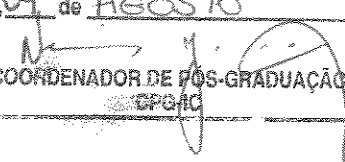


Este exemplar corresponde à redação final da
Tese/Dissertação devidamente corrigida e defendida
por: EMERSON YOSHIO TAKAHASHI

e aprovada pela Banca Examinadora.

Campinas, 09 de AGOSTO de 02


COORDENADOR DE PÓS-GRADUAÇÃO
CPGAC

**Integração de Correio Eletrônico em um Sistema de
Gerenciamento de Informações Digitais**

Emerson Yoshio Takahashi

Dissertação de Mestrado

Integração de Correio Eletrônico em um Sistema de Gerenciamento de Informações Digitais¹

Emerson Yoshio Takahashi

Abril de 2002

Banca Examinadora

- Prof. Dr. Rogério Drummond (Orientador)
- Prof. Dr. Roberto A. Hexsel²
- Prof. Dr. Edmundo R. M. Madeira³
- Prof. Dr. Ricardo O. Anido⁴

¹ Projeto parcialmente financiado pela CAPES, CNPQ e FAEP

² Departamento de Informática da Universidade Federal do Paraná

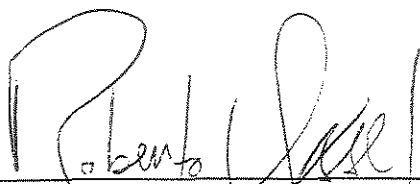
³ Instituto de Computação da UNICAMP

⁴ Instituto de Computação da UNICAMP

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 18 de abril de 2002, pela Banca Examinadora composta pelos Professores Doutores:



Prof. Dr. Roberto André Hexsel
UFPR



Prof. Dr. Edmundo Roberto Mauro Madeira
IC - UNICAMP



Prof. Dr. Rogério Drummond Burnier Pessoa de Mello Filho
IC - UNICAMP

UNIDADE 80
Nº CHAMADA TI-UNICAMP
T139i
V _____ EX _____
TOMBO BC/ 51313
PROC 16.837/02
C _____ DY _____
PREÇO R\$ 11,00
DATA 24/10/02
Nº CPD _____

CM00175024-9

BIBID. 265179

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Takahashi, Emerson Yoshio

T139i Integração de correio eletrônico em um sistema de
gerenciamento de informações digitais/ Emerson Yoshio Takahashi --
Campinas, [S.P. :s.n.], 2002.

Orientador : Rogério Drummond

Dissertação (mestrado) - Universidade Estadual de Campinas.
Instituto de Computação.

1.Sistemas de recuperação da informação. 2.World Wide Web
(Sistema de recuperação da informação). 3.Correio eletrônico. 4.Banco
de dados relacionais. I. Drummond, Rogério. II. Universidade Estadual
de Campinas. Instituto de Computação. III. Título.

Integração de Correio Eletrônico em um Sistema de Gerenciamento de Informações Digitais

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Emerson Yoshio Takahashi e aprovada pela Banca Examinadora.

Campinas, Abril de 2002



Prof. Dr. Rogério Drummond (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

00250067

Agradecimentos

Aos meus pais, que sempre deram total apoio nos momentos mais difíceis.

Aos professores do Instituto de Computação pelo desprendimento e diligência na árdua tarefa de transmitir e gerar conhecimentos.

Aos órgãos de fomento à pesquisa: CAPES, CNPQ e FAEP pelo apoio financeiro.

A todos do Laboratório A-HAND, especialmente a Adrian Carlos Ferreira, Alexandre Prado Teles, Ary Silva Dourado, Carlos Alberto Furuti, José Augusto da Costa Jales, Pedro de Espinosa Rangel, pelas críticas sempre construtivas, revisões, sugestões e incentivos.

Um agradecimento especial a Guilherme Rodrigues de Matos Jr. por toda sua dedicação e empenho na construção da parte servidora do INEX, sem a qual este trabalho não teria sido possível.

Ao professor Rogério Drummond, pela orientação e pela brilhante idéia do INEX.

Principalmente, é uma grande distinção tê-los como amigos.

Índice

RESUMO	X
ABSTRACT.....	XI
INTRODUÇÃO.....	1
CONCEITOS BÁSICOS.....	2
1.1 O QUE É INFORMAÇÃO?.....	2
1.2 QUALIDADES DA INFORMAÇÃO.....	2
1.3 META-INFORMAÇÃO	4
1.4 TECNOLOGIA E INFORMAÇÃO	5
1.5 GERENCIAMENTO DE INFORMAÇÕES.....	6
TECNOLOGIAS ATUAIS.....	10
2.1 CORREIO ELETRÔNICO.....	10
2.2 LISTAS DE DISCUSSÃO	11
2.3 NEWSGROUPS	12
2.4 WWW.....	12
2.5 CARACTERÍSTICAS DAS TECNOLOGIAS ESTUDADAS	13
2.6 INTRANETS.....	16
INEX.....	19
3.1 DESCRIÇÃO DOS TIPOS BÁSICOS INEX.....	20
3.2 CARACTERÍSTICAS DO SISTEMA	22
DETALHES DO SISTEMA.....	31
4.1 ARQUITETURA INEX.....	31
4.2 TECNOLOGIAS UTILIZADAS NO CLIENTE.....	32
4.3 PORQUE UTILIZAR NAVEGADORES PARA INTERFACE COM O USUÁRIO.....	36
4.4 SERVIDOR INEX.....	37
4.5 PERSISTÊNCIA DE INFORMAÇÕES	38
4.6 CAMADA DE EXTENSÃO.....	39
4.7 INTERAÇÃO ENTRE AS DIFERENTES CAMADAS.....	40
4.8 UM EXEMPLO DE INTERAÇÃO TÍPICA.....	40
INTEGRAÇÃO DO CORREIO ELETRÔNICO AO INEX.....	43
5.1 INTEGRAÇÃO COM A CAMADA DE EXTENSÃO	46
5.2 RECEBIMENTO DE MENSAGENS	47
5.3 ENVIO DE MENSAGENS	48
5.4 ARMAZENAMENTO DE MENSAGENS.....	49

5.5	UTILIZAÇÃO DE INEX EMULANDO <i>NEWSGROUPS</i>	49
5.6	INEX COMO GERENCIADOR DE LISTAS DE DISCUSSÃO	50
5.7	TESTES DE DESEMPENHO	51
CONSIDERAÇÕES FINAIS		55
6.1	CONCLUSÕES	55
6.2	TRABALHOS FUTUROS	56
REFERÊNCIAS BIBLIOGRÁFICAS.....		58
APÊNDICE A		62

Índice de Figuras

Figura 1.1 – Ciclo de vida da informação	6
Figura 2.1 – Diagrama exemplificando uma <i>intranet</i>	16
Figura 3.1 – Diagrama de classes básicas INEX.....	20
Figura 3.2 – Definição da classe Item.....	23
Figura 3.3 – Classe File.....	23
Figura 3.4 – Exemplo de extensão das classes básicas INEX	25
Figura 3.5 – Exemplo de extensão das classes INEX.....	26
Figura 3.6 – Exemplo de hierarquia INEX	28
Figura 4.1 – Arquitetura INEX.....	31
Figura 4.2 – Tecnologias utilizadas no cliente.....	32
Figura 4.3 – Diagrama da arquitetura RMI.....	34
Figura 4.4 – Diagrama do servidor INEX.....	37
Figura 4.5 – Mapeamento de classe INEX em tabela de banco de dados relacional.	38
Figura 4.6 – Diagrama da camada de extensão	39
Figura 4.7 – Diagrama de fluxo de uma operação típica	41
Figura 5.1 – Diagrama de comunicação entre correio eletrônico e o INEX.....	43
Figura 5.2 – Classes Email e Mailbox.....	44
Figura 5.4. – Diagrama das classes de extensão para Mailbox e EMail.....	46
Figura 5.5 – Arquivo de configuração para itens do tipo Mailbox e EMail [Mat01].....	47
Figura 5.6 – Recebimento de mensagens.....	48
Figura 5.7 – Envio de mensagens	49
Figura 5.8 – Tempo gasto para pesquisa de um termo.	53
Figura 5.9 – Tempo gasto para consulta de cada mensagem.	54
Figura 6.1 – Exemplo de aplicação usando o conceito de árvore hiperbólica.	56

Índice de Tabelas

Tabela 2-1 – Características das tecnologias atuais.....	14
Tabela 3-1 – Atributos de um tipo Item	23
Tabela 3-2 – Exemplo de atributos de objetos multimídia	25
Tabela 4-1 – Comparação DHTML+JavaScript com <i>applets</i> Java.....	36
Tabela 4-2 – Descrição dos passos de uma interação típica.	42
Tabela 5-1 – Atributos de itens do tipo EMail.....	45
Tabela 5-2 – Especificação das máquina usadas nos testes de desempenho.....	51

Resumo

O sistema de correio eletrônico é uma das ferramentas mais utilizadas para troca de informações e seu uso vem se expandindo a taxas excepcionais. Segundo projeções da International Data Corporation¹, no ano de 2005, o número de mensagens enviadas mundialmente num dia típico atingirá 35 bilhões. Um dos maiores problemas enfrentados atualmente por usuários de computadores em *intranets* é o gerenciamento do grande volume de dados gerado pelo correio eletrônico. As ferramentas de correio eletrônico existentes geralmente são proprietárias, restritas a uma plataforma de *hardware* ou *software* e não são apropriadas para a tarefa de gerenciar as informações contidas nas mensagens. Este trabalho apresenta uma extensão a um sistema de gerenciamento de informações digitais, com o objetivo de resolver esse tipo de restrição através do controle sobre o acesso e a distribuição de mensagens de correio eletrônico.

¹ <http://www.idcresearch.com>

Abstract

The electronic mailing system is one of the most successful tools for information exchange. According to International Data Corporation¹, the number of emails sent in an average day is expected to hit 35 billion worldwide by 2005. Managing the amount of data generated by electronic mail is one of the biggest problems users face today. Most of electronic mail tools currently available are proprietary, hardware and software dependent and inadequate to manage all information in messages. This work presents an extension to an electronic information management system that aims to solve that problem controlling the electronic mail messages access and sharing.

¹ <http://www.idcresearch.com>

Introdução

O crescimento explosivo das redes de computadores, especificamente a Internet, provocou uma revolução nos meios de comunicação. As redes de computadores tornaram muito fácil o acesso, criação e publicação de qualquer tipo de informação, a ponto de qualquer pessoa com acesso à rede poder publicar informações que são acessíveis a um grande número de outros usuários de computador.

No meio acadêmico e comercial, listas de discussão, publicações *on-line*, treinamento à distância, pesquisas cooperativas entre diferentes instituições nos mais variados ramos do conhecimento humano têm possibilitado o intercâmbio de informações numa escala sem precedentes.

Como exemplo podemos citar o sistema Rau-Tu¹ que “tem por objetivo possibilitar que um time de colaboradores possa responder perguntas colocadas por qualquer pessoa em um site da Web, cobrindo diversas áreas de conhecimento”.

A utilização maciça de redes de computadores tem inúmeras e inegáveis vantagens, mas um problema tem se mostrado cada vez mais em evidência: o excesso de dados.

Existe uma disponibilidade cada vez maior de dados em formato digital. Isso cria uma falsa impressão de abundância de informações, mas na realidade o que existe é uma quantidade enorme de dados que são inúteis para a maioria dos usuários, demandando tempo e esforço para se encontrar informação utilizável. Aliado a isso, há uma pressão considerável para o uso de informações no formato digital criando um ciclo que se auto-alimenta.

Novas ferramentas que otimizem a busca, armazenamento e distribuição são necessárias para gerenciar e organizar melhor a informação. Este trabalho propõe um abordagem e apresenta um sistema desenvolvido para atingir esse objetivo em relação ao sistema de correio eletrônico, especificamente no contexto de *intranets*. Inicialmente, será apresentado o conceito de informação e requisitos para se caracterizar “informação de qualidade” e as implicações da tecnologia aplicada à informação. Em seguida, serão apresentadas tecnologias atuais para fazer acesso às informações em redes de computadores e as limitações de cada uma delas. No final, será apresentada a ferramenta INEX (*Intranet Express*) que trata de alguns dos problemas abordados, detalhes de sua implementação, como ela pode ser usada para gerenciamento de mensagens de correio eletrônico, algumas considerações finais e as referências bibliográficas.

¹ <http://www.rau-tu.unicamp.br/>

CAPÍTULO 1

Conceitos Básicos

Neste capítulo caracterizaremos, para o contexto deste trabalho, alguns conceitos que serão utilizados posteriormente, delimitando assim, seus escopos de abrangência.

1.1 O Que é Informação?

Talvez o caminho mais lógico a ser seguido seja definir primeiramente o que se pode entender por *dado* e posteriormente o que se pode entender por *informação*. O que é dado? Um endereço é um dado. Um número de telefone é um dado. Uma referência bibliográfica é um dado. Dessa forma, tudo que descreva algo, qualquer coisa sobre a realidade é um dado.

“Intuitivamente entendemos que a informação é aquela parte do dado que tem impacto sobre nossas ações ou, no caso de falta ou não disponibilidade, também demanda novas ações. Para pessoas diferentes, ou até para a mesma pessoa em momentos diferentes, a mesma gama de caracteres poderia ser dado ou informação” [Gol92].

Assim, informação é um dado útil para alguém. Por exemplo, um endereço é um dado, mas para uma pessoa que procura por uma determinada empresa que se localiza no referido endereço, aquele endereço é uma informação.

1.2 Qualidades da Informação

Informação está em toda parte na Internet, existindo em grande quantidade e sendo continuamente criada e revisada. Esta informação existe em uma vasta gama de tipos (fatos, opiniões, histórias, interpretações, estatísticas), sendo criada com diversos propósitos (para informar, persuadir, vender, apresentar um ponto de vista, criar ou mudar uma atitude ou crença). Para cada um desses vários tipos e propósitos, a informação existe em vários níveis de qualidade e confiabilidade, variando do “muito bom” ao “muito ruim”, com todas as gradações entre esses extremos [Har97].

Nicholas Johnson [Joh93] destaca algumas propriedades que a informação deveria possuir, considerando-as como uma outra dimensão além das categorias nas quais as informações poderiam ser classificadas.

- **Precisão.** Existe uma distinção entre uma informação e uma informação precisa, e entre precisão e veracidade. Um termômetro digital com mau funcionamento pode continuar a ser bastante preciso (por exemplo, com dez casas decimais), mesmo que ele

esteja sendo precisamente errado. Uma história de um tablóide sensacionalista, repleta de detalhes, pode estar suficientemente errada para dar margem a um ação por difamação. Resultados de laboratório, seja de pesquisa pura ou aplicada, podem ter sido deliberadamente falsificados.

- **Oportunidade.** Uma das qualidades que distinguem alguns tipos de informação de outros é a sua adequação no tempo. As informações relativas ao movimento das ações na bolsa de valores, entre 10h e 11h de hoje, podem ser de grande interesse para um investidor. No entanto, estas mesmas informações podem não ter nenhuma utilidade para o mesmo investidor amanhã. A tabela periódica dos elementos químicos muda no tempo, mas muito mais vagarosamente que o movimento das ações na bolsa de valores. As informações de uma enciclopédia ou almanaque mudam muito rapidamente, mas ainda permanecem úteis por meses ou anos depois da publicação. Existem informações que possuem valor em um determinado espaço de tempo, após o qual podem se tornar inúteis (ou pior, “desinformação”) e este período de tempo é variável para cada tipo de informação e para cada tipo de usuário.
- **Relevância.** A informação pode ser precisa e apropriadamente oportuna, mas de pouca relevância. Um mapa atual e preciso não poderá ser de muita ajuda para encontrar um determinado caminho caso o usuário não saiba como se localizar. “Sobrecarga de informações” é outro nome para informações que as pessoas não consideram relevante naquele momento, como propaganda para produtos que não são necessários ou desejáveis, notícias sobre eventos que não afetam as pessoas em questão, ou programas de TV considerados tediosos ou ofensivos.
- **Eficiência.** Podemos definir a eficiência de uma informação como uma relação entre seu valor num determinado instante e seu custo. Existem custos associados com a informação: o preço de capa de uma revista, a matrícula para um curso, o acesso do computador à base de dados na Internet. Existem também custos associados a sua distribuição: a conta do telefone pelo acesso do computador, taxas de entregas por um livro, as taxas de entrega para equipamentos comprados. O armazenamento pode requerer outra estante ou arquivo. Se o usuário precisa mover a informação, outras características se tornam relevantes, por exemplo, tamanho e peso; um livro de capa dura é freqüentemente um meio muito eficiente de armazenamento de informações, mas uma biblioteca pode ser mais facilmente transportada como microfilme ou discos de CD-ROM do que como estantes repletas de livros. Existem ainda custos associados com acessibilidade que vão além dos pagamentos monetários: o tempo e o aborrecimento envolvidos para conseguir a informação. Acessibilidade não requer apenas dinheiro e facilidade de acesso, mas a capacidade de compreender, de extrair significado do que foi recuperado. Dados de satélite, fórmulas químicas, conjuntos de leis, tudo pode ser obtido com pouco ou nenhum custo e ainda ser virtualmente incompreensível para o leitor.
- **Formato.** Na era da multimídia, a forma na qual a informação é apresentada é outra importante qualidade. Desenhos, números, gráficos e diagramas, imagens estáticas e em movimento, cores, trilhas sonoras, algumas vezes ajudam muito na assimilação da

informação. Em outras ocasiões, elas podem variar entre algo irritante e um real obstáculo ao acesso à informação.

- **Nível de Abstração.** Um especialista em semântica poderia expressar a distinção entre dado, informação, conhecimento e sabedoria como diferenças nos "níveis de abstração". Qualquer que seja a forma como é chamada, distinções úteis podem ser estabelecidas entre descrições verdadeiras de eventos no espaço e no tempo e as generalizações e conclusões feitas a partir delas. Todos reconhecemos que a declaração "o senador José foi eleito com 42 por cento dos votos" é de alguma forma diferente de "o senador José não é muito popular" ou "o senador José é o pior senador do estado". O que diferencia as afirmações é o nível de abstração utilizado.

Além dos pontos levantados por Johnson, podemos destacar mais algumas características que deveriam ser levadas em consideração no momento em que se procura atribuir um determinado nível de qualidade a uma certa informação:

- **Persistência.** As diferentes mídias que podem ser utilizadas no armazenamento da informação têm a possibilidade de estabelecer uma natureza estática ou dinâmica para a informação. Podemos considerar um livro como possuindo uma informação de natureza estática, mas o mesmo não se pode afirmar a respeito, por exemplo, de uma página na Web que se encontra em construção, na qual as informações apresentadas são atualizadas frequentemente. Enquanto um filme ou documentário pode ser útil sendo reprisado, um noticiário normalmente perde sua utilidade após sua primeira exibição. O noticiário é uma informação de disponibilidade volátil, ao passo que o documentário poderia ser considerado persistente.
- **Maturidade.** A informação pode ser relevante, oportuna e precisa, mas pode ainda não ter atingido um nível de maturação que permita sua aplicação útil. Pode ser necessário um avanço tecnológico, agregação de mais informações ou idéias adicionais para que a informação em questão tenha realmente utilidade prática. Por outro lado, uma certa tecnologia pode se tornar obsoleta e um novo avanço nesse tipo de tecnologia se torna irrelevante, ou seja, o período de maturidade da informação contida na tecnologia foi ultrapassado.
- **Veracidade.** Uma informação, como já foi dito, pode ser precisa, mas não verdadeira. Dados bastante precisos mas não verdadeiros podem ser "plantados" por pessoas, empresas ou governos para ocultar, distorcer, ou induzir adversários a erro de julgamento. Nesse sentido, comunicações mentirosas, boatos, documentos falsos, propaganda enganosa e outros instrumentos não ortodoxos podem ser empregados na manipulação dos fatos. "A veracidade é a seu modo uma correspondência: a correspondência do afirmado (verdade) com a prova do afirmado" [Mor89].

1.3 Metainformação

O termo metainformação no contexto deste trabalho define toda informação que descreve outra informação ou conjunto de informações.

A importância da metainformação reside no fato de que, se disponível, ela provê um meio fácil de procurar por informações. Uma busca pode ser limitada pelas características das informações e não pela classificação nas quais estão inseridas. De acordo com Taylor [Tay99] “o objetivo do metadado é prover um nível intermediário de escolhas que podem ser feitas sobre um conjunto de informações que alguém deseja examinar ou investigar, sem procurar em uma grande quantidade de texto irrelevante”.

Metainformações provêem meios de classificar informações sem a criação de estruturas rígidas que limitem o campo de pesquisa em categorias ou classes mutuamente exclusivas.

1.4 Tecnologia e Informação

A informação tem muitas características que a diferencia das coisas tangíveis; a informação tende a aumentar de valor conforme é compartilhada e pode ser transferida a terceiros sem que ocorra uma perda para o possuidor. Aliada a esse fato, a tecnologia pode ser usada para agregar valor à informação através do armazenamento, codificação, processamento e apresentação de uma forma que aumente o conhecimento, com a restrição que a informação seja válida, precisa e aplicável para a tarefa em questão.

