de páginas que, supostamente, tenham significado.

A técnica principal utilizada nos MB é técnica dos algoritmos de Recuperação de Informação (RI), contudo, essa estratégia é bem sucedida para pequenas coleções de artigos, ou catálogos de livros em uma biblioteca. Neste caso, quando se menciona o ambiente Web, existem inúmeras informações sendo alteradas constantemente, e dados espalhados por milhões de computadores pelo mundo todo. Isso requer novas e desafiadoras técnicas para a RI.

Vários estudos tentam estimar o tamanho da Web. Embora seja um número bastante discutível e em constante crescimento, estima-se ao redor de um bilhão de páginas ativas. Se imaginarmos que cada página, na média, ocupa da ordem de 5 a 10 Kb, teríamos, somente com dados textuais, um total não inferior a 10 Terabytes (Tb).

O conteúdo da Web dobra, no mínimo, a cada 2 anos. E estima-se que deva perdurar na mesma taxa de crescimento pelos próximos anos. Para se ter uma idéia da constante atualização na Web, um estudo realizado em 500.000 páginas, por um período de 4 meses, concluiu que 23% dessas páginas mudavam diariamente. Nas “ponto-com” o índice chega a ser de 40%, e a média de vida dessas páginas é de 10 dias.

Os Mecanismos de Busca, conforme Arasu (2000), têm sua estrutura composta por três elementos básicos:

- Dispositivo de Coleta (*crawler*): robôs e/ou rastejadores, que obtêm parte ou toda informação dos servidores de páginas Web;
- Mecanismo de Indexação (*indexer*): organizam e atualizam dados recolhidos em um processo de coleta. O produto final será uma base de dados, um catálogo ou um diretório de assuntos;
- Sistema de Busca (*query engine*): permite vários usuários realizarem, simultaneamente, buscas em uma determinada base de dados, de várias formas, dentre elas a busca booleana.

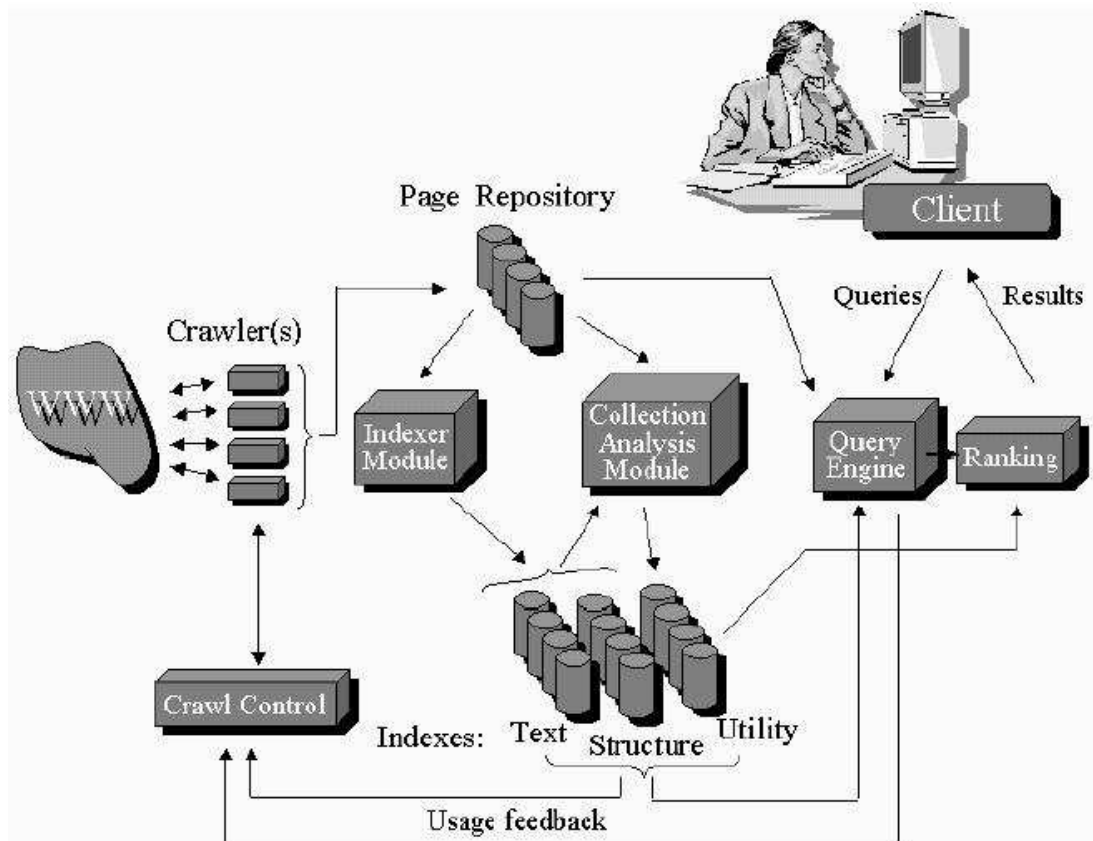


Figura 8: Arquitetura geral de um Mecanismo de Busca, retirado de Arasu (2000).

Na Figura 8, destaca-se a visão geral da arquitetura dos MB. Todo MB possui um ou mais módulos chamados *Crawlers* (ou *Spider*, ou ainda *roBot*) para fornecer a matéria prima para esta operação (mostrado no lado esquerdo da Figura 8).

Com base no artigo de Arasu (2000), pode-se afirmar que os *Crawlers* são pequenos programas que navegam pela Web, como se fossem seres humanos seguindo os *links* e passando pelas diferentes páginas da Web. Dado um conjunto de URL's (*Uniform Resource Locator* - Localizador Uniforme de Recursos), os *Crawlers* recuperam as páginas correspondentes na Web e, em seguida, repassam a informação para o módulo *Crawl Control*.

Este módulo determina quais *links* serão visitados na próxima vez, e quais serão retornados pelos *Crawlers*, ou seja, através deste módulo pode-se controlar o conjunto de URL's que deverão ser visitadas periodicamente. Conseguem também captar, através dos próprios *Crawlers*, novas URL's passíveis de serem visitadas no futuro, retornando ainda às páginas para o interior do *Page Repository* (Repositório de Páginas).

O *Crawl Control* é responsável por direcionar a operação dos *Crawlers*; neste sentido são eles explorados aqui, só que direcionados aos objetivos do Portal. O *Crawl Control* pode, por exemplo, usar um *link* gráfico detectado anteriormente na *Structure Index*, para decidir quais serão visitados ou não, no próximo ciclo de pesquisa.

O *Crawl Control* pode também usar padrões de pesquisa feitos anteriormente para guiar o processo, fazendo a conexão entre o *Query Engine* e o *Crawl Control*, da Figura 8.

O *Indexer Module* extrai todas as palavras-chaves de cada página, gravando-as com a sua respectiva URL. O resultado final é uma tabela de índices contendo todas as URL's que tenham as páginas de determinada palavra (*Text Index*, da Figura 8).

Devido aos fatores mencionados anteriormente, tais como o tamanho da Web e a rapidez de crescimento e renovação das páginas, este índice de texto possui certo grau de dificuldade para ser mantido atualizado. Como se não bastassem estas dificuldades, detecta-se ainda o problema dos vários tipos de índices na Web. Por exemplo, o *Indexer Module* pode criar um *Structure Index*, para refletir os *links* entre as páginas. Tais índices não seriam apropriados para uma coleção de textos que não apresentassem estes *links*.

O *Collection Analysis Module* é responsável justamente por criar essa variedade de índices. Na Figura 08, observa-se que o *Utility Index* foi criado pelo *Collection Analysis Module*. Por exemplo, o *Utility Index* pode ter páginas de tamanhos variáveis, páginas de uma determinada importância, ou com um certo número de imagens.

O módulo *Query Engine* é responsável por receber e preencher os pedidos de busca dos usuários. Para isso, o MB depende dos índices e do *Page Repository*. Devido ao tamanho da Web, quando o usuário entra com uma ou duas palavras-chave, os resultados são muito numerosos.

Cabe ao módulo *Ranking*, conseqüentemente, a tarefa de classificar esses resultados. Dessa forma, são apresentados, no topo da lista, aqueles que mais se assemelham ao que o usuário está procurando, os chamados *top ten* (os primeiros 10 resultados apresentados na primeira página da pesquisa).

O módulo *Query Engine* é importante porque utiliza técnicas tradicionais de RI. A maioria delas utiliza a estratégia de medir a similaridade de textos da pergunta, com os textos dos originais de uma coleção.

As pequenas buscas (*queries*) que ocorrem em cima das coleções típicas da Web, impedem que tais aproximações, baseadas na similaridade, filtrem as páginas relevantes, tanto em qualidade quanto em quantidade. Ao serem desdobrados, conjuntamente, com as técnicas de RI tradicionais, estes algoritmos de pesquisa melhoram a precisão da recuperação, em cenários de busca da Web.

3.1.2.2 Técnicas de Recuperação

As técnicas de recuperação utilizadas pelos Mecanismos de Busca, segundo Chu; Rosenthal (1996), são mencionadas a seguir:

Busca Booleana (*Boolean Search*): Aplicada na grande maioria dos Mecanismos de Busca. Os símbolos booleanos (+, -), ao contrário dos operadores (*and, or, not*), são usados para realizar facilmente a busca booleana, não havendo problemas por parte dos usuários. Por exemplo, no Alta Vista, o símbolo + significa *and*, e o símbolo - significa *not*, enquanto a ausência do uso de símbolos representará a união através do operador *or*. Em outros casos, os operadores booleanos são apresentados como itens nos menus.

Truncagem (*Truncation*) ou Coringa: É um meio de facilitar o usuário, muitas vezes utilizado pelos mecanismos na internet, lembrando o recurso utilizado nos Sistemas Operacionais tais como DOS, Windows, Unix, etc. Os símbolos adotados normalmente para o coringa são o asterisco (*), e, em alguns casos, o símbolo de interrogação (?). Se colocarmos o texto "car*", para pesquisa teríamos como resultados possíveis: car, carro, carroça, Carlos, etc.

Busca por Proximidade (*Proximity Search*): Na internet, a busca por proximidade é empregada, em escala limitada, em apenas alguns mecanismos (ex: Infoseek e Alta Vista) que utilizam o operador *near*. No caso do Google, teremos o sinal de adição (+). A busca por proximidade é uma expressão de busca no sistema de informação normal. Todavia, um grande número de mecanismos de busca da internet não possui esta capacidade, que é essencial para a recuperação de informação na rede.

Busca por Conceito (*Concept Search*): A recuperação Busca por Frase destina-se a recuperar informação relacionada a um conceito dado. Por exemplo, em uma busca onde a expressão é transporte público, informações que tratem de ônibus e metrô deverão ser recuperadas por mecanismos, como o Excite e o Infoseek, caracterizados principalmente por esta capacidade.

3.1.2.3 Tipologia dos Mecanismos de Busca

Apesar de serem relativamente recentes, os Mecanismos de Busca da internet podem ser divididos em grandes grupos, embora existam áreas em comum em muitos deles (CHU; ROSENTHAL, 1996):

Busca Indireta (*Subject Directory*): Possuem índices organizados hierarquicamente - listas seletivas com margem de erro humano - economizam tempo em buscas de informações relevantes - são conhecidos como Catalogs ou Subject Directory. Ex: Yahoo!

Busca Direta (*Search Engine*): Realiza busca por termos ou expressões - possui base de dados criada automaticamente por programas - a solicitação e a coleta de dados são realizadas pelas pesquisas - rastreamentos por *crawlers* - são conhecidos internacionalmente por *Search Engine*. Ex: Alta Vista e o famoso Google.

Mecanismos de Busca Múltiplo: Realizam buscas, simultaneamente, com diferentes Mecanismos de Busca. Realizam a coleta pelos últimos resultados das bases de dados, nos diferentes mecanismos recuperados. Ex: Metacrawler e o Mecanismo de Busca nacional Miner.

Apresenta-se a seguir, uma visão geral das principais categorias de ferramentas de busca da internet, suas características, semelhanças e diferenças, além das vantagens e desvantagens associadas a cada uma.

3.1.2.3.1 Busca Indireta - Diretórios

Intuitivamente, a primeira forma criada para localizar e organizar o conteúdo da internet, baseou-se na recuperação de informações semelhantes a uma biblioteca. E isso só se tornou possível, devido às dimensões da Web naquele período. Um bom exemplo é o Yahoo!, criado em 1994, a partir de um hobby de estudantes de doutorado, na Stanford University (CENDÓN, 2001).

Uma característica típica dos diretórios é a sua estrutura hierárquica permitir a fragmentação das diversas áreas do conhecimento humano (análogo a uma biblioteca real).

Um diretório gratuito e livre que se encontra em destaque é o *Open Directory Project* (<<http://dmoz.org/about.html>>), considerado o maior diretório da Web. Até o Google, no seu menu de diretório, e outros como Netscape Search, AOL Search, Lycos, HotBot, e DirectHit, baseiam a sua busca nesta ferramenta.

O *Open Directory Project* reconhece que as páginas Web crescem numa rápida progressão e os serviços automáticos de busca tem dificuldade em produzir resultados úteis nas pesquisas. As pequenas equipes editoriais, de diretórios comerciais como o do Yahoo!, não conseguem dar conta das submissões de novos *sites*, tornando a sua qualidade inferior. Em decorrência disso, muitos *links* não funcionam e o diretório não consegue acompanhar o crescimento da internet.

Em lugar de lutar contra o explosivo crescimento da internet, o *Open Directory* proporciona os meios para que ela se auto organize. À medida que a internet evolui, também cresce o número de internautas e portanto cada qual pode contribuir para organizar uma pequena parte na Web. Com isso, o restante da comunidade é beneficiada, ao mesmo tempo que esse internautas ajudam a expurgar material de má qualidade e inútil, sendo preservados apenas os *sites* de bom conteúdo.

Frente a essa realidade, o *Open Directory* foi fundado mediante o espírito do *Open Source Movement* (Movimento pelo software livre e com códigos fontes abertos), sendo o único grande diretório totalmente livre e gratuito. Em seu *site* afirma que não há, nem nunca haverá, nenhum custo para a submissão de *sites* ao diretório, e pelo uso de seus dados; fica, pois,

disponibilizado gratuitamente a qualquer usuário que concorde com os termos da licença.

Pode-se transformar num editor do *Open Directory* qualquer cidadão conhecedor de um dos seus tópicos. Esta entidade providencia ferramentas para adicionar, apagar e atualizar *links* em questão de segundos, àqueles que se candidatam a editores. Dessa forma, os editores especialistas nas mais diversificadas áreas passam a ser aceitos e reconhecidos pela comunidade.

3.1.2.3.2 Busca Direta - Motores de Busca

No caso dos Motores de Busca, os editores humanos dos diretórios são substituídos por pequenos programas que navegam pela Web, de forma automática, na procura de novos *sites*, ou alterações naqueles já cadastrados.

