UM AMBIENTE PARA MINERAÇÃO DE UTILIZAÇÃO DA WEB

José Roberto de Freitas Boullosa

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

_____

Prof. Jano Moreira de Souza, Ph.D.

_____

Prof. Geraldo Bonorino Xexéo, D.Sc.

_____

Prof. Geraldo Zimbrão da Silva, D.Sc.

_____

Profa. Ana Cristina Bicharra Garcia, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2002

BOULLOSA, JOSÉ ROBERTO DE FREITAS

Um Ambiente para Mineração de Utilização da Web [Rio de Janeiro] 2002

VIII, 120 p. 29,7 cm (COPPE/UFRJ, M.Sc., Banco de Dados, 2002)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Mineração de Dados

2. Mineração de Dados Web

I. COPPE/UFRJ II. Título (série)

A Mainha e Painho

Agradecimentos

A Xexéo, por ter vestido a camisa e me apoiado até o último minuto.

A Jano, pelo incentivo na reta final.

A Neca, pelos longos telefonemas interurbanos e depois internacionais para falar dos assuntos relacionados à tese.

A Zana, pelo carinho de sempre e pelas experiências semelhantes compartilhadas.

A Nando, pelo conselho sincrônico em 07/01: surtar.

A Lilian, Rosa, Zé, Henrique, Luciana, Alcina, Marta e Adriana, pelo amor que sempre demonstram.

A Larinha e Betinho, pela chegada durante esse período crítico.

A Peruca e Raul, por só me darem alegrias.

A Cláudia, pelos anos de convívio quase sempre pacífico, e pelo apoio sempre incondicional (e a Bê pelo incentivo adicional).

A Flávia, pela força mútua sempre construtiva.

A Lúcio, pela amizade e participação.

A todos os amigos e amigas que me incentivaram direta ou indiretamente, ou que tenham apenas me dado o prazer da sua convivência.

A Paty, Solange, Adilson e todos a quem dei trabalho e que me ajudaram bastante naquilo que puderam.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM AMBIENTE PARA MINERAÇÃO DE UTILIZAÇÃO DA WEB

José Roberto de Freitas Boullosa

Abril/2002

Orientadores: Geraldo Bonorino Xexéo
Jano Moreira de Souza

Programa: Engenharia de Sistemas e Computação

Ao planejar a estrutura de um site Web, o projetista necessita ter uma visão clara não somente dos objetivos que deseja atingir e dos perfis dos usuários que irão acessar o site, mas também da maneira pela qual estarão navegando pelas diversas páginas que o compõem. A análise da maneira pela qual um site Web é percorrido pelos visitantes pode fornecer pistas inestimáveis de como ele está atendendo aos requisitos aos quais se propõe. Tal análise envolve a transformação e interpretação dos registros de utilização armazenados nos logs de servidores Web e a conseqüente descoberta dos padrões de navegação implícitos e previamente desconhecidos, utilizando técnicas e ferramentas de mineração de dados e descoberta de conhecimento.

Este trabalho apresenta uma proposta de um ambiente para mineração de utilização de dados Web, o qual servirá como base para que um desenvolvedor possa implementar e testar novos métodos e algoritmos de mineração de utilização. Além disso, o trabalho mostra como o ambiente pode ser útil para um projetista Web que deseje construir sites e páginas que se adaptem automaticamente aos padrões de navegação do usuário.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AN ENVIRONMENT FOR WEB USAGE MINING

José Roberto de Freitas Boullosa

April/2002

Advisors: Geraldo Bonorino Xexéo
Jano Moreira de Souza

Department: Computer Science and System Engineering

When planning a Web site, designers should have not only a clear understanding of user's profiles and site objectives, but also an asserted knowledge of the way users will browse site pages. Analysis of a site visitors' behavior is a powerful tool that can be used to gather invaluable hints about how well the site is reaching its goals. Such analysis involves transformation and interpretation of Web server log records, in order to find hidden, implicit and previously unknown usage patterns, through the use of data mining and knowledge discovery tools and techniques.

This work proposes an environment for Web usage mining, such that it can be used as a basis for development, testing and implementation of new Web usage mining methods and algorithms. Furthermore, it shows how this environment can be useful for a Web designer that intends to build sites and pages that adapt themselves automatically according to user's needs.

Índice

1. Introdução

“Eu penso em Sara, o resto é fácil.”

J.W.Gordon

1.1. Motivação e objetivos do trabalho

O acesso diário à Internet via Web compreende um sem número de cliques em milhões de páginas diversas espalhadas por todo o mundo, cliques executados por uma quantidade de usuários cujo número é impossível contar com exatidão. Conteúdos os mais diversos, costumes e necessidades pessoais ainda mais diferentes constituem a realidade da Web, toda esta complexidade formidável resultando em padrões de utilização extremamente ricos e variados. Cada acesso, cada clique faz parte da ininterrupta e quase infinita seqüência denominada “*clickstream*” (KIMBALL & MERZ, 2000), o fluxo ou corrente incessante de acessos às páginas do universo Web.

Compreender os padrões de utilização das páginas Web, entender as motivações que impulsionam os usuários quando estão navegando, descobrir quais os modelos subjacentes a esta navegação tem sido a tarefa de uma legião de pesquisadores em áreas tão diversas como redes de computadores, bancos de dados, psicologia, inteligência artificial e outras.

Uma das bases para estas pesquisas está na análise e interpretação dos milhões de registros armazenados diariamente nos logs dos servidores Web, os quais, em sua totalidade, são, de uma maneira bastante desestruturada e desorganizada, o retrato nem sempre fiel da cadeia de acessos às páginas. Nem sempre fiel porque,

como será visto no presente trabalho, as diversas particularidades envolvidas no processo de navegar pela Web (a presença de navegadores, servidores Web, servidores *proxy*, servidores de cache e muitos outros fatores) fazem com que se torne virtualmente impossível descobrir com precisão quais foram as seqüências de páginas efetivamente visitadas.

A mineração de utilização da Web é a área que se dedica à atividade de investigação do "*clickstream*", tentando não só reconstituir os passos seguidos pelos usuários, mas também (e principalmente) descobrir quais os padrões de utilização que podem aflorar nessa reconstituição.

Contudo, como muitos dos inúmeros "buracos" nas seqüências de registros de utilização talvez nunca possam ser preenchidos com total segurança, o processo de mineração de utilização Web pode ser visto como uma verdadeira "arqueologia da Web": está sempre a fazer predições e deduções não só a partir das observações efetivamente realizadas, mas também a partir de suposições feitas na tentativa de preencher as lacunas.

A mineração de utilização apresenta-se, desta forma, como uma ferramenta fundamental não só aos estudiosos interessados em pesquisar e descobrir padrões de navegação. Pelo contrário, sua utilidade vai muito além disso, sendo também de inestimável valor para todos os indivíduos e organizações envolvidas no projeto e implementação de sites Web. O termo "*e-metrics*" vem sendo cada vez mais utilizado para denotar as medidas de desempenho das páginas armazenadas em um site Web, métricas estas que, para serem obtidas, não podem prescindir da mineração de utilização.

Os projetistas de sites Web, ao avaliarem como serão construídas as páginas e como serão as referências entre elas, precisam ter em mãos subsídios que permitam tomar decisões apropriadas sobre a estrutura ou topologia utilizada (SPILIOPOULOU, 2000). Sem o conhecimento descoberto a partir da mineração de utilização, o projeto

de um site dependeria apenas das suposições dos projetistas em relação às expectativas e padrões de comportamento do usuário.

Todavia, com este conhecimento em mãos, um conhecimento a partir de uma base fortemente empírica, o projetista passa a ter condições de tomar escolhas bem mais fundamentadas em relação ao design que será escolhido para cada página, link ou script disponibilizado em seu site.

A busca e a análise dos padrões de utilização são geralmente realizadas a partir dos dados brutos extraídos dos logs de um servidor Web, nos quais, após uma etapa de pré-processamento e transformação, serão utilizadas técnicas de mineração de dados voltadas especificamente para este tipo de tarefa.

Há atualmente uma profusão de ferramentas que permitem a extração e a análise dos dados armazenados nos logs de utilização. Tais ferramentas concentram-se, quase sempre, na disponibilização de dados de natureza estatística, tais como, por exemplo, quais páginas foram mais acessadas em determinado período, qual o tempo médio de visitação a uma página, ou quais os domínios de onde se originou a maior parte dos acessos a uma página.

Boa parte das ferramentas comerciais disponíveis não permite, contudo, a extração de padrões de utilização mais complexos. Por exemplo, quais as seqüências de páginas mais comumente visitadas? Além disso, as ferramentas comerciais possuem sempre uma arquitetura fechada, limitada em termos de customização, e que não permite modificações nos métodos de descoberta e análise dos padrões de uso.

Por outro lado, as ferramentas propostas em diversos trabalhos de pesquisa, apesar de se direcionarem à descoberta de padrões de utilização mais complexos, com o uso de técnicas avançadas de mineração de dados, são também fechadas no sentido de estarem voltadas especificamente para um determinado método ou análise.

O presente trabalho tem como objetivo fazer um estudo dos principais aspectos envolvidos na mineração de utilização da Web, identificando seus pontos fracos e pontos fortes, e, ao final, propor um ambiente chamado MineraWeb, que possibilitará o desenvolvimento e o uso de ferramentas e métodos de mineração de utilização Web construídas segundo os mais diversos modelos e algoritmos disponíveis.

Um dos problemas que encontram o pesquisador de mineração de utilização da Web é o de ter que enfrentar todas as dificuldades comuns inerentes à mineração de dados, tais como limpeza e filtragem de logs, identificação de usuários e sessões, sempre que decidir, por exemplo, apenas testar um novo algoritmo de mineração de padrões de utilização. Tais dificuldades, entretanto, são secundárias para ele, já que seu objetivo principal é o teste do algoritmo de mineração.

O ambiente aqui proposto virá, então, ao encontro dessa necessidade, provendo um arcabouço comum para tal teste, disponibilizando uma base de dados e ferramentas auxiliares que o livrarão da obrigação de implementar as tarefas acessórias ou secundárias.

Adicionalmente, será também sugerido como o ambiente MineraWeb pode servir de apoio para a criação de sites adaptativos, que se adeqüem às necessidades dos usuários, sempre em constante mutação.

1.2. Organização do trabalho

O trabalho está organizado da seguinte maneira: após esta breve introdução, o capítulo 2 fará uma descrição da área de mineração de dados, incluindo os tópicos relacionados *data warehousing* e OLAP, já que vários dos pontos envolvidos na mineração de utilização Web são derivados diretamente de ambas, especialmente da primeira. O capítulo mostrará, então, como as técnicas de mineração de dados podem

ser utilizados para a mineração de dados da Web, salientando as diferentes faces apresentadas por esse tipo de mineração.

Em seguida, o capítulo 3 detalhará uma dessas faces, a mineração de utilização da Web, abordando os principais temas envolvidos nesta atividade, as fases em que ela costuma se dividir, os modelos e algoritmos mais utilizados e algumas das ferramentas disponíveis.

O capítulo 4 mostrará a proposta do MineraWeb, um ambiente para mineração de utilização Web, útil não somente como base de uma implementação a ser usada pelos administradores de sites, mas também como plataforma destinada às pesquisas que desenvolvam e testem novas técnicas e algoritmos de mineração de utilização.

Finalmente, o capítulo 5 mostrará as conclusões atingidas no curso do desenvolvimento do trabalho, além de assinalar como ele se relaciona com algumas das propostas existentes, e propondo novos encaminhamentos para trabalhos futuros.

2. Mineração de Dados, Mineração de Dados da Web

Este capítulo mostra as principais idéias envolvidas nas áreas de mineração de dados e descoberta de conhecimento, avaliando como a mineração de dados pode ser utilizada para a busca de informações na World Wide Web, seja essa busca direcionada ao conteúdo, à estrutura ou à forma como é utilizada a Web. Serão introduzidos os conceitos de mineração da Web, mineração de conteúdo da Web e mineração de utilização da Web. Este último, por sua vez, será tratado mais detalhadamente no próximo capítulo.