Portanto, meros fatos têm pouco valor se não estão ligados a um escopo particular de um problema. Criar inferências a partir de dados para descobrir relacionamentos e sintetizar estes relacionamentos em conhecimento (que pode afetar decisões e aumentar a riqueza) são tarefas que podem ser transformadas de maneira bastante positiva com a utilização da tecnologia.

Assim, segundo Haeckel [Hae93+], “a tecnologia apropriadamente implementada e usada pode transformar a natureza da informação de um agente passivo para um agente ativo no processo de criação de riquezas”.

Para avaliar o impacto da tecnologia, vamos usar a definição de eficiência das informações pertencentes a uma organização segundo uma relação entre o valor da informação e seu custo num determinado instante.

Como o valor da informação é uma medida subjetiva e muitas vezes incomensurável em termos numéricos, podemos intuitivamente inferir que haverá um aumento na eficiência da informação ao alterar-se o componente *custo*, que pode ser decrescido com a utilização da tecnologia.

Um exemplo de como isso ocorre pode ser verificado pelo fato de que a simples aplicação de tecnologia à informação muda sua eficiência, porque seu processamento é mais rápido, dando às pessoas com poder de decisão mais tempo para avaliar as opções ou tomar decisões mais rapidamente.

Além disso, a tecnologia permite às organizações gerenciar, organizar e ligar múltiplas fontes de informação aos usuários adequados, possibilitando o trabalho em grupo. Conexão e compartilhamento de informações entre diferentes grupos podem dar aos participantes um senso unificado do objetivo da instituição. O resultado do uso da tecnologia para compartilhar informações e conectar usuários é um aumento do conhecimento ou da “inteligência total”.

No entanto, o aumento de investimentos em tecnologia não garante essa sinergia. Este investimento deve ser feito sobre ferramentas que garantam o gerenciamento e a distribuição da informação de maneira a atender aos processos da organização.

1.5 Gerenciamento de Informações

Através do gerenciamento de informações, podemos assegurar que o fluxo de informações de um grupo ou organização é “saudável” e está sendo tratado apropriadamente. Ferramentas de gerenciamento de informações podem assegurar que se está trabalhando com informação atualizada e que ela flui para onde é necessária.

O processo de gerenciamento de informações pode ser visto como uma série de ciclos, conforme ilustrado na **Figura 1.1**, que envolve as seguintes etapas: identificação das necessidades do usuário; aquisição da informação; organização e armazenamento da informação; distribuição da informação e o uso da informação, que continua constantemente refinada.

A concepção do gerenciamento de informação como um ciclo de atividades interrelacionadas para planejar, projetar e coordenar provê uma perspectiva que complementa a visão tradicional de gerenciamento de informação como gerenciamento da tecnologia de informação ou gerenciamento de recursos de informação [Cho98].

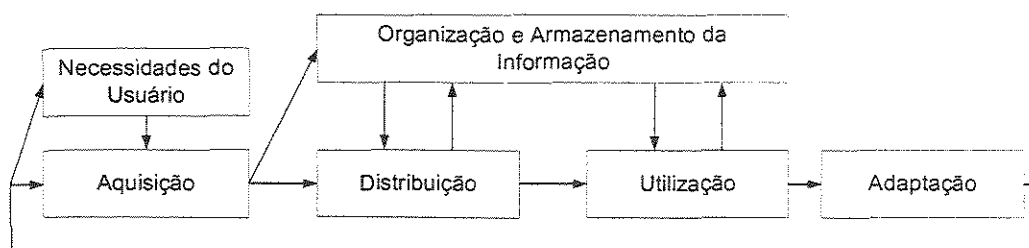


Figura 1.1 – Ciclo de vida da informação¹

1.5.1 Necessidades do Usuário

A identificação das necessidades do usuário deve ser suficientemente rica e completa na representação e elaboração das suas reais necessidades. Como o uso da informação normalmente tem lugar no contexto de uma tarefa ou problema, devemos reconhecer que as necessidades de informação consistem em duas partes inseparáveis [Tay91]: uma pertinente à questão central da necessidade (qual informação é necessária) e aquela que surge da própria situação de utilização da informação (por que é necessária e como ela será usada). Portanto, a tarefa de identificação das necessidades de informação não envolve somente a determinação dos tópicos de interesse do usuário, mas também os atributos ou qualidades da informação a ser fornecida que aumentem seu valor e utilidade.

¹ Adaptado de [Cho98]

1.5.2 Aquisição da Informação

A tarefa de aquisição de informação consiste em investigar o espaço de informações (livros, páginas HTML, mensagens) em busca da informação desejada de acordo com as necessidades do usuário. Existem várias razões que levam o usuário a buscar informações, que podem variar a cada momento, inclusive para o mesmo indivíduo. O que importa é o ponto onde uma informação se torna eficiente: se o custo de acesso à informação é maior que o seu valor, o usuário não a usará ou buscará outros meios de consegui-la.

Um poderoso meio de gerenciar uma grande variedade de fontes de informação é envolver o máximo possível de pessoas na tarefa de coletar informações, criando assim uma rede de coleta que inclua os principais interessados. Pessoas e não a informação impressa ou banco de dados eletrônicos, são a mais valiosa fonte de informações em qualquer organização [Cho95]. Pessoas são capazes de filtrar e resumir informações, ressaltar tópicos importantes, interpretar aspectos ambíguos e em geral fornecem uma visão mais rica e satisfatória sobre um tópico do que relatórios gerados por sistemas automatizados.

1.5.3 Organização e Armazenamento da Informação

A organização e o armazenamento da informação podem ser melhorados com a aplicação de tecnologias de informação. O uso de computadores gerou uma abundância de informações detalhadas sobre um amplo espectro de assuntos. No entanto, tais sistemas são direcionados para prover um grande volume de dados que são ineficientes e algumas vezes incapazes de devolver a informação que os usuários necessitam. Organizações com grandes volumes de informações precisam reorganizar e unificar dados de diversas fontes e prover ferramentas que possibilitem aos usuários “garimpar” os dados brutos em busca de informações úteis.

Em vez de eficiência no sentido de grandes volumes de dados retornados, o maior requisito de um sistema é a flexibilidade; os sistemas devem capturar dados de vários formatos, suportar múltiplas visões dos dados, unir itens que são funcional ou logicamente relacionados e assim por diante.

Como a mesma informação pode ser relevante em diferentes contextos de problemas, é necessário representar e organizar as informações de várias formas. O desenvolvimento de sistemas de indexação automática torna mais plausível a adoção de uma abordagem centrada no usuário, além do sistema centrado no conteúdo dos documentos. “Na abordagem centrada no usuário, a indexação pode ser feita em dois níveis: o primeiro reflete o assunto e outras características predeterminadas; o segundo é criado de acordo com os necessidades da situação, como o nível de tratamento, se geral ou específico” [Fid94]. Uma combinação das duas abordagens pode aumentar significativamente o desempenho na recuperação de informações, assim como a satisfação dos usuários.

1.5.4 Distribuição da Informação

O objetivo da distribuição de informações é encorajar o seu compartilhamento. Uma ampla distribuição de informações promove a criação de conhecimentos mais abrangentes e de forma mais freqüente, aumenta a eficiência na recuperação de informações relevantes e facilita a criação de novas visões a partir de dados aparentemente desconexos.

A distribuição da informação deve ser feita através de veículos e formatos que se encaixem nos hábitos e preferências dos usuários. A separação entre provedor de informação e usuário de informação deve ser dissolvida: ambos precisam colaborar como parceiros na disseminação e como indivíduos capazes de adicionar valor à informação, assegurando que a informação apropriada é vista pelas pessoas certas dentro de uma organização.

A remoção da barreira que separa provedores e consumidores de informação agrega a todos na rede de distribuição como participantes ativos que contribuem com idéias e auxiliam a refinar e filtrar as informações. Para encorajar os usuários a serem participantes ativos, deve ser fácil comentar, avaliar e redirecionar as informações recebidas.

Para um modelo de como essa idéia poderia funcionar, tomemos como base uma forma bastante popular de compartilhamento de informações – os grupos de *news* Usenet, conhecidos por *newsgroups* – detalhes técnicos dos *newsgroups* são discutidos no Capítulo 2. Nos *newsgroups*, a informação é postada no que é essencialmente um quadro de avisos eletrônico que é visto por todos no grupo. Cada texto tem um cabeçalho de assunto e os usuários podem examinar rapidamente estes títulos para escolher as mensagens a serem lidas. Depois da leitura, os próprios usuários participam através do envio de novas mensagens que respondem às questões formuladas, acrescentam novas explicações, sugerem alternativas, etc. Mensagens que tratam de um mesmo assunto são relacionadas em grupos sobre tópicos particulares. De tempos em tempos, as respostas às questões e um sumário das discussões podem ser reenviados ao grupo.

Podemos tomar a idéia de *newsgroups* como um modelo para a disseminação de informações na organização. Mensagens novas são distribuídas imediatamente, dependendo apenas do tempo de latência no tráfego pela rede. Os cabeçalhos informam sobre o assunto que a mensagem trata. Os usuários podem procurar, ler e discutir as mensagens uns dos outros. Múltiplas perspectivas e opiniões podem surgir de acordo com os diferentes níveis de experiência e conhecimentos dos usuários. Linhas de discussão podem ser resumidas por um usuário ou moderador com conhecimentos especiais sobre o assunto e esses resumos podem então ser reenviados. Essas trocas eletrônicas de informações podem levar ocasionalmente a discussões face a face, ou à formação de grupos especiais de interesse.

Diferente de reuniões e encontros formais, os *newsgroups* atuam como um “espaço compartilhado de colaboração na qual os participantes ativamente criam percepções compartilhadas” [Sch90]. Nesse espaço, as ferramentas são interativas, os participantes têm acesso igual e voluntário, as conversações são espontâneas, múltiplas linhas de discussão podem se desenrolar em paralelo e o foco está centrado no conteúdo das mensagens. Como um meio de distribuição e compartilhamento de informações, grupos de discussão bem organizados, com usuários motivados, podem ser capazes de adicionar um considerável incremento à base de conhecimentos de uma organização.

1.5.5 Uso da Informação

A utilização da informação é um processo dinâmico, interativo e social de investigação que pode resultar na criação de idéias ou tomada de decisões. Este processo de investigação é cíclico e varia entre considerações de partes e do todo e entre detalhes práticos e conceitos genéricos. Os participantes podem esclarecer dúvidas ou confrontar as idéias uns dos outros. Escolhas podem ser feitas por intuição pessoal, persuasão, assim como através de uma análise

racional. Usuários de informações trabalham em um ambiente que tem sido descrito como sofrendo de sobrecarga de informações (*information overload*): conforme novas informações são recebidas e os usuários refletem e atuam sobre os problemas, a percepção da situação muda, dando origem a novas incertezas. O contexto do problema a ser resolvido é refinado, usuários buscam novas informações e o ciclo recomeça até o problema ser considerado resolvido. A estrutura de informações de uma organização precisa ser aberta, flexível e robusta para fazer frente às necessidades dos usuários.

A participação ativa dos usuários é essencial para o melhor aproveitamento de um sistema de informações, visto que são eles os primeiros a compreender quais são as informações necessárias para a resolução de um problema e quais informações foram efetivamente utilizadas na resolução dos problemas. Dessa forma, os usuários finais são as melhores fontes para o refinamento e classificação das informações.

As organizações normalmente utilizam ferramentas de busca automáticas para localizar informações em suas *intranets* (mais detalhes são apresentados na seção 2.6), no entanto, muitas dessas organizações tem constatado que ferramentas automáticas (mesmo as mais recentes, chamadas “inteligentes”) não são suficientes. Por exemplo, ao se procurar uma informação, normalmente ferramentas automáticas retornam centenas de documentos como resultado e não há indicação de qual realmente possui a informação desejada, restando ao usuário verificar cada um dos documentos num processo que pode demandar um tempo excessivo até se encontrar a informação procurada.

De acordo com Brocolli [Bro99], isto ocorre devido à forma como ferramentas automáticas de busca operam, geralmente procurando por todas as ocorrências de um termo e simplesmente listando os documentos encontrados. No entanto, os termos procurados podem ser apenas citados nos documentos, não constituindo informação de real valor. O usuário também pode estar procurando por informações mais específicas relacionadas ao tópico e muitas vezes não sabe como restringir a extensão da pesquisa. Outra possibilidade é o documento usar sinônimos que não são grafados de forma semelhante, por exemplo, ao se fazer uma busca pelo termo “música”, não serão encontrados os documentos que possuem a palavra “canção”, mesmo que esses documentos sejam relevantes à pesquisa. Uma ferramenta de busca, como qualquer outro autômato, não sabe lidar com tais circunstâncias.

Em resumo, a utilização de informação para a criação de significados e compreensão requer processos e métodos que atendam a um alto grau de flexibilidade na representação da informação e que facilitem uma troca constante e avaliação de múltiplas representações entre usuários. A classificação e designação de conceitos e categorias deve ser relevante e inserida no contexto do usuário, sendo flexível e fácil de ser modificada.

1.5.6 Adaptação

Periodicamente é útil realizar uma reanálise formal sob uma perspectiva abrangente. Esta análise leva a um novo ciclo, com a organização tomando decisões e ações que irão realimentar o processo. Nessa transformação, podem ser alterados os critérios de obtenção, seleção, distribuição e as próprias informações já adquiridas, procurando adequar-se às necessidades dos usuários, de acordo com levantamentos e estudos realizados.

CAPÍTULO 2

Tecnologias Atuais

O termo "*networked information*" se aplica a um conjunto de informações eletrônicas e serviços atualmente disponíveis através da Internet, incluindo grupos de discussão, bases de dados, repositórios de *software* e cópias eletrônicas de livros e periódicos.

Entre os recursos utilizados para fazer acesso às informações na Internet podemos destacar o correio eletrônico, listas de discussão, *newsgroups* e a *Web* também conhecido como WWW ou *World Wide Web*.

2.1 Correio Eletrônico

Uma das aplicações mais populares utilizadas em redes de computadores é sem dúvida o correio eletrônico ou *email* (*electronic mail*), que permite que pessoas troquem informações na forma de mensagens de uma maneira semelhante ao sistema postal convencional.

O correio eletrônico é utilizado basicamente para troca de informações entre dois usuários; além disso, um usuário pode enviar a mesma mensagem para diferentes usuários usando o conceito de lista de distribuição (um nome que identifica um grupo) [Tan96].

Um problema das listas é a utilização excessiva de espaço de armazenamento devido à replicação da mesma informação. Suponhamos que uma lista de distribuição de alunos de uma universidade possua mil usuários cadastrados que possuem conta num mesmo servidor. Uma única mensagem postada para essa lista de distribuição criará mil cópias.

As listas de distribuição também têm limitações no seu uso, necessitando de constante intervenção humana para incluir ou remover interessados em receber as mensagens. Este controle sobre os destinatários normalmente é feito de forma artesanal, com o responsável incluindo ou removendo um a um endereços da lista.

Outra limitação do correio eletrônico é a ausência de um mecanismo de seleção das mensagens que o usuário quer receber; o máximo que se pode utilizar são filtros em alguns programas para leitura de *mail* que permitem, por exemplo, descartar automaticamente mensagens com certos assuntos (*subject*) ou que se originam de um determinado usuário, antes mesmo de mostrá-las. No entanto, esse tipo de controle é difícil de ser feito por usuários comuns devido à necessidade de conhecimentos técnicos para sua utilização correta e efetiva. Mesmo a utilização de ferramentas avançadas que contam com mecanismos simples para definir filtros e organização das mensagens não resolvem todos os problemas.

Outro problema, acarretado pelo próprio sucesso do correio eletrônico, é seu uso em funções para as quais não foi originalmente projetado, como por exemplo, gerência de tarefas e arquivo

pessoal de nomes e endereços[Whi96+]. Isto gerou o que se conhece por “*email overload*”¹, tendo como resultado usuários com centenas ou milhares de mensagens desordenadas, tratando dos mais diversos assuntos. Tentativas de organizar manualmente essas mensagens são freqüentemente mal sucedidas, com mensagens importantes sendo perdidas nos arquivos.

Diversas soluções têm sido propostas, como em Becker [Bec00], que propõem uma solução baseada em pastas virtuais, conjuntos de mensagens unidas através de um relacionamento lógico. Este relacionamento lógico é criado através de cláusulas SQL [Bec00, Fer97+] que recuperam mensagens e pastas através de critérios definidos pelos usuários.

Uma solução proposta por Whittaker [Whi96+] envolve a inserção automática pelo sistema de identificadores nas mensagens que permitiriam o agrupamento de mensagens correlatas.

No entanto, estas soluções ainda não são homologadas e carecem de uma maior disseminação.

2.2 Listas de Discussão

Listas de discussão (conhecidas também por *mailing lists*) podem ser consideradas uma extensão do correio eletrônico, uma vez que utilizam os mesmos conceitos e ferramentas. Se o correio eletrônico possibilita a comunicação “um para um”, listas de discussão permitem a comunicação “um para muitos” num determinado contexto que é definido pelo assunto da lista.

As listas de discussão são usadas basicamente para troca de informações ou discussões sobre um determinado assunto. Qualquer participante pode remeter à lista problemas ou dúvidas, por exemplo. Outros participantes podem enviar respostas e comentários para o remetente ou para a lista como um todo, permitindo dessa forma, uma réplica, contestação ou um debate sobre a validade ou mérito da resposta. Estes procedimentos permitem o compartilhamento de informações que potencialmente teriam utilidade para todos os participantes e aumentam a comunicação “um para muitos”.

Uma lista de discussão é na realidade um tipo especial de conta de usuário, onde ao invés de uma pessoa, há um programa que mantém a lista, armazena os arquivos e responde a certos comandos. A principal tarefa do programa é fazer a comunicação “um para muitos” entre os usuários. Quando uma mensagem é enviada para a lista, o programa a envia automaticamente para todos os participantes da lista.

Uma lista pode ser usada também como um serviço de notificações. Mensagens podem ser postadas na lista e os participantes (ou não-participantes em alguns tipos de lista) podem requerer uma cópia dos arquivos, notificações automáticas de atualizações de arquivos ou requisições para o envio de arquivos novos ou atualizados.

Como listas de discussão podem ser consideradas uma extensão do correio eletrônico, muitas das vantagens e desvantagens que se aplicam ao correio eletrônico se aplicam também às listas [Doh97].

¹ Whittaker [Whi96], usa o termo “*email overload*” para descrever a utilização de correio eletrônico em funções para as quais não fora projetado.

As mensagens de listas podem ocupar um espaço significativo de armazenamento, devido não apenas ao número de listas assinadas, mas também à redundância, quando várias pessoas conectadas a um mesmo servidor assinam uma mesma lista.

2.3 Newsgroups

Um *newsgroup* é um fórum de discussões sobre algum tópico específico. Pessoas interessadas nesse assunto podem se “inscrever” no *newsgroup*. Cada participante pode enviar artigos (mensagens) ao *newsgroup* que serão automaticamente enviados para todos os demais.

Como há uma grande variedade de assuntos, eles são organizados em uma estrutura hierárquica para facilitar o gerenciamento e acesso às informações. Isso permite também uma forma de seleção das informações a serem recebidas por parte do usuário. Num nível mais alto da hierarquia serão encontradas informações genéricas e nos níveis mais baixos, informações específicas. O usuário escolhe o nível das informações a receber. No entanto, deve-se observar que a assinatura de um determinado nível de um *newsgroup* não implica no recebimento dos artigos postados nos níveis hierárquicos inferiores ou superiores.

Diferente da filosofia das listas de discussão, cada servidor de *newsgroup* armazena as mensagens em um único diretório, por exemplo */news*, com vários subdiretórios correspondendo aos assuntos. Programas que fazem a leitura das mensagens simplesmente buscam os artigos de que necessitam e os apresentam ao usuário [Tan96]. Dessa forma, cada servidor precisa de apenas uma cópia de cada mensagem, não importando o número de usuários.

Entretanto, pode haver duplicação das mensagens, pois um usuário pode postar a mesma mensagem em vários assuntos ou em vários níveis da hierarquia, o que gera redundâncias que irão dificultar a busca de uma mensagem significativa nestes grupos por outros usuários.

Outro ponto a ser observado é a relativa rigidez das hierarquias, a criação de novos grupos é um processo bastante longo que envolve a submissão da proposta do novo grupo e sua apreciação pela comunidade que posteriormente irá votar aprovando ou reprovando o novo grupo.

2.4 WWW

A WWW (*World Wide Web*) compreende uma vasta coleção de documentos interconectados criados por pessoas ou organizações ao redor do mundo. Na definição de Tim Berners-Lee, pesquisador do CERN (Centro Europeu de Pesquisa Nuclear) que iniciou o projeto, a *Web* é um “sistema de hipermídia distribuído”.