Esses pequenos programas (robôs, ou na sua abreviatura em inglês: *bots*) podem receber vários nomes:

“Os robôs, também chamados de aranhas (*spiders*), agentes, viajantes (*wanderers*), vermes (*worms*), ou rastejadores (*crawlers*), são programas que o computador hospedeiro da ferramenta de busca lança regularmente na internet, na tentativa de obter dados sobre o maior número possível de documentos para integrá-los, posteriormente, à sua base de dados” (CENDÓN, 2001).

Dessa forma, os Motores de Busca não são organizados num formato hierárquico, potencializando seus robôs para o armazenamento estratégico de páginas que seus usuários irão captar através de palavras-chave.

Os robôs como são desenvolvidos para manifestarem comportamento mais próximo do ser humano, navegam pela Web através dos *links*, de tal forma

que os servidores da internet não percebam se estão sendo acessados por humanos ou por agentes. Também podem ser desenvolvidos para trabalharem em paralelo, ou seja, vários robôs atuarem simultaneamente, de forma autônoma, ou não, permitindo uma performance adequada do sistema de busca como um todo.

Somente por questão de segurança e de privacidade, as páginas podem possuir um pequeno arquivo intitulado “robot.txt” (estabelecido por normas internacionais em <http://www.robotstxt.org/>) que impedem o acesso desses *crawlers*.

Cendón (2001) ainda afirma que:

Todos os motores atuais utilizam robôs, sendo formados por quatro componentes: um robô, que localiza e busca documentos na Web; um indexador, que extrai a informação dos documentos e constrói a base de dados; o motor de busca propriamente dito; e a interface, que é utilizada pelos usuários.

A título de exemplo de Motores de Busca, cita-se o próprio Google, bem como AllTheWeb, Inktomi, Teoma, Lycos e o AltaVista. Pode-se ter idéia do montante de documentos indexados para cada Motor de Busca, no período de dezembro de 1995 até junho de 2003, ao observar o Gráfico 1.

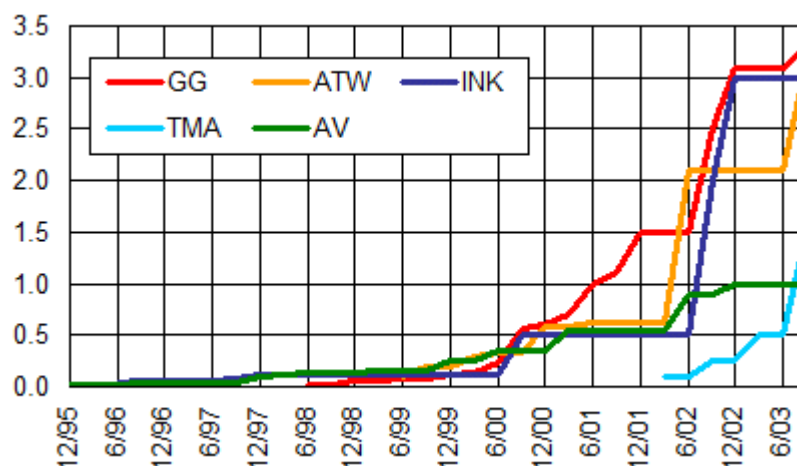


Gráfico 1 - Bilhões de documentos textuais indexados.

Fonte: SEARCH ENGINE WATCH. *Search engine sizes*. [online]. Disponível em: <<http://searchenginewatch.com/reports/article.php/2156481#key?>>. (21 dez.2003).

Legenda: GG=Google, ATW=AllTheWeb, INK=Inktomi, TMA=Teoma, AV=AltaVista.

3.1.2.3.3 Mecanismos de Busca Múltiplo - Metamotores

Na verdade, os Metamotores nada mais são do que a combinação dos anteriores, ou seja, numa única ferramenta, possibilitam a pesquisa em motores ou diretórios de terceiros, consolidando em lista as informações por elas geradas, com a eliminação de *links* redundantes.

A possível demora do procedimento acima, é um ponto negativo dessa estratégia. Como se fica aguardando o resultado das buscas em várias ferramentas, embora realizadas simultaneamente, existe uma espera em função da impossibilidade de sincronismo de todas elas, acrescida da necessidade de ser executada uma filtragem de *links* duplicados. Cita-se, como exemplo, um metamotor desenvolvido com sucesso no Brasil, cognominado de Meta Miner (<<http://miner.bol.com.br>>).

Outro tipo de estratégia que aciona os Mecanismos de Busca, sem a necessidade de estar com um *browser* aberto, são programas metamotores que podem ser “baixados” e executados diretamente na máquina dos

usuários, com a vantagem adicional de eliminar *links* mortos, como o Copernic (<<http://www.copernic.com>>).

Além disso, podem-se citar mecanismos de busca que se diferenciam dos modelos acima apresentados, tendo características originais tais como o MB francês Kartoo, com interface gráfica, que permite visualizar contextualmente, relacionamentos diretos e indiretos com as palavras-chave pesquisadas.

3.1.3 Agentes de Software

Face ao crescimento vertiginoso da internet e as aplicações típicas do ambiente Web, como Mecanismos de Busca, Comércio Eletrônico, organizações virtuais, governo eletrônico, os agentes de software ganharam destaque importante neste cenário.

Descreve-se, a seguir, o que são agentes de software e os principais modelos de arquitetura de agentes, no seu contexto geral, segundo a perspectiva da IA e dos Sistemas Distribuídos.

3.1.3.1 Arquitetura de Agentes

Como referido anteriormente, existem visões distintas de agentes e de suas características. Apresentam-se nesta seção, as arquiteturas de agentes mais representativas, embora Nwana (1996), tenha classificado os agentes nas seguintes dimensões:

- Mobilidade;
- Modelo de raciocínio;
- Função do agente;
- Autonomia;

- Cooperação;
- Aprendizagem;
- Características híbridas.

Combinando-se, assim, as características de autonomia, cooperação e aprendizagem, para formar a imagem da Figura 9.

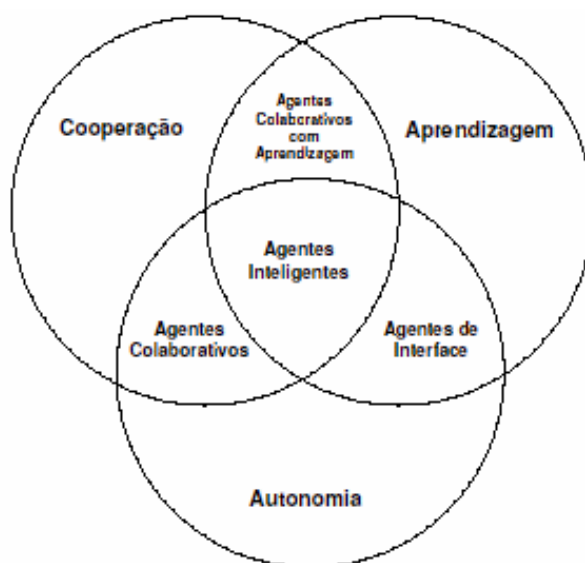


Figura 9: Combinação das características de autonomia, cooperação e aprendizagem conforme Nwana (1996).

Existem alguns pesquisadores, Wooldridge (1999) e Gilbert (1999), que apresentam as seguintes visões adicionais dos atributos de um agente: Inteligência, Reatividade, Habilidade Social, Pró-Atividade, Persistência, Representatividade, Confiabilidade e Comunicabilidade.

Portanto, com base no modelo de Nwana (1996) estabelecem-se os possíveis agentes:

- Intencionais;
- Reativos;
- Híbridos;
- De interface;

- Colaborativos;
- Móveis;
- De informação.

A IA procurou atribuir aos agentes, capacidades de raciocínio e de inteligência inerentes ao ser humano. É esta característica “inteligência” que, segundo a IA, distingue-os de outros componentes de software.

3.1.3.1.1 Agentes Intencionais, Reativos e Híbridos

Agentes Intencionais: Conforme Silva; Delgado (1998):

Nestes sistemas o mundo é representado simbolicamente por um número limitado de elementos que, ao serem combinados, permite que os agentes envolvidos possam operar sobre um conjunto simbolicamente codificado de instruções. As decisões com vista à execução de ações são baseadas em manipulação simbólica, raciocínio lógico e reconhecimento de padrões.

Agentes Reativos: Wooldridge (1999), cita em seu trabalho:

Com base na definição e nas limitações dos agentes intencionais, evolui-se para as arquiteturas reativas. Os agentes, neste caso, não possuem mais as representações internas simbólicas do seu ambiente e nem usam mecanismos de raciocínio simbólico. Por outro lado, interagem entre si segundo um padrão de estímulo-resposta ao estado corrente do ambiente em que se encontram integrados. Existe limitada aplicabilidade desses agentes, em geral voltada para jogos e simuladores.

Agentes Híbridos: A proposição de Maes (1994) foi de combinar os pontos positivos dos agentes anteriores:

Enquanto a representação simbólica permite o desenvolvimento de planos dinâmicos e a tomada de decisões sobre estes, a representação reativa com a sua rápida reação a eventos, sem a necessidade de manipulação de estruturas simbólicas complexas, permitiu a evolução para esse tipo de agente (SILVA; DELGADO, 1998).

Implementou-se este modelo em camadas nas arquiteturas de máquinas de Touring. Na camada superior, com uma máquina de processamento simbólico e na camada inferior, com uma máquina reativa. Mesmo assim o número de aplicações são bem reduzidas, abstraindo-se o problema da gestão das interações entre as diferentes camadas.

3.1.3.1.2 Agentes de Interface

Maes (1994) define agente de interface como um “assistente pessoal que colabora com o usuário no mesmo ambiente de trabalho”. Os agentes de interface caracterizam-se pela autonomia, capacidade de memorização e de aprendizagem, enfatizando as interações com os seus usuários (SILVA; DELGADO, 1998).

Uma característica deste agente é que o usuário simplesmente perceberá as funções das aplicações executadas de forma harmoniosa, e mesmo sendo eles móveis, reativos, cognitivos ou inteligentes (apenas como exemplo), o usuário desconhecerá as características e funcionalidades dos mesmos, ao trabalharem em conjunto para ele.

3.1.3.1.3 Agentes Colaborativos

A arquitetura de agentes colaborativos, também conhecida como multi-agentes, foca a atenção sobre as múltiplas relações e interações entre si. São entidades relativamente estáticas, em função de suas dimensões, que podem apresentar diferentes padrões de comportamento, e principalmente de cooperação e colaboração, obtendo uma representação dos objetivos dos outros. Dessa forma, suas ações, influenciam os objetivos dos outros agentes, como cada qual tem objetivos distintos, surge a questão da recompensa e da negociação entre os mesmos (SILVA; DELGADO, 1998).

Atualmente existe um grande número de aplicações baseadas em agentes colaborativos ou cooperativos. A crítica mais significativa aos sistemas multi-agentes, é de serem concebidos com o paradigma do pensamento simbólico. Um sistema baseado ainda nos princípios da IA produz aplicações, ou mesmo protótipos, com desempenho ineficiente ou pouco flexível.

3.1.3.1.4 Agentes Móveis

Agentes móveis são pequenos programas que têm como principal característica a possibilidade de navegarem por diferentes tipos de rede, através de seus nós. Atualmente seu “habitat natural” é a rede das redes, a Web.

Harrison; Kerhenbaum (1995) destacam quatro pontos em que os agentes móveis, por seguirem as características análogas ao paradigma clássico Cliente/Servidor, possuem vantagens: comunicação ponto-a-ponto (*peer-to-peer*), eficiência, comunicação assíncrona e tolerância a falhas.

3.1.3.1.5 Agentes de Informação

Pode-se destacar que o principal objetivo de um agente de informação é esconder de seu usuário a complexidade e heterogeneidade do acesso à informação. Permite também interagir com o maior número de recursos de informação disponíveis, e providenciar os mecanismos de compartilhamento e divulgação de tais dados (SILVA; DELGADO, 1998).

Este tipo de agente se utiliza, potencialmente, dos Mecanismos de Busca, tanto no desenvolvimento em contextos restritos, como na identificação dinâmica dos seus usuários, assim também na criação de grupos de usuários com interesses em comum.

Um dos focos deste trabalho foi justamente o desenvolvimento e adaptações de um Agente de Informação, trabalhando em conjunto com Mecanismos de Busca e Ontologias, que é o que se pretende relatar a seguir.

3.1.4 Ontologias

Ao se fazer uma retrospectiva da Web, relacionando-a com o tema ontologia, focam-se três gerações da internet (não se mencionando, nesse trabalho, sobre a Internet 2). Nos primórdios da rede, a principal funcionalidade da mesma era a troca de dados, permitindo vencer um grande desafio da época que era a interoperabilidade. Podiam-se movimentar os dados entre máquinas de tipos e fornecedores distintos, contribuindo para a efetiva comunicação, independente de hardware e software através da criação do modelo de referência OSI/ISO (Open Systems Interconnection - International Standardization Organization).

Na segunda geração, devido à criação de uma interface gráfica comum, através da linguagem HTML e a dita *World Wide Web*, a internet cria uma nova forma de relacionamento para grande número de pessoas, fornecendo variedade de dados e informações. Utilizado no começo, no mundo acadêmico, em seguida passou a ser explorado no mundo dos negócios para divulgação de produtos e serviços e também no próprio comércio eletrônico.

Os Mecanismos de Busca mencionados anteriormente, por mais que propiciem aos seus agentes o máximo de “inteligência” possível, ainda encontram, nesta geração, dificuldades em distinguir palavras escritas da mesma forma, mas com significados diferentes. Um bom exemplo é a palavra rede. Num contexto de pescadores, terá um significado diferente daquele de profissionais de informática preocupados em interligar computadores. E ainda poderá significar uma espécie de leito feito de tecido resistente, que se suspende pelas extremidades. Ou seja, dependendo da semântica, a palavra rede pode ser adequada a cada contexto.

Um dos defensores acirrados, bem como um dos divulgadores da Internet Semântica, encontra-se na pessoa de Sir Tim Berners-Lee, com este título recentemente conferido pela Inglaterra, além de ter sido um dos grandes pioneiros e criador da rede mundial.

Idealiza-se num futuro próximo, portanto, a existência de uma Web semântica, onde através de um contexto, ou melhor, através de uma ontologia, possa o software ajudar na importante tarefa da identificação semântica do conteúdo desejado. Nas palavras do próprio Berners-Lee (2001): “A rede semântica globaliza o hipertexto conceitual e a representação do conhecimento”.

A grande dificuldade neste projeto é a criação, desenvolvimento e manutenção de uma ontologia comum. Para cada área do conhecimento humano é passível de se criar uma ontologia correspondente. Mas, mesmo numa área específica, dependendo de como os especialistas estruturam sua área do conhecimento, pode haver distorções, ou polêmicas relacionadas aos vários tópicos.