2.1. Mineração de Dados e Descoberta de Conhecimento

A mineração de dados, ou "*data mining*", corresponde à atividade automática ou semi-automática de exploração e análise de grandes quantidades de dados com o propósito de neles descobrir regras e padrões significativos (BERRY & LINOFF, 1997). Sendo interdisciplinar em sua própria natureza, é influenciada ao mesmo tempo por áreas tais como a estatística, inteligência artificial, teoria dos grafos e, claro, a teoria de banco de dados.

O problema da mineração de dados parte do pressuposto de que os grandes bancos de dados do mundo real são verdadeiras "minas" de conhecimento (DEOGUN *et al.*, 1997), onde repousam informações de grande valor que podem ser encontradas através de técnicas e algoritmos adequados.

MANNILA (1997) define genericamente o problema da mineração de dados da seguinte forma: dados um conjunto de dados *d* e uma classe *P* de padrões ou

sentenças que descrevam as propriedades desses dados, pode-se determinar se um padrão $p \in P$ é interessante e ocorre com freqüência suficiente em *d*. Assim, a tarefa de mineração de dados volta-se para a descoberta do conjunto PI de padrões, onde:

$$PI(d, P) = \{ p \in P \mid p \text{ é interessante e ocorre em } d \text{ numa freqüência suficiente} \}$$

O desenvolvimento e a multiplicação das pesquisas em mineração de dados têm se dado a passos largos, não só pelos diversos desafios teóricos que se colocam na área, mas também pelas variadas aplicações práticas que podem ser associadas à descoberta do conhecimento previamente ignorado, armazenado nos grandes bancos de dados do mundo real.

A mineração de dados é, algumas vezes, também conhecida como *mineração de bancos de dados* ("*database mining*") ou ainda *descoberta de conhecimento em bancos de dados* ("*knowledge discovery in databases*") .

A expressão "descoberta de conhecimento em bancos de dados" foi lançada por PIATESTKY-SHAPIRO (2000), ao promover o primeiro "Workshop in Knowledge Discovery in Databases" na Conferência Internacional de Inteligência Artificial de Detroit, em 1989. Curiosamente, ele justificou a escolha dos termos afirmando que a expressão "mineração de dados", já muito utilizada naquela época, era pouco "sexy", além de ser utilizada pejorativamente por muitos estatísticos para criticar a área .

BERRY & LINOFF (1997) referem-se à descoberta de conhecimento como um dos "estilos" de mineração de dados, em contraposição aos testes de hipóteses ("hypothesis testing"). A primeira seria uma abordagem "bottom-up", em que, partindo-se dos dados, tenta-se chegar a um conhecimento previamente ignorado. A abordagem de testes de hipóteses, por sua vez, seria uma tentativa "top-down" de provar (ou refutar) idéias previamente concebidas.

Estas duas abordagens correspondem, respectivamente, aos dois tipos clássicos de inferência conhecidos como indução e dedução. Do ponto de vista do aprendizado,

a indução parte de casos particulares (dados de treino) para os gerais, tentando desenvolver um modelo explicativo. A dedução, ao contrário, parte do geral - um modelo prévio - para o particular - os dados (CHERKASSKY & MULIER, 1998).

No entanto, para muitos, a descoberta de conhecimento é uma área mais ampla, da qual a mineração de dados é tão somente um componente ou etapa que enfoca principalmente os **métodos** de produção de conhecimento: FAYYAD *et al.* (1996) definem a descoberta de conhecimento como a extração não-trivial de informações potencialmente úteis, previamente desconhecidas e implícitas em dados brutos. Por essa proposta, a descoberta de conhecimento divide-se nas seguintes etapas, que podem ser repetidas tantas vezes quanto sejam necessárias:

a) definição dos domínios onde serão realizadas as análises e quais os objetivos do processo de descoberta de conhecimento;

b) criação de um conjunto de dados, através da seleção entre as diferentes fontes de dados disponíveis;

c) pré-processamento dos dados, incluindo a limpeza dos dados desnecessários e o tratamento daqueles que estão indisponíveis ou que possam conter alguma incerteza;

d) transformação dos dados, adequando suas dimensões e variáveis de maneira apropriada, para que estes estejam coerentes com as necessidades dos métodos que serão utilizados na próxima etapa;

e) mineração de dados, etapa que envolve efetivamente as técnicas e algoritmos que produzirão o conhecimento procurado;

f) análise e interpretação dos resultados encontrados na etapa anterior.

A mineração de dados vale-se de diversos modelos, ferramentas e técnicas para chegar aos seus objetivos (BORGES & LEVENE, 1998, BERRY & LINOFF, 1997, FAYYAD *et al.,* 1996). Um modelo produz uma ou mais saídas para um determinado

conjunto de entradas, e seus resultados podem possuir mais ou menos acurácia. Os modelos utilizados na mineração de dados incluem os de classificação, os preditivos, os de agrupamento e os de séries temporais.

Os modelos de classificação tentam rotular e colocar registros em classificações previamente existentes, ou criar novas classificações para eles, além de lhes adicionar outras informações tais como a probabilidade de ocorrência em determinado contexto. Os modelos preditivos são similares aos de classificação, mas as suas saídas não se limitam a uma série de classes.

Modelos de agrupamento criam grupos menores a partir de grandes conjuntos de registros, levando em consideração as características em comum por eles compartilhadas. Modelos de séries temporais assemelham-se aos preditivos, porém os seus domínios incluem dados coletados ao longo do tempo.

2.2. Métodos para mineração de dados

2.2.1. Geração de regras de associação

A descoberta de regras de associação (SAVASERE *et al.*, 1995, AGRAWAL & SRIKANT, 1994) aplica-se a um banco de dados de transações onde cada transação é composta por um conjunto de itens, e no qual procura-se descobrir quando a presença de um conjunto de itens implica na presença de um outro item na mesma transação.

A geração de regras de associação é muito utilizada em sistemas comerciais, especialmente naqueles direcionados à área de vendas, e é conhecida popularmente como "*market basket analysis*", já que, para atingir o seu objetivo de encontrar grupos de itens que ocorram juntos, analisa uma situação semelhante à que ocorre quando se utiliza uma cesta de supermercado (a transação).

2.2.2. Análise de seqüências

Neste modelo, procura-se seguir os relacionamentos entre registros para desenvolver modelos a partir dos padrões seqüenciais encontrados. É uma aplicação da teoria dos grafos à mineração de dados. Os padrões seqüenciais (MANNILA *et al.*, 1995, SRIKANT & AGRAWAL, 1996) são encontrados entre transações, quando a presença de um conjunto de itens em uma transação implica no surgimento de outro item em uma transação posterior.

2.2.3. Classificação e agrupamento ("*clustering*")

Técnicas de classificação (HAN *et al.*, 1993) permitem o desenvolvimento de perfis de itens com atributos em comum que podem ser usados para classificar novos itens adicionados ao banco de dados. A classificação é feita a partir de um conhecimento apriorístico sobre as classes e categorias utilizadas.

Ao contrário, na análise de agrupamentos ("clustering analysis"), é feita a reunião de itens ou usuários com características semelhantes somente com base empírica, sem qualquer conhecimento prévio de quais serão os grupos formados. Neste caso, busca-se construir modelos que encontrem itens de dados similares entre si, a partir de alguma métrica de "distância" entre os itens, que serão assim colocados juntos em novos agrupamentos ("*clusters*") ou em agrupamentos já existentes. Para tanto, podem ser utilizados métodos geométricos, estatísticos ou mesmo redes neurais.

Ambas as técnicas podem ser úteis para a tomada de decisões de marketing ou para a mudança dinâmica dos conteúdos de um servidor Web, quando acessado por um cliente anteriormente analisado.

2.2.4. Memory–based reasoning

Estes métodos procuram fazer predições de novos itens de dados, a partir de itens já conhecidos, procurando pelos vizinhos mais próximos a estes últimos e combinando seus valores para encontrar valores de predição e classificação. Uma de suas vantagens é a possibilidade de aprendizado sempre que novos itens sejam acrescentados ao banco de dados (BERRY & LINOFF, 1997).

2.2.5. Árvores de decisão e indução de regras

Estes modelos envolvem o uso de árvores de regressão e classificação (CART – "*classification and regression trees*") e indução chi-quadrada automática (CHAID – "*chi-squared automatic induction*"). São úteis para uma mineração direcionada, especialmente com classificação. Os registros do conjunto de treino são divididos em subconjuntos disjuntos que serão descritos por regras simples sobre um ou mais campos. O ID3 (QUINLAN, 1983) e o seu sucessor, o C4.5 (QUINLAN, 1993), são os dois principais algoritmos utilizados para a descoberta de conhecimento baseada em árvores indutivas.

2.2.6. Redes neurais e algoritmos genéticos

As redes neurais são uma das técnicas mais comuns de mineração de dados, pela sua ampla aplicabilidade (MITCHELL, 1997). Utilizam modelos que procuram reproduzir de maneira mais simplificada as conexões neuronais do cérebro. Por esse método, procura-se, a partir de um conjunto de treino, aprender padrões gerais que possam ser aplicados à predição e classificação.

Dois dos principais problemas das redes neurais são as dificuldades associadas ao entendimento dos modelos por ela produzidos e sua grande sensibilidade ao formato dos dados de entrada.

Os algoritmos genéticos (GA – “genetic algorithms”) utilizam as idéias e mecanismos da genética e seleção natural (operações de seleção, “cross-over”, mutação) para encontrar os parâmetros ótimos que descrevem funções preditivas (MITCHELL, 1997). São desenvolvidas sucessivas gerações de soluções, até que apenas algumas “sobrevivam” e as funções encontradas convirjam para uma solução ótima.

2.3. Data Warehousing

Ainda que muitas das técnicas e algoritmos da mineração de dados sejam relativamente antigas, a área sofreu um grande impulso na última década, a partir da profusão cada vez maior não só das pesquisas, mas também das aplicações em *data warehousing*. INMON (1996) define um *data warehouse* como um conjunto de dados integrados, não-voláteis, orientados por assunto e variáveis no tempo, utilizados primordialmente como ponto de apoio a decisões gerenciais.

A área de *data warehousing* utiliza largamente a modelagem multidimensional de dados (KIMBALL, 1996), mais simples que a abordagem relacional quando o que se pretende é a obtenção de dados para a resolução de consultas analíticas. KIMBALL (1997) mostra as principais diferenças entre o modelo multidimensional e o modelo entridade-relacionamento, assinalando as vantagens que o primeiro apresenta quando se deseja construir um *data warehouse.*

No modelo multidimensional, a informação é representada e manipulada através de um **cubo** de n **dimensões**, cada uma das quais simboliza uma diferente perspectiva de como devem ser visualizados os dados principais, chamados **fatos**. Os fatos possuem uma série de atributos que permitem o uso de agregações para possibilitar uma melhor análise. As dimensões podem possuir atributos e ser divididas em hierarquias, aumentando ou diminuindo a riqueza das informações analisadas.

O cubo pode ser manipulado através de operações básicas tais como "*slicing and dicing*", "*roll-up*" e "*drill-down*", que permitem o encaminhamento através das dimensões e a seleção das dimensões e níveis de detalhamento que mais interessam ao usuário, além de possibilitarem a agregação dos dados sempre que possível e necessário.

Ao se representar um *data warehouse* no modelo relacional, um fato pode ser representado por uma tabela central, ligada a diversas tabelas periféricas que representem, cada uma, uma dimensão diferente. A esta configuração denomina-se esquema estrela (*star schema*). Pode-se, contudo, optar pelo esquema "flocos de neve" (*snow flake schema*), mais complexo, em que as tabelas que representam as dimensões são normalizadas, tornando-se, assim, centros de novas estrelas.

2.4 OLAP

OLAP (On-line Analytical Processing) é um conjunto de tecnologias que visam a possibilitar aos usuários o acesso e visualização dos dados de uma maneira analítica e multidimensional, realizando operações de classificação, comparação, taxas multidimensionais e variações percentuais, através de uma interface gráfica, interativa e amigável, com o propósito de realizar a descoberta manual de conhecimento a partir dos dados. Em contraponto a OLAP, OLTP (On-line Transaction Processing), é a tecnologia voltada para o acesso bruto aos dados, no nível mais básico das transações, com um forte enfoque na entrada e consulta planas (KIMBALL, 1996).