Documentos na *Web* são conectados uns aos outros na forma de um hipertexto. A vantagem do hipertexto na *Web* é que se o usuário deseja mais informações sobre um determinado assunto que é mencionado e que possui um *link* para outro documento, basta carregar a página referenciada pelo *link*, sendo que a página pode estar na mesma máquina do documento original ou em outra localizada a milhares de quilômetros de distância. Um *link* é similar a uma referência bibliográfica, mas cujo texto pode ser consultado instantaneamente.

O hipertexto permite a criação de documentos não-lineares que dão liberdade de escolha ao usuário possibilitando o acesso apenas às informações que ele considera relevantes. Pode se

considerar hipermídia como uma extensão do hipertexto, na qual uma página pode ter referências não somente para texto, mas também para imagens, sons ou animações [Bou97]. Documentos hipermídia têm sido usados com frequência cada vez maior na *Web*.

O acesso às informações na *Web* é feito através de um programa cliente WWW (os mais comuns são os *browsers* ou navegadores, por exemplo o Netscape Navigator¹) que é executado no computador do usuário. O programa cliente permite que os documentos hipermídia seja lidos, permitindo também a "navegação" através dos *links*.

A *Web* está se tornando um dos principais meios de disseminação de informações e devido ao baixo custo dessa distribuição é possível ter acesso a informações sobre praticamente todos os assuntos. Como é barato distribuir publicações na *Web*, uma gama enorme de publicações que seriam restritas sob outras formas (livros por exemplo), atingindo um público bastante limitado, podem se tornar disponíveis para um público muito maior.

Se por um lado isso representa uma "democratização da informação", por outro, representa mais dados que deverão ser filtrados para se conseguir informações de valor. Ferramentas de busca (Altavista, Lycos, etc) são utilizados para fazer essa filtragem. As ferramentas de busca trabalham normalmente através da indexação de palavras ou expressões contidas nas *Web pages*, mas com o crescimento explosivo que a *Web* vem experimentando, mesmo essas ferramentas poderão se tornar inadequadas para o gerenciamento das informações.

2.5 Características das Tecnologias Estudadas

Para definirmos as características desejáveis de um sistema para gerenciamento de informações, inicialmente vamos levantar as características das ferramentas existentes, verificando os pontos fortes e fracos para termos subsídios no processo de decisão dos atributos de tal sistema.

Podemos fazer uma caracterização das tecnologias existentes utilizando um sistema de três dimensões ortogonais a serem consideradas: controle de acesso, forma de comunicação e organização da informação.

Em relação à forma de comunicação da informação, os sistemas podem ser:

- 1-1: Um usuário envia informações a outro usuário;
- 1-N: Um usuário envia informações a vários usuários;
- 0-N: Usuários ativamente buscam informações disponíveis, não havendo necessidade do envio das informações por parte da origem.

A organização da informação pode ser:

- Plana: Todas as informações são dispostas em um único repositório;
- Hierárquica: As informações são organizadas em vários níveis.

Quanto ao controle de acesso, podemos ter acesso:

¹ www.netscape.com

- Controlado: Apenas usuários autorizados têm acesso às informações;
- Sem controle: Qualquer usuário pode ter acesso às informações.

Em relação às características das tecnologias estudadas podemos relacioná-las na tabela abaixo:

Tecnologia	Características		
	Comunicação	Organização	Acesso
Correio Eletrônico	1-1 ¹	Plana	Controlado
Listas de Discussão	1-N	Plana	Controlado ²
Newsgroup	1-N	Hierárquica	Sem Controle
WWW	0-N	Mista	Sem Controle ³

Tabela 2-1 – Características das tecnologias atuais

Cada uma dessas características possui suas vantagens e desvantagens dependendo da aplicação. Analisando o aspecto da comunicação, a abordagem “um para um” possui o mérito de ser uma forma de transferência bastante específica, onde dificilmente a informação será enviada a um destinatário não desejado. Exatamente por isso, limita-se a difusão, tornando o processo de distribuição lento e de pouca abrangência.

Examinando a comunicação “um para muitos”, verificamos que ela apresenta vantagens em relação ao enfoque “um para um”, notadamente no aspecto relativo à distribuição que pode ser de forma muito mais direta, atingindo um público amplo. Há no entanto o problema de informações indesejáveis que pode ser minimizado pela figura do moderador, encarregado de estabelecer quais dados podem ser distribuídos, visto que estão dentro do escopo estabelecido pelo grupo. Porém, isso também implica em custos maiores em termos de recursos humanos, deixando essa opção nem sempre viável.

Já a forma de comunicação que chamamos de “0 – N” depende da iniciativa do usuário de buscar a informação, dado que neste caso a informação possui uma natureza passiva e todo acesso a ela depende da ação do interessado. Se por um lado isso garante que o usuário não será perturbado com informações não solicitadas, por outro, pode privá-lo de conhecimentos essenciais para sua tarefa, pois nem todo usuário tem disposição para procurar pela informação e muitas vezes os usuários não têm conhecimento da sua disponibilidade.

Em relação à forma de organização, uma estrutura plana destaca-se pela sua simplicidade e facilidade de armazenamento, visto que todas as informações estão, do ponto de vista lógico, em um único local. Entretanto, seu uso torna-se muito mais complexo, visto que serão necessários índices ou algum tipo de estrutura sofisticada para ordenar e facilitar a busca de informações no repositório. Mesmo com a utilização de índices, não há a possibilidade de

¹ Através de múltiplos endereços é possível enviar uma mensagem de correio eletrônico a mais de um destinatário.

² Deve-se observar que de modo geral, o controle das assinaturas é feito através de um programa que normalmente aceita todas as requisições de assinatura.

³ Alguns servidores *Web* possuem mecanismos de autenticação.

agrupar conhecimentos correlatos sem o auxílio de estratégias complexas que comprometem a transparência e manutenção do sistema, o que em última instância caracteriza um sistema de organização misto.

Um organização hierárquica das informações facilita o processo de busca e recuperação, visto que pode se limitar o conjunto de dados a ser examinado a um grupo mais restrito e a própria estrutura da hierarquia pode servir como uma indicação de qual subconjunto deve ser pesquisado. De acordo com Shneiderman [Shn99+], hierarquias têm o potencial de reduzir a complexidade pela organização de informações correlatas em estruturas compreensíveis. Há também o fato que estruturação hierárquica tem sido longamente usada pelo ser humano na resolução de problemas [Chm82, New94], o que torna o processo de manipulação desse tipo de estrutura mais intuitivo.

No entanto, o processo de catalogação de uma informação no repositório será mais dispendioso que em uma estrutura plana, dado que essa atividade envolve uma classificação prévia da informação em um grupo definido ou mesmo a criação de uma nova categoria de informações, tarefas estas que não são satisfatoriamente executadas sem a intervenção de um operador humano, tornando o processo oneroso.

Quanto à questão do controle de acesso, mesmo em ambientes controlados, como em redes privadas, é fundamental que haja algum tipo de mecanismo que permita a seleção de quais informações ficarão disponíveis aos diversos segmentos de usuários, de acordo com suas necessidades ou política de segurança¹ da organização. Informações sensíveis não podem ou não devem ser disponibilizadas sem que haja um controle sobre os usuários que as acessam.

Também é desejável que haja diversas “granularidades” no controle de acesso, permitindo que num nível mais elementar, esse controle seja feito sobre grupos de usuários e grupos de informações. De forma semelhante, é interessante, através de um “ajuste fino”, o controle individual sobre informações e usuários.

Com essas observações, concluímos que uma ferramenta de gerenciamento de informações a ser utilizado por um grupo limitado de usuários (grupo de trabalho, departamento), deve possuir as seguintes características:

- Comunicação 1-N;
- Organização hierárquica; e
- Controle de acesso.

A comunicação 1-N atuaria como um canal de distribuição. Uma vez que um usuário consiga uma informação relevante, ele pode distribuí-la aos interessados, evitando um trabalho redundante de busca pela mesma informação.

¹ Por “política de segurança”, entendemos um conjunto de regras relacionadas à acesso, distribuição e modificação de informações que deve ser implantado e seguido em circunstâncias que envolvam informações sensíveis.

Com uma organização hierárquica representando as estruturas de relações entre as informações armazenadas, facilita-se o processo de busca por uma informação específica, não sendo necessário pesquisar todo o repositório, mas apenas a hierarquia correspondente.

Com controle de acesso pode-se se permitir o acesso a certos conjuntos de informações apenas para determinados grupo de usuários, reduzindo problemas relacionados à segurança e integridade das informações.

2.6 Intranets

A despeito do tema “*intranets*” ter recebido muita atenção por parte da literatura especializada [Ben98, Hol96, Spa97] nos últimos anos, a solução não é nova, sendo bastante conhecida pelos profissionais da área de tecnologia em redes de computadores.

Uma *intranet* (Figura 2.1) é uma infraestrutura de comunicação. Ela é baseada nos padrões de comunicação da Internet e nos padrões de conteúdo da *Web*. Dessa forma, as ferramentas usadas para criar e usar uma *intranet* são idênticas às usadas para Internet e aplicações *Web*. A característica peculiar à *intranet* é que o acesso à informação publicada é restrita aos clientes de um grupo. Historicamente, isto tem sido conseguido através do uso de redes protegidas por *firewalls* [Tel96, Wha00].

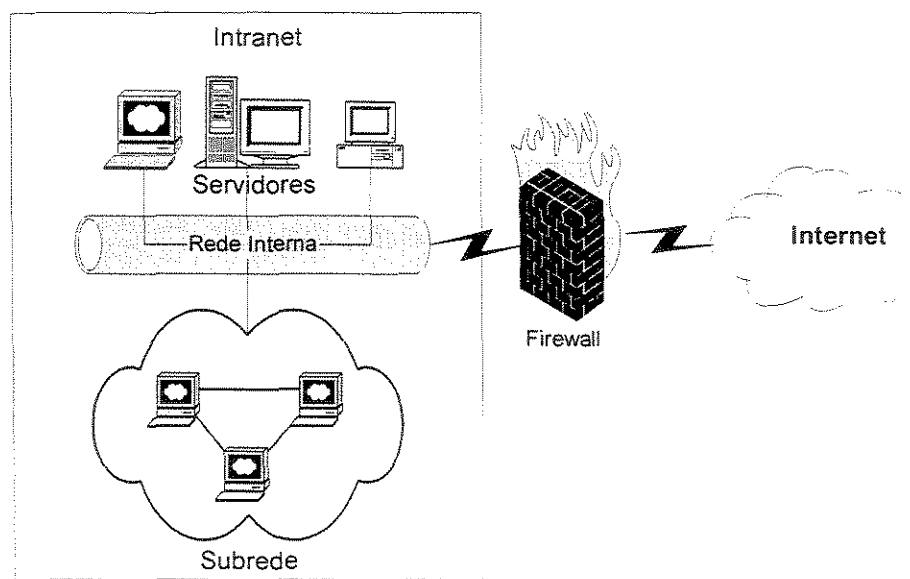


Figura 2.1 – Diagrama exemplificando uma *intranet*

Apesar da infraestrutura usada na *Web* sozinha não criar uma *intranet*, ela deve ser a interface central que os usuários empregam para acesso à *intranet*. Um navegador *Web* é fácil de aprender e simples de usar, bastando ativar um item para chegar a outra página. Para voltar à página anterior, simplesmente clica-se no botão “voltar”, ou seja, requer pouco ou nenhum treinamento para se usar e possibilita a uma grande gama de usuários, dos mais inexperientes aos mais técnicos, procurar por informações [Ric96+]. Os navegadores mais recentes ainda

provêem meios de acesso a uma série de aplicações não-*Web*, incluindo correio eletrônico e *newsgroups*. Todas estas vantagens podem ser totalmente aproveitadas em *intranets*.

Outra das grandes potencialidades das *intranets* está relacionada à redução da necessidade de papel. Como os navegadores utilizados para acessar a *intranet* podem ser usados em praticamente todas as plataformas de *software* e *hardware*, a mesma informação pode ser acessada por todos os usuários. Isto significa que relatórios, manuais de procedimentos, lista de telefones, materiais de treinamento, formulários, etc, podem ser colocados na forma eletrônica e constantemente atualizados a um custo mínimo. Um ponto a ser considerado é a manutenção da consistência dos documentos publicados e seu interrelacionamento, remoções ou alterações de documentos devem ser feitos de maneira a não criar *links* inconsistentes, documentos “orfãos”, etc.

O fato da informação ser apresentada da mesma forma virtualmente idêntica em qualquer computador ainda apresenta a vantagem de possibilitar que todos os computadores e programas de uma organização façam parte de um único sistema, no qual os usuários podem encontrar as informações independente da sua localização.

Outro ganho decorrente da aglomeração das aplicações é a facilidade de manutenção dos sistemas, não exigindo equipes de desenvolvedores geograficamente dispersas. Como os programas também podem ser executados em diferentes arquiteturas de *hardware*, não exigem equipes específicas para cada plataforma.

Por outro lado, um dos problemas é a forma quase artesanal de desenvolvimento dos sistemas para *intranets*, onde muitos são codificados em linguagens como Perl ou PHP. Se por um lado essas linguagens permitem uma grande flexibilidade podendo ser utilizados em uma grande variedade de plataformas, não possuem uma grande variedade de ferramentas avançadas de desenvolvimento.

Este cenário tem mudado ultimamente com ferramentas que são direcionadas especificamente para o desenvolvimento de sistemas para Internet/*intranets*. Ferramentas como ColdFusion¹, Visual InterDev², WebObjects³ e outros permitem um rápido desenvolvimento de aplicações e possuem uma série de bibliotecas que suprem as funcionalidades normalmente mais requeridas.

No entanto, esse enfoque ordinariamente tem a desvantagem de tornar os sistemas dependentes da ferramenta na qual foram construídas, visto que muitas utilizam tecnologia proprietária e não padronizada. Existe ainda o agravante que essas ferramentas normalmente suportam um leque muito estreito de plataformas, criando mais uma dependência.

Contudo, apenas ferramentas não são capazes de criar uma *intranet* útil. Além de servidores, redes de computadores, protocolos, uma *intranet* produtiva é antes de tudo conteúdo. Não é raro os profissionais responsáveis pelas informações que irão compor a *intranet* serem deixados ao largo na sua construção, visto que de modo geral não são especialistas em roteadores, domínios, TCP/IP, etc.

¹ <http://www.allaire.com/Products/coldfusion/>

² <http://msdn.microsoft.com/vinterdev/prodinfo/default.asp>

³ <http://www.apple.com/webobjects/>

As consequências vão desde uma *intranet* subutilizada a um aglomerado confuso de dados que irá sobrecarregar o usuário típico. Ou seja, outro problema que as *intranets* sofrem é o mesmo da Internet: excesso de dados. Elas podem ficar sobrecarregadas de dados que os usuários são obrigados a vasculhar para encontrar informações de interesse, sendo que uma *intranet* pela sua própria natureza deveria ser mais focalizada na organização, livre de elementos dispersivos e com o propósito de aumentar a eficiência da organização como um todo.

Esta sobrecarga normalmente ocorre devido à ausência de uma política para publicações de informações. Assim, uma análise detalhada de um problema publicada por um usuário que a considera de vital importância será apenas um obstáculo para outro. Multiplicado pelo número de pessoas utilizando o sistema isso pode se tornar um problema de grandes proporções. “Pelo menos dez milhões de dólares por ano são perdidos em produtividade por uma empresa de 10.000 funcionários devido a má navegabilidade e ausência de padrões de projeto. No mundo inteiro, o custo da má usabilidade de *intranets* irá atingir cerca de US\$ 100 bilhões no ano de 2001, caso não sejam criados melhores sistemas de navegação nem estabelecidos padrões de projetos mais rígidos” [Nil99].

No centro da questão, está o fato de que não há ferramental adequado para gerenciar de modo efetivo as informações na maioria das *intranets*. A simples definição de quais usuários podem publicar e quais podem modificar ou ter acesso a uma informação é custosa e normalmente envolve administradores de sistema que precisam modificar arquivos de configuração não acessíveis a usuários comuns.

Outro ponto ainda a ser resolvido é a vulnerabilidade à ataques externos e utilização indevida do sistema. Pelo fato de usar as mesmas tecnologias da *Web*, as *intranets* também estão sujeitas às mesmas deficiências de segurança que a Internet. As vulnerabilidades relacionadas à privacidade e integridade são ainda mais acentuadas, visto que as informações normalmente são sigilosas e estão direcionadas a um público restrito.

O próximo capítulo apresenta o sistema INEX que foi desenvolvido com o objetivo de resolver alguns dos problemas levantados em relação às tecnologias atualmente utilizadas para gerenciamento de informações.

CAPÍTULO 3

INEX

Conforme descrito no capítulo anterior, foi demonstrado que as soluções tecnológicas disponíveis não satisfazem os critérios para organizar e compartilhar informações de qualidade. A partir desse fato optou-se pela criação de um sistema totalmente novo que abordasse de forma efetiva o problema do gerenciamento de informações.

Inicialmente, foi feito um levantamento das características que um sistema para gerenciamento de informações deveria ter e chegou-se a uma lista de atributos desejáveis:

- Informações organizadas de forma hierárquica;
- Suporte a múltiplas visões;
- Supressão de redundâncias;
- Mecanismos avançados de busca;
- Controle de acesso;
- Possibilidade de postar comentários relacionados às informações;
- Controle sobre a validade de uma informação.

Com estes objetivos em mente foi concebido o INEX ou *Intranet Express*, um projeto do laboratório A-HAND que tem como objetivo o gerenciamento e a organização de informações digitais em *intranets* que são armazenadas em uma hierarquias de listas. Uma lista é uma abstração que agrupa itens sob algum aspecto similares e que são acessíveis a um ou mais grupos de usuários, sendo possível definir privilégios de acesso aos usuários individualmente.

Cada lista possui um administrador ou grupo de administradores responsável pelo gerenciamento dos itens nela contidas. Usuários poderão criar suas próprias visões das hierarquias, agrupando componentes de várias listas, suprimindo itens ou mesmo criando sua própria hierarquia de listas conforme suas necessidades.

Inicialmente, itens são componentes estáticos com um estrutura predefinida. No entanto, os usuários têm a possibilidade de estender os tipos básicos através de subclasses. Como tipos ou itens básicos entendemos listas, documentos e comentários (**Figura 3.1**).

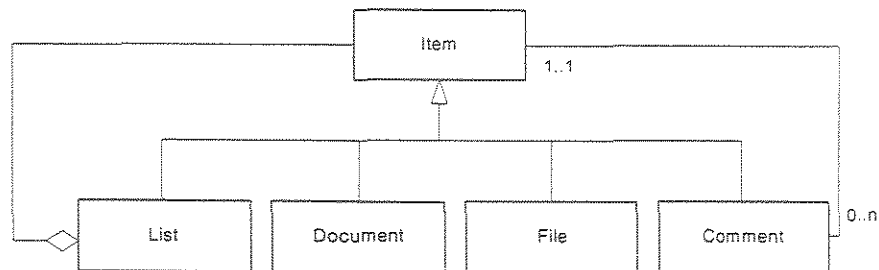


Figura 3.1 – Diagrama de classes básicas INEX

3.1 Descrição dos Tipos Básicos INEX

3.1.1 Item

Um Item é basicamente um conjunto de dados associado a uma ou mais listas dentro das hierarquias, sendo composto por dois grupos de elementos. O primeiro, utilizado para fins de gerenciamento, contém dados tais como: proprietário, data de criação, data de expiração da validade do item, permissões de acesso, etc. O segundo, com dados como autor, palavras-chave, resumo e outros, é empregado em critérios para buscas.

Estritamente falando, um Item básico não possui informação propriamente dita, apenas dados que permitem que a informação seja facilmente organizada e localizada. O papel de repositório de dados é representado pelas classes especializadas. Nesse sentido, a classe Item atua na realidade como uma metainformação para as classes que a estendem.

A classe Item também provê uma série de serviços básicos e uma interface bem definida que as classes especializadas poderão sobrecarregar caso a implementação corrente não seja satisfatória.

Dessa forma, as principais funções são previamente definidas e os serviços mais comumente utilizados e requisitados são implementados num nível mais alto da hierarquia de classes, facilitando o trabalho de desenvolvimento de novas extensões que não necessitam implementar estes serviços, dando assim ao desenvolvedor de extensões mais liberdade para focalizar seus esforços nos aspectos inerentes ao seu problema.

Outro benefício decorrente dessa abordagem é a uniformidade com que o servidor trata os diferentes tipos de itens que podem vir a ser criados, visto que todos devem implementar a mesma interface. Deste modo, as operações para busca, acesso e armazenamento são padronizados e não requerem alterações no servidor, mesmo que o tipo do item seja inteiramente novo.

3.1.2 List

É um repositório que armazena um conjunto de Itens agrupados a partir de algum critério. Uma lista está inserida dentro de uma ou mais hierarquias de tópicos relacionados. Associadas a

listas estão algumas informações a respeito de sua natureza e utilização, tais como um título e um preâmbulo.

Tipicamente hierarquias organizam listas através da especialização e generalização dos tópicos envolvidos. Dessa forma, sub-listas contém itens que tratam de assuntos que são subconjuntos ou especializações do tópico da lista onde estão inseridas.