O avanço tecnológico, rumo à terceira geração, ainda é bastante árduo, por vários motivos. Pode-se ilustrar melhor a problemática utilizando as próprias palavras de Freitas (2002): “organizar toda a internet ontologicamente

esbarra em problemas de várias naturezas” e prossegue, detalhando suas justificativas nos itens abaixo:

O usuário comum que navega na rede e publica páginas, mesmo com auxílio de interfaces gráficas, teria dificuldade em formular consultas que envolvessem regras de lógica e ontologias. Teria, também, problemas para lidar com as complexidades da especificação de ontologias ou instanciá-las em suas próprias páginas, ainda mais utilizando padrões que se sobrepõem em várias camadas, como o trio OIL-RDF-XML;

Novos problemas surgiriam, relacionados à veracidade e correteza das especificações contidas nas páginas;

As próprias páginas possuem, às vezes, conteúdo ambíguo e vago, e isso pode fazer parte da própria informação, seja por seu conteúdo (como, por exemplo, em poesia), ou pela localidade do vocabulário empregado, mantendo o problema da linguagem natural;

Difícilmente, um padrão ontológico para a codificação de páginas, tanto no que se refere à linguagem de markup, como de ontologias a referenciar, será adotado pela “rede da liberdade” num curto espaço de tempo;

Ainda que o fosse, problemas de escalabilidade, tanto das ontologias como da indexação ontológica, se fariam presentes;

Além do mais, a maior parte das páginas que já estão publicadas dificilmente será alterada, ou o será dentro de um tempo suficiente para manter um mercado propício exclusivamente aos agentes que processam ontologias dentro de páginas.

Portanto, encontram-se ainda nos dias atuais, problemas técnicos e sociais na implementação efetiva das ontologias. Embora tenham-se explorado ferramentas de criação de ontologias tais como a *Ontolíngua* e o *Protégé*, percebe-se a inexistência de métodos de desenvolvimento, verificação, validade, documentação para efetivo emprego dos princípios da metodologia orientada a objetos, e o seu próprio reuso.

Na busca por uma ontologia de Engenharia de Software, foram encontradas três visões sobre esta área do conhecimento: no SWEBOK ligado ao IEEE – Computer Society, na ACM – *Association for Computing Machinery*, e na *NEC Research Institute* através do CiteSeer (destacam-se estas ontologias, com maior riqueza de detalhes, nos próximos capítulos).

Uma questão significativa que Freitas, autor acima mencionado, coloca em seu trabalho é: “Até que ponto as ontologias devem refletir as peculiaridades dos métodos de inferência sobre os quais será usado o conhecimento contido nelas?”. Procura ele demonstrar que embora as ontologias possam representar uma certa área do conhecimento, indaga se essas “regras” de inferência contêm em si mesmas esta essência. Paira no ar a polêmica de um possível padrão de criação de uma ontologia.

Acredita-se que as ontologias ainda estão vivendo sua primeira geração. E assim como a internet passou por várias fases, e atualmente, encontra-se evoluindo para a Internet 2, as ontologias deverão receber ainda várias implementações mundiais. Como o processo de desenvolvimento de software *Open Source* cresce a cada ano, e a filosofia do trabalho colaborativo (assim com esta dissertação) tem estimulado vários grupos, visualiza-se maior contribuição mundial no desenvolvimento das ontologias, como conclui Freitas:

As ontologias já começam a desempenhar o papel de conhecimento estruturado disponível para reuso em larga escala por sistemas e programas, um acontecimento sem paralelo na história da ciência da computação. Isso representa para a história da informática o que, para a história humana, representou a criação de escolas, enciclopédias e bibliotecas. Enfim, o armazenamento de conhecimento, já que o conhecimento agora trafega entre computadores e sistemas que podem lançar mão deles, manipulá-los e aplicá-los no cumprimento de suas funções. As ontologias também têm servido, em certas áreas, para unificar o conhecimento e para formar consensos sobre certos conceitos, causando uma integração de grupos de pesquisas e sendo utilizadas, inclusive, com propósitos educativos.

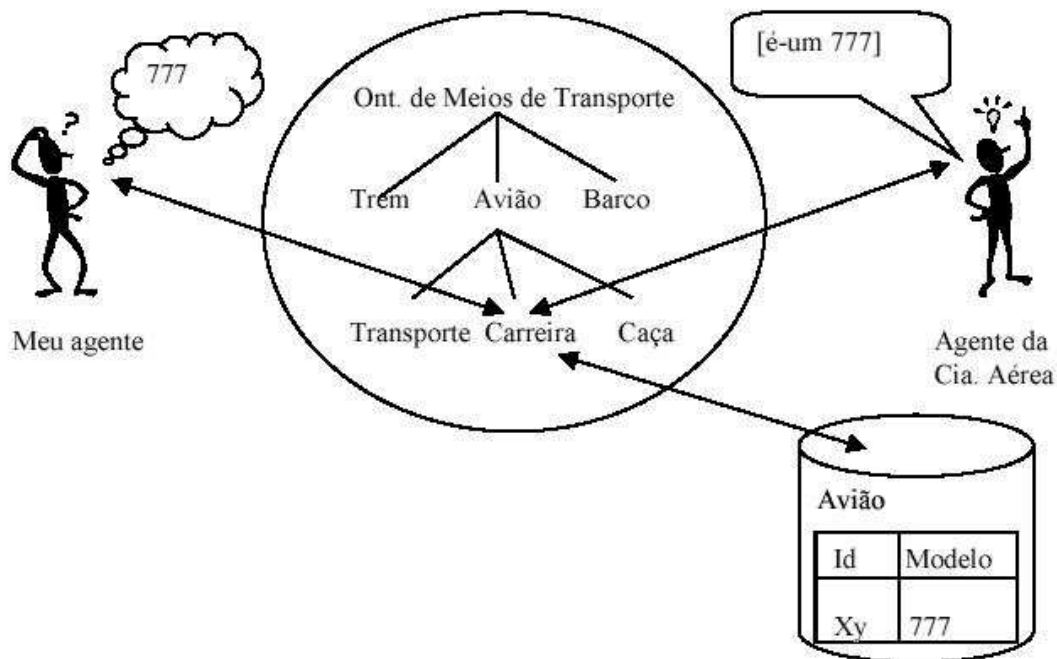


Figura 9: Sistemas Multi-agentes cognitivos para a recuperação, classificação e extração integradas de informação da Web - Freitas (2002).

3.1.5 Ambiente Wiki Wiki

Em novas modalidades de desenvolvimento de projetos tais como Engenharia Simultânea, ou mesmo em práticas da nova metodologia de desenvolvimento de software chamada *eXtreme Programming* - XP (Programação Extrema), o processo do trabalho colaborativo é vital.

A infra-estrutura necessária para dar suporte ao trabalho colaborativo segue uma tendência que tipicamente utiliza o ambiente Web, permitindo que através de *browsers* (Internet Explorer, Netscape, Mozilla ou Opera - citando apenas os mais famosos) e uma conexão à internet, pessoas possam trabalhar de forma produtiva, de qualquer lugar e a qualquer hora.

Para isso, os próprios criadores do *Extreme Programming*, liderados por Ward Cunningham, estruturaram uma ferramenta que permite que qualquer

usuário, mesmo que seja leigo em ferramentas computacionais, pudesse inserir dados através da Web, e alimentar uma base comum de dados vitais para um projeto colaborativo bem sucedido. Cunningham (2001) denominou esta ferramenta de WikiWikiWeb, fazendo uma analogia à palavra Wiki Wiki, que em havaiano significa "muito rápido" (para simplificar, refere-se ao longo do trabalho, somente como Wiki).

Este sistema aberto e democrático que permite criar, gerenciar e publicar conteúdo de um *site*, faz do Wiki uma espécie de plataforma de cooperação e interatividade. Transcende-se o usuário da posição de leitor/observador de páginas para a de criador/autor/colaborador. Assim, esta ferramenta poderia ser utilizada por uma comunidade universitária, na qual os alunos publicassem seus projetos, os professores dialogassem, inserindo comentários para que os alunos reformulassem seus trabalhos, e especialistas pudessem também fazer comentários.

O Wiki também é muito interessante para uso de um grupo comunitário e toda e qualquer organização coletiva que necessite de trabalho colaborativo, pois, com essa ferramenta de colaboração online, pode-se criar e desenvolver projetos em conjunto e escrever textos coletivos de forma bastante dinâmica (Cadernos Eletrônicos - comunidades virtuais - listas, chats).

Em 1995, Cunningham já havia criado no *site* desenvolvido especialmente para esse projeto: <<http://c2.com/cgi/wiki?WikiWikiWeb>>, a filosofia do Wiki, que se transformou, rapidamente, numa das ferramentas mais populares para a criação coletiva de conteúdo, gerando projetos de grande destaque e valor, como o *WIKIPEDIA*, uma enciclopédia criada coletivamente, contendo mais verbetes do que a *Enciclopédia Britânica*.

Pregando a simplicidade e granularidade de comunicação entre desenvolvedores de um projeto, o Wiki se encaixa como uma ferramenta

perfeita para ser usada por programadores que utilizam a metodologia XP em seus projetos, e esta foi rapidamente assimilada por outros usuários na Web.

Cunningham; Leuf (2001), no livro *The Wiki Way*, definem Wiki como:

uma coleção livremente expansível de páginas Web interligadas num sistema de hipertexto para armazenar e modificar informação num banco de dados, onde cada página é facilmente editada por qualquer usuário com um *browser* com suporte a formulários.

Páginas no Wiki podem ser controladas, criadas, conectadas, editadas, apagadas, movidas e renomeadas, por uma linguagem de programação ou geração de *scripts*, e são armazenadas ou como simples arquivos de texto puro no formato ASCII, ou num banco de dados relacional externo, como MySQL, Oracle, ou PostgreSQL.

Páginas Wiki são formatadas ou visualizadas como HTML através de *templates* pelo servidor Web Wiki. Deve-se ressaltar que o Wiki, originado por Cunningham, foi escrito em PERL.

Desde 1995, foram criados muitos outros clones do Wiki, ou aplicações para gestão de conteúdo (CMS) com características similares ao Wiki, na sua grande maioria com licenças de software aberto ou livre, em diversas linguagens que rodam em qualquer plataforma, incluindo PDAs e telefones celulares.

3.1.5.1 Estrutura de um Wiki

Não existe dificuldade ou maior complexidade para montar um Wiki. Basta ter um servidor acessível na Web e alguns conceitos básicos. O primeiro passo, é identificar qual é o software (ambiente) de Wiki mais adequado às necessidades.

Descreve-se, mais adiante, alguns dos mais relevantes e ativos projetos de software livre que se dedicam a construir um Wiki Engine, um servidor de Wiki. Alguns fatores podem ajudar na decisão. Para alguns puristas, não deveria existir controle de acesso ao Wiki, pois todas as páginas seriam livres para todos editarem. Porém, alguns softwares de Wiki implementam esquemas alternativos de segurança, onde algumas páginas podem ser bloqueadas para escrita, ou até mesmo para leitura.

Outro ponto interessante é permitir a carga de arquivos (*up-load*) para o Wiki. Este recurso facilita a criação de um repositório de documentos, mas traz riscos de segurança ao servidor. No caso do Portal Colaborativo, adota-se esse recurso apenas para usuários devidamente autorizados.

Além disso, é importante analisar a linguagem usada no Wiki. Alguns Wikis permitem que o usuário crie páginas usando HTML, por exemplo. Na adaptação que foi feita ao Portal, permitiu-se, por existirem pessoas da comunidade conhecedoras de HTML, a possibilidade de inserir diretamente o código nativo, nesta linguagem.

Questões técnicas também devem ser consideradas. Alguns wikis são *scripts* em PERL que podem rodar na grande maioria dos provedores. Outros talvez necessitem de bancos de dados instalados, ou aplicações mais complexas. A escolha pode variar de acordo com a linguagem de programação usada no Wiki (para que possam ser feitas alterações no software) e, também, com a especificação do servidor onde deverá ser

implantado o Wiki. Baseando-se nesses pontos, escolhe-se o software de Wiki mais propício a uma comunidade específica.

Um outro problema levantado é a possibilidade dos textos do *site* poderem ser desvirtuados, por usuários mal intencionados. Especialistas argumentam que a própria comunidade toma conta do Wiki, corrigindo ou apagando o que foi colocado com propósitos inconvenientes. Além disso, a facilidade de alterar o *site* acaba funcionando como uma defesa contra *crackers*, pois não há nenhum desafio envolvido em mudar os dados de um *site* Wiki, conforme cita Caio Barra Costa, administrador do Wiki brasileiro: *CuDeBebado*.

Realmente, os *crackers* preferem agir em *sites* em que podem demonstrar sua habilidade em driblar sistemas de segurança. Mesmo assim, existem alternativas, como *backups* e bloqueio de IPs, que podem ser usadas no caso de depredação do Wiki. Adotou-se este tipo de segurança dentro do Portal.

Mesmo com todas essas facilidades, é estranho que o conceito, que gerou tantos *sites* no exterior, desde 1995, ainda tenha pouco uso no Brasil, sendo praticamente desconhecido pela comunidade técnica.

3.2 Infra-estrutura Externa (Síncronas e Assíncronas)

Foi denominado Infra-estrutura Externa, neste trabalho, toda ferramenta computacional que auxilie ou forneça suporte aos recursos humanos (usuários, pesquisadores, pessoal técnico) envolvidos no Portal Colaborativo, formado basicamente de ferramentas de *Groupware*.

Uma sub-divisão necessária foi a criação do processo de sincronização de interação na comunidade. Ou seja, se esse processo está sendo realizado

de forma síncrona (em tempo real quando um usuário pode entrar em contato on-line com outro); ou, em caso contrário, de forma assíncrona. Observando-se as principais ferramentas utilizadas no Portal, destaca-se que a grande maioria delas trabalha de forma assíncrona.

Embora a tecnologia tenha avançado bastante em ferramental síncrono (um exemplo típico seria a videoconferência) ainda apresenta um custo considerável em qualquer projeto. Um tipo simples de ferramenta de conferência síncrona é representado pelos IRC's (Internet Relay Chat). Os IRC's permitem a comunicação entre vários participantes através de uma janela comum, onde tudo o que é escrito por cada participante pode ser lido imediatamente pelos demais, como no popular ICQ. No caso do Portal Colaborativo, preferiu-se adotar um Chat, desenvolvido na linguagem padrão do Portal, a linguagem PERL.

Num experimento realizado entre Alexandre Freire, estudante colaborador do Instituto de Matemática e Estatística, Universidade de São Paulo (IME/USP), e o autor deste trabalho, visando o desenvolvimento colaborativo à distância de um artigo técnico, verificou-se que, utilizando somente o ambiente Wiki, em muitos momentos, o mesmo deixava a desejar por ser inteiramente assíncrono. Se, na época, houvesse um simples Chat instalado, alguns problemas que ocorreram, poderiam ter sido facilmente sanados.

A instalação de videoconferência foi descartada, num primeiro momento, devido ao seu alto custo, e pelo número ainda reduzido de pesquisadores na Web que possuem este recurso. No entanto, não se descartou sua utilização num futuro próximo.