CODD *et al.* (1993) propuseram um conjunto de 12 regras para a avaliação de produtos OLAP, das quais a primeira é a necessidade de uma visão conceitual multidimensional, num artigo em que traçam o caminho percorrido pela comunidade de bancos de dados desde o advento do modelo relacional, idealizado pelo próprio CODD (1970), no início dos anos 70, até o aparecimento da modelagem multidimensional como uma ferramenta para agregar mais semântica às aplicações de suporte à

decisão, vindo assim como um suplemento ao ambiente de processamento on-line de transações.

Por armazenar os dados (além das suas agregações– totais, subtotais, médias, etc.) em uma estrutura multidimensional, a tecnologia OLAP apresenta um desempenho bastante superior para os tipos de consultas que necessitam de uma análise dos dados como um todo, voltadas, portanto, para o processo de tomada de decisão. OLAP é, portanto, uma ferramenta complementar à área de *data warehousing*, utilizando as mesmas operações básicas do cubo multidimensional vistas na seção anterior. Assim, o usuário de um sistema OLAP pode fazer seleções dos subconjuntos de dados que lhe interessam entre as múltiplas dimensões do cubo, inclusive a temporal.

Há três principais categorias de OLAP, quanto à estrutura física de armazenamento:

- OLAP Multidimensional (MOLAP) – os dados e agregações são armazenados fisicamente numa estrutura multidimensional.
- OLAP Relacional (ROLAP) – os dados são armazenados em um SGBD relacional.
- OLAP Híbrido (HOLAP) – os dados são armazenados em um SGBD relacional e as agregações em uma estrutura multidimensional.

OLAP pode ser vista como uma ferramenta complementar à mineração de dados. O seu ponto forte é a característica descritiva, enquanto, na mineração de dados, o ponto forte é a descoberta automática de padrões.

Ambos, OLAP e mineração de dados, podem se juntar à estrutura mais ampla de um *data warehouse*, como ferramentas úteis à descoberta de conhecimento (informação) a partir das vastas quantidades de dados nele armazenados.

2.5. Mineração de dados da Web

Os princípios e técnicas de mineração de dados e descoberta de conhecimento podem ser aplicados com sucesso na World Wide Web para extrair conhecimentos úteis e interessantes sob diversos aspectos, seja do ponto de vista dos usuários ou dos desenvolvedores de sites Web. A este tipo de aplicação da mineração de dados dá-se o nome genérico (COOLEY et al, 1997) de mineração da Web ("*Web mining*"), que possui, no entanto, diversas ramificações com variados matizes.

2.5.1.Termos e conceitos úteis

O grande número de termos utilizados nos estudos da Web pode levar a confusões. Por isso, é interessante que alguns desses termos sejam bem definidos para evitar dúvidas.

O World Wide Web Consortium, ou W3C, organização responsável pela padronização da Web, foi criado em 1994, a partir de um acordo entre o MIT (Massachusetts Institute of Technology) e o CERN (European Organization for Nuclear Research), berço da própria Web, onde foral realizados os trabalhos pioneiros de Tim Berners-Lee em 1989.

Na home-page do W3C, podem ser encontrados documentos e rascunhos contendo definições úteis à área, algumas das quais (especialmente as associadas à mineração da Web) serão aqui descritas, antes de se prosseguir com o aprofundamento dos assuntos abordados no trabalho (W3C, 1999).

- Visita a uma página ("*page view*") - é o conjunto de arquivos que contribuem para a construção de uma página Web no navegador do cliente, como resultado de um acesso do usuário, ao realizar um clique ou "*hit*"; neste trabalho, usaremos da mesma forma os termos "acesso" e

"visita". Os arquivos são requisitados pelo navegador através do protocolo HTTP *(Hypertext Transfer Protocol*) (FIELDING et al., 1997).

- *Clickstream* – a seqüência de visitas a páginas de um usuário; pode ser usado também para denotar o fluxo constante de acessos de todo o universo de usuários de um site ou da própria Web.
- Sessão – o conjunto dos acessos a páginas feitos por um usuário ao realizar uma determinada visita a um site Web. Observe-se que, aqui, o termo "visita" está sendo usado para denotar uma visita a um site, e não a uma página individual. Ou seja, a visita a um site é composta por um conjunto de visitas às suas páginas (portanto, é uma sessão).
- Episódio – um subconjunto das páginas acessadas em uma sessão.
- URL (*Universal Resource Locator*) – endereço de localização de um documento na Internet. Tem o formato genérico ***esquema://host:porta/path/querystring***, onde ***esquema*** indica o protocolo utilizado (HTTP, FTP, etc.), ***host*** é o nome completo (ou endereço IP) da máquina onde está a informação, ***porta*** é porta que deve ser utilizada na comunicação entre a máquina que busca o documento e a que o armazena, ***path*** é o caminho completo do documento dentro da máquina origem, e ***querystring*** é o conjunto de parâmetros passados opcionalmente ao se acessar o documento, sendo utilizado sobretudo ao se acessar documentos dinâmicos, montados a partir da execução de programas ou scripts.
- URI (*Universal Resource Identifier*) – é uma categoria mais ampla que inclui tanto as URLs quanto os URNs (Universal Resource Names). Na prática, nos últimos anos, tem-se utilizado cada vez mais os termos URI e URL de maneira quase idêntica (W3C, 2001).

- Referidor (*referrer*) – a página de onde partiu a referência para a visita atual. Por exemplo, se um usuário está na página de um site de busca e clica em um link que o transporta para o site de uma universidade, o servidor da universidade reconhecerá a página do site de busca como referidora para aquele acesso.

- *Cookie* - Os *cookies* são marcadores usados para registrar e rastrear automaticamente os usuários de um site. Na prática, são arquivos que ficam armazenados localmente no computador do cliente, guardando informações que serão usadas na sua identificação (NETSCAPE, 1999). Apesar de não fazerem parte do padrão atual do protocolo HTTP (versão 1.1), são adotados pelos principais servidores e navegadores Web do mercado.

- Programas CGI (*Common Gateway Interface*) – programas usados para a construção dinâmica de páginas Web. O acesso a uma URL associada a um programa CGI faz com que o servidor Web execute aquele programa, o qual será então responsável por montar, dinamicamente, toda a estrutura e o conteúdo da página, que só então será enviada ao cliente.

2.5.2. Modelos de navegação e classificação das páginas Web

Os sistemas que analisam a Web necessitam, muitas vezes, de um modelo que procure descrever e explicar como atuam os usuários, classificando os diversos padrões de navegação pelas páginas, abrangendo o uso dos navegadores, seus botões de navegação, suas preferências, etc. Além disso, é necessária uma classificação ou tipagem das páginas, seja a partir de seus atributos, de seus conteúdos ou dos relacionamentos e links entre elas.

Pode-se considerar a Web como um sistema hipermídia aberto, colaborativo e altamente dinâmico, uma "ecologia de informações dinâmica", nas palavras de CATLEDGE & PITKOW (1995). Segundo eles, por ser um sistema hipermídia, encontram-se, na Web, os mesmos tipos de estratégias de utilização de outros sistemas da mesma espécie: busca e navegação.

Na estratégia de busca, há uma orientação quanto ao objetivo, um direcionamento; na de navegação, esta se dá através de fontes de informações que contêm itens com um alto grau de similaridade quanto ao interesse do usuário. COVE & WALSH (1988) acrescentam ainda uma terceira estratégia, a navegação serendípica ("serendipitious browsing"), puramente aleatória, apropriando-se do termo cunhado por Horace Walpole (BRITANNICA, 2002) para denotar o processo de descoberta casual ou acidental de princípios ou coisas que não se estavam buscando.

Apesar disso, as atividades de busca e navegação não são mutuamente exclusivas, os usuários estão constantemente a mudar o seu foco de uma para a outra. Assim, pode-se observar na Web um continuum entre, por um lado, a atividade de navegação como algo extremamente direcionado e focado a um determinado objetivo e, por outro, a navegação completamente livre, aleatória e sem qualquer compromisso a não ser com o próprio ato de navegar.

LEVENE & LOIZOU (2001) mostram bem essa interpenetração constante, esse ir e vir entre busca e navegação, ao dividir em quatro etapas o processo de busca de informações na Web:

a) especificação da consulta, quando o usuário define o que está buscando;

b) recuperação da informação, onde o sistema faz a disponibilização dos resultados obtidos, normalmente dando ao usuário um conjunto de links a serem seguidos, muitas vezes de acordo com critérios de importância;

c) navegação, quando o usuário repete alternadamente as tarefas de escolher uma das páginas disponibilizadas para visitar, e, a partir dali, ir para novas páginas relacionadas;

d) modificação da consulta, quando se retorna ao ponto de partida para refinar ou modificar o objetivo inicial.

Para satisfazer às necessidades de milhares de usuários diferentes, o projeto de um site Web deve tentar, na medida do possível, suportar essas diferentes estratégias, evitando assim que o visitante veja-se subitamente "perdido no hiperespaço" (NIELSEN, 1990). Há porém, uma certa tensão entre projetar um site tendo em vista os usuários que desejam somente fazer buscas e projetá-lo para aqueles que desejam apenas navegar.

Uma estrutura de site hierárquica à maneira de um diretório de arquivos - por exemplo, um site como o YAHOO! (2002), ou ainda um engenho especializado como o GOOGLE (2002), são extremamente adequados para usuários interessados em buscas. Contudo, tais estruturas são indesejáveis para o usuário que procura o inesperado, e que se sente insatisfeito com a precisão requerida nesses sites.

Atualmente, cada vez mais sites abrem a possibilidade de serem completamente indexáveis e, como tais, acessíveis a diversas espécies de buscas, ao mesmo tempo em que oferecem aos seus visitantes uma estrutura agradável e não totalmente direcionada. Para se avaliar a exata proporção das necessidades dos usuários é que se mostra útil a análise dos seus padrões de utilização dos sites Web, como será visto nas próximas seções.

Para obter subsídios sobre os padrões de navegação de um usuário Web, TAUSCHER & GREENBERG (1997) analisaram seis semanas de utilização de 23 usuários de um navegador comercial. A análise dos dados mostrou que 58% dos acessos eram de páginas já visitadas anteriormente. As pessoas tendem a revisitar

um número considerável de páginas, acessar algumas poucas páginas ou grupos de páginas com muita freqüência, além de gerar seqüências curtas de caminhos de páginas. A maior parte das revisitas inclui as poucas páginas acessadas recentemente.

A revisitação de páginas já acessadas na mesma seção é tão freqüente que todos os principais navegadores possuem métodos de predição para auxiliar os usuários quando estes desejam voltar para um ponto já percorrido. Contudo, o método de predição de páginas baseado em pilhas (cada página visitada é adicionada a uma pilha de páginas acessadas recentemente), geralmente usado pelos navegadores comerciais, é inferior à abordagem mais simples de mostrar apenas as URLs visitadas mais recentemente, excluindo-se as páginas duplicadas.

Analisando os próprios dados e os de CATLEDGE & PITKOW (1995), TAUSCHER & GREENBERG (1997) chegaram a uma taxa de recorrência R, que representa a probabilidade de que uma página acessada já tenha sido visitada anteriormente. Para os dados do próprio estudo, foi encontrada R=58%, com desvio padrão de 9%. Para os dados de CATLEDGE & PITKOW (1995), encontrou-se R=61%, com desvio de 9%. Ou seja, a atividade de navegação Web é um exemplo de **sistema recorrente** (%29), no qual o usuário está sempre a repetir atividades já realizadas anteriormente, ao mesmo tempo em que executa novas atividades entre as muitas possíveis.

As principais razões dadas pelos usuários estudados para o fato de estarem revisitando páginas incluíram as seguintes: as informações contidas nas páginas mudam; eles desejam explorar as páginas mais detalhadamente; as páginas possuem algum propósito especial (páginas de busca, por exemplo); são páginas que eles estão criando ou editando; as páginas fazem parte do caminho de navegação para uma outra página revisitada.