Além disso, listas permitem a organização de informação em hierarquias superpostas. Cada usuário pode criar as suas próprias hierarquias de classificação de acordo com sua visão pessoal sem interferir nas hierarquias já estabelecidas ou nas visões de outros usuários.

Outra vantagem desta abordagem é o fato das listas oferecerem um conceito uniforme para a organização de informações de diversos tipos. Não importando o tipo de documento, a abstração é sempre a mesma: listas organizadas em uma estrutura hierárquica cuja “profundidade” indica o nível de especialização das informações.

A administração das hierarquias criadas pelo próprio usuário é de sua inteira responsabilidade, sendo o mesmo responsável pela organização, manutenção das listas e das informações nelas contidas.

3.1.3 Document

Um documento é uma entidade que pode conter uma grande variedade de informações na forma de texto simples, tais como um relatório ou uma lista de telefones.

3.1.4 File

O tipo File representa qualquer tipo de arquivo que se queira armazenar. O conteúdo deste tipo de item não é armazenado no sistema, apenas sua descrição é guardada. Seu armazenamento é feito diretamente no sistema de arquivos do sistema operacional na qual está sendo executado o servidor. A segurança destes arquivos deve ser assegurada através da garantia de acesso apenas através do INEX, não permitindo que os usuários possam acessá-los diretamente.

3.1.5 Comment

Se um usuário tem acesso a um item, ele poderá postar comentários sobre o conteúdo do mesmo. O nome e outras informações de identificação do usuário são automaticamente armazenados no sistema para fins de registro e acompanhamento.

Caso o item ao qual o comentário se refere seja removido, todos os comentários associados serão automaticamente descartados. Os comentários também podem ser removidos pelos usuários que os postaram ou por algum usuário com permissão de administração na lista na qual o item e por conseguinte o comentário está inserido, caso o conteúdo do comentário seja considerado inválido, impróprio, etc.

Comentários poderão ser usados como critérios para avaliação do valor de uma informação. Itens com mais comentários provavelmente serão mais relevantes e através das opiniões expressas, o usuário poderá considerar se a informação corresponde ao procurado sem a

necessidade de verificar o item. Essa abordagem pode ser interessante, por exemplo, nos casos de artigos muito extensos que demandem muito tempo para serem lidos.

Pelo autor da comentário, pode-se avaliar a relevância da observação: caso o autor seja um especialista na área em questão, possivelmente o seu comentário terá mais credibilidade para validar ou desacreditar determinada informação. No entanto, cabe ao usuário decidir pela credibilidade dos comentários emitidos.

3.2 Características do Sistema

Nesta seção apresentaremos as principais características do sistema, destacando como proporcionam um melhor gerenciamento de informação.

3.2.1 Extensibilidade

As ferramentas para gerenciamento de informações atuais geralmente manipulam documentos estáticos com uma estrutura predefinida. No entanto, as necessidades dos usuários mudam, assim como os tipos de documentos existentes. Pode ser necessário, por exemplo, adicionar um novo tipo de documento ou aperfeiçoar um já existente.

Assim, no INEX um item é uma entidade de dados genérica com os dados básicos que permitem a sua classificação em uma hierarquia e além disso, permite a criação de especializações que abordam aplicações específicas a serem implementadas. A tabela abaixo descreve os campos que compõem uma classe Item:

Atributo	Descrição
ID	Identificador único do item.
Type	Tipo do item
Subject	Assunto do item.
Description	Descrição resumida do item.
Author	Nome do autor do item.
Owner	Nome do usuário INEX que inseriu o item no sistema.
Creation	Data e hora da criação/inserção do item.
Modification	Data e hora da última atualização do item.
Lastaccess	Data e hora da última consulta ao item.
Expiration	Data e hora de expiração da validade das informações armazenadas no item.

Keywords	Conjunto de palavras-chave relacionadas ao item.
Permissions	Conjunto de permissões. Como estas permissões estão associados a uma relação entre o usuário atual e o Item, este conjunto é criado dinamicamente e varia para cada usuário.

Tabela 3-1 – Atributos de um tipo Item

A classe Java (linguagem escolhida para implementar o servidor) que define o tipo Item é apresentada a seguir¹:

```
package inex;
public class Item {
    private String ID;
    private String type;
    private String subject;
    private String description;
    private String author;
    private String owner;
    private java.sql.Timestamp creation;
    private java.sql.Timestamp modification;
    private java.sql.Timestamp lastaccess;
    private java.sql.Timestamp expiration;
    private Keyword[] keywords;
    private char[] permissions;
}
```

Figura 3.2 – Definição da classe Item

Para a criação de um tipo de item mais específico, basta estender a classe Item acrescentando os atributos necessários ao novo tipo. Assim, a classe File é uma extensão de Item, herdando todos os atributos de Item e definindo outros que são específicos da sua categoria:

```
public class File extends Item {
    private String originalName;
    private byte[] fileContent;
    private String fileType;
}
```

Figura 3.3 – Classe File

¹ Deve se notar que os códigos apresentado não correspondem à implementação real, sendo os códigos simplificados e as definições dos métodos suprimida, visto que não são necessários à compreensão das idéias expostas.

Com essa abordagem, a criação de itens de informação mais complexos não implica em altos custos relacionados à modificações e recompilação das classes já existentes. Basta estender as classes básicas e inserir novos atributos e métodos que contemplem o comportamento esperado.

Como o Item possui uma interface para sua manipulação bem definido, comportamentos que desviam desse conjunto básico são contemplados através de classes que são designadas especificamente para manipular os tipos que são extensões das classes básicas. Maiores detalhes sobre o tratamento a extensões serão descritos nos **Capítulos 4 e 5**.

Para exemplificar como poderia ser feita a extensão a uma classe básica, vamos supor que um usuário queira um sistema para controlar arquivos multimídia. Um grande problema é a variedade de tipos e formatos diferentes dos arquivos. Alguns dos atributos mais comuns associados a arquivos multimídia estão relacionados na tabela seguinte:

Atributo	Observação
File name	Cada plataforma (Unix, Mac e PC) possui regras diferentes para nomes de arquivos.
File size	O tamanho dos arquivos em <i>bytes</i> .
Last modification	Data da última modificação.
Description	Uma descrição do conteúdo do arquivo.
Keywords	Palavras-chave para auxiliar na busca.
File type or format	Existem vários tipos de arquivos e outros podem ser criados.
Image dimension	Tamanho da imagem para apresentação normal (em <i>pixels</i>).
Color depth	Número de <i>bits</i> para representar a cor de cada ponto.
Compression level	Alguns formatos de vídeo e imagens possuem múltiplos níveis de compressão.
Resolution	Resoluções em unidades reais.
Sampling rate	Arquivos de sons podem estar em mono ou estéreo e ter várias taxas de amostragem, normalmente 11 kHz, 22 kHz, 32 kHz, 44.1 kHz e 48 kHz.
Video frame rate	Algumas possibilidades são: 30, 29.97, 29, 25 e 24 fps (quadros por segundo).
Codec	Algoritmo de codificação (subcategoria de formato).
Source	Livro, CD, DVD ou outra fonte do material original.

Duration	Tempo normal de apresentação..
Plug-in location	Página <i>Web</i> onde pode ser encontrado o <i>plug-in</i> ¹ necessário para execução do arquivo.

Tabela 3-2 – Exemplo de atributos de objetos multimídia

Uma rápida análise na tabela permite verificar que existem claramente pelo menos três tipos de objetos - arquivos de vídeo, arquivos de som e arquivos de imagem. Cada tipo de mídia no entanto, tem atributos específicos que são inerentes apenas ao seu tipo. Tomando como exemplo o atributo que especifica a duração, ele é importante para um arquivo de som ou vídeo, mas não tem sentido associado a um arquivo de imagem.

Um solução para armazenar esses itens no INEX seria a criação de uma classe para cada tipo de mídia (imagem, vídeo, som) que estendesse a classe File, visto que ela possui como atributos alguns dos dados necessários e os serviços básicos para manipulação de arquivos. Assim teríamos o seguinte conjunto de classes:

<pre> public class Image extends File { private int imageHeight; private int imageWidth; private int colorDepth; private int compressionLevel; private int resolution; private String plugInLocation; } public class Sound extends File { private float samplingRate; private int length; private String plugInLocation; } </pre>	<pre> public class Video extends File { private int imageHeight; private int imageWidth; private int colorDepth; private int compressionLevel; private float samplingRate; private float frameRate; private String codec; private int duration; private String plugInLocation; } </pre>
--	---

Figura 3.4 – Exemplo de extensão das classes básicas INEX²

¹ *Plug-ins* são programas que podem ser incorporados a outros programas e que aumentam sua funcionalidade provendo funções não originalmente disponíveis.

² A classe Video poderia ser uma especialização das classes Image e Sound, no entanto, Java não suporta herança múltipla.

Dessa forma, o diagrama de classes apresentaria a seguinte configuração:

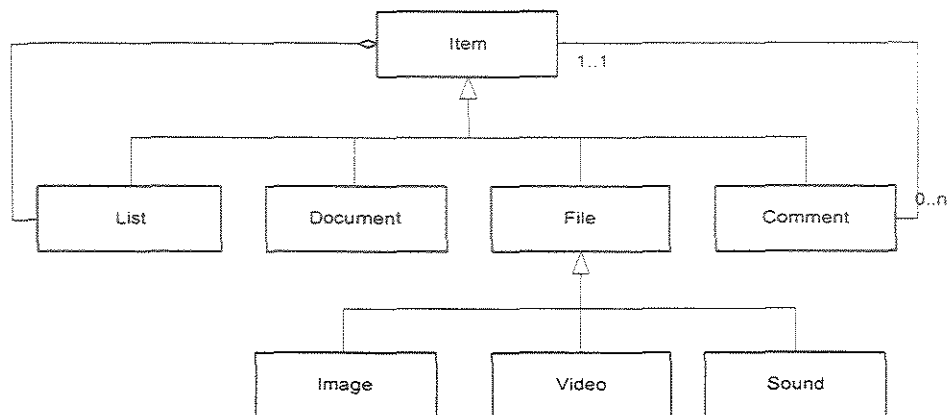


Figura 3.5 – Exemplo de extensão das classes INEX

Deve se observar, entretanto, que a simples criação de novas classes não é suficiente para tornar o sistema funcional. Ela deve vir acompanhada de mudanças nas classes que implementam a interface com o usuário para que haja o suporte necessário à representação dos novos tipos e visualização do seu conteúdo, assim como de eventuais itens associados.

No exemplo anterior, há uma clara necessidade de adicionar características à interface que possibilitem a exibição de filmes ou imagens e a execução de arquivos sonoros.

3.2.2 Flexibilidade

Pelo fato de suportar diversos tipos de documentos e possibilitar a adição de novas classes de informação, a estrutura do INEX é facilmente adaptável, mesmo para o gerenciamento de informações em sistemas preexistentes. Uma série de processos legados podem ter benefícios utilizando o INEX.

- Qualquer processo que envolva a produção, requisição, distribuição e atualização de informações dinâmicas que tenham sido tradicionalmente publicadas em papel. Exemplos incluem manuais de usuário, descrição de normas e procedimentos, relatórios, etc.
- Processos que envolvem a consolidação de informações de múltiplas fontes, comunicação e colaboração entre diversos usuários. Os ganhos podem ser ainda maiores caso os participantes estejam geograficamente dispersos.

Enfim, qualquer processo que envolva armazenamento, busca e requisição de informações pode ser automatizado em maior ou menor grau usando o INEX.

3.2.3 Independência de Plataforma

O servidor do sistema INEX foi implementado usando a linguagem Java [Fla97, Eck98], o que permite sua utilização em inúmeras plataformas de *hardware* e *software*. Como os programas Java não são compilados para o código nativo de uma arquitetura específica e sim para um conjunto de *bytecodes* que representam instruções para uma máquina virtual, o único requisito para se executar o servidor INEX em um ambiente é a disponibilidade de uma máquina virtual Java

para este ambiente. Atualmente existem máquinas virtuais para as plataformas mais populares, possibilitando a execução do servidor em virtualmente qualquer plataforma.

Em relação ao cliente, o requisito mais importante para sua utilização é a existência de um navegador que possua a capacidade de processar *applets* Java. A comunicação entre o cliente e o servidor é feita através de chamada a métodos remotos (RMI¹). O mecanismo de RMI permite a comunicação entre máquinas virtuais Java em diferentes servidores, possibilitando que referências a objetos em servidores remotos sejam usados como referências locais. O RMI permite a chamada de métodos em objetos remotos, a passagem de objetos como argumentos e o envio de objetos como valor de retorno.

A necessidade do navegador com suporte a *applets* deriva da necessidade de um *proxy*² entre o cliente e o servidor que seja responsável pelo mapeamento das chamadas dos clientes em chamadas remotas no servidor. No entanto, isto não deve ser encarado como um problema, visto que, mais de 90% dos navegadores utilizados possuem suporte a *applets* segundo dados da BrowserWatch³.

3.2.4 Gerenciamento Efetivo de Informações

O INEX pode ser configurado e utilizado na criação e administração de conjuntos de dados baseados em uma hierarquia de tópicos, permitindo gerenciar coleções de itens de informação (artigos, páginas *web*, e-mails, etc), organizando-as em uma estrutura hierárquica de assuntos e oferecendo mecanismos para armazenamento, recuperação e pesquisas sobre as mesmas.

A inserção de novas informações é feita apenas por usuários com permissão de escrita nas listas. Os usuários com permissão de escrita devem ter conhecimento da área da qual trata a lista, sendo responsáveis pela qualidade das informações inseridas.

A concessão de permissão para inserção de informações é feita pelos administradores de cada lista, sendo que estas permissões podem ser revogadas. Na configuração inicial do sistema, são definidos um ou mais usuários que serão os administradores e que estarão responsáveis por designar novos administradores para as hierarquias.

A pesquisa de uma informação pode envolver qualquer combinação dos atributos que compõem a classe Item, podendo ser utilizados também os atributos específicos das classes estendidas, desde que estas implementem os métodos necessários. Esta pesquisa pode ser local em apenas uma determinada lista ou pode ser global, envolvendo toda a base de dados, a única condição verificada é a permissão de acesso, sendo a pesquisa restrita aos itens aos quais o usuário tenha pelo menos acesso de leitura.

O controle de acesso sobre os itens é feito através de um sistema de listas de controle de acesso que definem quais permissões um determinado usuário possui sobre determinado item. As permissões podem ser também definidas em termos de grupo de usuários. Deve se notar,

¹ Remote Method Invocation - RMI

Uma descrição mais detalhada de RMI e *applets* é feita no Capítulo 4 - Detalhes do Sistema, onde são discutidas as tecnologias utilizadas.

² Um *proxy* neste contexto, é um processo que atua como intermediário na comunicação entre dois processos.

³ <http://browserwatch.internet.com/stats/stats.html>

entretanto, que as permissões individuais de cada usuário possuem prioridade sobre as permissões do grupo ao qual o usuário pertence.

Assim, se um usuário possui permissão de escrita sobre determinado item, mesmo que o grupo ao qual o usuário pertence tenha apenas permissão de leitura, o que prevalece é a permissão de escrita do usuário. Da mesma forma, permissões revogadas individualmente não são sobrepostas pelas permissões do grupo ou grupos aos quais pertence o usuário.

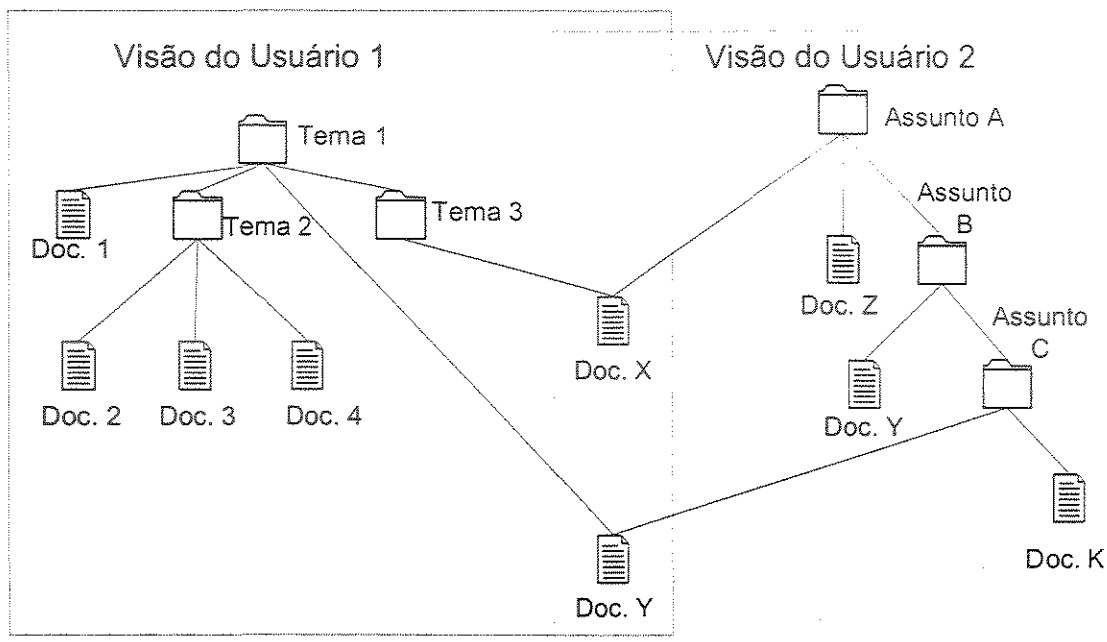


Figura 3.6 – Exemplo de hierarquia INEX

A arquitetura INEX permite a organização de diversas hierarquias de acordo com as necessidades de cada usuário (Figura 3.6), permitindo que os usuários classifiquem as informações da maneira mais natural e de acordo com sua cultura ou visão pessoal, visto que um engenheiro provavelmente não enxergará um conjunto de informações sob a mesma ótica que uma pessoa do departamento de *marketing*, por exemplo.

No exemplo da Figura 3.6, para o “Usuário 1”, a informação “Doc. X” se encaixa numa lista denominada “Tema 3”, no entanto, para o “Usuário 2”, esta mesma informação é mais natural ou conveniente que esteja localizada numa lista denominada “Assunto A”.

O usuário tem total controle sobre as suas hierarquias, sendo automaticamente designado administrador nas hierarquias por ele criadas.

Esta flexibilidade de organização das informações permite aos usuários criar “visões” pessoais, que são independentes e não interferem com as hierarquias já estabelecidas. Deve-se ressaltar, no entanto, que essa independência se refere à disposição das informações. A inserção e remoção de itens são operações que podem ter impacto em diversas hierarquias de diversos usuários.

A distribuição em uma estrutura hierárquica tem ainda o benefício de oferecer um conceito uniforme para a organização de informações de diversos tipos. Não importando o tipo de documento, a abstração é sempre a mesma: listas organizadas em uma estrutura hierárquica cuja profundidade indica o nível de especialização das informações. Conforme anteriormente citado, estruturas hierárquicas são um expediente longamente utilizado pelo ser humano para resolução de problemas e sua manipulação é bastante intuitiva.

Outra característica que permite melhorar o gerenciamento das informações consiste em um identificador único para cada informação que é armazenada. Isto evita que múltiplas cópias de um mesmo documento sejam armazenadas em diferentes lugares na hierarquia. Desta forma, mesmo que conceitualmente um documento possa aparecer em diversas visões ou hierarquias, há apenas uma única cópia armazenada que recebe diferentes referências. Note no exemplo da **Figura 3.6** que o item “Doc. Y” aparece em duas listas do “Usuário 2” e ainda é referenciado por uma terceira lista pertencente ao “Usuário 1”. Caso o item seja removido, todas as referências são automaticamente eliminadas. A remoção de um item pode ser feita pelo proprietário do item ou por um usuário com direitos de administração sobre a hierarquia na qual o item foi originalmente inserido.

A política de remoção de itens pode ser reavaliada caso se mostre confusa para o usuário. Uma possibilidade seria vetar ou prevenir os usuários sobre a remoção de itens que possuam referências em outras hierarquias.

Os administradores de uma lista são também responsáveis por definir quais usuários ou grupos de usuários podem ter acesso a determinadas listas ou itens e quais informações são relevantes ou devem ser descartadas. Dessa forma, minimiza-se o problema de sobrecarga de informações, havendo uma prévia seleção de quais informações podem ser úteis a determinados usuários ou grupos.

Isto sem dúvida, representa uma sobrecarga sobre os responsáveis pelas hierarquias; no entanto, os ganhos decorrentes da economia de tempo e eficiência provavelmente superam os custos. Esta abordagem envolvendo intervenção humana também tem a vantagem de levar a um maior envolvimento dos usuários que, cientes de que as informações passarão por uma avaliação, deverão se preocupar mais com a qualidade das informações que publicam.

Com esta abordagem, podemos esperar que os requisitos para informação de qualidade conforme definido no Capítulo 1 sejam atendidos.