Zaina; Bressan; Ruggieko (2002), afirmam que a utilização do Chat é prática e efetiva na construção do conhecimento, em grupos de 5 a 7 pessoas.

Grupos maiores podem provocar verdadeiro caos nos encontros em tempo real.

Outro ponto a ser considerado é que nem sempre se deseja, no momento da pesquisa e da análise de informações, que o pesquisador venha a interromper constantemente, de forma on-line, outro colega. Portanto, a forma mais produtiva é disponibilizar suas informações, ou mensagens, através do Portal, e, no momento apropriado, permitir que outros pesquisadores interajam com aqueles dados.

Partindo-se da premissa de que o Portal necessita da maior interação de todos na construção do conhecimento daquela área, é desejável que hajam poucas ferramentas síncronas, permitindo que a grande maioria possa trocar informações, sem utilização de grandes recursos técnicos.

Principais ferramentas computacionais utilizadas no Portal Colaborativo:

SÍNCRONAS

Chat

ASSÍNCRONAS

Fóruns Eletrônicos, ou Listas de Discussão;
Wiki como ferramenta externa de comunicação;
Email direto do Portal;
Blogs (Web Log).

3.2.1 Wiki como ferramenta externa de comunicação

Ao longo dos anos, muitas comunidades surgiram em torno de Wikis públicos, com resultados variados. Discutem-se, a seguir, as características

determinantes de um software Wiki e como funciona a dinâmica das comunidades que conseguiram fazer seus Wikis evoluírem com sucesso.

Cunningham, autor anteriormente citado, definiu o Wiki como a base de dados online mais simples que poderia funcionar, tendo a mesmo conseguido popularidade imediata na comunidade devido, principalmente, à disseminação do uso da internet e ao entusiasmo de vários de seus colaboradores.

Elaborou também o *site*, que foi e permanece dedicado à PeopleProjectsandPatterns, e à estrutura do wikiwikiweb, que o opera. Wiki se configura como um sistema de criação e de edição de páginas na, e para, internet ou intranet. Por meio dele, grupos virtuais de colaboradores registram e atualizam informações de forma rápida e fácil, e podem modificar, por acréscimo ou supressão, os textos ali exibidos mesmo que tenham sido editados por outrem. Como Ivanoff (2003) indica:

Wiki é um sistema de composição; é um meio de discussão; um depósito; é um sistema de correspondência; é uma ferramenta de colaboração. Na verdade, temos dificuldade em defini-lo, mas é uma forma divertida para se comunicar de forma assíncrona pela rede.

No *site*, intitulado IberiaWiki, existe interessante definição de Wiki como sendo uma minúscula peça de software, inserida num servidor Web, que permite a quaisquer usuários criar e editar páginas da internet livremente, usando apenas o seu *browser*. E continua:

A novidade do Wiki, entre os mecanismos de comunicação de grupos, está em permitir a organização de contribuições a serem editadas em acréscimo ao próprio conteúdo. Como outros conceitos simples, a edição aberta tem alguns efeitos profundos e sutis sobre a utilização do Wiki. Como permite aos usuários criar e editar rotineiramente página em um *site* na Web, torna-se excitante e encoraja o uso democrático da Web, promovendo a elaboração de conteúdo por usuários não técnicos.

<<http://www.openspaceworld.org/iberia/iberia.cgi?ComoUsarEsteWikiWeb>>.

Qualquer um pode facilmente aprender a usar e contribuir para um ambiente Wiki, precisando apenas assimilar regras simples de formatação (lamentavelmente ainda não estarem padronizados os ambientes Wikis existentes).

3.2.2 Diferenças entre Blog e Wiki

Baseado num artigo de Mattison, segue-se breve comparação de Blog e Wiki, para visualizar os pontos positivos e negativos de cada ferramenta <<http://www.infotoday.com/searcher/apr03/mattison.shtml>>. (30out.2003):

1. Wikis e blogs são exemplos de *Groupware*; cada qual contém elementos de uma arquitetura para trabalho colaborativo;
2. O primeiro Wiki <<http://c2.com/cgi-bin/wiki>> entrou on-line em março de 1995. Os primeiros blogs reconhecidos entraram on-line a partir de 1997 <http://www.rebeccablood.net/essays/weblog_history.html>;
3. Uma característica interessante dos Wikis é que pela sua estrutura, eles podem até ser usados como um blog. No entanto, com os recursos técnicos de um blog, não se permitiria a construção de um Wiki;
4. Tanto Wikis quanto blogs são extensíveis o suficiente para se tornarem indistinguíveis. O princípio de arquitetura geral de Wikis e blogs continua o mesmo: Wikis promovem conteúdo sobre forma, blogs promovem forma (organização temporal) sobre conteúdo;
5. Todos os Wikis incluem, por *default*, um mecanismo de busca integrado para achar conteúdo; nem todas as aplicações de autoria e serviços de hospedagem de blogs incluem mecanismos de busca integrados;
6. Wikis, em sua grande maioria, são abertos a qualquer um, no domínio do servidor do Wiki, mas não têm, geralmente, mecanismos de segurança contra usuários não autenticados (não convidados);
7. Blogs, por *default*, são seguros contra colaboração aberta, mas podem ser administrados para permitir colaboração limitada, normalmente via adição de comentários, ou num fórum com discussões estruturadas por tópicos.

Adotou-se, no Portal Colaborativo, um Blog com a finalidade de posicionar os usuários sobre as últimas implementações realizadas dentro do próprio Portal.

4 EXPERIMENTOS NO PORTAL COLABORATIVO

Para a concepção do Portal Colaborativo houve a necessidade de pesquisar as possíveis ferramentas necessárias para a sua estruturação, além de explorar infra-estruturas de software, hardware e de comunicação necessárias para a sua construção (linguagem PERL, Mecanismos de Busca, Agentes de Software, ontologias e o ambiente Wiki).

Uma das primeiras providências práticas tomadas foi a aquisição de um domínio próprio na internet via *The Internet's Network Information Center - InterNIC* (NOVAINTER.NET - diretamente cadastrado nos Estados Unidos), devido à menor exigência de burocracia e documentação formal. Verificou-se que, se não fosse feito desta forma, dificilmente se teria a oportunidade de instalar de forma adequada o Portal num provedor gratuito (em particular, o acesso à área de CGI).

Após resolver os problemas típicos de transferência de arquivos, acessos inadequados, e até troca de provedor, conseguiu-se estabilizar as instalações numa empresa especializada em hospedagem de *sites* chamada *Internet Host* (<<http://www.internethost.com.br/>>). Os principais fatores técnicos que pesaram na escolha desta empresa foram:

- Sistema Operacional Linux;
- Servidor Apache;
- Servidores no Brasil;
- Servidores com conexões redundantes;
- Configuração de domínio grátis (Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP e *InterNIC*);
- Domínio acessível tanto com ou sem *www*;
- Conexão direta a um *backbone*;
- Suporte técnico via e-mail, atendimento on-line ou telefone;

Painel de Controle para administração da conta contendo:

- Administração de arquivos;
- Mudança de senhas;
- Administração de ilimitados e-mails;
- Grupos de correspondência;
- Diretório CGI-BIN próprio, PHP4, PERL 5.6.1 e Python;
- Diretórios protegidos (*Protect Directories*);
- Acesso ilimitado por FTP e SSH;
- Logs de acesso não formatados;
- Subdomínios;
- Configuração das páginas de erro;
- Sala de Bate Papo;
- Enquetes;
- Servidor de banco de dados MySQL;
- Estatísticas de acesso (*Webalizer* e *Analog*);
- Servidor SSL (*Secure Sockets Layer*);
- SSI (*Server Side Includes*);
- Backups diários.

Dentro dos nove tipos de planos de hospedagem, foi escolhido inicialmente o plano de 300Mb em disco. Com o decorrer dos testes, ampliou-se o plano para 500Mb (R\$49,90 mensais, em fev.2004), pois o mesmo permitia a transferência mensal de até 3072Mb, para atender uma demanda crescente.

Criou-se um subdomínio intitulado: <<http://wiki.novainter.net>>, para facilitar o processo mnemônico dos usuários que estivessem utilizando o ambiente colaborativo Wiki.

Na configuração do portal utilizou-se a área de CGI com PERL, acessos via FTP, vários *scrips*, e administração de arquivos.

Para os testes executados com a linguagem PERL e CGI, houve a necessidade de instalar o servidor *Apache*, facilitando os testes locais antes de encaminhar para o servidor principal.

O ambiente Wiki que apresentou maior aderência ao projeto foi o UseModWiki, desenvolvido em PERL. Como se procurava uma linguagem adequada para a personalização do agente, e depois da incorporação do ambiente Wiki, optou-se por padronizar totalmente a linguagem do Portal em PERL.

4.1 Experimentos com Wiki

Embora o Wiki seja considerado uma solução simples para resolver os problemas de criação de ambientes colaborativos, ele "revolucionou" o modo de trabalho e a forma de visualizar a internet como um todo. Depois da compreensão desta ferramenta computacional, não se consegue mais navegar pelos diversos *sites* de forma indiferente. Basta existir alguma contribuição em algum *site*, para haver o desejo de alterar, ou incluir, esta nova informação.

Mesmo sendo o ambiente da Web tipicamente interativo, quando comparado com outros meios de comunicação (TV, rádio, jornais), o Wiki pode transformá-la num ambiente ainda mais interativo, juntando comunidades virtuais com interesses específicos.

No Brasil, o Wiki ainda é uma ferramenta colaborativa pouco conhecida e explorada. No entanto, ela é ideal para administrar projetos, propiciar um ambiente de desenvolvimento do conhecimento (KM) de forma colaborativa, e registrar, automaticamente, toda a documentação necessária para o bom gerenciamento de um portal.

Como no desenvolvimento do Portal Colaborativo existiu a necessidade de estruturação e desenvolvimento do conhecimento, o Wiki veio atender esta necessidade, substituindo até ferramentas computacionais de e-Learning mais complexas.

Alguns pontos negativos que se percebe no atual estágio dos Wikis é a sua falta de padronização na formatação dos textos. Ou seja, utilizando-se de um ambiente Wiki e ao mudar para outro ambiente, pode-se ter problemas na formatação do texto. Uma possibilidade que permitirá resolver isto será através da utilização de ferramentas de conversão.

Um outro ponto é o fator cultural: as pessoas estão tão acostumadas a navegar por um *site* e interagir com o mesmo de forma passiva, que quando entram num ambiente Wiki, aparentemente sem controle, têm um grande receio de modificar o conteúdo do *site*, trazendo suas idéias para benefício da comunidade.

Para possibilitar a exploração mais completa do ambiente Wiki, e vivenciar melhor todas as suas possibilidades, foram realizados vários experimentos tais como:

- Geração de um artigo de cunho científico em conjunto com outro colega Alexandre Freire, citado anteriormente, do IME-USP, usando apenas este ambiente como base de comunicação e também de construção do artigo;
- Montagem e estruturação do Portal *CarpeDiem* (<<http://kiwi.arca.ime.usp.br/?CarpeDiem>>), em julho de 2003, originado da Lista de Discussão *Competitive-Knowledge*, onde dois participantes decidiram dedicar-se com mais ênfase ao tema da aprendizagem, bem como produzir conteúdos e experiências práticas, indo além do simples debate e troca de informações.

Este projeto chegou a ser composto de 34 profissionais de várias áreas do conhecimento, com o objetivo comum de colaborar para a construção de um processo educacional mais sistêmico e dinâmico;

- Embora possuindo um *site* pessoal, foi reconstruído o mesmo, tendo como base o ambiente Wiki. Isto foi muito útil na estruturação, composição e apresentação de aulas, pesquisas científicas, e organização pessoal;
- Construção, apresentação e discussão com o orientador durante a elaboração desta dissertação, reduzindo a necessidade de impressão de material e permitindo discussão, à distância, entre uma reunião de encontro de trabalho e outra.

Mais um item importante, no desenvolvimento do Wiki, para o Portal Colaborativo, foi o fato dele ter servido tanto para a estrutura de montagem do próprio portal, análogo ao cimento e tijolos na construção de um edifício, como também de uma Central de Telefonia PABX nos processos de comunicação existentes dentro do próprio conceito Wiki. Este processo de comunicação foi útil tanto para comunicações internas, atualizando-se o portal e seus usuários, como para comunicações externas, referenciando-se a outros *links* e *sites* na internet.

4.2 O ambiente Wiki: UseModWiki

O ambiente Wiki adotado no Portal Colaborativo, intitulado UseModWiki (<<http://www.usemod.com/cgi-bin/wiki.pl?UseModWiki>>), foi programado na linguagem PERL, e escrito por Clifford Adams, baseando-se no conceito original do Wiki, criado por Ward Cunningham, anteriormente citado. Por se

achar inserido na filosofia do código aberto, foi possível adaptar e alterar o programa original para atender ao modelo proposto do Portal Colaborativo.

Os programas fonte do UseModWiki estão muito bem documentados, e embora o programa seja extenso (mais de 5000 linhas), foi relativamente fácil estudar, alterar e adaptar o programa original para a proposição do portal. O código fonte alterado está incluído no CD-ROM anexado a este trabalho.

Tornou-se necessário testar cada adaptação em um ambiente Web, com vários *uploads*. Em cerca de dois meses obteve-se o núcleo básico do wiki construído.

Este ambiente possui uma parte inicial de personalização, permitindo alterar o plano de fundo, incluir logotipos e imagens, criar áreas de *upload* de arquivos, e montagem de um ambiente relativamente seguro. Neste tocante, o ambiente faculta ao administrador de sistema, ou aos editores de páginas, uma vez identificados por *login*, alterar os status das páginas, possibilitando que elas sejam livres, ou seja, com permissão de alterar, ou de travá-las para edição.

Um outro recurso personalizável que foi adotado (facilitado somente para administradores) foi à permissão de apagar páginas. Dentro do conceito original do Wiki este ponto é bastante polêmico (pois se as páginas estão abertas para todos, alguém poderá querer interferir naquela página que está sendo apagada). Mas, por questões organizacionais, e dentro do Modelo proposto, preferiu-se adotar o critério de incluir a opção *Delete this page*.

Um recurso bastante poderoso, e útil dentro do contexto, foi explorar o módulo de Mecanismo de Busca Interno, como já se tinha mencionado detalhadamente no capítulo 3. Como a pretensão do portal é acumular as várias contribuições dos participantes, gerando conhecimento cumulativo,

nada melhor do que ter um recurso que permita ao usuário localizar, com facilidade, qualquer conteúdo que aí exista.

Outro recurso que foi disponibilizado para o administrador foi o *Edit Banned List*. Este recurso permite, uma vez identificado o IP de algum usuário indesejado, ou não autorizado, a sua inclusão num arquivo especial que contém uma lista negra de IPs. Uma vez inserido nesta lista, o usuário com aquele IP específico não terá acesso ao portal.

Para o administrador, ainda existem os recursos *Edit/Rename pages* e o *Run Maintenance*. O primeiro tem o intuito de permitir a alteração de qualquer nome das páginas do portal, enquanto o segundo visa realizar manutenções administrativas no *site* como um todo.