Por outro lado, eles visitam novas páginas pelas seguintes razões: suas necessidades de informação mudam, desejam explorar um novo site, a página foi recomendada por algum amigo, ou notaram uma página interessante enquanto estavam navegando por outra página.

Foram identificados por TAUSCHER & GREENBERG (1997) sete padrões de navegação ou visitação de páginas:

a) visitas iniciais a um grupo de páginas;

b) revisitas a páginas;

c) visitas a páginas que estão sendo criadas ou editadas;

d) visitas a páginas produzidas por aplicações;

e) navegação "*hub-and-spoke*" – visitas a uma página central ("*hub*") com muitos links para novas páginas, como numa pesquisa por largura ("*breadth-first*");

f) navegação dirigida, nas quais as próprias páginas possuem links estruturados para navegação (por exemplo, links do tipo "próxima página"), e os usuários decidem seguir esses links;

g) navegação de profundidade ("*depth-first*"), quando o usuário segue uma cadeia de links até o fim, antes de retornar à página central.

Em BORGES & LEVENE (1998) e LEVENE & LOIZOU (1999) é apresentado um modelo de navegação alternativo baseado na teoria de cadeias de Markov. Os trabalhos consideram que um banco de dados hipertexto consiste de um repositório de informações cujo conteúdo é armazenado na forma de páginas. Um grafo direcionado descreve a estrutura do banco de dados, sendo os nós correspondentes às páginas e as arestas correspondentes aos links entre essas páginas. A um conjunto de trilhas ou caminhos no grafo denomina-se "visão web".

Finalmente, pela própria característica estocástica da navegação, representa-se cada página como um estado e associam-se probabilidades aos links. Tais probabilidades denotam as freqüências com que os usuários utilizam os links e o peso relativo que dão a esses links para uma determinada consulta. Com isso, obtém-se um modelo de cadeia de Markov a partir do qual podem ser desenvolvidos algoritmos de mineração de padrões de navegação, assim como métodos para automatizar a navegação a partir das consultas dos usuários.

Esta característica probabilística da navegação Web é reforçada por HUBERMAN *et al.* (1998), os quais mostram que os padrões de navegação da Web possuem uma acentuada regularidade do ponto de vista estatístico. Analisando os comportamentos de usuários, desenvolveram um modelo de navegação que atingiu altos graus de correlação entre os cliques observados e os previstos pelo próprio modelo.

Além disso, os autores indicam que o comportamento do usuário se pauta pela busca constante de maximizar a utilidade ou o valor das páginas: cada página tem um valor associado, e há um custo ou esforço na navegação. Clicar na próxima página supõe que ela também terá um certo valor esperado, ainda que desconhecido. Portanto, um indivíduo navegará numa certa direção até que o custo exigido seja maior do que o valor esperado.

O seu trabalho sugere ainda que o número de cliques por página obedece a uma distribuição semelhante à da lei de Zipf, o que foi mais tarde demonstrado analiticamente por LEVENE & BORGES (2001). A probabilidade de uma trilha (seqüência de páginas) de comprimento t ser percorrida por um usuário é de $\boldsymbol{t}^{-3/2}$. Por essa distribuição, os usuários da Web tendem a preferir as trilhas curtas às longas. Uma explicação para esse fato é que, na topologia da Web, o número de trilhas curtas é exponencialmente menor que o de longas, e assim é bem mais fácil encontrar uma trilha relevante curta do que uma longa; a razão entre o valor agregado e o esforço

despendido ao navegar por uma seqüência é maior para as trilhas curtas do que para as longas.

PIROLLI *et al.* (1996) procuram fazer uma classificação das páginas Web em níveis cada vez maiores de abstração, a fim de facilitar a análise dos padrões de navegação dos usuários. A partir da extração da estrutura de sites Web e do estudo de logs de utilização, chegam a uma classificação de páginas baseada em fatores tais como a similaridade textual e os graus e padrões de conectividade entre elas.

COOLEY *et al. (*1999) abordam as estratégias de navegação a partir dos diferentes tipos de páginas Web: utilizam uma classificação de páginas modificada a partir daquela proposta em PIROLLI *et al.* (1996). As páginas podem ser de cinco tipos: páginas de cabeçalho, de conteúdo, de navegação, de 'look-up" ou pessoal, cada uma com uma característica física particular (**tabela** 1).

Tabela 1: Tipos de página Web (COOLEY *et al.,* 1999)

Tipo de página	Características físicas	Características de uso
Cabeçalho	. Contém links de entrada partindo da maioria das páginas do site . Raiz da estrutura de páginas do site	. Página inicial nas sessões dos usuários
Conteúdo	. Alta quantidade de textos e gráficos em relação a links	. Tempo médio de visitação longo
Navegação	. Pequena quantidade de textos e gráficos em relação a links	. Tempo médio de visitação curto . Não é referência posterior máxima
Look-up	. Pequeno número de links de entrada . Poucos ou nenhum link de saída . Conteúdo bastante reduzido	. Tempo médio de visitação curto . Costuma ser referência posterior máxima
Pessoal	. Não possuem características gerais em comum	. Baixa freqüência de visitação

Muitas páginas podem cair em mais de uma categoria: por exemplo, uma página pode ser, ao mesmo tempo, de conteúdo e de navegação. As páginas pessoais não costumam apresentar características em comum, além de não serem controladas pelos projetistas dos sites. Por outro lado, por terem uma taxa de utilização menor do

que a média, elas serão provavelmente desconsideradas na mineração de padrões de navegação, pois serão filtradas por medidas de controle tais como o suporte e a confiança (estas medidas serão explicadas mais adiante, no capítulo sobre mineração de utilização da Web, quando for abordada a etapa de descoberta de padrões).

Uma página de "referência posterior máxima" é, simplificadamente, aquela que finaliza uma transação de usuário: a última página "nova" acessada pelo usuário antes dele decidir acessar uma página já visitada anteriormente. Esse conceito será mais esclarecido quando forem descritos os métodos de identificação de transações, também no próximo capítulo.

A classificação das páginas de um site pode ser especificada não só manualmente, uma a uma, pelo próprio projetista, mas também realizada automaticamente por algum tipo de algoritmo, como o C4.5, ou ainda estabelecida (mais uma vez pelo projetista) a partir de metadados – e.g., *meta-tags* HTML ou uso de esquemas de codificação XML baseados em RDF (*Resource Description Framework,* W3C, 2002).

Do ponto de vista da utilização, podem ainda ser reconhecidas certas características para cada um dos tipos de página Web, como, por exemplo, a duração de referência ("*reference length*"): o tempo durante o qual uma página é vista pelo usuário (a duração de um *"page view")*. Este tempo pode ser determinado, com uma certa precisão, a partir da análise das entradas ou registros dos logs de servidores Web.

Esses modelos de navegação e de classificação de páginas Web serão bastante úteis na mineração de dados da Web, seja ao se fazer mineração do conteúdo das páginas (por exemplo, para agrupar as páginas num sistema de recomendação), seja na investigação dos padrões de navegação: por exemplo, na descoberta dos tipos de transações e de quais transações estão presentes no log de um servidor Web.

Alguns métodos de mineração dos padrões de utilização das páginas também recorrem a estes modelos: por exemplo, na descoberta de regras de associação, as páginas de conteúdo é que interessam ao algoritmo de busca; as demais páginas são consideradas como auxiliares da navegação. Um ponto importante a ser salientado é que uma página de conteúdo para um usuário pode ser de navegação para outro, e vice-versa (COOLEY *et al.,* 1997a, 1999).

As formas pelas quais os modelos de navegação e classificação das páginas Web podem ser utilizados na mineração de utilização da Web serão detalhadas no próximo capítulo.

2.5.3. Tipos de mineração de dados da Web

Uma das classificações propostas para a mineração da Web (ZAÏANE, 1999, ZAÏANE, 2000) distingue entre mineração de conteúdo ("*Web content mining*"), mineração de estrutura ("*Web structure mining*") e mineração de utilização ("*Web usage mining*").

Por essa classificação, na **mineração de conteúdo**, procura-se extrair informações relevantes do próprio conteúdo dos documentos da Web. Aí estão incluídas a mineração de dados textuais na Web, a mineração baseada em indexação de conceitos e as tecnologias baseadas em agentes. É uma mineração voltada precipuamente para os usuários finais da WWW.

A **mineração de estrutura** procura inferir o conhecimento com base na própria organização dos documentos Web e nos links entre eles. É voltada principalmente para os desenvolvedores e projetistas de sites e páginas Web.

Finalmente, a **mineração de utilização** é aquela que procura extrair conteúdos relevantes a partir dos logs de utilização, ou seja, serão minerados os próprios logs de

acesso dos servidores Web na busca de padrões de uso. Assim como a anterior, também volta-se primordialmente para os desenvolvedores e projetistas.

Uma classificação mais simplificada, porém, diferencia apenas entre **mineração de conteúdo** e **mineração de utilização** (COOLEY *et al.*, 1997). Aqui também, a **mineração de conteúdo** tem, como foco principal, a busca do conhecimento presente nos próprios dados contidos nas páginas Web. Porém, nesta classificação, as atividades que seriam de mineração de estrutura estão incluídas na **mineração de utilização**, já que os próprios métodos de preparação de dados, análise e busca de conhecimento sobre a utilização das páginas Web salientam a necessidade de se ter informações estruturais sobre os sites.

Tanto a atividade de mineração de conteúdo quanto a de mineração de utilização (pode-se dizer que mais especialmente esta última) apresentam uma série de problemas para a análise dos dados: necessidade de filtragem e integração de várias fontes de dados (logs de acesso, perfis de usuários, etc.); dificuldades na identificação dos usuários pela falta de atributos diferenciadores; necessidade de identificar as sessões ou transações dos usuários a partir de logs de acesso, topologias de sites e modelos do comportamento dos usuários.

Tais problemas têm sido abordados em anos recentes por uma série de trabalhos, e várias propostas têm sido feitas para contorná-los, algumas das quais, na área de mineração de utilização, estarão sendo analisadas no próximo capítulo. KOSALA & BLOCKHEEL (2000) e WANG (2000) descrevem diversas destas abordagens.

2.6. Mineração de conteúdo da Web

A mineração de conteúdo sofre bastante pela falta de estrutura dos documentos armazenados na Web. Contudo, a recuperação de informações na Web tem, nos

últimos anos, se beneficiado do desenvolvimento de agentes inteligentes e da tentativa de se conseguir um maior grau de estruturação dos dados disponíveis.

Os agentes Web podem ser divididos em três categorias distintas (COOLEY *et al.,* 1997): agentes de busca inteligentes, agentes baseados em filtragem/categorização da informação e agentes personalizados. Em MLADENIC (1999), pode-se encontrar um amplo levantamento de vários desses agentes.

Os agentes de busca inteligentes procuram informações relevantes baseados em características de domínios e perfis de usuários para organizar e interpretar as informações encontradas. Por exemplo, o Parasite, desenvolvido por SPERTUS (1998) permite a interpretação de documentos da Web codificados em tabelas em um banco de dados relacional, sendo baseado no Squeal, um sistema desenvolvido para permitir que sejam feitas consultas SQL à Web.

Já o Shopbot (DOORENBOS *et al.,* 1997) é um exemplo dos agentes que aprendem a partir de fontes de dados de estrutura previamente desconhecida. O agente busca na Web e oferece ao usuário os preços mais baratos para um determinado produto (por exemplo, um CD). Portanto, no caso do Shopbot, o aprendizado se dá a partir de informações genéricas sobre os domínios dos produtos à venda.

Vários agentes utilizam técnicas de recuperação de informação, aliadas às características de hiper-texto dos documentos Web, para realizar a busca, filtragem, e classificação desses documentos. Os "*web crawlers*", "*robots"*, ou "*spiders*" (KOBAYASHI *et al.,* 2000), são agentes especializados que podem ser utilizados para a extração das estruturas ou topologias dos sites Web.