A precisão e veracidade da informação deve ser garantida pelos próprios usuários e pelos administradores que são encarregados de administrar as listas, não havendo na versão atual um sistema automático de filtragem das informações.

O atributo que define a data de expiração de um item assegura que se está trabalhando com informação oportuna e válida para o período em questão.

A classificação em hierarquias e o controle de acesso com definição clara de quais usuários devem ter acesso às informações, proporciona informações com mais relevância, contribuindo para diminuir a “sobrecarga de informações”.

Hierarquias também possibilitam aos usuários classificar as informações de acordo com o nível de abstração desejado. Como os usuários podem definir sua própria hierarquia de

informações, informações mais genéricas podem ser inseridas em níveis mais elevados da hierarquia e informações mais especializadas em níveis inferiores da hierarquia, o nível de abstração é definido de acordo com a preferência pessoal do usuário.

Outra vantagem de hierarquias definidas de acordo com a preferência do usuários e com controle de acesso é a eficiência na recuperação das informações, o controle de acesso garante uma filtragem prévia das informações, não possibilitando acesso às informações não consideradas relevantes para o usuário em questão. A definição das hierarquias pelos próprios usuários garante uma maior familiaridade com a estrutura utilizada e facilidade de localizar as informações desejadas.

A utilização de comentários permite dar um dinamismo maior às informações, novas idéias podem ser acrescentadas e compartilhadas. Usuários com permissão de escrita podem alterar as próprias informações.

A possibilidade de estender os tipos básicos de informação acrescentando novas funcionalidades permite que novos formatos e tipos de dados sejam utilizados.

No próximo capítulo são apresentados as tecnologias utilizadas na implementação do sistema INEX e as vantagens oferecidas pelas tecnologias escolhidas.

Detalhes do Sistema

Neste capítulo apresentaremos alguns detalhes de implementação, a arquitetura do sistema, tecnologias utilizados no servidor e no cliente, motivações para a escolha dessas tecnologias e outros detalhes que serão importantes para melhor situar o capítulo seguinte.

4.1 Arquitetura INEX

O INEX foi implementado seguindo uma arquitetura de três camadas proposta por Orfali [Orf98+], com ligeiras modificações¹:

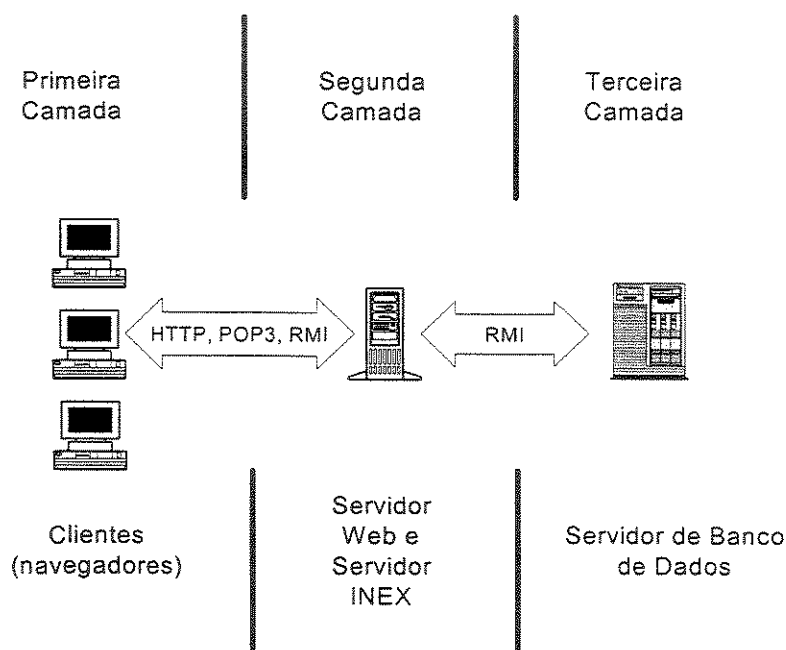


Figura 4.1 – Arquitetura INEX

- Primeira Camada

A primeira camada, ou cliente, são as aplicações em HTML, DHTML, JavaScript e Java com as quais o usuário interage. Esta camada é baseada em navegadores *Web*.

¹ No INEX é utilizado RMI em substituição à CORBA.

- Segunda Camada

A segunda camada é composta pelo *hardware* e *software* que ficam entre o cliente e os servidores de dados. O *software* inclui um servidor *Web* que recebe requisições do cliente e fornece os *applets* responsáveis pela comunicação com os servidores de banco de dados. Os clientes se comunicam com o servidor *Web* através de protocolos padrão, tais como HTTP. Esta camada também hospeda o servidor INEX, responsável por fornecer a infra-estrutura de serviços acessíveis aos clientes.

- Terceira Camada

A terceira camada é composta pelo *software* que provê os serviços de banco de dados, acessíveis através de conexões entre o servidor *Web* da segunda camada e um servidor de banco de dados. Note que a **Figura 4.1** descreve as três camadas dispostas em *hosts* distintos; no entanto, o modelo suporta implementações nas quais um *host* hospeda mais de uma camada, ou uma configuração onde uma camada pode estar distribuída por vários *hosts*.

4.2 Tecnologias Utilizadas no Cliente

Existe um grande conjunto de tecnologias suportadas por navegadores. Esta seção apresentará as principais tecnologias usadas no contexto do cliente, dispostas no diagrama abaixo (**Figura 4.2**).

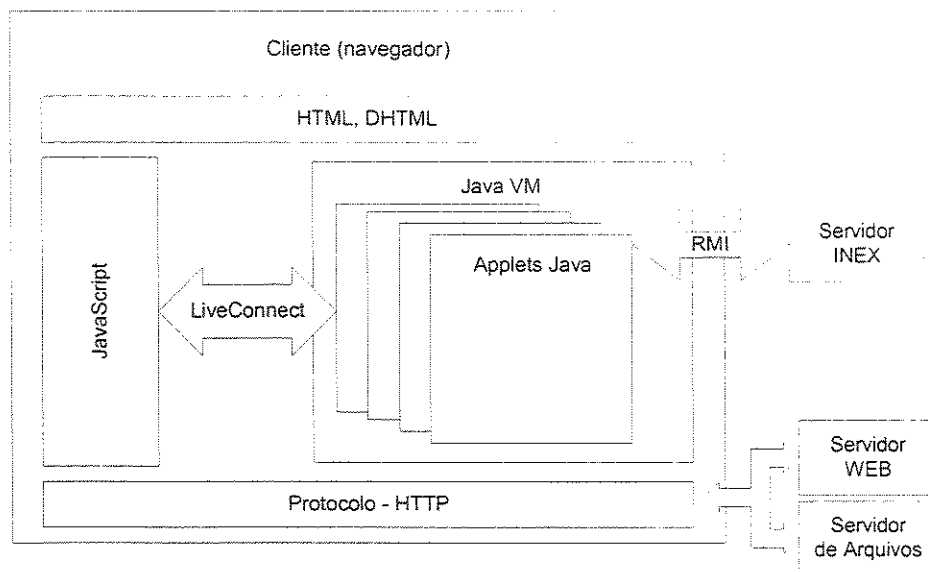


Figura 4.2 – Tecnologias utilizadas no cliente

4.2.1 HyperText Markup Language (HTML)

Um documento HTML é um arquivo texto na qual certos marcadores (*tags*) delimitam regiões do documento atribuindo significados especiais aos mesmos.

Algumas interfaces com o usuário (IU) são construídas usando apenas HTML. Este enfoque tem várias vantagens, como o tamanho reduzido dos arquivos e a possibilidade de utilizar ferramentas simples e baratas para a sua criação. Como a linguagem é composta apenas de marcadores, não necessita de programadores altamente qualificados para sua codificação.

A especificação HTML [W3C99a] define elementos de IU com suporte a texto com vários tipos e cores, listas, tabelas e formulários (campos de texto, botões, *checkbox*, *radiobuttons*). Estes elementos são adequados apenas para criar uma interface com o usuário em sistemas simples, visto que não dispõem da flexibilidade necessária em aplicações complexas. Dessa forma, HTML é combinado com outras tecnologias para se conseguir o resultado desejado.

Uma interação INEX típica inicia-se com o navegador requisitando a página inicial do sistema ao servidor *Web*. Esta página inicial é um documento HTML estático, contendo os *scripts* JavaScript, *applets* Java e DHTML que posteriormente serão utilizados para a geração da interface com o usuário.

4.2.2 JavaScript

JavaScript é uma linguagem interpretada independente de plataforma e baseada em objetos. Ela é composta de um núcleo de objetos comuns que são executados tanto no cliente como no servidor, assim como um conjunto de objetos específicos do navegador e do servidor. Os objetos do núcleo incluem *arrays*, datas, números e *strings*. Na parte cliente, existem documentos, formulários, *frames* e janelas. Estes objetos permitem a manipulação de documentos HTML (verificação de campos de formulários, submissão de formulários e a criação de páginas dinamicamente) e o controle do navegador (mudar a página corrente, mostrar mensagens, etc). JavaScript também permite que diversos eventos gerados pelo *mouse* e teclado sejam capturados e tratados.

Existem atualmente diversas versões de JavaScript. ECMAScript é uma versão que tem sido padronizada pela ECMA¹, baseada no JavaScript 1.1, inclui somente os objetos do núcleo JavaScript. Como não existem objetos específicos para o cliente ou o servidor, ou mesmo funções, estão sendo feitos esforços para definir o ECMAScript V2 que deverá resolver estas deficiências [IBM98a].

4.2.3 Dynamic HTML (DHTML)

DHTML é um conjunto de tecnologias que permitem uma grande flexibilidade no projeto e na apresentação da interface com o usuário. Em particular, DHTML inclui *Cascading Style Sheets* (CSS) que facilita definições uniformes para detalhes de apresentação das várias partes da página a ser criada. Estes elementos podem ser precisamente posicionados usando coordenadas absolutas.

Outra vantagem do DHTML é o aumento no nível de funcionalidade de uma página HTML através de um modelo de objetos do documento. Os objetos do documento permitem às linguagens de *scripting* (como JavaScript), o controle de partes da página HTML. Por exemplo, textos ou imagens podem ser movidos na tela, ocultos ou expostos de acordo com comandos

¹ European Computer Manufacturers Association

do *script*. Estes *scripts* também podem ser usados para validação dos dados de formulários sem a necessidade de consulta ao servidor.

Infelizmente, existem algumas desvantagens no uso de DHTML. A maior delas é o fato de existirem duas diferentes implementações do modelo de objetos, Netscape e Microsoft, que juntas detêm cerca de 90% do mercado de navegadores de acordo com dados do BrowserWatch¹. Para resolver estes problemas, um *Document Object Model* (DOM) foi definido pelo W3C [W3C99b] para padronizar as diferentes implementações. Este modelo prevê que qualquer objeto em um documento HTML pode ser consultado, modificado, removido ou adicionado.

Uma segunda desvantagem com DHTML deriva do fato de que apenas versões mais recentes dos navegadores oferecerem este suporte. Isto pode ser encarado como um problema menor, visto que o INEX é direcionado para *intranets*, onde os navegadores tendem a ser padronizados ou mesmo impostos pela gerência da organização.

4.2.4 RMI

O mecanismo de RMI permite que objetos em uma máquina virtual Java invoquem métodos de objetos em uma máquina virtual Java remota.

O arquitetura RMI consiste de três camadas: a camada *stub/skeleton*, a camada de referência remota e a camada de transporte.

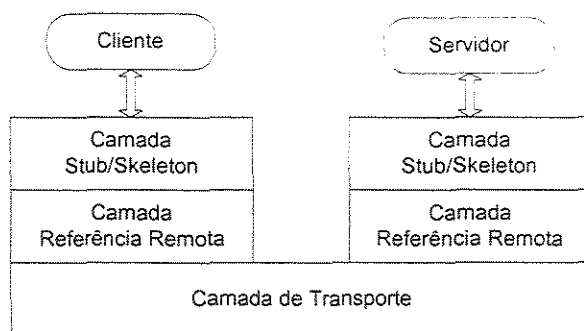


Figura 4.3 – Diagrama da arquitetura RMI

Cada camada é definida por uma interface e um protocolo específicos, sendo independentes entre si. Dessa forma, uma camada pode ser substituída por uma implementação alternativas sem reflexos em outras camadas.

A camada de *stubs/skeletons* fica diretamente abaixo da aplicação do usuário. O *stub* possui o conhecimento de como fazer a transferência de invocação de métodos entre os objetos participantes. O *skeleton* é uma classe auxiliar, sendo responsável por ler os parâmetros da invocação do método, fazer a invocação remota do método, receber o valor de retorno e repassá-lo para o *stub*.

A Camada de Referência Remota define e provê o suporte para a semântica da invocação da conexão RMI.

¹ <http://browserwatch.internet.com/stats/stats.html>

A Camada de Transporte faz efetivamente a conexão entre as máquinas virtuais. Todas as conexões são feitas sobre o protocolo TCP/IP.

4.2.5 Applets Java

Um *applet*, como o nome implica, é um tipo de mini-aplicação, projetado para ser executado por um navegador *Web*, ou no contexto de algum “visualizador de *applet*” [Fla97]. *Applets* oferecem a mais flexível das tecnologias de IU. Java provê um rico conjunto de elementos de IU que incluem o equivalente a cada um dos elementos HTML. Somando-se a isso, como Java é uma linguagem de programação, um conjunto infinito de elementos de IU podem ser construídos e usados. Existem também diversas bibliotecas que oferecem elementos de interface com o usuário tais como tabelas, planilhas, gráficos, editores, etc.

Uma desvantagem no uso de *applets* Java é a necessidade de suporte por parte dos navegadores. Os navegadores mais usados têm suporte a Java, mas apenas alguns têm suporte às versões mais recentes. Dessa forma, a versão utilizada no desenvolvimento do *applet* acaba impondo o navegador a ser utilizado pelos usuários.

Outra desvantagem dos *applets* é o fato do código fonte ser compilado para *bytecodes* e ser interpretado pela máquina virtual Java do navegador, isto torna sua execução muito mais lenta que código nativo da plataforma e, devido a razões de portabilidade, Java normalmente não faz uso de bibliotecas dinâmicas que estão disponíveis nas diversas plataformas.

As interfaces executam mais eficientemente se implementadas usando DHTML e JavaScript do que *applets*. Por outro lado, com *applets* é possível criar interfaces mais elaboradas com maior controle sobre os elementos gráficos.

A tabela a seguir sintetiza os principais pontos e as diferenças entre DHTML/JavaScript e *applets* Java:

	DHTML e JavaScript	Applets Java
Recursos gráficos	Poucos elementos de alto nível	Muitos elementos de baixo e alto nível
Criação dos elementos gráficos	Iniciados por DHTML e JavaScript, sendo interpretados e executados por bibliotecas gráficas do navegador	Iniciados por código Java (JVM) e executados por bibliotecas Java (JVM)
Bibliotecas gráficas	Código nativo suportado pelo navegador	Código Java em <i>bytecode</i> independente de plataforma, mais uma parte dependente de plataforma implementada pela própria JVM.
Processamento das bibliotecas gráficas	Rápido (código nativo)	Lento (<i>bytecode</i>)

Processamento de código não gráfico	Lento	Mais rápido que JavaScript pois o <i>bytecode</i> é parcialmente compilado (via JIT) e portanto mais eficiente
-------------------------------------	-------	--

Tabela 4-1 – Comparação DHTML+JavaScript com *applets* Java

4.2.6 LiveConnect

LiveConnect é uma tecnologia desenvolvida pela Netscape e envolve um conjunto de objetos que fazem parte do navegador e classes que podem ser incorporadas aos *applets* que permitem a comunicação entre *scripts* JavaScript e *applets* Java. Entre outras facilidades, LiveConnect permite [Net99]:

- Usar código JavaScript para acesso direto a variáveis, métodos, classes e *packages* Java.
- Controlar *applets* Java ou *plug-ins* com JavaScript.
- Usar código Java para acessar métodos e propriedades JavaScript.

4.3 Porque Utilizar Navegadores para Interface com o Usuário

O uso de navegadores para hospedar a interface do sistema com o usuário resulta em uma série de benefícios.

- Existem atualmente muitos navegadores para os sistemas operacionais mais usados, incluindo Windows, UNIX, OS/2 e MacOS.
- O ambiente operacional do autor de um documento, do servidor e do cliente são independentes entre si. Documentos escritos usando HTML e padrões relacionados, armazenados em um servidor HTTP, podem ser vistos por qualquer cliente, não importando o ambiente no qual foram criados ou onde estão armazenados.
- Como o sistema utiliza protocolos padronizados de comunicação tais como TCP/IP e HTTP, dispositivos simples como telefones celulares e PDAs¹ têm o potencial de acesso a uma aplicação *Web* da mesma forma que os navegadores em um computador.
- Usuários normalmente têm familiaridade com navegadores, facilitando o aprendizado na utilização do novo sistema.

Essa independência de plataforma, aliada à crescente popularização da *Web* e paralelamente à linguagem HTML e o protocolo HTTP, abrem novas perspectivas de acesso e distribuição de aplicações.

- O encapsulamento de aplicações dentro de páginas HTML permite a portabilidade dessas aplicações entre múltiplas plataformas de *software* e *hardware* sem a necessidade de recodificação.

¹ Personal Digital Assistant

- Como as aplicações ficam armazenadas em servidores, os custos de distribuição, atualização e correção de erros são menores que em aplicações que normalmente são armazenadas nos discos locais de cada máquina-cliente.
- A atualização das aplicações no servidor (ou um pequeno número de servidores) torna a aplicação atualizada instantaneamente acessível a todos os usuários sem os custos envolvidos na atualização individual do sistema para cada usuário e os problemas decorrentes de configuração e peculiaridades de cada sistema operacional ou plataforma de *hardware*. Todo o gerenciamento do sistema fica restrito aos servidores.

4.4 Servidor INEX

No servidor se encontra toda a lógica da aplicação e a semântica do funcionamento do sistema. O servidor provê um conjunto de funções para clientes definidas num alto nível de abstração, facilitando o desenvolvimento do cliente e, se necessário, a mudança para um cliente totalmente novo sem a necessidade de grandes mudanças no código da aplicação cliente.

Com o objetivo de isolar a semântica da aplicação de eventuais mudanças na implementação ou particularidades do banco de dados, salvaguardando os clientes de futuras modificações, o servidor foi dividido em uma camada de aplicação e uma camada de banco de dados que denominamos de *apLayer* e *dbLayer* respectivamente.

A *apLayer* contém a interface pública que será utilizada pelos clientes, sendo responsável por manipular os objetos recebidos e moldá-los no formato esperado pelos métodos da camada *dbLayer*. Da mesma forma, os dados recebidos da *dbLayer* são transformados nos objetos que são enviados aos clientes. As operações para manipulação de dados implementadas nessa camada fazem chamadas de alto nível para o gerenciamento de transações e o controle de acesso.

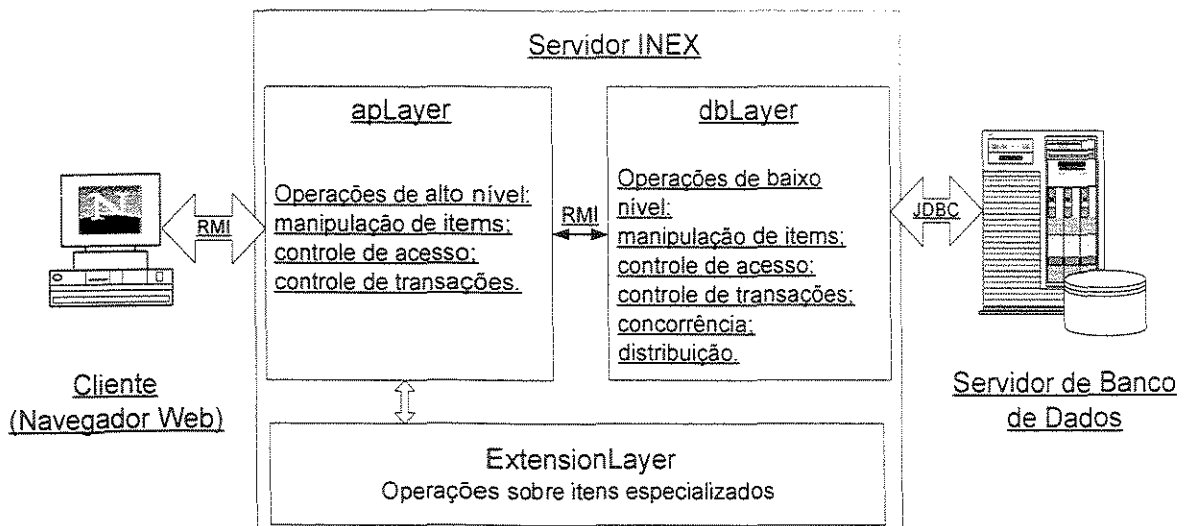


Figura 4.4 – Diagrama do servidor INEX

A `apLayer` garante a atomicidade das operações feitas no banco de dados através da API da `dbLayer` que implementa funções de mais baixo nível levando em consideração as características proprietárias do SGBD utilizado¹. A `apLayer` implementa um *mini-monitor* transacional que garante a disponibilidade da base de dados para os vários clientes do sistema. Antes de chamar operações não atômicas na `dbLayer`, a `apLayer` lhe indica que está iniciando uma transação. A `dbLayer` encarrega-se então de garantir que mecanismos de controle de concorrência sejam efetivamente utilizados no acesso ao SGBD.