Para permitir a melhor identificação dos usuários que estão no sistema, existe o recurso intitulado *personalização*, e também o item chamado *LOGIN*. O primeiro atende aos usuários genéricos, e o segundo realiza a identificação dos administradores ou editores especiais de páginas.

Como qualquer ambiente Wiki, o portal possui um *RecentesEdicoes* para o acompanhamento de todas as páginas editadas, nos últimos sete dias, ou conforme as opções de 1, 3, 30 ou 90 dias anteriores.

Ainda existe, nas páginas do portal, um recurso convencional que permite visualizar as últimas revisões realizadas naquelas páginas, servindo tanto para acompanhar a evolução das mesmas, como, também, em caso de erro, para recuperar os dados anteriores. Uma outra aplicação para esse recurso é, no caso de algum usuário mal intencionado ter danificado uma página, permitir ao administrador recuperar facilmente a versão anterior.

Inseriu-se, ainda, o recurso *todas PÁGINAS* permitindo a qualquer usuário o acesso à árvore, ordenada de forma alfabética, de páginas do portal. Esta

árvore permite visualizar a página principal e todas as sub-páginas dela dependentes. Por uma limitação do sistema, pode-se construir só dois níveis (página principal e suas sub-páginas). Este recurso também informa a quantidade de páginas que já foram geradas pelos usuários.

4.3 A Linguagem PERL, Expressões Regulares e CGI

A linguagem de programação PERL foi selecionada para a construção do Portal Colaborativo, por vários motivos. O principal, foi em decorrência do próprio ambiente Wiki que foi escolhido, pois o UseModWiki foi escrito originalmente nesta linguagem.

Outros motivos para tal escolha, além de permitir uma padronização na linguagem de desenvolvimento, foi a facilidade de aprendizagem. Embora o autor nunca a houvesse estudado de forma regular, os conceitos e a prática adquiridos com outras linguagens de programação permitiram construir rapidamente um domínio básico do PERL.

Uma outra justificativa para a escolha, foi pela opção natural de poder trabalhar no conceito de *Client/Server*, ou seja, em camadas. Por ser o CGI um protocolo de comunicação, conforme já mencionado, através do qual o servidor Web realiza a transferência de informações entre um programa no servidor e um cliente HTTP (via *browser*), e como a grande maioria dos seus programas são feitos em PERL, houve mais este argumento contundente.

Ao explorar os recursos disponíveis na linguagem PERL, encontrou-se um recurso poderoso chamado de Expressões Regulares. Com elas tornou diminuta a complexidade dos programas, obtendo algoritmos mais sintéticos e objetivos. Foram utilizadas Expressões Regulares intensivamente no desenvolvimento dos agentes e na filtragem da base de dados do Google.

Portanto, além de se utilizar a linguagem PERL como apoio na adaptação do ambiente Wiki, explorou-se a mesma na construção de uma série de programas tais como: de comunicação – chat e fórum; utilitários – e-mail direto pelo *site*.

Um destaque especial deve ser dado ao programa de tratamento das informações provenientes do Google (Anexo A).

Conforme a Figura 13 (apresentada nas próximas páginas), pode-se observar que, no servidor NOVAINTER.NET, existem basicamente dois programas: o WIKI.PL (ambiente UseModWiki) e o META.CGI (agente de informação). Para o desenvolvimento do programa META.CGI utilizou-se a base de um sistema *open-source*, que depois de toda a adaptação e desenvolvimento, resultou nos módulos PERL: `websearch-google.pl` e `websearch-acm.pl`, e nas páginas auxiliares HTML: `metasearch_error.html`, `metasearch_results.html`, e `web_link.html`.

4.4 Construção do Agente para o Portal

Com base nas arquiteturas de agentes, anteriormente mencionadas, foi possível caracterizar o agente como Móvel e de Informação. Referências interessantes sobre Agentes de Informação foram obtidas de Daniel Gomes, do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, por e-mail.

Como este professor se encontrava em fase final do lançamento de um *crawler*, intitulado *Tarântula* (Sistema de Recolha de Documentos da Web), chegou-se a discutir a possibilidade de uso do mesmo para a presente dissertação.

As indicações do Prof. Daniel Gomes, levaram a *sites* específicos sobre o tema que serve de apoio à personalização do Agente de Informação: [<http://www.2.cs.cmu.edu/~rcm/websphinx/index.html>](http://www.2.cs.cmu.edu/~rcm/websphinx/index.html) e [<http://research.compaq.com/SRC/WebL/>](http://research.compaq.com/SRC/WebL/) .

Uma das propostas iniciais deste trabalho foi a utilização de multi-agentes, que constituem uma sociedade de agentes de informação, onde cada qual, individualmente, possui conhecimento sobre determinada classe de páginas (WOOLDRIDGE, 1999).

Através das ontologias, tais bases de conhecimento propiciam que os agentes identifiquem as páginas mais relevantes e que possam ser recuperadas, através dos Mecanismos de Busca, dentro de uma semântica contextualizada.

As mensagens devem referir-se a um contexto e vocabulário comuns para que a troca de mensagens possa ser efetuada dentro de uma semântica bem definida e sem ambigüidades. Esse contexto é, normalmente, fornecido pelo que é intitulado de ontologia compartilhada ou reusável. Assim, já existe tecnologia disponível para a troca de conhecimento declarativo entre agentes, permitindo cooperação entre eles na execução de suas metas comuns ou individuais, alcançando, por conseguinte, distribuição e concorrência (FREITAS, 2002).

Para a personalização do agente, conforme os objetivos do portal, adotou-se a estrutura do Agente LWP::UserAgent, uma classe da Biblioteca LWP, implementada na linguagem PERL, especificado na documentação on-line o *site*: [<http://search.cpan.org/~gaas/libwww-perl-5.69/lib/LWP/UserAgent.pm>](http://search.cpan.org/~gaas/libwww-perl-5.69/lib/LWP/UserAgent.pm).

4.5 Ontologia SWEBOK

Para que o agente desenvolvido em PERL pudesse ter uma referência na área do conhecimento focada no trabalho - Engenharia de Software, houve a necessidade de pesquisar e estudar uma ontologia adequada.

Uma das primeiras ontologias encontradas e estudadas nesta área, foi a do SWEBOK, desenvolvida por especialistas voluntários, do mundo todo, com apoio do IEEE-Computer Society e outras entidades.

O projeto do SWEBOK ainda está em desenvolvimento (SOFTWARE ENGINEERING COORDINATING COMMITTEE, 2001), abordando três fases, representadas pelas versões *Straw Man*, *Stone Man* e *Iron Man* (com duas sub-fases). Foi utilizada, neste trabalho, a versão do SWEBOK na fase *Stone Man*.

O SWEBOK pretende definir os conceitos básicos da disciplina de Engenharia de Software, obtidos através de consenso entre especialistas do mundo inteiro. O Corpo de Conhecimentos do SWEBOK inclui informações essenciais ao Engenheiro de Software, subdividido em dez Áreas de Conhecimento (*Knowledge Areas – KA*), cuja descrição permite que pesquisadores encontrem rapidamente o caminho para assuntos de seu interesse.

As principais *Knowledge Areas* pregadas pelo SWEBOK, conforme a versão 1.00, de maio de 2001, são as seguintes:

- Software Requeriments;
- Software Design;
- Software Construction;
- Software Testing;

- Software Maintenance;
- Software Configuration Management;
- Software Requirements;
- Software Design;
- Software Engineering Management;
- Software Engineering Process;
- Software Engineering Tools and Methods;
- Software Quality.

A navegação pelo Portal Colaborativo é baseada nesta estrutura de tópicos do SWEBOK, por estar muito bem documentada e determinar com bastante clareza as especificações de cada KA.

4.5.1 Ontologia ACM

Considerada a primeira Sociedade em Computação, a Association for Computing Machinery (ACM), foi fundada em 1947 e possui mais de 75.000 membros.

A ACM tem uma ontologia intitulada Computing Classification System (CCS), que apresenta uma árvore bem completa, sendo vista em <<http://www.acm.org/class/1998/ccs98.txt>>. Todos os pesquisadores ao encaminharem trabalhos para a ACM, são obrigados a classificá-los conforme este sistema.

Bonato; Ferreira (2001) afirmam que a partir desta ontologia distingue-se Nodos, Sucessores e Relacionamentos, existentes entre os assuntos. Por Nodo, entende-se um determinado assunto e por sucessor os assuntos que estão incluídos na área de conhecimento desse Nodo. Já as relações

exemplificam as ligações existentes entre Nodos que abordam os assuntos distintos.

A ACM possui uma divisão de primeiro nível (Nodo) assim constituída:

- A. General Literature
- B. Hardware
- C. Computer Systems Organization
- D. Software
- E. Data
- F. Theory of Computation
- G. Mathematics of Computing
- H. Information Systems
- I. Computing Methodologies
- J. Computer Applications
- K. Computing Milieux

Como para o intuito deste trabalho o foco está no Nodo D observa-se a seguir os Sucessores do Nodo de Software:

- D.0 GENERAL
- D.1 PROGRAMMING TECHNIQUES (E)
- D.2 SOFTWARE ENGINEERING (K.6.3)
- D.3 PROGRAMMING LANGUAGES
- D.4 OPERATING SYSTEMS (C)
- D.m MISCELLANEOUS

Destaca-se o conceito de Relacionamento, no Nodo “D.1” onde ele se associa com outro Nodo da rede, representado por “E”, sem que exista entre eles uma relação de sucessão.

Neste trabalho, foi extraído da árvore o nodo “D.2” que é a estruturação da ENGENHARIA DE SOFTWARE (Anexo B). Percebe-se a evolução da ontologia, desde 1964, com mudanças na sua estrutura. Por exemplo: o item *Rapid prototyping* era classificado como sucessor de *D.2.m Miscellaneous*, até janeiro de 1998; atualmente, é classificado como sucessor de *D.2.1 Requirements/Specifications*, como um dos assuntos incluídos em *Elicitation Methods*.

Quando o pesquisador opta pela pesquisa na ACM, os agentes fazem a busca na base *THE GUIDE TO COMPUTING LITERATURE*, que possui em seu acervo mais de 600.000 citações e cerca de 3.000 publicações. Porém, por ser uma base paga, se o pesquisador não possuir uma conexão autorizada, o acesso direto ao documento integral só será possível mediante pagamento.

4.5.2 Outras Ontologias

Existem algumas outras ontologias que devem ser ressaltadas. Uma delas é a Scientific Literature Digital Library (CiteSeer), conforme Almeida (sd) trata-se de um projeto da *NEC Research Institute* que objetiva melhorar a disseminação, recuperação e acesso à literatura científica, podendo-se dizer que tem a forma de um “índice de citação eletrônico”. Possui uma biblioteca digital que indexa aproximadamente 250.000 artigos relacionados com a área de computadores, contendo mais de 3.000.000 de referências.

E Almeida (sd) continua citando mais detalhes do CiteSeer:

Localiza artigos científicos na Web, extrai informações tais como citações, contexto da citação, título do artigo, etc; e executa uma indexação de texto completo e indexação de índice autônomo. Fornece algoritmos, técnicas e software para uso em outras bibliotecas digitais.

O *Open Directory Project* é um diretório de recursos da internet, organizado hierarquicamente por assuntos, indo do mais geral ao mais específico, semelhante a uma biblioteca de referência. O próprio Google se utiliza deste diretório como uma opção as suas pesquisas.

A COMPUTER.ORG não possui propriamente uma ontologia, mas uma base de dados intitulada *Digital Library* do IEEE *Computer Society*. Não é uma base de dados gratuita, mas de acesso aos seus assinantes ou mediante pagamento, por texto, disponibilizado na íntegra.

4.6 Visão Geral

Conforme a Figura 11, pode-se ver a estrutura básica de pesquisa dos agentes na internet. O primeiro ponto em destaque é que, por mais que se queira vasculhar a Web, como um todo, existem "buracos negros" que nem os mais poderosos Mecanismos de Busca conseguem acessar, às vezes por questões de segurança, outras por questão de privacidade, e ainda até por problemas técnicos.

Outra conclusão que se tira dessa imagem, é a sobreposição de conteúdo. Ou seja, alguns repositórios têm uma abrangência maior, e outros menores, mas normalmente existe algo em comum entre as bases de dados. Pode-se destacar que o Google é, hoje, o mecanismo de busca que consegue indexar o maior número de endereços na internet. A biblioteca digital da ACM, por ser um conteúdo especializado e pago, tem suas limitações.

Finalmente, o *CiteSeer* tem a pretensão de apresentar um conteúdo técnico, de alto nível, com abrangência maior que a ACM, com a vantagem de fornecer acesso gratuito.

No Portal Colaborativo, o pesquisador tem a possibilidade de iniciar a busca em uma das duas bases de dados. O resultado da busca está sempre associado à base pesquisada, podendo ser realizada nas outras bases, sempre que o pesquisador assim o desejar.

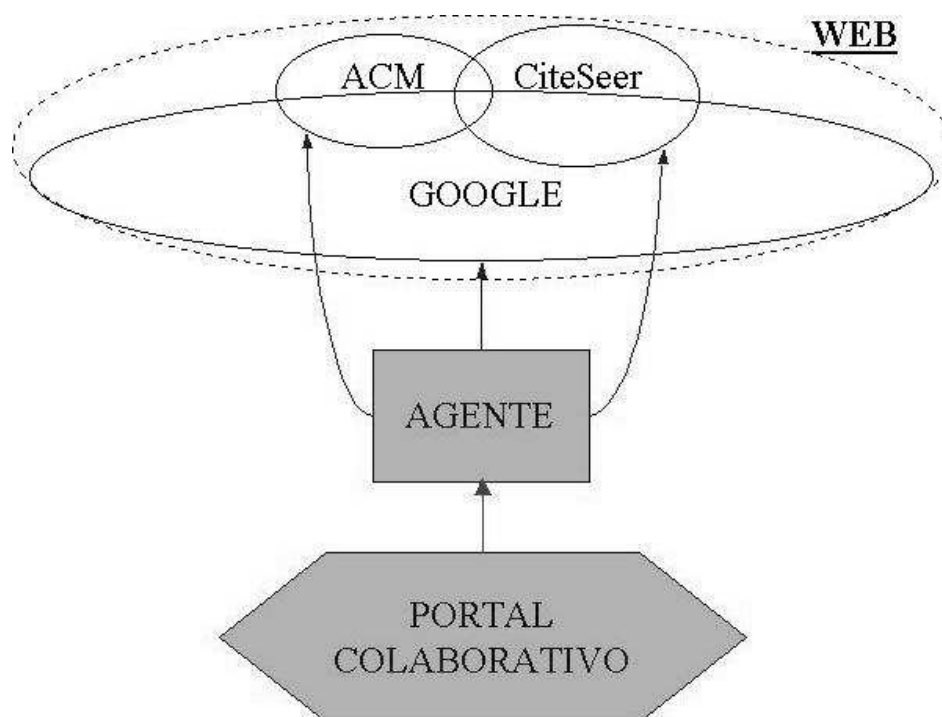


Figura 11: Estrutura básica dos agentes.