Estas aplicações analisam o código de uma página HTML e criam uma estrutura com todos os seus links, seguindo então cada um deles e continuando o processo por todas as páginas do site, até que todo ele seja mapeado. Mas os *crawlers* também são

largamente utilizados na indexação de páginas Web para a mineração de conteúdo. Os principais mecanismos de busca indexada da Web, como, por exemplo, o ALTAVISTA (2002), costumam recorrer a esse artifício para automatizar a construção de seus bancos de dados de páginas.

Em PIROLLI *et al.* (1996), por exemplo, é utilizada uma ferramenta chamada "*the walker*", um agente autônomo que produz uma matriz de adjacências a partir de um determinado ponto de um site, extraindo e guardando informações sobre o nome, título, lista de links, tamanho e a data da última modificação de cada página. Estas informações poderão mais tarde ser utilizadas tanto para mineração de utilização e estrutura, quanto como apoio à mineração de conteúdo, criação de perfis, etc.

Agentes personalizados aprendem as preferências do usuário e buscam fontes de informação a partir de tais preferências, bem como a partir de outros usuários com interesses semelhantes, através de técnicas de filtragem. Por exemplo, o Letizia (LIEBERMAN, 1995), desenvolvido no MIT, foi um dos primeiros agentes desse tipo, auxiliando a navegação a partir dos hábitos do próprio usuário, sem o uso de palavras-chaves ou *rating.*

Já o WebWatcher (JOACHIMS *et al.*, 1997, ARMSTRONG *et al.*, 1995) aprende a partir dos hábitos de navegação de toda a sua comunidade de usuários, e, assim, lhes faz recomendações de quais são as páginas consideradas mais "interessantes". A pesquisa de tais agentes tem muitos pontos em comum com a mineração de utilização, pois em ambos os casos há um interesse em **como** o usuário navega.

Cumpre notar que a utilização de agentes que assistem a navegação é uma área tão próxima à da mineração de utilização que chega a ser difícil, senão impossível, traçar uma linha divisória clara o bastante entre ambas. Por exemplo, os agentes podem ser um mecanismo para a coleta de informações de utilização que servirão de entrada para uma posterior mineração, assim como podem se valer das informações obtidas durante a mineração de utilização, como será visto no próximo capítulo.

Uma abordagem alternativa aos agentes é aquela baseada em bancos de dados (COOLEY *et al.,* 1997), que tenta estruturar em níveis mais altos os dados semi-estruturados da Web e/ou utilizar mecanismos de consulta e técnicas de mineração de dados para analisá-los. Várias destas abordagens propõem a criação de bancos de dados multi-camadas, onde a camada inferior é composta pelos documentos Web semi-estruturados e as camadas superiores são meta-dados ou generalizações criadas a partir da organização dos níveis mais baixos (HAN *et al*.,1993).

Outras abordagens, tais como UnQL (BUNEMAN *et al.,* 1996) e W3QL (MENDELZON *et al.*, 1996) provêem sistemas de consulta e linguagens baseadas ou no conteúdo ou na estrutura dos documentos. UnQL aproxima-se da álgebra relacional e vale-se de um modelo baseado em grafos para fazer as suas consultas. Em W3QL, as informações sobre as páginas Web são guardadas em tabelas, apoiando-se também em estruturas de grafos, e consultadas a partir de uma linguagem semelhante a SQL, integrando tanto recuperação de informações textuais quanto consultas baseadas na topologia dos dados.

3. Mineração de Utilização da Web

Neste capítulo, são mostrados os principais aspectos que envolvem a mineração de utilização da Web, avaliando-se as etapas, tecnologias, métodos e trabalhos encontrados na área.

3.1. Aspectos gerais

Tanto organizações comerciais quanto não-comerciais possuem os mesmos objetivos relativos à organização de seus sites: os visitantes devem acessar as páginas "importantes", os links relevantes devem ser exibidos em cada página, e deve-se evitar que os visitantes fiquem desorientados.

Do ponto de vista do projetista Web, a estrutura de um site reflete a forma esperada de comportamento dos seus visitantes. Os conteúdos e referências entre as páginas mostram, portanto, estas expectativas do projetista. Contudo, ao montar um site Web, os projetistas baseiam-se muito mais em premissas, análises e investigações feitas sobre os interesses e perfis dos potenciais usuários do que na análise dos padrões reais de acesso destes (SPILIOPOULOU *et. al*, 1999).

Com o crescimento da Web e a complexidade cada vez maior de se projetar e implementar os sites, é patente a necessidade de informações sobre os padrões de utilização dos usuários. Essa análise pode auxiliar na reestruturação e projeto físico dos sites, além de servir como base para ferramentas de apoio à navegação.

A mineração de utilização da Web procura descobrir os padrões de navegação dos usuários dos servidores Web. Esta idéia foi apresentada inicialmente por CHEN *et*

al. (1996), MANNILA & TOIVONEN (1996) e YAN *et al*. (1996). Os dados utilizados na descoberta desses padrões estão concentrados principalmente nos logs dos servidores Web, que armazenam as interações dos usuários com as páginas neles armazenadas, mas também podem ser encontrados nas próprias estruturas dos sites Web (informações sobre as referências e links entre as páginas) ou obtidos dos usuários a partir do uso de programas CGI, *cookies*, agentes e outros mecanismos.

A análise de todos esses dados pode, desta maneira, ser uma ferramenta de grande utilidade no entendimento não só do comportamento subjacente aos usuários, mas também da própria estrutura da Web, ajudando também a organizar e modificar tal estrutura.

A mineração de dados de utilização Web apresenta, então, estes dois aspectos complementares: ao mesmo tempo em que é apropriada para analisar sistematicamente o comportamento passado dos usuários, serve como apoio na tomada de decisões sobre o que deve ser modificado em um site.

Há duas faces a serem analisadas no comportamento do visitante de um site Web (SPILIOPOULOU *et. al*, 1999):

a) seus interesses e as informações que acessa;

b) a maneira pela qual essas informações são acessadas.

Para uma análise do primeiro aspecto, podem ser utilizados questionários, pesquisas e outros mecanismos que permitam a classificação dos usuários segundo diferentes perfis, o que não se caracteriza propriamente como análise de utilização da Web. No, segundo aspecto, contudo, entra em cena a investigação de como se comporta o usuário ao navegar, o que pode ser feito a partir da interpretação dos logs de uso gravados pelos servidores Web. Esses dois aspectos podem ser considerados complementares, mas o presente trabalho concentrou-se especialmente no segundo deles.

Para COOLEY *et al. (*1999), a mineração de utilização Web vem reconciliar duas diferentes perspectivas:

a) como o projetista imagina que deva ser utilizado o site;

b) as maneiras reais pelas quais os visitantes estão efetivamente utilizando-o.

PIROLLI *et al.* (1996) levantam também a questão de que, na reconciliação destas duas perspectivas, é necessário o conhecimento da topologia (estrutura) do site Web como parte fundamental do processo de descoberta dos padrões de utilização seguidos pelos visitantes.

A mineração de utilização da Web pode ser classificada em dois ramos principais (COOLEY *et al*., 1997):

a) descoberta de padrões de acesso gerais;

b) descoberta de padrões customizados.

No primeiro caso, procura-se analisar os logs em busca de padrões e tendências de utilização genéricas. No segundo, tenta-se identificar os padrões específicos de um determinado usuário, a fim de se adaptar o próprio servidor Web ao mesmo. Este ramo está intrinsecamente ligado ao desenvolvimento de sites adaptativos (MAEDCHE *et al*, 2001, PERKOWITZ & ETZIONI, 1997, 1998, 1999).

Há uma ampla variedade de aplicações para as informações obtidas a partir da análise dos dados de utilização de um site Web, indo desde a utilização em campanhas promocionais, análise de estratégias de marketing, reestruturação e adaptação automática do próprio site, até o gerenciamento mais efetivo das comunicações de um grupo de trabalho e da infraestrutura organizacional, passando ainda pela distribuição de propaganda para usuários específicos e venda de espaços de publicidade.

Existem muitos produtos comerciais que fazem análises dos logs Web, tais como o NetGenesis (CUSTOMERCENTRICS, 2002), o WebTrends (WEBTRENDS, 2002) e

o NetTracker (SANE, 2002). Já o Analog (ANALOG, 2002) é um produto disponível gratuitamente e de código aberto, sendo um dos mais populares analisadores de arquivos de log.

A maioria destes produtos descreve a atividade dos usuários nos servidores e oferece várias opções de filtragem que permitem, por exemplo, saber o número de acessos ao servidor ou a determinados arquivos, o momento dos acessos, os nomes dos domínios e URLs dos usuários, além de prover outros tipos de estatísticas de tráfego e acesso a páginas ou pequenas seqüências destas.

Mas boa parte destas ferramentas diz muito pouco quanto aos relacionamentos mais profundos entre os arquivos acessados, os tipos de usuários e a estrutura dos sites Web. ZAÏANE *et al.* (1998) descrevem várias delas e terminam por concluir que não são adequadas a uma análise mais profunda e detalhada dos padrões de utilização.

Por exemplo, a simples contagem do número de acessos a uma página pode dar uma perspectiva enganadora a respeito de sua importância (FULLER & DE GRAAFF, 1996). Se uma página muito acessada só pode ser alcançada através uma seqüência de outras páginas, essas outras páginas também terão, como efeito colateral, o seu número de acessos bastante aumentado, ainda que não possuam a mesma importância da primeira.

As ferramentas para mineração de utilização da Web que têm surgido nos últimos anos podem ser agrupadas em duas categorias principais, segundo COOLEY *et al* (1997):

a) ferramentas para descoberta de padrões;

b) ferramentas para análise de padrões.

As primeiras extraem conhecimento sobre os padrões de uso a partir de técnicas de inteligência artificial, mineração de dados, utilizando subsídios de áreas tais como a

psicologia e a teoria da informação. O sistema WebMiner, em seus estágios iniciais (MOBASHER *et al*., 1996), seria um exemplo desta categoria, disponibilizando uma arquitetura genérica para a mineração de utilização da Web, que permite a descoberta automática de regras de associação e padrões seqüenciais a partir dos logs dos servidores Web.

As ferramentas da segunda categoria, mais voltadas para análise de padrões, tal como o WebViz (PITKOW & BHARAT, 1994), procuram auxiliar principalmente na visualização, entendimento e interpretação dos padrões obtidos, seja a partir de técnicas OLAP ou outros mecanismos de consulta: o WebLogMiner (ZAÏANE *et al.,* 1998), por exemplo, faz uso de técnicas de OLAP na visualização dos dados.

Seria útil acrescentar a esta classificação uma terceira categoria de ferramentas mistas, que perfazem tanto uma como outra atividade, pois há hoje cada vez mais dificuldade em dividir as duas tarefas, já que muitos dos trabalhos mais recentes procuram integrá-las. O próprio WebMiner (MOBASHER *et al*., 2000), por exemplo, veio mais tarde a apresentar estas duas características, pois, além da busca de padrões, passou a utilizar uma linguagem de consulta semelhante à SQL para analisar as informações descobertas, além de integrar outras estratégias como a descoberta de perfis de usuários, a mineração de conteúdo e a personalização de páginas.

3.2. Etapas da mineração de utilização da Web

A mineração de utilização da Web pode ser dividida em pelo três menos etapas distintas, cada uma delas com suas próprias características, procedimentos, métodos, entradas e saídas (COOLEY *et al.*, 1997, COOLEY, 2000).

1) **preparação de dados**, quando são realizadas a leitura, filtragem, limpeza e integração das diferentes fontes de dados de utilização Web; além disso, esta fase pode incluir ainda a identificação dos usuários e

de suas sessões (conjunto de acessos a páginas), para o caso das fontes de dados que não contenham uma correta ou precisa identificação. Pode ser necessária, também, a identificação de transações de usuários;

2) **descoberta dos padrões de utilização**, quando são efetivamente aplicados métodos e algoritmos de mineração de dados em busca de padrões úteis e significativos;

3) **análise e visualização dos padrões,** quando são disponibilizados e identificados os principais padrões encontrados, a partir das preferências e definições dos usuários.