A `dbLayer` isola a `apLayer` de eventuais mudanças de banco de dados e de suas particularidades. Mudanças na infra-estrutura implicarão na alteração da implementação da `dbLayer`, mas não em sua interface.

4.5 Persistência de Informações

O armazenamento das informações na implementação atual é feito sobre um conjunto de tabelas, armazenados em um banco de dados relacional, que são utilizados pelo Servidor INEX para persistência dos objetos que representam os itens INEX.

Ao ser solicitado o armazenamento de um item, o servidor, através de reflexão, irá determinar o nome da classe do objeto corrente e seus atributos e os armazenará na tabela correspondente. Há um mapeamento direto entre o nome das classes e o nome das tabelas no banco de dados que serão utilizadas para armazenar os atributos das instâncias dessas classes. Dessa forma, para armazenar objetos do tipo `Item`, há uma tabela “ITEM” no banco de dados (**Figura 4.5**) cujas colunas possuem os mesmos nomes e tipos compatíveis com os atributos da classe.

<pre>package inex; public class Item { private String ID; private String type; private String subject; private String description; private String author; private String owner; private Timestamp creation; private Timestamp modification; private Timestamp lastaccess; private Timestamp expiration; private Keyword[] keywords; private char[] permissions; }</pre>	<pre>CREATE SCHEMA INEX AUTHORIZATION INEX CREATE TABLE ITEM (ID CHAR(64) NOT NULL, TYPE CHAR(30) NOT NULL, SUBJECT CHAR(100) NOT NULL, DESCRIPTION VARCHAR(250), AUTHOR VARCHAR(100), OWNER CHAR(30) NOT NULL, CREATION TIMESTAMP NOT NULL, MODIFICATION TIMESTAMP, LASTACCESS TIMESTAMP, EXPIRATION TIMESTAMP NOT NULL)</pre>
---	--

Figura 4.5 – Mapeamento de classe INEX em tabela de banco de dados relacional.

¹ Na implementação corrente Oracle 8.0.5

No momento da consulta de um item, os dados armazenados são recuperados e a partir destes, uma instância do tipo do item é criada e devolvida pela camada de aplicação ao cliente.

Deve se notar que as permissões de acesso em relação aos itens são pertinentes a cada usuário e carregados no momento da recuperação do item em questão, dessa forma, elas não fazem parte dos dados das tabelas de armazenamento de itens, sendo armazenadas em tabelas relacionadas.

Da forma semelhante, as palavras-chaves relacionadas a cada item também são armazenadas em tabelas relacionadas e carregadas no momento da recuperação do item.

4.6 Camada de Extensão

Com a finalidade de permitir a manipulação de novas classes de informação que necessitem de tratamento específico, a INEX implementa uma camada de extensão que possibilita que as novas classes tenham um tratamento diferenciado e possam ter um comportamento diferente dos demais itens previamente definidos.

Esta camada segue o padrão *observer* [Gam94+], permitindo seu desacoplamento dos tipos de itens que futuramente sejam implementados. Dessa forma, para criar um novo tipo de item INEX que necessite de operações imprevistas basta que uma nova classe implemente a interface *ExtensionContract* (Figura 4.6) e fique responsável pela manipulação do novo item.

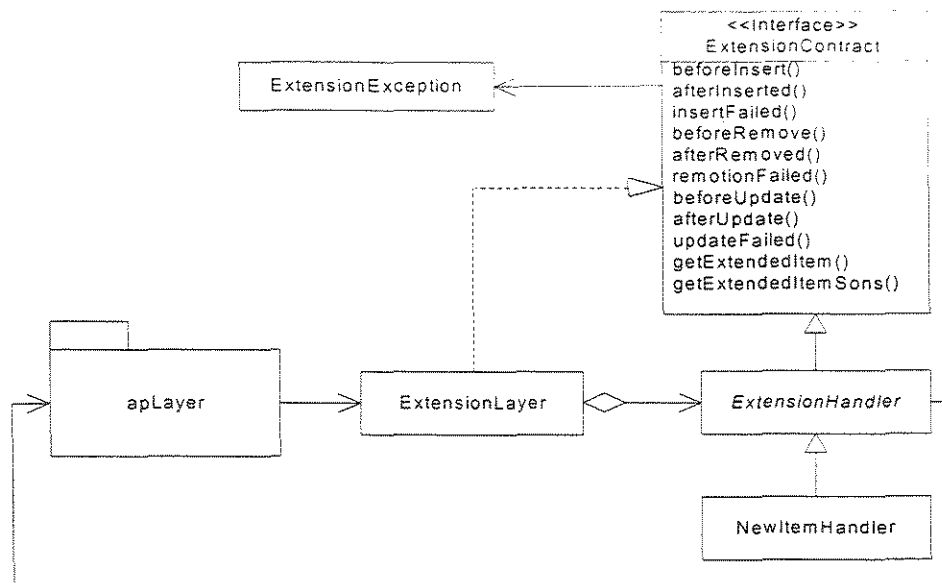


Figura 4.6 – Diagrama da camada de extensão¹

Como há uma referência da camada de extensão à *apLayer*, é possível aos novos itens o acesso aos serviços desta camada e todos os meios de que dispõem as classes básicas.

¹ Adaptado de Matos [Mat01].

A persistência de novos tipos de itens é feita sem a necessidade de alterações nas classes que implementam o servidor INEX. Conforme já descrito no tópico anterior, o armazenamento é feito utilizando-se reflexão, assim, concomitantemente com a criação de classes que implementam novos tipos, basta criar tabelas apropriadas capazes de armazenar corretamente os atributos dos novos objetos.

Conforme observado anteriormente, deve-se estender os módulos que implementam a interface com os usuários, para que seja dado suporte aos novos tipos de itens possibilitando a visualização ou representação das informações e sua correta manipulação.

4.7 Interação Entre as Diferentes Camadas

- Comunicação cliente/servidor

A comunicação entre o cliente e o servidor é feita através da invocação de métodos remotos usando RMI. Para isso, há no cliente, um *applet* que implementa a interface pública do servidor e recebe as solicitações do cliente repassando-as ao servidor através de RMI.

Este *applet* atua como um *proxy* que isola totalmente o cliente de detalhes como configurações de rede, endereço do servidor e outros detalhes de baixo nível. Assim, para o cliente as solicitações ao servidor são transparentes, e do ponto de vista lógico, as chamadas são locais, existindo virtualmente um servidor em cada cliente.

- Comunicação servidor INEX/servidor BD

A interação entre o servidor INEX e o servidor de banco de dados é feita através de JDBC¹, o que resulta em uma grande flexibilidade na escolha do banco de dados a ser usado.

As operações de alto nível utilizadas pela *apLayer* são mapeadas em chamadas JDBC ao banco de dados. “A API JDBC é uma API Java que permite acessar virtualmente qualquer tipo de dado em formato tabular [Whi99+]”. Ela é composta de um conjunto de classes e interfaces escritas em Java que provêem uma API padrão para desenvolvedores, dessa forma, o mesmo programa pode acessar qualquer base de dados, bastando que exista o *driver* correspondente.

4.8 Um Exemplo de Interação Típica

Numa interação típica, o fluxo normal de uma operação, pode ser estruturado sob o ponto de vista do usuário da seguinte forma:

1. O usuário entra na página inicial do INEX e digita sua conta e senha.
2. O sistema, após validar o login e senha, retorna uma relação de listas.
3. A partir do resultado retornado, o usuário pode solicitar uma operação sobre algum item.
4. Usuário verifica o resultado da operação.

¹ Java Database Connectivity

O fluxo de dados e controles que ocorrem no processamento são mostrados no diagrama (Figura 4.7) e tabela (Tabela 2-1) a seguir.

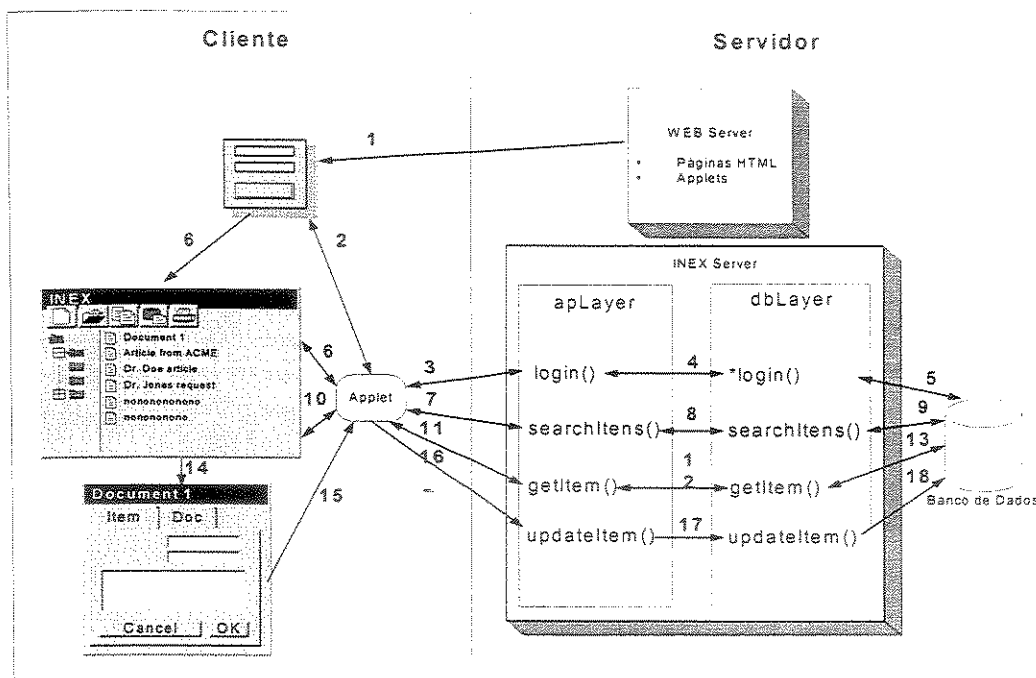


Figura 4.7 – Diagrama de fluxo de uma operação típica¹

#	Interação com usuário	Fluxo da Aplicação
1	Usuário solicita a página inicial do sistema.	O navegador do usuário solicita a página principal da INEX ao servidor <i>Web</i> que devolve a página inicial juntamente com os applets e scripts do sistema.
2	Usuário digita conta e senha	O código Javascript do formulário HTML, através de LiveConnect ou outra tecnologia para comunicação com <i>applets</i> , solicita validação de usuário e senha para o <i>applet</i> de conexão com servidor INEX.
3		O <i>applet</i> recebe os dados e através de RMI solicita ao servidor INEX validação do usuário e senha.
		A <i>apLayer</i> recebe a solicitação, converte em operações de baixo nível e repassa a chamada para a <i>dbLayer</i> através de RMI.
4		Usando JDBC, a camada <i>dbLayer</i> executa comandos

¹ Deve se observar que o diagrama foi simplificado para facilitar a visualização e que a correspondência um para um entre as chamadas da *apLayer* e a *dbLayer* não corresponde à implementação real, visto que uma chamada a um método na *apLayer* pode corresponder a várias chamadas de vários métodos na *dbLayer*.

		SQL para recuperar conta e senha do BD e validar o <i>login</i> do usuário.
5		Com os dados recuperados, a <i>dbLayer</i> valida o usuário, retornando o resultado para a <i>apLayer</i> que repassa ao <i>applet</i> , que por sua vez o devolve ao JavaScript.
6		O navegador abre a janela com tela principal do sistema e solicita listas para serem exibidas ao usuário.
7		O <i>applet</i> solicita as listas para o servidor.
8		<i>apLayer</i> recebe solicitação e repassa para a <i>dbLayer</i> .
9		<i>dbLayer</i> executa comandos SQL para recuperar as informações, os resultados são transferidos através das diversas camadas até o navegador que se encarrega de exibí-los.
10	Usuário solicita informações sobre um item.	Código JavaScript solicita informações detalhadas sobre o item ao <i>applet</i> de conexão.
11		O <i>applet</i> repassa a solicitação à <i>apLayer</i> .
12		A <i>apLayer</i> repassa a solicitação à <i>dbLayer</i> .
13		<i>dbLayer</i> executa comandos SQL para recuperar as informações, os resultados são transferidos através da <i>apLayer</i> até o <i>applet</i> .
14		Navegador abre uma nova janela com todas as informações.
15	Usuário altera alguma informação do item.	Código JavaScript solicita atualização das informações ao <i>applet</i> de conexão.
16		O <i>applet</i> repassa a solicitação à <i>apLayer</i> .
17		A <i>apLayer</i> repassa a solicitação à <i>dbLayer</i> .
18		<i>dbLayer</i> executa comandos SQL para atualizar as informações. Os dados alterados são atualizados no BD.

Tabela 4-2 – Descrição dos passos de uma interação típica.

Integração do Correio Eletrônico ao INEX

Com o intuito de validar as idéias propostas, apresentaremos o uso do INEX para o gerenciamento de mensagens de correio eletrônico, demonstrando assim, sua flexibilidade no tratamento de tipos de informação não previamente definidos.

O SMTP¹, protocolo utilizado para fazer a transmissão de mensagens de correio eletrônico, é um padrão bem definido e largamente utilizado. No entanto, não provê suporte a diversos campos que estão associados aos itens armazenados no INEX, por exemplo, hierarquias de assuntos ou comentários. Para resolver este problema, utilizamos um *gateway* do tipo *backbone* para fazer a conversão dos dados INEX para SMTP.

Esse *gateway* é responsável por encapsular os dados associados a um documento INEX em uma mensagem *multipart mime* e enviá-los através de um protocolo padrão, no caso POP3 ou IMAP, para um servidor de correio eletrônico. Este por sua vez, o envia ao destinatário através do protocolo SMTP conforme figura abaixo:

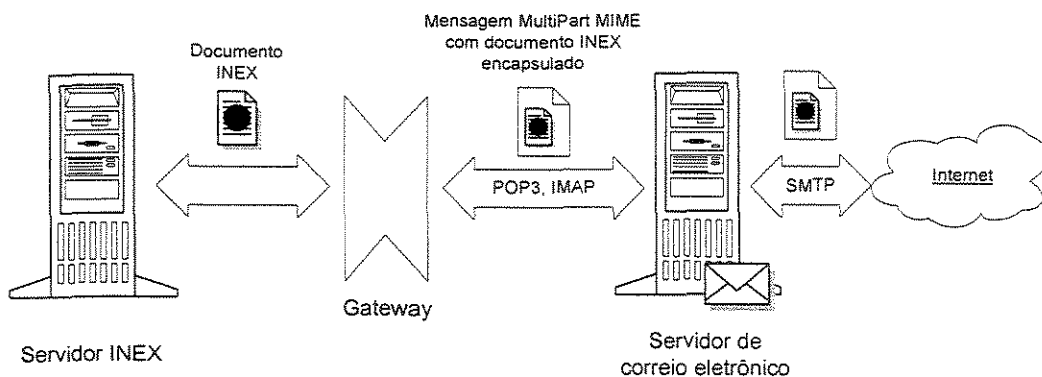


Figura 5.1 – Diagrama de comunicação entre correio eletrônico e o INEX

Quanto uma mensagem é recebida, o *gateway* é responsável por restaurar os dados encapsulados na mensagem para sua forma original, caso seja originário de um domínio INEX. O *gateway* é

¹ Simple Mail Transfer Protocol

² Um *gateway* do tipo *backbone* converte mensagens de um formato proprietário para um formato comum ou padrão e vice-versa.

implementado em Java e trabalha em estreita relação com o servidor INEX, usando a camada de extensão (ver **Capítulo 4**) para manipular dados do tipo Mailbox e EMail (**Figura 5.2**).

Quando o servidor INEX é executado, o módulo MailboxHandler fica responsável por gerenciar itens do tipo Mailbox e responde por toda manipulação deste tipo de item. Esse módulo compreende o *gateway* e toda a lógica necessária às operações com mensagens de correio eletrônico.

Ao ser aberta uma lista do tipo Mailbox, o MailBoxHandler solicita ao servidor de correio eletrônico uma lista de todas as mensagens correspondentes. Caso o usuário tenha optado por armazenar a senha de acesso à conta, uma *thread* é criada e fica responsável por monitorar o *mailbox* correspondente, periodicamente contatando o servidor de correio eletrônico e interpelando sobre a chegada de novas mensagens.

```
public class EMail extends
    inex.Item {
    private String received;
    private String destination;
    private String replyTo;
    private String errorsTo;
    private String cc;
    private String bcc;
    private String sender;
    private String messageID;
    private String date;
    private String status;
    private String mimeType;
    private String contentType;
    private String transferEncod;
    private int contentLength;
    private String attach;
    private byte[] content;
    private String xInexVersion;
    private String xOriginalName;
    private String xOriginalListID;
    private String xOriginalItemID;
    private String xMailer;
    private String xPriority;
}
```

```
public class Mailbox extends
    inex.List {
    private String eMail;
    private String accountName;
    private String replyTo;
    private String outServer;
    private String inServer;
    private int inServerPort;
    private String inServerUsername;
    private String inServerPassword;
    private String inServerType;
    private String leaveMsgOnServer;
    private String storePassword;
    private int checkServerInterval;
    private int outServerPort;
    private String outServerUsername;
    private String outServerPassword;
}
```

Figura 5.2 – Classes Email e Mailbox

Caso haja uma nova mensagem, a *thread* solicita-a ao servidor e a converte em um item do tipo EMail, inserindo-a na lista (Mailbox) correspondente. O processo de “conversão” de *email* em um EMail envolve a separação do cabeçalho do corpo da mensagem e a fragmentação em suas partes componentes que são então utilizados na criação do cabeçalho do item INEX. O corpo da mensagem é armazenado sem nenhuma modificação.

A RFC 822 [RFC822] descreve de forma detalhada como deve ser o formato dos cabeçalhos das mensagens Internet, mas não faz maiores considerações sobre o corpo da mensagem, assumindo que é composto apenas por texto usando o formato ASCII¹. No entanto, com o surgimento de novos tipos de dados além de texto simples, foram necessárias extensões, padronizadas por um conjunto de documentos coletivamente chamados de *Multipurpose Internet Mail Extensions* (MIME) [RFC2045, RFC2046, RFC2047, RFC2048, RFC2049].

O MIME prevê novos tipos de informações que podem ser transmitidas através de correio eletrônico e possibilita novas extensões, acomodando novos tipos de objetos, conjuntos de símbolos e métodos de acesso que têm a possibilidade de serem padronizados através de registro junto à IANA². Assim, todo documento enviado através do INEX utiliza o tipo MIME “multipart/mixed”, assegurando sua conformidade aos padrões.

Um item do tipo EMail é semelhante a qualquer outro *email* padrão utilizado na Internet, utilizando protocolos e infra-estrutura padrões para ser transferido de um servidor a outro. O que distingue um item EMail de uma mensagem de correio eletrônico comum é o cabeçalho, no qual são acrescentados um conjunto de campos usados para controle por parte do sistema INEX. A definição de atributos da classe Java que implementa um item do tipo EMail é mostrada na **Figura 5.2**.

Pode se identificar os atributos comuns a todos os *emails* e os atributos que são inerentes apenas aos itens EMail do INEX que são precedidos pela letra “x”, os quais, durante o processo de envio são convertidos em campos do tipo “X-”, indicando um campo experimental [RFC2048].

Dessa forma, aos atributos padrão dos cabeçalhos dos emails, foram acrescentados os seguintes tipos “experimentais” descritos na tabela a seguir:

XInexVersion	Indica a versão da Inex utilizada.
xInexOriginalName	O nome original do item.
xInexOriginalItemID	O ID utilizado pelo item.
xInexOriginalListID	A identificação da lista na qual o item se localizava.
Xmailer	O programa responsável pelo envio do item.
Xpriority	A prioridade da mensagem.

Tabela 5-1 – Atributos de itens do tipo EMail

¹ American Standard Code for Information Interchange

² Internet Assigned Numbers Authority

Esses atributos permitem rastrear os itens que são trocados através de redes com INEX e criar relacionamentos entre diversas mensagens. O campo “References” [RFC822] não foi utilizado para esta função por ser opcional e não prover toda a funcionalidade requerida para rastrear os itens, não sendo possível por exemplo, determinar a lista de origem da mensagem.

Deve-se ressaltar que nem todos os atributos da classe EMail são convertidos em campos de uma mensagem de correio eletrônico: alguns atributos como attach e status, são apenas utilizados para controle interno do sistema.

5.1 Integração com a Camada de Extensão

O suporte aos tipos EMail e Mailbox é feito através das classes de extensão EMailHandler e MailboxHandler, que estendem a classe ExtensionHandler e portanto, indiretamente implementam a interface ExtensionContract (Figura 5.3).