4.6.1 Detalhes do Funcionamento dos Agentes

Detalha-se o funcionamento básico do agente, e como a linguagem PERL apóia esse processo. Conforme a Figura 12, podem-se visualizar seis etapas estratégicas para o correto funcionamento do agente.

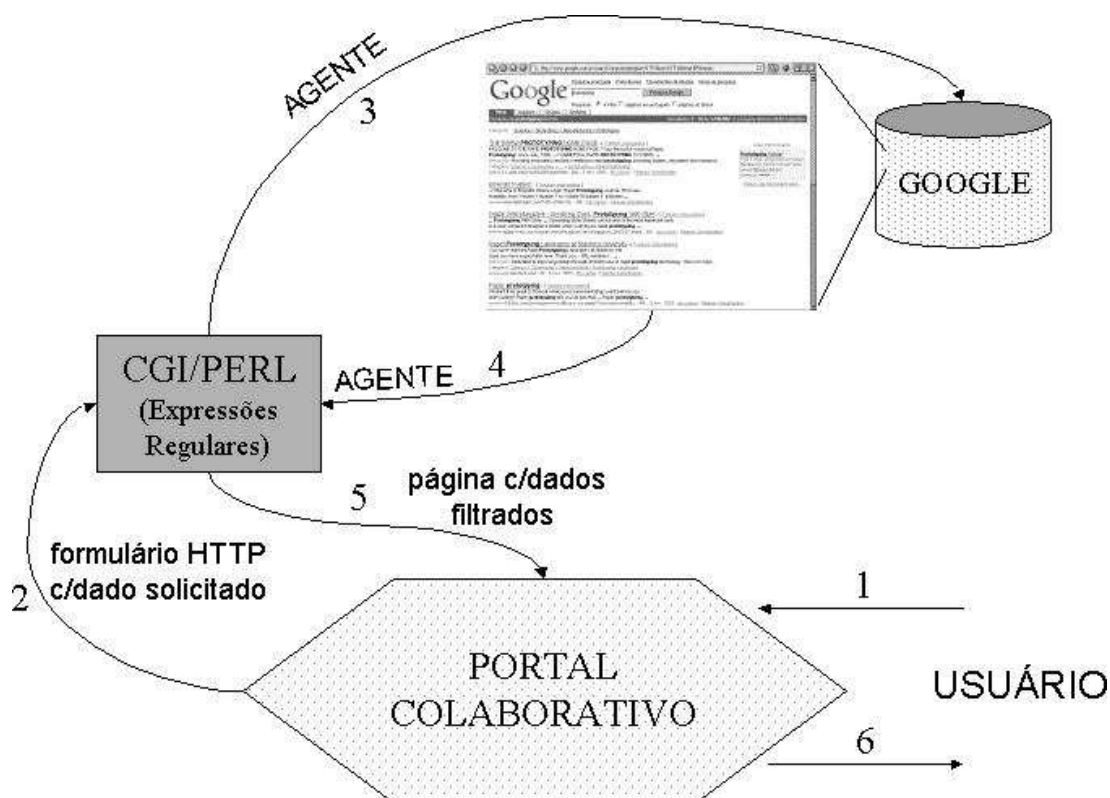


Figura 12: Detalhes básicos do agente.

Na primeira etapa, o usuário do portal, tipicamente pesquisador, lança uma palavra-chave, que pode ser um termo da ontologia do SWEBOK, escolhendo em que base irá pesquisar (na Web ou na ACM).

Na segunda etapa, são encaminhados, via formulário HTTP para o *Host*, os dados selecionados ou digitados pelo usuário. Trabalhando-se com o conceito Cliente/Servidor, a estação Cliente é aquela onde o usuário estará utilizando um *browser*, e o Servidor a máquina que hospeda o domínio *novainter.net*. Ao clicar no botão de acionamento do agente, são enviados os dados para o servidor via interface CGI.

Na fase 3 é lançado, pelo programa em PERL, um agente personalizado que vai pesquisar a base desejada. A título de exemplo, foi escolhida a base do Google. Estudando o funcionamento do Mecanismo de Busca do Google,

surgiu a oportunidade de fazer um primeiro filtro nesta etapa, utilizando recursos de sintaxe, montados em tempo real, na URL de pedido (Anexo A).

No momento seguinte, na etapa 4, encontra-se a resposta do próprio Google a URL enviada. O Mecanismo de Busca gera uma página em HTML com os dados pedidos, que o agente encaminha para o servidor. De posse desta página no servidor da *novainter-net*, o programa em PERL começa a trabalhar utilizando o mecanismo de Expressões Regulares para adequar e filtrar estrategicamente a solicitação do usuário do portal. Alguns critérios foram adotados, para se obter uma página nova, mais adequada ao perfil dos pesquisadores.

Na etapa 5, gera-se uma página em HTML para a estação cliente, adotando-se uma formatação o mais simples e intuitiva possível. O número de *links* apresentados é variável, em função da quantidade de filtros que forem atuar na página original do Google. Finalmente, na última etapa, apresentam-se para o usuário, os resultados de busca.

Na Figura 13, existe um aprofundamento dos detalhes mencionados anteriormente. Destaca-se a relação Cliente/Servidor entre as máquinas envolvidas, os dois principais softwares que estão instalados no Provedor *Novainter-net* (*wiki.pl* e *meta.cgi*), os aplicativos que são utilizados dentro do Portal Colaborativo (Blog, Correio Eletrônico, Chat), o método utilizado nas páginas solicitantes em HTML, e a estratégia de inserir na URL, os dados do primeiro filtro encaminhados para o Google.

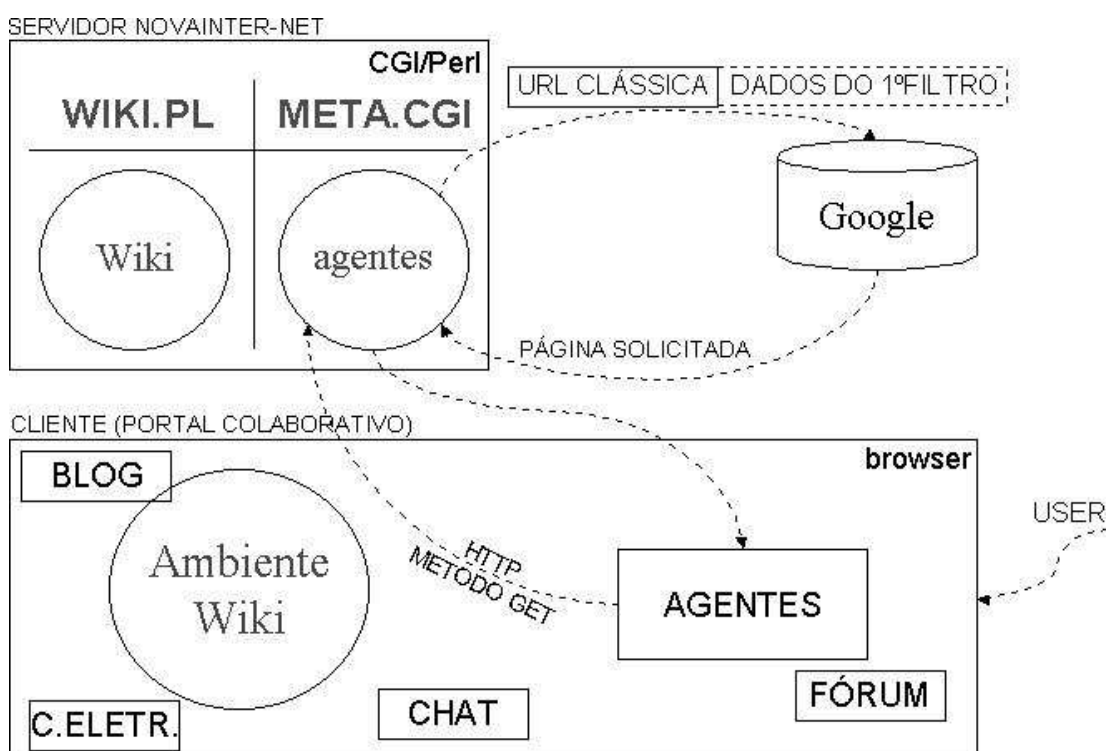


Figura 13: Visão detalhada dos principais programas usados no Portal Colaborativo.

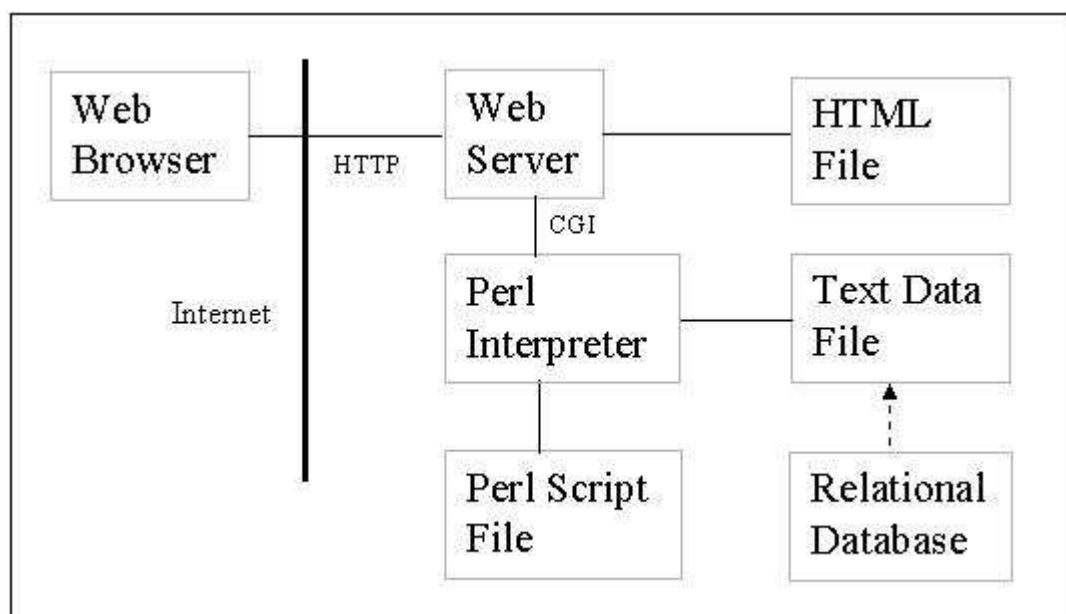


Figura 14: Comparativo com uma aplicação CGI típica. [online]. Disponível em: <http://developer.netscape.com/viewsource/lazar_cgi.html>.

4.6.2 Processos do Modelo

Para se visualizar todo o fluxo de informações e dados que circulam neste arcabouço, detalha-se a seguir, numa visão *top-down*, cada processo envolvido (vide Figura 4):

1. COMUNIDADE DE COLABORADORES ACESSANDO O PORTAL: os usuários participam basicamente de duas formas. De forma passiva (navegando pelo portal) ou de forma ativa (enviando mensagens para o Líder do PORTAL, alterando conteúdos específicos, através do Wiki, comunicando-se via Chat com outros colaboradores).

De qualquer forma, o usuário deixará, indiretamente, registros no sistema de como foi a sua navegação (base de dados de visita), ou diretamente, colocando suas experiências, dúvidas e sugestões no portal.

Por qualquer um destes caminhos, o Líder do PORTAL poderá, através do sistema, analisar e ponderar a maneira como foi feita a navegação e realimentará, ou mesmo poderá redefinir a estrutura do portal.

2. UTILIZAÇÃO DA BASE DE DADOS DE VISITA: praticamente é realizado um *WebMining* do portal. Devido ao registro da navegação do usuário aí nesta base, obtêm-se, estatisticamente: as principais páginas de interesse, principais termos que os usuários procuram, e mapeamento de como o usuário típico navega pelo portal. Graças a tais informações, o Líder pode ajustar o portal para atender melhor a COMUNIDADE.

3. ALIMENTAÇÃO DO PORTAL: através do binômio CGI e PERL, o portal pode ser atualizado automaticamente a partir do ambiente colaborativo Wiki. Isto é o que permitirá o dinamismo do portal, mantendo-o sempre o mais atualizado possível. Todos os personagens podem, com os graus variados de intervenção, contribuir para sua atualização.

4. ALIMENTAÇÃO/ATUALIZAÇÃO DO PORTAL COLABORATIVO: neste processo o Líder do PORTAL é o principal responsável em manter uma estrutura consistente. Além de dialogar com o COMITÊ de especialistas, ponderando o que será necessário para o aprimoramento do Portal, o mesmo possui várias funções técnicas/administrativas de vital importância em todo o processo. Com base no Agente de Informação (MULTI AGENTES) seleciona o material mais adequado para atualização do Portal. E com os Dados de visita, refina a estruturação do Portal, e informa ao COMITÊ de novas estratégias a serem tomadas. O sistema possui capacidade de aprendizagem a partir do feedback automático dado pelos utilizadores sobre a relevância dos documentos.

5. ALIMENTAÇÃO/ATUALIZAÇÃO DAS ONTOLOGIAS: embora tendendo a sofrer poucas alterações, o Líder do PORTAL deverá sempre estar atento aos órgãos internacionais, detentores de ONTOLOGIAS, para alimentá-las e atualizá-las adequadamente.

6. ESTRUTURAÇÃO DO AGENTE DE INFORMAÇÃO: contando com a experiência pessoal na área, com as informações do COMITÊ e informações dos Usuários, o Líder do PORTAL cria estratégias especiais para personalizar os Agentes de Informação. Neste processo incluem-se: relação das principais palavras-chave da área de estudo, uma lista de *sites* (URL's) que são importantes, inseridas na área de conhecimento, seleção das línguas e países que serão pesquisados, determinação do grau de relevância, profundidade e periodicidade das pesquisas.

Em função das ONTOLOGIAS, o Agente de Informação possui, como se fosse um roteiro, os *sites* (URL's) que visitará na WEB, o grau de profundidade com o qual deverá realizar o seu trabalho, e o período. Ou seja, as ONTOLOGIAS determinarão o escopo do Agente de Informação.

Este modelo possui características de feedback automático, permitindo que haja melhora contínua na qualidade e abrangência das informações disponibilizadas. Para isso, depende da participação de voluntários que contribuam colaborativamente, utilizando-se da internet e de uma ferramenta simples (*browser*).