Vale notar que estas correspondem, em certa medida, àquelas propostas por FAYYAD *et al.* (1996) para a descoberta de conhecimento. Aqui, porém, já estão de antemão definidos os domínios e os conjuntos de dados a serem minerados.

3.2.1. Preparação de Dados

Idealmente, a mineração de padrões deveria ser feita sobre um conjunto de sessões ou transações de usuários, contendo informações precisas e detalhadas sobre quem acessou o site, que páginas foram visitadas e em que ordem, e por quanto tempo cada página foi vista (COOLEY *et al.,* 1999). Considera-se uma sessão de usuário como composta por todas as páginas acessadas por um usuário em uma determinada visita ao site. Uma transação é um agrupamento semanticamente significativo de páginas, como será visto mais à frente.

Porém, devido à diversidade de fontes de dados e ao fato destas fontes nem sempre serem precisas e detalhadas como se desejaria, a atividade inicial a ser realizada antes de qualquer processo de mineração de utilização deve ser sempre o pré-processamento dos dados a serem minerados, incluindo aí o desenvolvimento de

um modelo de dados para os logs de acesso, a filtragem e limpeza dos dados brutos, a identificação de usuários, sessões e transações.

As fontes de dados de utilização de um site Web podem ser produzidas por agentes autônomos ou outras interfaces que façam o registro direto das ações dos usuários e que incluam uma correta e precisa identificação dos mesmos, assim como de suas sessões. Pode-se, por exemplo, ter um sistema implantado em um provedor de acesso Internet e que faça a gravação dos dados de uso diretamente, permitindo, desta forma, identificar exatamente qual usuário está acessando cada página em cada instante (SHAHABI *et al.* 2001).

Em outros casos, os dados de uso poderiam ser gravados em arquivos ou bancos de dados por páginas com scripts que utilizassem *cookies* ou outros mecanismos tais como identificadores de sessões (HALLAM-BAKER & CONNOLLY, 1996a). Nestes casos, a identificação posterior de usuários e sessões será desnecessária. A identificação de transações poderá também ser dispensável, caso os sistemas em questão já façam previamente tal identificação. O mesmo não se pode dizer, porém, da filtragem dos acessos, já que possivelmente estarão sendo gravadas visitas não só às próprias páginas Web, como também a arquivos de imagens, post-script, etc.

Vale ressaltar que a utilização de *cookies* e mecanismos de identificação explícita pode levar às mesmas preocupações de privacidade que serão abordadas mais adiante, ao se levantar os problemas na identificação de usuários em logs de servidores Web.

Entretanto, as principais fontes de dados de utilização de um site não são aquelas customizadas e obtidas por sistemas, scripts, páginas ou agentes, mas sim os próprios logs brutos gerados pelos servidores Web, devido, principalmente, à facilidade de obtenção dos mesmos. Os principais servidores Web do mercado

disponibilizam logs de diversos tipos, registrando cada um dos acessos realizados pelos usuários ao visitarem os sites.

Tais logs, porém, possuem sérias deficiências em termos de detalhamento e reconhecimento dos usuários e sessões, a ponto de alguns estudos argüirem que eles nem mesmo podem ser considerados como fontes válidas para a descrição e análise dos padrões de utilização de um site.

Com efeito, os dados contidos em um log de servidor Web não representam com total confiabilidade as sessões dos usuários devido a uma série de fatores: não só a presença de grande número de itens irrelevantes (da mesma forma que nas outras fontes de dados citadas anteriormente), mas também (e principalmente) a ausência de identificação única dos usuários, sessões e transações, bem como a inexistência dos registros das visitas a inúmeras páginas, devido a fatores tais como o uso de cache e servidores proxies.

3.2.1.1. Filtragem dos dados

A limpeza ou filtragem das fontes de dados é fundamental pelo fato delas conterem muitas informações irrelevantes, tais como acessos a arquivos de imagens, som, vídeo, animações, etc. Além disso, é importante salientar que tal limpeza é útil para qualquer tipo de análise dos logs de servidores Web, não somente para a mineração de utilização.

Os logs de servidores Web apresentam vários formatos, o que é uma séria preocupação a ser levada em conta por uma ferramenta de mineração de utilização Web que faça a filtragem de dados. Alguns trabalhos (por exemplo, COOLEY *et al.,* 1999) analisam apenas arquivos que estejam no formato padrão Common Log Format – CLF (LUOTONEN *et al*, 1995), especificado pelo CERN e NCSA como parte do protocolo HTTP.

Entretanto, este padrão, por ser bastante limitado em termos das informações que armazena, foi ampliado posteriormente pelo W3C para o Extended Log Format - ECLF (HALLAM-BAKER & BEHLENDORF, 1996), que adiciona, por exemplo, informações sobre a referência de origem da página (referidor ou "*referrer*"), ou seja, a página de onde partiu o acesso. Ao se utilizar o Extended Log Format, é possível que se especifiquem quais os campos que se deseja gravar no log.

Uma entrada de log de utilização contém, normalmente, o registro do percurso de uma página de origem até uma página de destino, incluindo o IP da máquina cliente que originou a chamada, o tipo de acesso (POST ou GET) realizado, além de outros dados, a depender do padrão utilizado pelo servidor Web.

No protocolo HTTP (FIELDING *et al.,* 1997), um cliente faz ao servidor uma requisição para cada arquivo necessário à visualização de uma determinada página. Assim, um acesso a uma simples página Web provoca a gravação de várias entradas de log no servidor, para cada uma das imagens, scripts ou outros arquivos carregados juntamente com a página. Em geral, somente as entradas de log associadas aos acessos às páginas HTML serão de interesse na mineração de utilização, pois os demais arquivos, especialmente imagens, são baixados automaticamente à revelia do usuário. Assim, nem sempre serão úteis para um sistema que procure minerar os padrões de navegação do usuário, já que não foram explicitamente solicitados por este.

Para remover imagens, uma abordagem simples é retirar do log todas as entradas associadas às extensões conhecidas para elas, tais como GIF, JPG, JPEG, etc. O mesmo pode ser feito em relação a arquivos de sons ou outras fontes multimídia. O ideal, contudo, é que o sistema permita a configuração de quais serão os sufixos de arquivos que devem ser ignorados no processo de limpeza (COOLEY *et al.,* 1999). O analista, muitas vezes, pode estar interessado em acessos explícitos a determinados tipos de arquivos ou imagens. Pode haver até mesmo, adicionalmente,

uma lista com os nomes de arquivos que devam ser explicitamente ignorados na mineração.

3.2.1.2. Identificação de usuários

Após a limpeza do log, o próximo passo a se considerar é a identificação de usuários, caso as fontes de dados em questão não contenham em si mesmas informações de identificação. Notadamente, os logs de servidores Web são bastante incompletos em relação a isto, ao contrário de fontes de dados geradas diretamente por outros meios.

No caso da análise de logs de servidores Web, várias dificuldades podem ser levantadas, principalmente o uso de cache e a presença cada vez maior de servidores *proxy* (PIROLLI *et al.*, 1996, PITKOW, 1997). Estes dois fatores podem distorcer sobremaneira os dados dos logs.

O cache, seja localizado no navegador ou no próprio provedor Internet (aqueles que oferecem servidores de cache aos usuários) faz com que a mesma página possa ser acessada várias vezes pelo usuário, sem que o servidor Web registre no log tais acessos. O cache local do navegador é um método para tornar mais eficiente o acesso a páginas muito visitadas pelo usuário e que não sofram muitas modificações. A página fica armazenada no disco local, e, com isso, nas próximas vezes em que for acessada, não será feita uma nova requisição ao servidor Web.

O uso de servidores de cache em muitos provedores de acesso é ainda mais emblemático desse problema, pois, aí, as páginas de maior utilização de **todos** os usuários do provedor serão armazenadas para uso futuro. Com isso, mesmo que um dado usuário nunca tenha acessado uma determinada página, ao fazê-lo pela primeira vez não será gerada uma entrada no log do servidor Web onde ela se localiza, caso a mesma já tenha sido armazenada no servidor de cache do provedor, por ter sido muito acessada por outros usuários.

Os servidores *proxy* também colaboram para tornar mais difícil a tarefa de identificação do visitante, pois o mesmo número IP em diversas entradas de log pode estar associado a diferentes máquinas clientes. O servidor *proxy* é uma maneira eficiente e segura que muitas organizações encontram para compartilhar o uso de um número IP por diversas máquinas da sua intranet. Assim, para os computadores da rede externa, todas aquelas máquinas serão vistas como uma só, já que utilizam o mesmo número.

Para contornar os problemas causados pelo uso de cache, pode-se forçar, na página Web, o "*by-pass*" do cache: obriga-se o navegador a recarregar a página sempre que ela for visitada, num processo conhecido como "*cache busting*".

A identificação do usuário pode ainda ser facilitada com o uso de *cookies*, da mesma forma que nas fontes de dados geradas por agentes outros, diferentes dos servidores Web. As entradas dos logs terão, assim, armazenadas as informações sobre quais *cookies* foram utilizados nos acessos.

Outra possibilidade é o registro explícito do usuário, que será convidado a entrar com os seus dados pessoais e senha sempre que iniciar a navegação por um determinado site. Assim, os logs registrariam também um identificador do usuário. Esta solução também é semelhante à adotada em outras fontes de dados, como visto anteriormente. O registro explícito é uma solução colaborativa; sem dúvida, a solução mais simples de se utilizar, mas nem sempre a mais factível.

Quando os usuários requerem explicitamente o uso do sistema, a identificação das sessões é bastante simplificada. Além disso, não existem aí as mesmas questões de privacidade advindas do uso de *cookies* e outros agentes, já que o usuário escolhe utilizar o sistema em troca dos seus benefícios. Contudo, no caso de sistemas que não forneçam um feedback imediato, o registro das ações do usuário levanta mais uma vez as mesmas questões.

SHAHABI *et al*. (1997) mostram uma outra solução em que é enviado ao cliente um agente Java que será responsável por mandar de volta ao servidor informações precisas sobre a navegação do usuário.

PITKOW (1997) questiona, porém, que essas soluções possuem limitações. Os *cookies*, por exemplo, podem ser removidos pelo usuário, ou podem ser desabilitados na configuração do navegador. O “by-pass” de cache também pode ser desabilitado e, além disso, é um mecanismo que termina por tirar a principal vantagem do uso do cache, que é a maior velocidade de navegação. Finalmente, o registro explícito, apesar das vantagens que traz pelas informações demográficas adicionais que oferece, levanta questões de privacidade que podem fazer com que muitos usuários desistam de navegar num site ou mesmo forneçam informações incorretas sobre si mesmos.

Haja vista todas essas limitações, deve-se apelar, na maioria das vezes, a heurísticas que permitam identificar se uma requisição de página gravada no log veio do mesmo usuário. Pode-se usar algoritmos ou estratégias que permitam testar diferentes combinações de números IP, nomes de máquina e informações temporais para se identificar o usuário (PITKOW, 1997).

PIROLLI *et al.* (1996), por exemplo, propõem que se identifique uma mudança de usuário sempre que houver uma mudança nas entradas do log para os campos que guardam o agente, software cliente ou sistema operacional, ainda que essas entradas provenham do mesmo número IP. Naturalmente, nesse caso, é necessário que os logs de utilização estejam gravando essas informações, o que nem sempre é o caso.

Podem ainda ser feitas considerações sobre o intervalo de tempo entre dois acessos consecutivos ou sobre quais páginas deveriam ser acessadas por duas entradas consecutivas para serem consideradas como originadas pelo mesmo usuário.

Uma outra heurística possível utiliza, além do log, a topologia do site, na tentativa de reconstituir os caminhos percorridos por cada usuário (PIROLLI *et al.,* 1996). A extração da topologia ou estrutura do site, como visto no capítulo precedente, pode ser feita com o uso de *crawlers.* Se, pela análise da topologia do site, uma página visitada não puder ser acessada através de qualquer seqüência de links que parta de outra página acessada anteriormente, então isso indicará que esta visita foi realizada por um segundo usuário utilizando o mesmo IP.