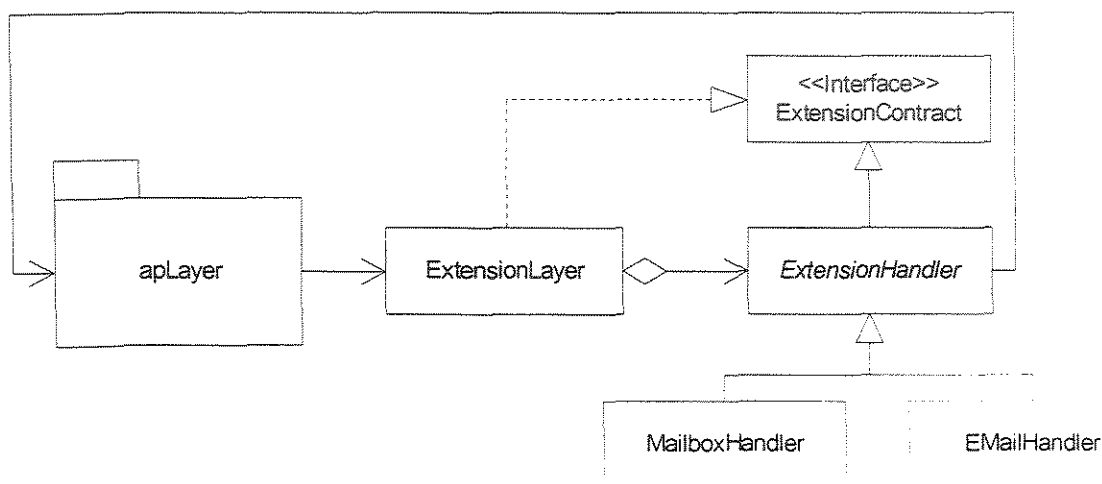


Figura 5.3 – Diagrama das classes de extensão para Mailbox e EMail¹

O arquivo de configuração `inex_extension.ini` (Figura 5.4) define quais são as classes estendidas e suas respectivas classes de manipulação. Este arquivo também define quais métodos (`insert`, `update`, etc) deverão ter um tratamento diferenciado e assim, suas chamadas são repassadas às classes de extensão que implementam o comportamento desejado.

¹ Adaptado de Matos [Mat01].

```

# definicao da classe de extensao MailboxHandler
mailbox.class      = inex.common.Mailbox
mailbox.handler    = inex.server.MailboxHandler
mailbox.insert     = false
mailbox.update     = false
mailbox.remove     = false
mailbox.get        = false
mailbox.children   = true

#definicao da classe de extensao EmailHandler
email.class        = inex.common.EMail
email.handler      = inex.server.EmailHandler
email.insert       = true
email.update       = false
email.remove       = true
email.get          = false
email.children     = false

```

Figura 5.4 – Arquivo de configuração para itens do tipo Mailbox e EMail [Mat01]

5.2 Recebimento de Mensagens

Ao solicitar o conteúdo de uma lista do tipo Mailbox, o usuário dá início a uma série de tarefas que serão delegadas às classes de extensão, que por sua vez se encarregarão de contactar o servidor de correio eletrônico através do protocolo POP3 ou IMAP, solicitando o envio das mensagens armazenadas (Figura 5.5).

No Apêndice A encontra-se uma breve descrição das classes definidas em `javax.mail` que são utilizadas nos exemplos desta seção (Figura 5.5, Figura 5.6).

Caso o usuário tenha optado por manter a sua senha de acesso ao servidor de correio eletrônico armazenada no próprio Mailbox, o processo de envio é totalmente automatizado, não exigindo nenhuma intervenção do usuário, caso contrário, apenas será solicitada a senha de acesso para permitir a conexão ao servidor.

Se uma mensagem possui um único destinatário, a mensagem é recebida, transformada em um item do tipo EMail e armazenada na lista Mailbox correspondente.

Caso uma mensagem possua mais de um destinatário, apenas uma única cópia é armazenada em uma área do sistema e diversos *links* são criados possibilitando que os destinatários tenham acesso à mensagem.

A garantia de que apenas uma única mensagem é armazenada, mesmo que possua diversos destinatários, é feita através da verificação do campo MESSAGE-ID, que contém um identificador único referente à versão corrente da mensagem, sendo gerado pelo servidor na qual a mensagem foi originalmente postada [RFC822].

```
public class MailboxHandler extends ExtensionHandler {
    (...)
    public getExtendedItemChildren(Mailbox mbox) {
        (...)
        // conecta-se ao servidor de correio eletronico
        store.connect();
        inbox = store.getFolder("INBOX");
        count = inbox.getMessageCount();
        // solicita as mensagens
        for (int i = 1; i <= count; i++) {
            message = (MimeMessage)inbox.getMessage(i);
            eMail = new EMail(message);
            // se ha mais de um destinatario
            if ((message.getRecipients(RecipientType.TO)).length() > 1) {
                // se a mensagem ja esta armazenada na area comum,
                // apenas um link e' criado
                if (server.searchItems(eMail, criteria, commonMbox.ID) {
                    Link link = new Link(eMail);
                    server.addItem(commonMbox.ID, link);
                } else
                    // caso contrario insere a mensagem na area comun
                    inxServer.addItem(commonMbox.ID, eMail);
            } // se ha apenas um destinatario, a mensagem
            // e' armazenada no proprio Mailbox
            } else
                inxServer.addItem(mbox.ID, eMail);
        }
        (...)
    }
}
```

Figura 5.5 – Recebimento de mensagens

5.3 Envio de Mensagens

O processo de envio de uma mensagem de correio eletrônico é feito através da inserção de um item do tipo Email em uma lista do tipo Mailbox. O processo é transparente para o usuário e totalmente similar à criação de qualquer outro tipo de item INEX.

Internamente, todo o processo de conexão e envio de mensagens para o servidor é executado pela classe MailboxHandler.

Antes do envio, o item EMail é transformado (Figura 5.6) em uma instância da classe MimeMessage (ver Apêndice A) que atende a todos os requisitos de um mensagem padrão e portanto pode ser utilizado em ambientes fora do INEX.

```
public class MailboxHandler extends ExtensionHandler {
    (...)
    public afterInserted(Mailbox mbox, Email inexEmail) {
        (...)
        Properties props = new Properties();
        // define o servidor smtp
        props.put("mail.smtp.host", mbox.getOutServer());
        // cria uma nova sessao
        Session session = Session.getDefaultInstance(props(), null);
        // cria uma nova mensagem
        MimeMessage msg = new MimeMessage(session);
        // preenche os campos do cabeçalho
        inexEmail.putHeaders(msg);
        // preenche os campos inex (x-inex...) do cabeçalho
        inexEmail.putInexHeaders(msg);
        // preenche o conteúdo da mensagem
        MyMultipartDataSource ds = new
            MyMultipartDataSource(inexEmail.getContent());
        MimeMultipart mp = new MimeMultipart(ds);
        msg.setContent(mp);
        // send the message
        Transport.send(msg);
    }
}
```

Figura 5.6 – Envio de mensagens

5.4 Armazenamento de Mensagens

As mensagens recebidas ou enviadas são armazenadas no banco de dados, como qualquer outro tipo de item do INEX. Como o corpo da mensagem é armazenado sem nenhuma modificação, o processo de recriar um *email* a partir desse item EMail é bastante simples, bastando que os cabeçalhos sejam preenchidos e o texto do corpo anexado à mensagem.

O armazenamento em banco de dados permite aos usuários buscas por todo o texto da mensagem (assumindo-se que a maior parte das mensagens de correio eletrônico são compostos por textos simples), o que flexibiliza a tarefa de procurar uma determinada mensagem, visto que parte das ferramentas não dispõem dessa facilidade. O fato de usar um banco de dados permite também a criação de índices que otimizam o processo de busca.

5.5 Utilização de INEX Emulando *Newsgroups*

Uma hierarquia do INEX poderia ser usada como um canal de distribuição de mensagens semelhantes à *newsgroups*. Como as informações são armazenadas hierarquicamente, os *newsgroups* poderiam ser criados utilizando listas que representariam a hierarquia de assuntos.

Esta abordagem possui todas as vantagens de que desfrutam as listas Mailbox, ou seja, haveria recursos para buscas por diversos campos, normalmente a busca seria muito mais rápida que em clientes normais, visto que as mensagens teriam a opção de armazenamento em banco de dados, dispondo de todos os recursos oferecidos pelo banco utilizado. Este enfoque possui diversas questões a serem resolvidas; por exemplo, ausência na presente versão de integração com um servidor de *newsgroups*, dessa forma, as discussões seriam limitadas ao grupo de usuários do INEX; mesmo recursos simples, mas bastante úteis, como por exemplo suporte a *threads*¹, não estariam disponíveis. Para resolver estas questões, seriam necessárias alterações no cliente e na forma como são criados relacionamentos entre diversas mensagens.

Considerando estes pontos e o fato do suporte a protocolos adicionais poder ser facilmente implementado, outra abordagem para criar o suporte a *newsgroups* seria através da implementação do próprio protocolo NNTP² e a criação de novos tipos que atendessem a esse protocolo.

Dessa forma, haveria uma lista “NewsBox” que poderia conter os *newsgroups* assinados pelo usuário com toda a lógica e semântica para se conectar a um servidor NNTP e solicitar as mensagens correspondentes. Com esse enfoque, funcionalidades que fossem levantadas pelos usuários também poderiam ser implementadas.

5.6 INEX como Gerenciador de Listas de Discussão

O INEX pode simular uma lista de discussão através de listas do tipo Mailbox que são acessíveis a um grupo de usuários que seriam os “assinantes” da lista. Para isso, basta criar uma lista do tipo Mailbox com permissões de escrita e leitura a todos os participantes. Dessa forma, um documento enviado a essa lista se tornaria acessível a todos e não haveria a necessidade de se criar uma cópia do documento para cada um dos participantes. A única restrição seria quanto aos usuários que poderiam participar da lista de discussão, os quais, obrigatoriamente teriam que ser usuários do domínio INEX em questão.

Um dos problemas dessa abordagem é a ausência de suporte por parte do cliente para a criação de *threads* que permitiriam um melhor acompanhamento das discussões pelos usuários. Uma organização de mensagens na forma de *threads* permite contextualizar uma mensagem e o ambiente na qual ela se insere. O acesso às mensagens através de *threads* permite ao usuário selecionar qualquer mensagem, consultar todas as relacionadas e dessa forma estabelecer o contexto de cada uma dessas mensagens. Esta funcionalidade é equivalente a ter todo o histórico de uma discussão em uma única mensagem [Whi96+].

O suporte a *threads* no INEX poderia ser conseguido com pequenas alterações nas classes EMail, adicionando campos que permitissem criar relações entre uma mensagem de resposta com a mensagem a qual esta se refere. Para a exibição deste tipo de organização, um cliente

¹ Neste contexto, por *thread*, entendemos um sequência de mensagens em resposta a uma questão inicial, possibilitando a um usuário seguir esta “corrente” de mensagens e acompanhar o desenvolvimento da discussão.

² Network News Transfer Protocol.

O protocolo NNTP é muito similar ao protocolo para envio e recebimento de correio eletrônico. Seu funcionamento é baseado em servidores que são os responsáveis pela transferência das mensagens e por clientes que se conectam a esses servidores e solicitam a leitura e envio de mensagens, entre outros serviços.

que modificasse a forma de apresentar os itens seria adequado, visto que a implementação atual não possibilita uma estrutura hierárquica entre itens, apenas entre listas ou tipos derivados de listas.

No entanto, essa solução funcionaria apenas no contexto do INEX, visto que não há campos homologados para esta função. Dessa forma, qualquer campo adicionado no cabeçalho que fosse usado nesse sentido seria normalmente removido pelo programa cliente do usuário externo.

5.7 Testes de Desempenho

Considerando o fato de que as mensagens de correio eletrônico recebidas são automaticamente processadas e armazenadas no sistema, não exigindo intervenção humana, sob a perspectiva do usuário, a tarefa de aquisição de uma informação consiste na pesquisa por esta informação no conjunto de mensagens armazenadas. Sob esta ótica, a questão relevante é o tempo gasto na busca.

Dessa forma, foram feitas medições para se determinar o tempo necessário para localizar mensagens que contivessem um termo específico e que estivessem em um pasta predefinida. Estas medições foram produzidas através de um programa simulando uma operação de pesquisa sendo conduzida por um usuário típico.

Não foram feitos testes utilizando usuários reais devido à instabilidade do sistema cliente. Muitas das tecnologias utilizadas no cliente eram recentes e necessitavam de uma maior maturação e robustez.

O ambiente de teste usado para avaliação do desempenho do sistema foi composto de diversas máquinas-cliente e um servidor cujas especificações são dadas a seguir, conectados através de uma rede Fast Ethernet 100Mbit:

	Clientes	Servidor
Processador	AMD Duron 600/750 Mhz	AMD Athlon 1000 Mhz
Placa Mãe	ASUS A7V133	ASUS A7V133
Memória	256 Mb PC133	512Mb PC133
Disco Rígido	40 Gb 5400RPM	2x40Gb 7200 RPM (RAID 0)
Sistema Operacional	Windows 2000	Linux (RedHat 6.2)
Base de Dados		Oracle 8.1.7

Tabela 5-2 – Especificação das máquina usadas nos testes de desempenho.

Foram utilizados de uma a cinco máquinas clientes, em cada máquina cliente foram criadas de uma a cinco instâncias do programa de simulação.

Deve-se observar que o teste aplicado não está considerando o tempo necessário para o usuário escolher um Mailbox que o mesmo considere mais apropriado para iniciar uma busca. De acordo com estudos conduzidos por Bälter [Bäl00], esta seleção demanda entre alguns milissegundos a 10 segundos, dependendo do número de opções disponíveis¹.

No entanto, como a seleção é feita de forma interativa pelo usuário, o tempo gasto nesta atividade não deve ser “percebido” pelo usuário, apenas após o início da busca da informação pelo sistema, o tempo decorrido será considerado e terá impacto na satisfação do mesmo com o desempenho do sistema. Segundo Hansen [Han98], a atividade de selecionar “alvos” para a pesquisa de uma informação contribui para satisfação do usuário com o sistema, pois permite um maior controle sobre a interação e uma participação mais efetiva.

Inicialmente, foram coletadas mensagens enviadas em uma lista de discussão², sendo posteriormente distribuídas entre diversos Mailboxes. Esta distribuição foi feita de forma a criar diversos Mailboxes contendo entre dez a quinhentas mensagens, possibilitando avaliar o tempo necessário em diversas situações no tocante ao universo de mensagens a ser pesquisado.

O maior conjunto utilizado nos testes, composto de quinhentas mensagens, totalizou 1.6 Mb de dados, com uma média de 3415 caracteres por mensagem, com a menor mensagem sendo composta de 615 caracteres e a maior mensagem sendo composta de 13391 caracteres.

Para simular a pesquisa, foi desenvolvido um programa que solicitou ao sistema a busca exaustiva por um termo nos diversos Mailboxes. As simulações foram executadas um número predeterminado de vezes³, sendo calculado a média aritmética simples dos tempos gastos em cada simulação. Deste modo, procurou-se evitar resultados distorcidos por algum fato incomum.

As simulações foram feitas utilizando entre um a vinte e cinco clientes simultâneos. Os resultados dos tempos obtidos nestas simulações estão expostos na Figura 5.7.

¹ No estudo foram considerados ambientes com mensagens subdivididas em no máximo cinquenta pastas.

² A lista de discussão Linux-BR (linux-br@bazar.conectiva.com.br) é uma lista destinada à discussão de sistemas operacionais baseados no Linux.

³ Cada conjunto de simulações foi executado vinte vezes.

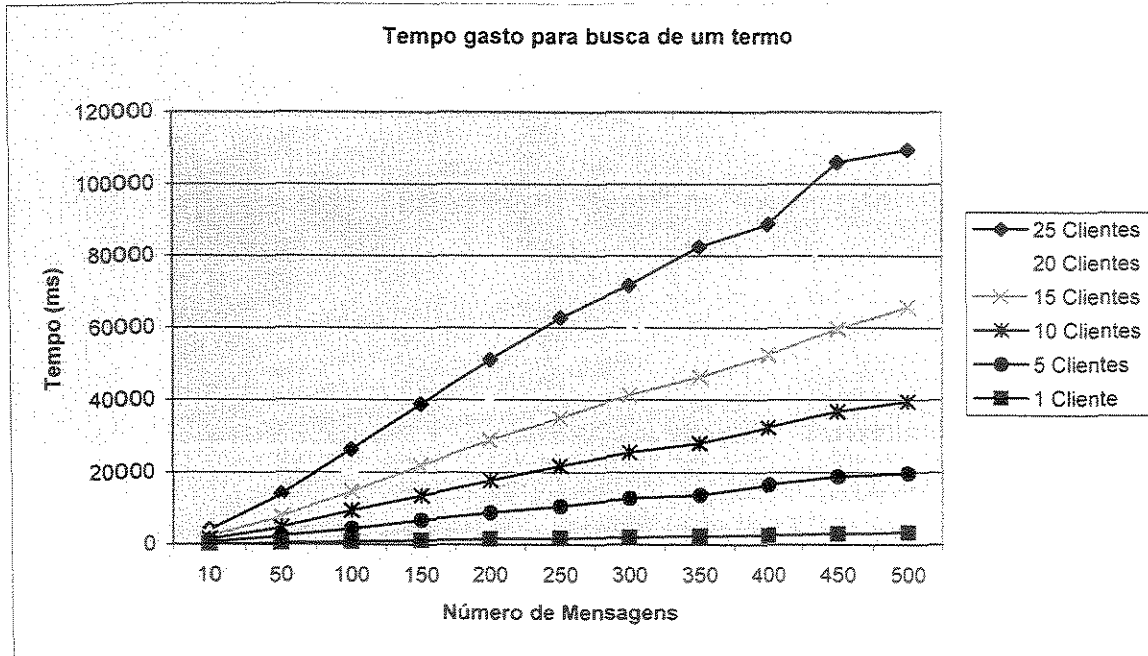


Figura 5.7 – Tempo gasto para pesquisa de um termo.

A partir do gráfico, observa-se um aumento quase linear no tempo necessário para fazer a busca, sendo a progressão, diretamente proporcional ao número de mensagens em cada Mailbox e ao número de clientes simultâneos.

Não se observa uma degradação irregular do tempo necessário para a pesquisa ou algum tipo de comportamento anômalo do sistema.

No entanto, as buscas com um número grande de clientes simultâneos em conjuntos extensos de mensagens demandou um tempo excessivamente elevado. Nas simulações, uma pesquisa em um Mailbox com quinhentas mensagens utilizando vinte e cinco clientes simultâneos exigiu cerca de dois minutos em cada cliente, porém Shneiderman [Shn84] recomenda que o tempo de resposta em sistemas interativos não seja superior a dois segundos. Por conseguinte, seria aceitável a utilização do sistema apenas por um número restrito de usuários ou em casos onde o número de mensagens a ser pesquisado não fosse elevado. Mas, de acordo com Nielsen [Nil96] em sistemas *Web*, considera-se aceitável um tempo de resposta de até quinze segundos.

O problema de tempo de resposta elevado pode ser minimizado através de interfaces com o usuário que apresentem indicadores de progresso ou mecanismos que reduzam a percepção do tempo de espera por parte do usuário. Este problema também pode ser atacado através de programas clientes que solicitem *a priori* um certo número de informações do servidor, antecipando-se a necessidade do usuário e eliminando o tempo de busca das mesmas no momento da requisição.

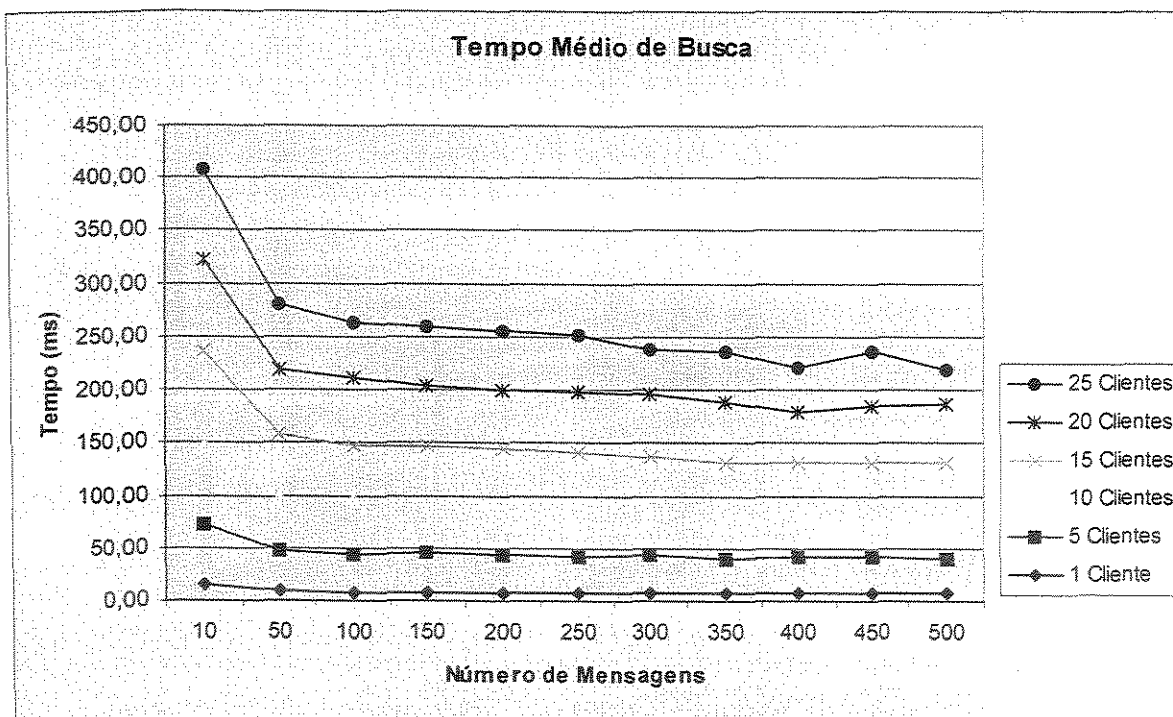


Figura 5.8 – Tempo gasto para consulta de cada mensagem.