4.7 Decisões Técnicas Tomadas e Justificativas

A título de justificar, de forma sintética, as várias tomadas de decisão, quanto a escolhas das ferramentas computacionais adotadas, na construção do Portal Colaborativo, destaca-se no quadro sinóptico abaixo, um resumo de das decisões:

FERRAMENTAS COMPUTACIONAIS	JUSTIFICATIVAS
Linguagem PERL	Embora possam ser utilizados outros tipos de linguagens no desenvolvimento do Portal, como PHP, escolheu-se o PERL. Isto acontece por que as principais ferramentas adotadas possuem maior disponibilidade de Software Livre, desenvolvido nesta linguagem. Na internet, o par casado PERL e CGI, é o que possui maior riqueza de softwares.
WIKI	Foi escolhido o ambiente UseModWiki pois além de ter sido criado originalmente em PERL, é um dos ambientes Wiki, com código livre, mais utilizado na Web.
Agente LWP::UserAgent	Adotou-se este agente, por ser a biblioteca LWP um padrão, na comunidade dos desenvolvedores PERL.
Ontologia SWEBOK	Das ontologias utilizadas no Portal, a do SWEBOK se destaca por ser a mais bem trabalhada, por vários especialistas mundiais em Engenharia de Software, e com documentação disponível na internet.
Ferramentas Síncronas e Assíncronas	No desenvolvimento do Portal, pelo fato de ter sido implementado em PERL, optou-se preferencialmente por ferramentas desenvolvidas nesta linguagem. No entanto, por serem

	ferramentas de apoio e suporte ao Portal, podem ser adotadas as que o desenvolvedor tiver mais familiaridade, sem prejudicar a filosofia do modelo proposto.
--	--

5 CONCLUSÕES FINAIS

Com a pesquisa bibliográfica, e os experimentos desenvolvidos, verificou-se que é factível a montagem de Portal Colaborativo, para um público especializado, conforme a proposição original.

Do ponto de vista técnico, os recursos necessários para atender tal modelo, são relativamente fáceis de encontrar na internet, sendo os principais programas adaptáveis às necessidades específicas de um Portal especializado.

Verificou-se, também, que do ponto de vista financeiro, o modelo é viável, por basear-se em softwares *open-source*. Os custos mensais são de pouca monta, tendo em vista que os valores aplicados atualmente na hospedagem de um provedor na internet são ínfimos; inclusive podendo o portal ser hospedado num servidor patrocinado por terceiros.

Como vantagem adicional, o modelo pode ser aplicado facilmente a qualquer área do conhecimento humano, bastando pesquisar e adaptar uma ontologia específica àquela área, ou mesmo criá-la, no caso de sua ausência.

Na construção de um Portal Colaborativo, exigem-se profissionais com conhecimento especializado nas ferramentas mencionadas neste arcabouço. E o uso, pela comunidade de interesse, é bem simples, sendo necessário, basicamente, o conhecimento do uso de um *browser* padrão, e a navegação pela Web.

No fechamento deste trabalho, teve-se acesso a algumas tecnologias computacionais correlatas, com a proposta desta dissertação. Por serem tecnologias ainda recentes não foram devidamente pormenorizadas. Portanto, podem-se citar os seguintes trabalhos paralelos:

- MOODLE: ambiente GMS contendo Wiki, e com possibilidade de três tipos de desenvolvimento;
- ORKUT: para o desenvolvimento de comunidades;
- PLONETARIUM: ambiente para o desenvolvimento de portais comunitários e colaborativos da incubadora virtual da FAPESP;
- DEBIAN: controladora de Software Livre que possui modelo semelhante ao apresentado nesta dissertação, quanto ao controle de pacotes de software e mantenedores.

5.1 Desdobramentos Futuros

Como a pretensão modesta deste trabalho era demonstrar a viabilidade da construção de um Portal Colaborativo, com o uso de ferramentas *open-source* disponíveis na internet, e o uso de ontologias conhecidas na área de Engenharia de Software, podem decorrer no futuro os possíveis trabalhos:

- Maior exploração das ontologias, permitindo a interação com os pesquisadores, criando novas formas de ontologias, e atualizando-as conforme o avanço da tecnologia e do próprio conhecimento;
- Com base no item anterior, seria factível a criação de uma ontologia específica de Engenharia de Software, em língua portuguesa, fomentando pesquisas nacionais;
- Após o período de maturação do Portal, captar as várias contribuições dos pesquisadores, detectando os pontos positivos e negativos, criticando o modelo proposto neste trabalho e aprimorando-o;

- Evoluir tecnicamente o Portal, com ferramentas mais complexas, analisando inclusive a adoção de um Banco de Dados de maior porte. Com isso, poder-se-ia alterar a estrutura dos registros, ampliando o armazenamento das informações, contribuindo para os pesquisadores quanto à atualização e comunicação via e-mail, e no andamento em suas áreas de estudo.

Referências

ALMEIDA, M.B. sd. CiteSeer – Universidade Federal de Minas Gerais. [online]. Disponível em: <<http://www.mbalmeida.pro.br/trab1.html>>. Acesso em: 20dez.2003.

AL-OMAIM, N. 1997. What is *Groupware*? [online]. Disponível em: <http://www.usfca.edu/fac-staff/morriss/478/projects_972/webdoc4.htm>. Acesso em: 17mar.2003.

ARASU, A; CHO, J.; GARCIA-MOLINA, H.; PAEPCKE A.; RAGHAVAN, S. Searching the Web. Computer Science Department, Stanford University, 2000. [online]. Disponível em: <<http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en&doc=2000-37&format=pdf&compression=>>>. Acesso em: 6jun.2004.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. Scientific American, maio 2001. Disponível em: <<http://www.scientificamerican.com/2001/0501issueberners-lee.html>> Acesso em: 8jan.2003.

BONATTO, D.T.; FERREIRA, M. da S. 2001. Redes de conhecimento compartilhado. In: OFICINA DE INTELIGÊNCIA ARTIFICIAL, 5, 2001, Pelotas, RS. [online]. Disponível em: <http://gpia.ucpel.tche.br/~daniel/docs/cricte_2002.pdf>. Acesso em: 23jun.2003.

CANDOTTI, C.T.; HOPPEN, N. 1999. Reunião virtual e o uso de *Groupware* - uma nova possibilidade de realizar trabalho em grupo. ENANPAD.

CANHOS, D.A.L. sd. Biodiversidade: sistemas de informação: conceito, infra-estrutura e política. [online]. Disponível em: <<http://www.bdt.fat.org.br/publicacoes/padct/bio/cap2/3/dora.html>>. Acesso em: 27jul.2004.

CARVALHO, R.B. ; FERREIRA, M.A.T. 2000. Análise de software de gestão do conhecimento. XXI Simpósio de gestão da inovação tecnológica. São Paulo.

CENDÓN, B.V. Ferramentas de Busca na Web - Escola de Ciência da Informação da Universidade Federal de Minas Gerais – Ci. Inf., Brasília, v.30, n.1, p. 39-49, jan/abr.2001.

CHU, H.; ROSENTHAL, M. 1996. Search engines for the World Wide Web: a comparative study and evaluation methodology. In: ASIS 1996 ANNUAL CONFERENCE, 1996, New York. Proceedings. New York: ASIS. [online].

Disponível em: <<http://www.asis.org/annual-96/EletronicProceedings/chu.html>>. Acesso em: 22abr.2003.

CINTRA, G. 2001. Programação CGI usando a linguagem PERL. [online]. Disponível em: <<http://www.ime.usp.br/~glauber/perl/>>. Acesso em: 3mar.2004.

CUNNINGHAM, W.; LEUF, B. 2001: The Wiki Way. Quick Collaboration on the Web. Addison-Wesley.

ELIOT, T.S. 1934. The Rock - A Pageant Play. London: Faber & Faber.

FILIPO, R. 2000. Curso de PERL-básico. [online]. Disponível em: <<http://www.gulux.unerj.br/apostilas/PERLBasico.pdf>>. Acesso em: 5set.2003.

FOX, G. 2001. Architecture and Implementation of a Collaborative Computing and Education Portal. Florida State University.

FREITAS, F.L.G. de. 2002. Sistemas multi-agentes cognitivos para a recuperação, classificação e extração integradas de informação da Web. s.l.: UFSC.

FUKUYAMA, F. 1996. Trust: the social virtues and the creation of prosperity. sl: Free Press.

GILBERT, D. et al. 1999. The role of intelligent agents in the information infrastructure. 9p. [online]. Disponível em: <<http://www.ajrhem.com/iagents.pdf>>. Acesso em: 13dez.2003.

GUELICH, S.; GUNDAVARAM, S.; BIRZNIKES, G. 2001. Programação CGI com PERL. Rio de Janeiro: Ciência Moderna.

HARRISON, C.; CHESS, D.; KERHENBAUM, A. 1995. Mobile agents: are they a good idea? s.l.: IBM.

IVANOFF, G.B.; OqueEUmWiki, 2003. [online]. Disponível em: <<http://www.ilanet.com.br/cgi-local/twiki/bin/view/Main/OQueEUmWiki>>. Acesso em: 11jul.2004.

JARGAS, A.M. Expressões Regulares, 2003. [online]. Disponível em: <<http://guia-er.sourceforge.net/guia-er.html>>. Acesso em: 26jun.2004

LÉVY, P. 1999. A Inteligência Coletiva: por uma Antropologia do Ciberespaço. 2ed. São Paulo, Loyola.

MAES, P. 1994. Agents that reduce work and information overload. *Communications of the ACM*, Cambridge: MIT, Media Laboratory, v.37, n.7, p.31-40, Jul.

MURRAY, G. 1999. *The portal is the desktop*. Intraspect, May/June. [online]. Disponível em: <<http://www.e-promag.com/eparchive/index.cfm?fuseaction=viewarticle&ContentID=166&websiteid=>>>. Acesso em: 12mar.2003.

NAGAMACHI, J. 2000. Formulários e CGI-BIN - Alguns conceitos importantes. [online]. Disponível em: <<http://socrates.if.usp.br/~jngamac/form/>>>. Acesso em: 21jul.2003.

NAVEGA, S.C. 2002. Projeto Cyc: confundindo inteligência com conhecimento. In: WORKSHOP BRASILEIRO DE INTELIGÊNCIA COMPETITIVA E GESTÃO DO CONHECIMENTO, 3. 2002. São Paulo; CONGRESSO ANUAL DA SOCIEDADE BRASILEIRA DE GESTÃO DO CONHECIMENTO, 1. 2002. São Paulo. Anais. 19p. [online]. Disponível em: <<http://www.intelliwise.com/reports/kmbscn.pdf>>. Acesso em: 21ago.2004.

NWANA, H. 1996. Software agents: an overview. *Knowledge Engineering Review*, v.11, n.3, p.1-40.

RAYMOND, E.S. 2000. The Cathedral and the Bazaar. [online]. Disponível em: <<http://catb.org/~esr/writings/cathedral-bazaar/>>>. Acesso em: 22jun.2003.

SAAB, W.G.L. 2000. E-Business: o uso corporativo da internet. 6p. [online]. Disponível em: <<http://www.bndes.gov.br/conhecimento/setorial/get4is15.pdf>>. Acesso em: 22jun.2003.

SCHEER, S. Tópicos especiais em Educação à Distância. In: IAHN, L.F.; MARTINS, O.B.; POLAK, Y.N. de S. Capacitação de tutores na modalidade de Educação à Distância. Curitiba: Apta, 2002.

SEVERINO, A.J. 2000. Metodologia do trabalho científico. 22ed. São Paulo: Cortez.

SILVA, A. R.; DELGADO, J. AgentSpace versus Aglets: Infraestruturas de Agentes para as Futuras Aplicações da Internet. (SBC – SEMISH'98, Brasil, Belo Horizonte), *Anais do 18º Congresso da Sociedade Brasileira de Computação – Rumo à Sociedade do Conhecimento*, 1998.

SHANNON, C. 1948. A mathematical theory of communication. *Bell Systems Technical Journal*, v.27, p.379-423; 623-656.

SOFTWARE ENGINEERING COORDINATING COMMITTEE. 2001. Guide to the Software Engineering Body of Knowledge - SWEBOK. IEEE Computer Society.

TERRA, J.C.C.; GORDON, C. 2002. Portais corporativos: a revolução na gestão do conhecimento. São Paulo: Negócio.

TUTHIL, G.S. 1990. Knowledge engineering: concepts and practices for knowledge-based systems. s.l.: Tab Books Inc.

URRUTIA, G.J.A. 2002. CGI/PERL páginas dinâmicas com banco de dados fácil de aprender. São Paulo: Érica.

VILLAS BÔAS, A.A. et al. 2001. CGI com PERL: conceitos, programação e aplicação. São Paulo: Érica.

WOOLDRIDGE, M. 1999. Intelligent agents. In: WEISS, G. (Ed.). Multiagent systems: a modern approach to distributed artificial intelligence. s.l.: MIT Press.

ZAINA, L.A.M.; BRESSAN, G.; RUGGIEKO, W. 2002. Aplicação das ferramentas interativas na construção do conhecimento em cursos à distância. [online]. Disponível em: <<http://www.asee.org/international/INTERTECH2002/582.pdf>>. Acesso em: 11jun.2004.

Referências consultadas

ABRAN, A.; MOORE, J.W. (Eds.). Swebok: guide to the software engineering body of knowledge: trial version; version 1.000. [online].

Disponível em:

<http://www.Swebok.org/stoneman/version_1.000/SWEBOK_w_correct_copyright_web_site_version.pdf>.

Acesso em: 10out.2003.

AMSTRONG, R. et al. 1995. Webwatcher: A learning apprentice for the World Wide Web. In: SYMPOSIUM ON INFORMATION GATHERING FROM HETEROGENEOUS, DISTRIBUTED ENVIRONMENTS, 1995. Proceedings. s.l.: AAAI Press.

BALABANOVIC, M.; SHOHAM, Y.; YUM, Y. 1995. An adaptative agent for automated web browsing. Journal of Image Representation and Visual Communications, v.6.

BATES, J. 1994. The role of emotion in believable agents. Communications of ACM, v.37, n.7, p.122-125.

BATES, M.E. Super searchers do business: the online secrets of top business researchers. s.l.: CyberAge? Books, sd. 206p.

BAUMANN, J.; TSCHUDIN, C.; VITEK, J. 1996. In: ECOOP WORKSHOP ON MOBILE OBJECT SYSTEMS, 2. 1996. Linz, Australia. Proceedings. Dpunkt.

BLATTMANN, U.; FACHIN, G.R.B.; RADOS, G.J.V. Recuperar a informação eletrônica pela internet. [online]. Disponível em:

<<http://www.ced.ufsc.br/~ursula/papers/buscanet.html>>. Acesso em: 8jun.2000.

BOND, A.; GASSER, L. 1988. An analysis of problems and research in DAI. Distributed artificial intelligence. s.l.: Morgan Kaufmann.

BURKLE, P. 1995. The distributed asynchronous scheduler. In: ZWEBEN, M.; FOX, M. Intelligent scheduling. s.l.: Morgan Kaufmann.

CALISHAIN, T.; DORNFEST, R. 2003. Google hacks: 100 industrial-strength tips and tricks. Sebastopol, CA., O'Reilly. [online]. Disponível em: <<http://novainter.net/arq/GoogleHacks.pdf>>. Acesso em: 1fev.2004.

CAMPOS, E.; TEIXEIRA, F.L.C. 2001. A adoção da Tecnologia de Informação nas Organizações: uma proposta de análise para o "Modelo de Estruturação da Tecnologia", através da implementação de Sistemas de *Groupware*. In: Resumo dos Trabalhos XXV Enanpad. Rio de Janeiro: Anpad.