Contudo, há limitações para tais análises: por exemplo, se dois usuários com o mesmo navegador, mesmo número de IP e a partir do mesmo tipo de máquina acessarem o mesmo conjunto de páginas, a probabilidade de ambos serem confundidos é grande. Por outro lado, um usuário com dois navegadores na mesma máquina ou que digite diretamente as URLs no campo de endereço do navegador poderia ser erroneamente tomado como múltiplos usuários.

3.2.1.3. Identificação das sessões

O processo de identificação das sessões é semelhante ao de identificação dos usuários. Considera-se a sessão como uma passagem completa de um usuário por um site, desde a página por onde ele iniciou a passagem até a última página acessada. A sessão, portanto, inclui todas as páginas percorridas em uma visita completa do usuário ao site.

Para a identificação de sessões individuais de cada usuário pode-se adotar o método mais simples de utilizar-se um *time-out* de controle: sempre que o tempo entre dois acessos consecutivos para um determinado usuário for maior do que este *time-out*, assume-se que ele iniciou uma nova sessão. Muitos produtos comerciais utilizam, como *time-out,* 30 minutos; CATLEDGE & PITKOW (1995) chegaram ao valor de 25,5 minutos, a partir de dados empíricos. Após a análise estatística de um site, pode-se

calcular um valor apropriado para este tempo, que será utilizado pelo algoritmo de identificação de sessões.

Em SPILIOPOULOU *et. al* (1998, 1999), são adotados dois critérios complementares para identificação de sessões: além de considerar o *time-out* entre dois acessos consecutivos, como descrito acima, assume-se também que uma nova sessão é iniciada quando a duração total de uma seqüência de acessos exceder um determinado limiar.

Um problema adicional na identificação de sessões é a descoberta dos acessos que não foram registrados nos logs, devido a dois mecanismos que já causaram problemas anteriormente na identificação de usuários: o uso de cache (local ou não) e a presença de servidores *proxy* (COOLEY *et al.,* 1999). Ainda que se possa aqui utilizar soluções semelhantes, tal como a supressão do cache, pode-se também adotar um algoritmo de completamento de caminhos para tentar suprir essas lacunas.

Por exemplo, se for acessada uma página não diretamente alcançável a partir da página imediatamente anterior, pode-se assumir que o usuário usou o botão de "VOLTAR" do navegador, passando a percorrer páginas que estavam no seu cache, até que, a partir de uma dessas páginas anteriormente visitadas, acessou a página atual.

DIX & MANCINI (1997) assinalam que os modelos de hipertexto adotados pelos principais navegadores suportam a possibilidade de voltar diretamente para uma página que está mais distante no passado, não apenas para a página imediatamente anterior.

Além disso, o usuário do navegador tem à sua disposição a opção de recarregar a página atual, o que fará com que uma nova entrada seja registrada no log do servidor. Isto, juntamente com os freqüentes retornos a páginas em cache torna ainda mais problemática a análise do log.

É importante observar que o uso constante do botão de retorno do navegador e do botão de recarregar pode significar um projeto ineficiente do site, o que é uma informação bastante significativa para o projetista. Contudo, é difícil deduzir tal observação apenas a partir da análise dos logs de uso.

COOLEY *et al.* (1999) consideram que, assim como na identificação dos usuários, para se chegar à correta identificação das sessões não basta tão somente uma simples varredura dos logs de utilização, mas também é necessário o conhecimento da topologia do site. Portanto, as páginas faltosas não encontradas nos logs serão deduzidas a partir da análise da estrutura do site, sendo então adicionadas ao arquivo de sessões.

Para descobrir o tempo de utilização dessas novas páginas encontradas, pode-se tomar como base o tempo médio de acesso às páginas de navegação identificadas no log, já que as páginas faltosas podem ser consideradas como páginas de navegação, e não páginas de conteúdo.

Ao final do processo de identificação de sessões, será gerado um arquivo contendo todas as sessões dos usuários, o qual servirá como entrada para a fase de descoberta de padrões, ou para a etapa ainda preparatória de identificação de transações. Esse arquivo deve ser formatado convenientemente em certos casos: por exemplo, para os algoritmos de identificação de regras de associação, não serão necessários dados temporais, que deverão ser removidos. Alternativamente, os dados sobre as sessões descobertas podem ser armazenados diretamente em um SGBD.

3.2.1.4. Identificação de transações

A identificação de transações será necessária para o descobrimento de regras de associação na próxima fase de descoberta de padrões. Uma transação difere-se da sessão, por ser um agrupamento semanticamente significativo de referências de páginas (COOLEY *et al.,* 1999), podendo incluir desde apenas uma até todas as

páginas acessadas em uma sessão, o que dependerá dos critérios usados para identificá-la.

Entretanto, alguns trabalhos e autores dão outro sentido ao termo: em FOSS *et al* (2001), por exemplo, consideram-se transações as próprias entradas dos logs depois de filtradas, as quais serão, posteriormente, agrupadas em sessões.

A partir dos diferentes tipos de páginas - páginas de navegação e de conteúdo pode-se tentar identificar as transações de duas maneiras (COOLEY *et al.,* 1997a, 1999): na primeira abordagem, considera-se uma transação como o conjunto de acessos sucessivos a páginas auxiliares, até se chegar a uma página de conteúdo, que indica o final da transação. Na segunda, a transação inclui todas as páginas de conteúdo (e somente elas) visitadas pelo usuário.

A mineração, na primeira abordagem, revela os caminhos comuns até uma determinada página, através da análise das chamadas transações "de navegação" ou "auxiliares/de conteúdo". Na segunda, mostra os relacionamentos e associações entre as páginas de conteúdo, sem se interessar pelos caminhos que conduzem até elas, analisando apenas as transações ditas "de conteúdo".

Algumas dificuldades surgem na identificação de regras a partir de transações de conteúdo ou auxiliares/de conteúdo: por exemplo, seja uma regra de associação $A \rightarrow B$, onde A e B são páginas acessadas por um usuário, significando que, quando a página A é acessada, isso implicará no acesso também à página B. Se tal regra foi deduzida a partir de transações de conteúdo, seu significado será bem específico: A implica B somente quando ambas são acessadas enquanto páginas de conteúdo.

Porém, se for feita uma análise que leve em conta as páginas de navegação (auxiliares), esta regra poderá ser ignorada, não sendo descoberta pelo algoritmo de busca. Isto porque uma grande quantidade de usuários pode utilizar a página A apenas como uma página de navegação, e, ao fazê-lo, muitos deles não acessarão a

página B. Com isso, o grau de confiança da regra A $\rightarrow$ B poderá ser tão reduzido a ponto dela não ser reconhecida pela mineração.

Isto pode ser uma vantagem ou desvantagem, o que dependerá do que se deseja analisar. Por exemplo, sempre que se desejar ignorar tais tipos de regras, serão utilizadas na mineração as transações auxiliares/de conteúdo. Por isso, uma das chaves do sucesso da identificação de transações é saber quando uma página é de conteúdo e quando é somente auxiliar para um determinado usuário (COOLEY *et al.,* 1997a).

O processo de identificação de transações envolve tanto a divisão de transações em outras menores quanto o agrupamento de transações pequenas em outras maiores, numa seqüência de passos que conduza aos tipos de transações apropriados para o algoritmo de mineração que será utilizado na próxima etapa.

As duas abordagens, de divisão e de agrupamento de transações, possuem em comum o fato de que suas entradas e saídas constituem-se de uma lista de transações. Portanto, ambas as abordagens poderão ser utilizadas sucessivamente, em diferentes ordens. Porém, como a entrada inicial é o arquivo de sessões com todas as referências de páginas, a primeira etapa será sempre uma abordagem de divisão.

Considere-se *L* um conjunto de entradas de log de utilização Web, cada entrada $l \in L$ incluindo o número de IP do cliente, *l.IP*, o identificador do usuário, *l.uid*, a URL da página acessada, *l.URL*, e o momento do acesso, l.*tempo*, além de outros campos, tais como o método (GET, POST) de acesso, mas que não precisam ser considerados para efeito de identificação de transações.

Uma transação pode ser então considerada uma tripla (COOLEY *et al.,* 1997a):

$$t = < ip_t, uid_t, \{(l_{t1}.url, l_{t1}.tempo), ..., (l_{tm}.url, l_{tm}.tempo)\} >$$

onde, para $1 \leq k \leq m, l_{tk} \in L, l_{tk}.ip = ip_t, l_{tk}.uid = uid_t$

COOLEY *et al. (*1997a, 1999) comparam três principais métodos de identificação de transações:

a) identificação por duração da referência;

b) identificação por referências posteriores máximas;

c) identificação por janelas de tempo.

Identificação por duração da referência

O método de identificação por duração da referência baseia-se na suposição de que o tempo gasto numa página correlaciona-se com o fato desta ser uma página de conteúdo ou de referência para o usuário.

Identificação por referências posteriores máximas

O método de identificação por **referências posteriores máximas** ("*maximal forward references*") foi introduzido inicialmente por CHEN *et al.* (1996) para dividir as sessões dos usuários em transações. Ele considera que uma transação é composta por uma seqüência de páginas visitadas a partir de uma página inicial, até a última página acessada imediatamente antes da próxima **referência reversa**.

Uma **referência reversa** ("*backward reference*") é definida como uma página já presente no conjunto de páginas visitadas na transação. Por sua vez, uma **referência posterior** ("*forward reference*") é uma página ainda não visitada na transação. Uma **referência posterior máxima** é, portanto, a última página acessada pelo usuário antes dele tentar voltar para uma página já visitada na transação, denotando assim o final desta.

O início da próxima transação corresponderá à **primeira referência posterior** acessada logo após o final da transação atual. Portanto, após a detecção do final da transação, ao ser atingida a primeira **referência reversa**, devem ser ignoradas as

próximas **referências reversas**, até que se chegue a uma nova **referência posterior**, quando então se considera iniciada a nova transação. Este modelo pressupõe que as páginas de **referências posteriores máximas** são páginas de conteúdo, e as demais páginas que conduzem a elas são páginas auxiliares.

O algoritmo MF ("*maximal forward*"), introduzido por CHEN *et al.* (1996), tem como objetivo dividir o conjunto original de entradas de log em um conjunto de seqüências de percorrimento ("*traversal paths*"), cada uma delas representando as páginas acessadas até se encontrar uma **referência posterior máxima** a partir do início da sessão, as quais se denominarão **seqüências posteriores máximas**. A seguir, outros algoritmos devem identificar, entre estas **seqüências posteriores máximas**, padrões de percorrimento freqüentes, chamados **seqüências longas de referências**, ou seja: uma seqüência de referências que apareça um número suficiente de vezes no banco de dados (uma "referência" corresponde ao acesso a uma página, um "*page view*").

Tal problema é similar ao de se encontrar grandes conjuntos de itens para algoritmos de mineração de regras de associação, onde um conjunto grande de itens contém itens que apareçam num número suficiente de transações. A diferença, contudo, é que os grandes conjuntos de itens são apenas uma combinação de itens em uma transação, ao passo que uma seqüência longa de referências deve conter referências **consecutivas** até se encontrar uma **referência posterior máxima**. Isso faz com que novos algoritmos tenham que ser criados para a descoberta das **seqüências longas de referências**.

CHEN *et al.* (1996, 1998) desenvolveram dois algoritmos para esta tarefa: o FS ("*full-scan*"), que usa técnicas de *hashing* e *pruning* para resolver as discrepâncias entre os padrões de percorrimento e as regras de associação e o SS ("*selective-scan*"), que utiliza um conjunto de seqüências candidatas para determinar novos conjuntos de seqüências candidatas. O algoritmo SS deixa para o final a determinação

dos conjuntos de seqüências longas, quando então será varrido todo o banco de dados. No FS, ao contrário, cada passo necessitará de uma varredura completa do banco de dados, tornando-o, por isso, mais oneroso.