O gráfico da Figura 5.8 apresenta o tempo consumido na pesquisa de cada Mailbox dividido pelo número de mensagens, dessa forma, pode-se acompanhar o tempo gasto na verificação individual de cada mensagem pelo termo solicitado.

Neste gráfico, observa-se que o tempo necessário para a verificação de cada mensagem pelo termo procurado diminui com o aumento do número de mensagens em cada Mailbox, ocorrendo uma tendência à estabilização com um número maior de mensagens. A grosso modo, podemos considerar que para o mesmo número de clientes, o tempo necessário para a verificação de cada mensagem é constante.

A maior discrepância é observada nos casos dos Mailboxes com apenas dez mensagens, que possivelmente é decorrente do tempo necessário para solicitação da pesquisa e o envio da resposta, que não pode ser “dispersado” por um número maior de mensagens.

Como o conjunto de dados utilizado nos testes foi relativamente pequeno, possivelmente todo o conjunto pôde ser armazenado na memória principal do servidor, gerando o tempo constante para verificação das mensagens. No entanto, para conseguir a saturação da memória do servidor, seriam necessários testes com milhares de mensagens, o que tomaria um tempo excessivamente elevado para realização dos testes.

Considerações finais

6.1 Conclusões

Neste trabalho indicamos as propriedades que caracterizam uma informação de qualidade e os meios que podem ser usados para atingi-la através das diferentes fases do seu ciclo de vida. Mostramos também que as ferramentas mais utilizadas para distribuição de informações não são adequadas para as tarefas de organização e gerenciamento e, por conseguinte, falham no esforço de distribuir informações de qualidade.

A partir dessa premissa, apresentamos o projeto INEX que procura resolver alguns dos problemas expostos. Destacamos as suas principais características e como elas se aplicam na resolução dos problemas levantados.

Finalmente, como prova de conceito, demonstramos a sua utilização em um problema concreto: o gerenciamento de mensagens de correio eletrônico em uma *intranet*. Dessa forma, confirmamos que a ferramenta pode ser facilmente estendida, podendo ser utilizada na resolução de problemas nas mais variadas situações e aplicações.

A utilização em larga escala da ferramenta, o que possibilitaria o levantamento de falhas e poderia fornecer informações em termos numéricos dos ganhos decorrentes do seu uso, não foi possível. Durante o desenvolvimento do prototótipo, muitas das tecnologias¹ utilizadas na parte cliente do sistema estavam em evolução e apresentando diversos problemas, resultando em uma grande instabilidade no cliente, não permitindo seu uso em um ambiente controlado. Acreditamos que isso não invalida as propostas apresentadas, visto que são problemas técnicos e não conceituais. Com a maturação das tecnologias e novas plataformas de *hardware*, possivelmente a maior parte dos problemas será resolvida.

Para contornar os contratempos, os testes para validação do sistema foram feitos através de programas que simulavam um cliente executando chamadas aos métodos que seriam utilizados pelos usuários e verificando a validade dos resultados.

Os resultados dos testes de desempenho demonstraram que ocorre um aumento linear do tempo necessário para uma pesquisa no servidor diretamente proporcional ao número de clientes simultâneos e ao número de mensagens a ser pesquisado.

¹ Javascript, DHTML, *applets*.

A partir dos dados levantados, conclui-se que o sistema INEX é adequado para ambientes com um número baixo de acessos simultâneos, ocorrendo uma queda de desempenho com o aumento de acessos simultâneos.

6.2 Trabalhos Futuros

- *Java Servlets*

A implementação atual do projeto INEX utiliza um *proxy* e *applets* Java na camada cliente para acessar os dados. É possível mover uma boa parte da funcionalidade provida por *applets* e pelo *proxy* para *servlets* que ficariam no servidor.

O modelo de *servlets* oferece algumas vantagens em relação à implementação baseada em *applets*, incluindo aspectos relacionados à segurança, integração com o servidor HTTP e menores requisitos de *hardware* no cliente [Hun98].

- Interfaces Alternativas

Outra abordagem para a interface com o usuário seria a utilização de formas alternativas de apresentação dos dados. A interface atual utiliza um sistema de listas hierárquicas semelhantes aos gerenciadores de arquivos usados nos sistemas Microsoft Windows e Apple MacOS. Uma possibilidade é a utilização de estruturas de árvores hiperbólicas para exibição dos dados (**Figura 6.1**).

“A essência dessa abordagem é a distribuição da hierarquia uniformemente em um plano hiperbólico e o mapeamento deste plano em uma região circular de visualização. A projeção em um disco provê um mecanismo natural para conceder mais espaço para uma porção da hierarquia que está inserida num contexto muito mais amplo. A mudança no foco é feita através da mudança na estrutura no plano hiperbólico, o que permite uma transição suave sem comprometer a apresentação do contexto” [Lam94+].

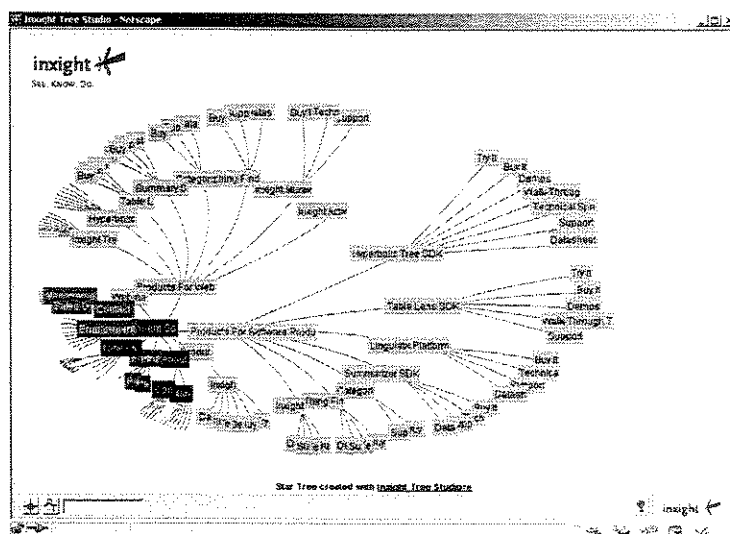


Figura 6.1 – Exemplo de aplicação usando o conceito de árvore hiperbólica.

- Hierarquias distribuídas entre vários domínios.

Uma das considerações feitas sobre o *Intranet Express* seria a possibilidade de armazenar hierarquias de listas e informações de maneira distribuída. Vários servidores *Intranet Express* poderiam ser interligados de forma a oferecer, transparentemente, acesso a informações armazenadas em diversas máquinas servidoras, tanto dentro de uma *intranet* quanto entre *intranets*.

Essa idéia cria uma série de possibilidades interessantes. A hierarquia (ou hierarquias) de listas de um determinado servidor INEX poderia ser integrada com parte ou toda a hierarquia (ou hierarquias) de listas de outro domínio de forma transparente. Isto é, ao navegar pela hierarquia, o usuário poderia acessar listas e documentos que se encontram distribuídas por vários domínios sem uma transição aparente. Neste caso, a partir do acesso a uma hierarquia em um servidor, o usuário poderia entrar em uma sub-lista que esteja localizada em outro domínio. Outra possibilidade seria a de permitir que listas fossem "espelhadas" em diversos domínios de maneira a permitir o acesso à mesma informação a partir de diversos pontos.

Seria necessária a implementação de um sistema de controle de acesso mais refinado e que permitisse a existência de diversos domínios de hierarquias de grupos e usuários. Esquemas de criptografia, autenticação e certificação precisariam ser integrados ao sistema para garantir a integridade da informação distribuída pelas diversas INEX interligadas.

- *Links* entre documentos INEX e HTML

Não há na versão atual um mecanismo que permita que documentos armazenados nas hierarquias façam referências relativas a documentos externos ao INEX. Uma solução pode ser a integração com um servidor *web*.

- Controle de versões

A implementação atual da INEX não contempla um sistema de controle de versões de documentos que permita manter múltiplas variantes de um documento e um controle sobre os responsáveis por alterações. Este tipo de sistema pode ser útil para documentos compartilhados por várias pessoas ou grupos com permissão de escrita ou em casos onde se deseja que certos documentos tenham dupla custódia, com determinada pessoa responsável pela escrita e outra responsável pela revisão.

- Suporte a *threads* dentro de Mailboxes

A possibilidade de organizar as mensagens de correio eletrônico em *threads*, conforme anteriormente discutido, resulta em diversos benefícios. O usuário pode determinar o estado de uma discussão apenas verificando a última mensagem postada relacionada ao tópico em questão, permitindo ao mesmo avaliar a conveniência de uma nova colocação ou resposta. Conjuntos inteiros de mensagens podem ser arquivados ou mesmo removidos após a finalização das discussões.

Estas são apenas algumas possibilidades de extensões e a lista acima não pretende ser uma relação exaustiva; a abrangência do projeto permite que inúmeras outras possam ser levantadas.

Referências Bibliográficas

- [Bäl00] Bälter, O., Keystroke level analysis of email message organization, in *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, Haia, Holanda, 2000.
- [Bec00] Becker, K., Cardoso, M., Mail-by-Example: a Visual Query Interface for Managing Large Volumes of Electronic Messages, in *Proceedings of the International conference on Advanced Visual Interfaces (AVI2000)*, ACM Press, Palermo, 2000.
- [Bec96+] Becker, K., Ferreira, S.N., Virtual folders : database support for electronic messages classification, in *Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications*, Kyoto, 1996.
- [Ben98] Bender, M., *Your First Intranet*, in PC Magazine, August, 1998.
<http://www.zdnet.com/pcmag/stories/reviews/0,6755,340028,00.html>
- [Bro99] Broccoli, K., *Improving Information Retrieval with Human Indexing*, 1999.
<http://idm.internet.com/features/humanindex-1.shtml>
- [Cho95] Choo, C. W., Information Management for the Intelligent Organization: Roles and Implications for the Information Profession, in *1995 Digital Libraries Conference, Faculty of Information Studies*, University of Toronto, 1995.
<http://choo.fis.utoronto.ca/FIS/ResPub/DLC95.html>
- [Cho98] Choo, C. W., *Information Management for the Intelligent Organization*, Information Today/Learned Information, New Jersey, 1998.
<http://choo.fis.utoronto.ca/fis/imio/>
- [Chm81] Chomsky, N., *Regras e representações: a inteligência humana e seu produto*, Zahar, Rio de Janeiro, 1981.
- [Die85] Diebold, J., *Business in the Age of Information*, AMACOM, New York, 1985.
- [Doh97] Doherty, G., *Comparison of X.400 to the Internet Mail Protocols*, 1997.
<http://ganges.cs.tcd.ie/4ba2/x400/opinions.html>
- [Eck98] Eckel, B., *Thinking in Java*, Prentice Hall Computer Books, 1998.
- [Eps86] Epstein, I., *Teoria da Informação*, Editora Ática, São Paulo, 1986.
- [Fla97] Flanagan, D., *Java in a Nutshell*, 2nd Edition, O'Reilly, 1997.
- [Fer97+] Ferreira, S.N., Becker, K., A query language for retrieving information in electronic mail environments, in *Proceedings of the 11th Brazilian Symposium on Database Systems (SBBD97)*, Fortaleza, 1997.
- [Fid94] Fidel, R., User-Centered Indexing, in *Journal of the American Society for Information Science*, Volume 45, Número 8, 1994.

- [Gam94+] Gamma, E., Helm R., Johnson R. and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, 1994.
- [Gol92] Goldratt, E. M., *A Síndrome do Palheiro: Garimpando Informações Num Oceano de Dados*, Educator Editora, São Paulo, 1992.
- [Hae93+] Haeckel, S. H., Nolan, R. L., The Role of Technology in an Information Age: Transforming Symbols into Action, in *1994 Annual Review of the Institute for Information Studies*, Northern Telecom Inc. and The Aspen Institute, 1994.
- [Han98] Hansen, P., *Evaluation of Information Retrieval User interface - Implications for User Interface Design*. Swedish Institute of Computer Science, 1998.
<http://www.hb.se/bhs/ith/2-98/ph.htm>
- [Har97] Harris, R., *Evaluating Internet Research Sources*, Vanguard University of Southern California, 1997.
<http://www.sccu.edu/faculty/R.Harris/evalu8it.htm>
- [Hol96] Holts, S., Intranets: What's All the Excitement?, in *Communication World*, Junho/Julho 1996.
<http://www.holtz.com/byshel/intranets.htm>
- [Hun98] Hunter, J. with Crawford, W., *Java Servlet Programming*, O'Reilly, 1998.
- [IBM98a] IBM Corporation, *IBM Application Framework for e-business - Web Application Programming Model*, International Business Machines Corporation, 1998.
<http://www-4.ibm.com/software/ebusiness/pm.html>
- [Joh93] Johnson, N., The Knowledge Economy: The Nature of Information in the 21st Century, in *1994 Annual Review of the Institute for Information Studies*, Northern Telecom Inc. and The Aspen Institute, 1994.
- [Lam94+] Lamping, J., Rao R., Laying out and Visualizing Large Trees Using a Hyperbolic Space, in *Proceedings of the ACM Symposium on User Interface Software and Technology*, ACM Press, Novembro 1994.
- [Mat01] Matos Jr., G. R., *Componentes para o Desenvolvimento de Intranets*, Dissertação de Mestrado, Instituto de Computação (IC), Universidade Estadual de Campinas (UNICAMP), Maio 2001.
- [Men96+] Meng, R., Maier, R., Donger, E., *Managing Information Overload*, 1996.
http://cascadel.se.cs.meu.edu/MEDD_96/Rapid_Fire_Web_site/project.html
- [Mor89] Mora, J. F., *Diccionario de Filosofia Abreviado*, Sudamericana, Buenos Aires, 1989.
- [Net99] Netscape Communications Corporation, *JavaScript Guide*, 1997.
<http://developer.netscape.com/docs/manuals/js/client/jsguide/contents.htm>
- [New94] Newell, A., *Unified theories of cognition*, Harvard University Press, 1994.
- [Nil94] Nielsen, J., *Top Ten Mistakes in Web Design*, Jakob Nielsen's Alertbox, Maio 1996.
<http://www.useit.com/alertbox/9605.html>

- [Nil99] Nielsen, J., *Intranet Portals: The Corporate Information Infrastructure*, Jakob Nielsen's Alertbox, 1999.
<http://www.useit.com/alertbox/990404.html>
- [Orf98+] Orfali, R., Harkey D., *Client/Server Programming with Java and Corba*, 2nd Edition, Wiley Computer Publishing, New York, NY, 1998.
- [RFC822] Crocker, D., *Standard for the format of ARPA Internet text messages*, Dept. of Electrical Engineering, University of Delaware Newark, 1982.
<http://www.rfc-editor.org/rfc/rfc822.txt>
- [RFC2045] Freed, N., Borenstein N., *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Network Working Group, 1996.
<http://www.rfc-editor.org/rfc/rfc2045.txt>
- [RFC2046] Freed, N., Borenstein N., *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, Network Working Group, 1996.
<http://www.rfc-editor.org/rfc/rfc2046.txt>
- [RFC2047] Moore, K., *Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text*, Network Working Group, 1996.
<http://www.rfc-editor.org/rfc/rfc2047.txt>
- [RFC2048] Freed, N., Klensin J., Postel J., *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*, Network Working Group, 1996.
<http://www.rfc-editor.org/rfc/rfc2048.txt>
- [RFC2049] Freed, N., Borenstein N., *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples*, Network Working Group, 1996.
<http://www.rfc-editor.org/rfc/rfc2045.txt>
- [Ric96+] Rice, J., Farquhar A., Piernot P., Gruber T., Using the Web instead of a Window System, in *1996 CHI Electronic Proceedings*, 1996.
http://www.acm.org/sigchi/chi96/proceedings/papers/Rice/jpr_txt.htm
- [Sch90] Schrage, M., *Shared Minds: The New Technologies of Collaboration*, Random House, New York, NY, 1990.
- [Shn84] Shneiderman, B., Response Time and Display Rate in Human Performance with Computers, in *ACM Computing Surveys*, Setembro 1984.
- [Shn99+] Shneiderman, B., Feldman D., Rose A and Grau F. X., *Visualizing Digital Library Search Results with Categorical and Hierarchical Axes*, Human-Computer Interaction Laboratory, Department of Computer Science and Institute for Systems Research, University of Maryland, 1999.
- [Spa97] Spangles, T., Pure Intranets, in *PC Magazine*, Maio 1997.
<http://www.zdnet.com/products/content/pcmg/1610/pcmg0044.html>
- [Tan96] Tanenbaum, A. S., *Computer Networks*, Third Edition, Prentice Hall, New York, 1996.

- [Tay91] Taylor, R. S., *Information Use Environments*, in Progress in Communications Science, ed. B. Dervin and Melvin J. V, Ablex Publishing Corporation, Norwood, NJ, 1991.
- [Tay99] Taylor, A. G., *The Organization of Information*, Libraries Unlimited Inc., Englewood, CO, 1999.
- [Tel96] Tellen, S. L., *The Intranet ArchitectureTM: Managing information in the new paradigm*, Amdahl Corporation, 1996.
<http://www.amdahl.com/doc/products/bsg/intra/infra.html>
- [W3C99a] World Wide Web Consortium, *HTML 4.0 Specification*, 1999.
<http://www.w3.org/TR/REC-html40/>
- [W3C99b] World Wide Web Consortium, *Document Object Model (DOM)*, 1999.
<http://www.w3.org/DOM/>
- [Wha00] Whatis?com, *Internet Technology Terms: Intranet*, 2000.
<http://whatis.techtarget.com/>
- [Whi96+] Whittaker, S., Sidner C., Email Overload: Exploring Personal Information Management of Email, in *Conference Proceedings on Human Factors in Computing Systems*, Vancouver Canada, 1996.
- [Whi99+] White, S., Fisher, M., Cattell, R., Hamilton, G., Hapner, M., *JDBC API Tutorial and Reference, Second Edition – Universal Data Access for the Java 2 Platform*, Addison-Wesley, 1999.

Apêndice A

O pacote `javax.mail`

O conjunto de classes opcional definido em `javax.mail` é uma extensão padrão definida pela Sun¹ que modela um sistema de correio eletrônico. Esta biblioteca de classes provê uma série de classes abstratas que permitem a leitura, composição, envio e recebimento de mensagens de correio eletrônico. Nesta seção, apresentaremos as principais classes utilizadas nesta tese.

`javax.mail.Session`

Esta classe fornece métodos para carregar e controlar classes que implementam um determinado protocolo. Por exemplo, uma instância da classe `javax.mail.Store` descrita abaixo é obtida através de uma `Session`.

`javax.mail.Store`

Representa um repositório de mensagens. A implementação da classe deve ser feita para atender aos requisitos de um protocolo específico. Por exemplo, a implementação que provê o acesso às mensagens através do protocolo POP3 deve suprir todos os métodos que permitam a monitoração, leitura, envio e busca de mensagens no servidor definidas neste protocolo.

`javax.mail.Folder`

Também deve ser implementada pelo fornecedor da solução para um protocolo específico e provê meios de acesso às mensagens como instâncias da classe `javax.mail.Message`.

`javax.mail.Message`

A classe `javax.mail.Message` modela todos os componentes de uma mensagem de correio eletrônico real, tais como, remetente, destinatário, assunto, etc. Também deve ser implementada pelo fornecedor da solução para o protocolo sendo usado.

`javax.mail.MimeMessage`

Esta classe estende a classe `javax.mail.Message` e representa um mensagem de correio eletrônico no formato MIME.

`javax.mail.Transport`

É uma classe abstrata que também deve ser implementada para um protocolo específico e sua principal função é prover métodos para conexão ao MTA².

¹ <http://java.sun.com/products/javamail/>

² Mail Transfer Agent. Um MTA é o responsável pela transferência das mensagens de um servidor a outro. Um exemplo de MTA é o programa `sendmail`.