- CARDELLI, L. 1994. Obliq: a language with distributed scope. s.l.: Digital Equipment Corporation, Systems Research Center. (Digital White Paper).
- CHU, H. 1997. Internet search tools: what can they offer to users?: database access. In: NATIONAL ONLINE MEETING, 18., 1997, New York. Proceedings. Medford: Information Today. p.73-80.
- CIANCARINI, P.; TOLKSDORF, R.; VITALI, F. 1996. PageSpace?: an architecture to coordinate distribute applications on the web. Computer Networks and ISDN Systems, v.28, p.941-952.
- DAVIES, J.; WEEKS, R. 1995. Jasper: communicating information agents. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 4, 1995.
- DEMAZEAU, Y.; MULLER, J. (Eds.). 1990. Decentralized artificial intelligence. s.l.: Elsevier.
- ETIZIONI, O.; WELD, D. 1994. A softbot-based interface to the internet. Communications of the ACM, v.37, n.7, p.72-76, Jul.
- FELT, E. Analysis of web robots. List and Analysis of Web Robots by Elizabeth Felt and Jane Scales at Washington State University. [online]. Disponível em: <<http://www.wsulibs.wsu.edu/general/robots.htm>>. Acesso em: 30nov.2003.
- FENSEL, D. 2000. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin.
- FIRBY, R.; HANKS, S. 1987. A simulator for mobile robot planning. In: DARPA KNOWLEDGEBASED PLANNING WORKSHOP, 1987.
- FRANKLIN, S.; GRAESSER, A. 1996. Is it an agent, or just a program?: a taxonomy for autonomous agents. In: INTERNATIONAL WOKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 3. 1996. Proceedings. Springer-Verlag. [online]. Disponível em: <<http://www.msci.memphis.edu/~franklin/AgentProg.html>>. Acesso em: 22jan.2004.
- GARMAN, N. 1999. Meta search engines. Online, v.23, n.3, p.75-78, May/Jun.
- GENESERETH, M.; KETCHPEL, S. 1994. Software agents. Communications of the ACM, v.37, n.7, p.48-53, Jul.
- GRAY, R. 1995. Agent Tcl: a transportable agent system. In: CIKM WORKSHOP ON INTELLIGENT INFORMATION AGENTS, CIKM'95, 1995.

HAHN, T.B. 1998. Text retrieval online: historical perspective on Web Search Engines. *Bulletin of the American Society for Information Science*, v.24, n.4, p.7-10, Apr./May, 1998.

HALVORSON, T.R. 1999. *Law of the super searchers: the online secrets of top legal researchers*. s.l.: Cyberage Books. 360p.

HERMANN, E. 1977. *Aprenda em uma semana programação CGI com PERL 5*. Rio de Janeiro: Campus.

HEWITT, C. 1977. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, v.8, n.3, p.323-364.

HOCK, R. 1999. Web search engines: features and commands. *Online*, v.23, n.3, p.24-28, May/Jun.

INTRODUÇÃO à linguagem PERL. In: *Desenvolvendo Web sites interativos com CGI*. Capítulo 5, 46p. [online]. Disponível em: <http://www.argonavis.com.br/cursos/web/w500/cgi_02.pdf>. Acesso em: 19fev.2004.

JENNINGS, N. 1992. On being responsible. In: *EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTS AND MULTI-AGENT WORLDS*, 3. 1992. Proceedings.

KIMMEL, S. 1997. WWW search tools in reference services. *The Reference Librarian*, v.57, p.5-20.

KOCH, T. et al. 1996. The building and maintenance of robot based internet search services: A review of current indexing and data collection methods. (Prepared by Work Package3 of EU Telematics for Research, project DESIRE). [online]. Disponível em: <<http://www.lub.lu.se/desire/radar/reports/D3.11v0.3/index.html>>. Acesso em: 12abr.2004.

KOZIEROK, R.; MAES, P. 1993. A learning interface agent for scheduling meetings. In: *ACM SIGCHI INTELLIGENT WORKSHOP ON INTELLIGENT USER INTERFACES, 1993*, Orland, Florida. Proceedings. New York: ACM Press. p.81-88.

LABROU, Y.; FINN, T. 1994. A semantic approach for KQML: a general purpose communication language for software agents. In: *INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, CIKM'94*, 3. 1994. Proceedings. [online]. Disponível em: <<http://www.cs.umbc.edu/kqml/papers/cikm94.pdf>>. Acesso em: 23set.2003.

LANDES, D.S. 1998. *The wealth and poverty of nations: why some are rich and some are poor*. s.l.: W.W. Norton.

- LENAT, B. 1995. CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, v.38, n.11, p.33-38, Nov.
- LIMA, C.M. O que é a internet e como utilizá-la para pesquisa? [online]. Disponível em: <<http://www.elo.com.br/~cynthia/interpesq.htm>> Acesso em: 9dez.2002.
- NAGOA, K.; TAKEUCHI, A. 1995. Social interaction: multimodal conversation with social agents. s.l.: s.c.p.
- NORA, P. et al. 1999. Great scouts!: cyberguides for subject searching on the web. s.l.: Cyberage Books. 200p.
- NORMAN, D. 1994. How might people interact with agents. *Communications of the ACM*, v.37, n.7, p.68-71, Jul.
- O'BRIEN, P.; WIEGAND, M. 1996. Agents of change in business process management. *BT Technology Journal*, v.14, n.4.
- OTSUKA, J.L. sd. Fatores determinantes na efetividade de ferramentas de comunicação mediada por computador no Ensino à Distância. [online]. Disponível em: <http://penta.ufrgs.br/pesquisa/joice/joice_ti.html#sumula>. Acesso em: 17mar.2004.
- POULTER, A. The design of World Wide Web search engines: a critical review. *Program*, v.31, n.2, p.131-145, Apr.1997.
- PRESSMAN, R. 2002. Engenharia de software. 5ed. Rio de Janeiro: Makron Books. 843p.
- RAN hock's extreme searcher's guide to web search engines: a handbook for the serious searcher. 2ed. s.l.: A CyberAge? book from Information Today, 2001.
- RAO, A.S.; GEORGEFF, P. 1995. BDI agents: from theory to practice. In: INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, 1. 1995. San Francisco. Proceedings. (Technical Note 56). [online]. Disponível em: <<http://www.cs.nott.ac.uk/~mhl/archive/Rao+Georgeff:95a.pdf>>. Acesso em: 12jan.2004.
- REZENDE, S.O. (Org.). 2003. Sistemas inteligentes: fundamentos e aplicações. Barueri, SP: Manole.
- ROSENSCHEIM, G.; ZLOTKIN, J. 1994. Designing conventions for automated negotiation. *AI Magazine*, v.15, n.3, p.29-46.
- SCHLEIN, A.M. 1999. Find it online: the complete guide to online research. Tempe: Facts on Demand Press. 506p.

- SCHWARTZ, C. 1998. Web search engines. *Journal of the American Society for Information Science*, v.49, n.11, p.973-982.
- SEARCH, J. 1969. *Speech acts*. s.l.: Cambridge University Press.
- SHERMAN, C. 1999. The future of Web search. *Online*, v.23, n.3, p.54-61, May/Jun.
- SHOHAM, Y. 1993. Agent oriented programming. *Artificial Intelligence*, v.60, n.1, p.51-92.
- SICHMAN, J. et al. 1994. A social reasoning mechanism based on dependence networks. In: EUROPEAN CONFERENCE OF ARTIFICIAL INTELLIGENCE, ECAI94, 11. 1994. [online]. Disponível em: <<http://citeseer.nj.nec.com/cache/papers/cs/1089/http:zSzzSzwww-leibniz.imag.frzSzMAGMAzSzPAPERSzSzSzsichman.ecai94.e.pdf/sichman94social.pdf>>. Acesso em: 21jul.2004.
- STRASSER, M.; BAUMANN, J.; HOHL, F. 1996. Mole: a zava-based mobile object system. In: ECOOP WORKSHOP ON MOBILE OBJECT SYSTEMS, 2. 1996. Linz, Australia. Proceedings. Dpunkt.
- SULLIVAN, D. (Ed.). *Search engine watch: tips about internet search engine*. [online]. Disponível em: <<http://www.searchenginewatch.com>>. Acesso em: 9jun.2000.
- SULLIVAN, D. 1999. Crawling under the hood: an update on search engine technology. *Online*, v.23, n.3, p.30-38, May/Jun.
- TERRA, J.C.C. 2000. *Gestão do conhecimento: o grande desafio empresarial; uma abordagem baseada no aprendizado e na criatividade*. São Paulo: Negócio.
- TERRA, J.C.C. 2001. A criação de portais corporativos de conhecimento. In: SIMPÓSIO INTERNACIONAL DE GESTÃO DO CONHECIMENTO E GESTÃO DE DOCUMENTOS. 2001. Curitiba. Anais. Curitiba: PUC-PR/CITS.
- VERE, S.; BICKMORE, T. 1990. A basic agent. *Computational Intelligence*, v.6, p.41-60.
- WELLMAN, M. Wellman. 1993. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of AI Research*, n.1, p.51-92.
- WERNER, E.; DEMAZEAN, Y. (Eds.). Northland Pub. p.93-102. [online]. Disponível em: <<http://www.ecs.soton.ac.uk/~nrj/pubs.html>>. Acesso em: 4abr.2004.

WHAT is the internet? [online]. Disponível em:
<<http://www.walthowe.com/navnet/faq/internet.html>>. Acesso em:
22nov.2003.

WHITE, J. 1994. Telescript technology: the foundation for the electronic marketplace. s.l.: General Magic White Paper, General Magic.

WOOLDRIDGE, M.; JENNINGS, N.1995. Intelligent agents: theory and practice. Knowledge Engineering Review, v.10, n.2, p.115-152.

Anexos

A. Hacking Google URLs

Conforme o livro *Google Hacks* (CALISHAIN, T.; DORNFEST, R. 2003), tem-se complexa sintaxe na URL de pedido de pesquisa ao Google. Destacam-se as principais utilizadas nos experimentos (algumas não citadas no livro, mas com base nos estudos deste trabalho):

as_epq=query

delimita-se com chaves o texto digitado para pesquisa, aumentando a precisão.

ir=lang_en

opta-se por escolher somente material escrito em língua inglesa.

safe=off

elimina-se material pornográfico.

start=\$page

controla-se a variável da página de início.

sa=N

num=\$results

numeram-se resultados encaminhados por vez, ao ser consultada a Base de Dados.

filter=0

as_qdr=m6

especifica-se o tempo da pesquisa, filtrando material mais recente e atualizado, ao adotar a variável sugerida pelo livro de 6 meses.

as_occt=body

determina-se que a pesquisa deve ser realizada com base no texto inserido no corpo das páginas dos *sites* cadastrados, e não no título (existe o costume, não ético, de colocar palavras-chaves mais freqüentemente utilizadas na Web no título da página, para aumentar o nível de audiência da mesma).

OBS.: Utilizou-se um dos métodos do HTTP. Atualmente, dois métodos são empregados para submeter dados de formulários: POST e GET. No caso, o método GET permitiu que os dados de entrada do script façam parte da URL, associados à consulta enviada para o servidor.

A cadeia de entrada é colocada na variável de ambiente QUERY_STRING, com um limite de 128 caracteres. Como os dados da consulta fazem parte da URL, pode ela ser encapsulada em uma URL, para o *link* do Google.

B. Relação da Ontologia de Engenharia de Software da ACM

D.2 SOFTWARE ENGINEERING (K.6.3)

D.2.0 General (K.5.1)

Protection mechanisms

Standards

D.2.1 Requirements / Specifications (D.3.1)

Elicitation methods (e.g., rapid prototyping, interviews, JAD) (NEW)

Languages

Methodologies (e.g., object-oriented, structured) (REVISED)

Tools

D.2.2 Design Tools and Techniques (REVISED)

Computer-aided software engineering (CASE)

Decision tables

Evolutionary prototyping (NEW)

Flow charts

Modules and interfaces

Object-oriented design methods (NEW)

Petri nets

Programmer workbench**

Software libraries

State diagrams (NEW)

Structured programming**

Top-down programming**

User interfaces

D.2.3 Coding Tools and Techniques (REVISED)

Object-oriented programming (NEW)

Pretty printers

Program editors

Reentrant code**

Standards

Structured programming (NEW)

Top-down programming (NEW)

D.2.4 Software / Program Verification (F.3.1) (REVISED)

Assertion checkers

Class invariants (NEW)

Correctness proofs

Formal methods (NEW)

Model checking (NEW)

Programming by contract (NEW)

Reliability

Statistical methods (NEW)

Validation

D.2.5 Testing and Debugging

Code inspections and walk-throughs

Debugging aids

Diagnostics

Distributed debugging (NEW)
 Dumps**
 Error handling and recovery
 Monitors
 Symbolic execution
 Testing tools (e.g., data generators, coverage testing) (REVISED)
 Tracing
 D.2.6 Programming Environments
 Graphical environments (NEW)
 Integrated environments (NEW)
 Interactive environments (REVISED)
 Programmer workbench (NEW)
 D.2.7 Distribution, Maintenance, and Enhancement (REVISED)
 Corrections**
 Documentation
 Enhancement**
 Extensibility**
 Portability
 Restructuring, reverse engineering, and reengineering (REVISED)
 Version control
 D.2.8 Metrics (D.4.8)
 Complexity measures
 Performance measures
 Process metrics (NEW)
 Product metrics (NEW)
 Software science**
 D.2.9 Management (K.6.3, K.6.4)
 Copyrights**
 Cost estimation
 Life cycle
 Productivity
 Programming teams
 Software configuration management
 Software process models (e.g., CMM, ISO, PSP) (NEW)
 Software quality assurance (SQA)
 Time estimation
 D.2.10 Design** (D.2.2)
 Methodologies**
 Representation**
 D.2.11 Software Architectures (NEW)
 Data abstraction (NEW)
 Domain-specific architectures (NEW)
 Information hiding (NEW)
 Languages (e.g., description, interconnection, definition) (NEW)
 Patterns (e.g., client/server, pipeline, blackboard) (NEW)
 D.2.12 Interoperability (NEW)
 Data mapping (NEW)
 Distributed objects (NEW)

Interface definition languages (NEW)
D.2.13 Reusable Software (NEW)
Domain engineering (NEW)
Reusable libraries (NEW)
Reuse models (NEW)
D.2.m Miscellaneous
Rapid prototyping**
Reusable software**

*Indicates that the classification is no longer used as of January 1991, but that the item is still searchable for previously classified documents.

**Indicates that the classification is no longer used as of January 1998, but that the item is still searchable for previously classified documents.

C. Listagem dos Programas em PERL

Em anexo, no CD-ROM, apresentam-se:

- Listagem do Programa Wiki;
- Listagem do Programa Meta;
- Listagem dos Módulos do Programa Meta;
- Principais telas do Portal Colaborativo.