Ao contrário dos sistemas que utilizam referências reversas para identificar o aparecimento de novas transações (por exemplo, COOLEY *et al.,* 1997 e CHEN *et al.*, 1996), outras propostas, como a de SPILIOPOULOU *et. al* (1999), consideram tal artifício inválido, pois argumentam que uma referência reversa poderia simplesmente corresponder a um padrão de navegação "dirigida" (TAUSCHER & GREENBERG 1997), como visto no capítulo anterior.

Identificação por janelas de tempo

O método de identificação de transações por janelas de tempo é o mais simples: particiona a sessão em intervalos com duração menor do que um determinado parâmetro. Aqui, não se utiliza o modelo de páginas de conteúdo e auxiliares, mas sim a suposição de que uma transação tem um tempo médio de duração. Por isso, não serão produzidos os dois tipos diferentes de transações encontrados pelos métodos anteriores.

Análise dos métodos

A análise dos diferentes métodos de identificação de transações feita em COOLEY *et al.* (1997a, 1999) concluiu que o método de duração de referência é o único a encontrar regras que não poderiam ser deduzidas a partir de um simples escrutínio da estrutura do log.

Já o método de referências posteriores máximas não obteve bons resultados na tentativa de encontrar transações de conteúdo em sites que possuem alto grau de conectividade entre as páginas, o que se explica pelo fato de que nem sempre a última

página é a mais importante em um determinado percurso do usuário. Por outro lado, a descoberta de transações auxiliares/de conteúdo por este método levou a um número exagerado de regras, limitando o valor da mineração.

Tanto o método de duração da referência quanto o de referências posteriores máximas produzirão dois tipos de transações: as de conteúdo e as auxiliares/de conteúdo. Além disso, ambos são métodos que perfazem abordagens de divisão das transações.

O método de janelas de tempo pode ser usado tanto como uma abordagem de divisão, quanto como uma abordagem de agrupamento de transações: para um conjunto de entrada formado por transações curtas, caso seja utilizada uma janela de maior tamanho, muitas das transações serão provavelmente agrupadas em outras maiores. Este método pode, por isso, ser utilizado juntamente com os métodos anteriores, em etapas sucessivas, para, por exemplo, garantir que as transações tenham um tamanho mínimo. Vale notar que uma janela de tempo suficientemente grande fará com que uma sessão seja considerada uma única transação.

3.2.2. Descoberta de padrões

Após a identificação das sessões ou transações, há vários tipos possíveis de mineração a realizar nos dados: além de simples análises estatísticas, podem ser tentadas análises dos caminhos percorridos, descoberta de regras de associação e padrões seqüenciais, agrupamento e classificação, sempre utilizando as técnicas de mineração de dados já estabelecidas, modificando-as para o fim pretendido, ou, eventualmente, desenvolvendo-se novos métodos específicos para a mineração de utilização da Web.

A análise estatística procurará simplesmente dados de caráter geral, tais como o número de “hits” por página, as páginas acessadas mais freqüentemente, as páginas

mais usadas como ponto de partida ou de saída no site, o tempo médio de acesso de cada página, etc. É o tipo de padrão mais difundido nas ferramentas de mineração de utilização disponíveis.

A análise dos caminhos pode gerar grafos direcionados onde cada nó é uma página e cada aresta representa uma referência entre as páginas. Alternativamente, as arestas poderiam representar similaridades entre páginas ou o número de usuários que saíram de uma para outra página. Pode-se tentar determinar padrões de percorrimento freqüentes, ou seqüências longas de referência a partir da estrutura dos grafos obtidos, assim como os caminhos mais percorridos, além de outros padrões úteis.

A descoberta de regras de associação, como visto no capítulo 2, pode ser aplicada a um banco de dados de transações onde cada transação é composta por um conjunto de itens. O que se deseja descobrir é quando a presença de um determinado conjunto de itens implica na presença de um outro item na mesma transação. No caso de regras de associação aplicadas à utilização das páginas armazenadas nos servidores Web, cada item representa uma página acessada e uma transação é o conjunto de acessos de um usuário em uma determinada visita ao servidor, como visto na última seção.

Para diminuir o impacto do tamanho dos bancos de dados, costuma-se utilizar medidas de confiança e suporte dos itens baseadas no número de ocorrências das transações dentro dos logs.

Confiança é o percentual entre o número de transações que contêm todos os itens de uma regra e o número de transações que contêm os antecedentes da regra. Suporte é o percentual de transações que contém determinado padrão.

LI (2001) assinala que, na descoberta de regras de associação para a navegação Web, os antecedentes das regras a serem avaliados incluem não apenas

as páginas individuais, mas também subconjuntos (sem ordenação) de páginas e subseqüências ordenadas das mesmas. Além disso, acrescenta um critério para a seleção de regras baseado na comparação dos comprimentos das páginas.

As regras de associação descobertas podem ser de grande utilidade para o comércio eletrônico, como também para servir de apoio à organização do próprio servidor.

As técnicas de descoberta de padrões seqüenciais também serão úteis nesta fase da mineração de utilização. Na Web, os padrões seqüenciais encontrados podem, por exemplo, mostrar que um certo percentual de usuários que acessaram determinada página num servidor também fizeram uma compra on-line em uma outra página no intervalo de uma semana.

Outro tipo de padrão seqüencial associado ao tempo é aquele em que se determina qual o intervalo em que determinadas páginas foram mais acessadas, ou procura as características em comum dos clientes que visitaram estas páginas em um intervalo específico.

Nesse ponto, uma diferenciação importante entre a mineração de dados tradicional e a mineração de utilização da Web é que os mineradores de seqüências convencionais são projetados para a descoberta de seqüências freqüentes, não sendo, em geral, adequados para encontrar seqüências raras, porém confiáveis (SRIKANT & AGRAWAL, 1996). ZAKI *et al.* (1998) tentam contornar tal problema removendo as seqüências não interessantes em sessões consecutivas de mineração e pós-mineração. A abordagem adotada no sistema WUM (SPILIOPOULOU *et. al*, 1999), entretanto, é mais elegante, sendo detalhada mais tarde, quando forem descritos os trabalhos relacionados.

As técnicas de classificação e agrupamento também podem ser utilizadas nesta fase, para não só reunir as páginas semelhantes entre si, seja a partir de critérios pré-

definidos ou não, mas também para fazer o mesmo com as seqüências de páginas, facilitando, assim, a análise dos padrões de navegação e permitindo, além disso, a comparação destes com os perfis de usuários, se disponíveis. SU *et al.* (2001) propõem um algoritmo (RDBC – Recursive Density Based Clustering) para fazer o agrupamento de páginas com base na freqüência de suas utilizações, e não no seu conteúdo. Pode-se ainda classificar as páginas visitadas num servidor a partir de informações demográficas sobre os usuários.

COOLEY *et al. (*1999) advogam o uso de um filtro de sites que possa ser aplicado aos algoritmos de mineração, diminuindo assim o seu tempo de processamento. Como o filtro reduzirá o número de regras inúteis encontradas, pode-se diminuir as medidas de suporte e confiança dos algoritmos de mineração, a fim de aumentar a quantidade de regras e padrões úteis. Além disso, ele pode auxiliar a apontar certas características de uso do site.

O filtro poderia checar, por exemplo, se a página inicial está realmente sendo utilizada como ponto de partida pelos usuários, ou se outras páginas o estão, apontando tais discrepâncias. Poderia também ignorar as regras triviais, tais como uma regra que apenas confirme um link direto entre duas páginas, ou, ao contrário, perceber a falta de uma regra esperada, no caso contrário, quando tal link direto não está sendo utilizado pelos visitantes.

3.2.3. Análise dos padrões

Depois de encontrados os padrões de utilização, serão necessárias ferramentas de análise que permitam um melhor entendimento destes, tais como programas estatísticos, gráficos, de visualização e consulta. As análises serão feitas através da comparação do conhecimento descoberto sobre a utilização do site com a perspectiva que os seus projetistas possuem sobre como ele deveria ser utilizado.

É importante notar que, além dos padrões mais triviais, deve-se também procurar descobrir padrões de acesso com propriedades estatísticas interessantes (SPILIOPOULOU *et. al*, 1999). A dominância estatística não é sempre interessante, já que os padrões dominantes quase nunca acrescentam conhecimento a um analista expert. As características que tornam um padrão interessante são de caráter mais genérico, tais como caminhos quase nunca percorridos ou que percorram páginas com um assunto em comum.

A meta a ser perseguida pode ser, por exemplo, a descoberta de subcaminhos com propriedades estatísticas ou estruturais interessantes, a partir de um dado número de caminhos percorridos. Estão aí incluídos os subcaminhos que passem por páginas com determinadas características, que sejam percorridos por um número mínimo ou máximo de usuários que mostrem uma relação estatística confiável entre quaisquer duas ou mais páginas do caminho.

KATO *et al.* (2000) apresentam uma ferramenta voltada especificamente para auxiliar o projetista na análise dos padrões de utilização. Para avaliar as expectativas do projetista ao desenhar o site, a ferramenta analisa a relevância conceitual entre páginas e a conectividade dos seus links. Por outro lado, medindo a co-ocorrência de acessos entre diferentes páginas, ela avalia os padrões de uso dos visitantes.

Comparando esses dois aspectos, mostra ao administrador do site quais são os pontos falhos no seu projeto, apontando as páginas que não estão sendo efetivamente úteis (aquelas que não apresentam uma suficiente correlação entre a relevância conceitual e a co-ocorrência de acessos). A ferramenta disponibiliza de maneira gráfica os resultados das análises, através de um sistema de coordenadas polares que torna fácil a comparação das trilhas seguidas pelos usuários com os problemas detectados, dando pistas bastante úteis de como o site pode ser reestruturado.

Na área de visualização, o WebViz (PITKOW & BHARAT, 1994) utiliza um paradigma baseado nos percursos ou caminhos da Web (as seqüências de links entre as páginas), pelos quais se extraem, a partir dos logs de utilização, subseqüências dos padrões de percorrimento das páginas ou "Web paths". A Web, ali, é vista como um grafo cíclico direcionado, no qual os nós são páginas e as arestas são referências entre elas. O usuário pode, assim, visualizar e analisar os trechos de seu interesse, desprezando os irrelevantes.

As linguagens de consulta também acrescentam poder de análise aos padrões minerados (MOBASHER *et al.,* 1996), seja através de restrições criadas em alguma linguagem declarativa e armazenadas no banco de dados a fim de restringir a área a ser minerada, seja utilizando uma linguagem voltada especificamente para consultas nos padrões minerados, como é o caso da linguagem de consulta do WebMiner, baseada em SQL.

Outro exemplo de linguagem de consulta é MINT (SPILIOPOULOU & FAULSTICH, 1998, SPILIOPOULOU *et. al*, 1999), suportada pelo sistema WUM, que localiza os padrões correspondentes a valores especificados pelo usuário para determinados critérios (o próprio sistema provê valores apropriados nos casos não definidos explicitamente). Tais critérios incluem a especificação do conteúdo, estrutura e estatística dos padrões de navegação. Em MINT é utilizado ainda um conceito definido como "*interestingness*" (PIATESKI-SHAPIRO & MATHEUS, 1994) que permite ao analista especificar quais padrões serão úteis, a partir de critérios subjetivos.

As consultas em MINT possuem predicados que especificam o conteúdo, correspondente às propriedades das páginas (por exemplo, URL, tamanho), a estrutura, equivalente às posições relativas das páginas dentro dos padrões, e as estatísticas das páginas dentro dos padrões minerados.

Uma diferença importante entre consultas em MINT e regras de associação é que estas últimas não possuem qualquer ordenação, podendo levar a conclusões errôneas quanto ao comportamento dos usuários. Além disso, as consultas em MINT geram padrões de navegação que contêm os nós intermediários, que podem ser úteis para o analista descobrir, por exemplo, as partes em comum nos caminhos percorridos entre dois nós de seu interesse.

O processamento de uma consulta em MINT inclui os seguintes passos: 1) gerar um conjunto de "descritores" de todos os padrões de navegação candidatos, ou seja, aqueles que satisfazem às especificações de conteúdo e estrutura; 2) construir o padrão de navegação de cada um desses "descritores", testando os critérios estatísticos.

Outra área de grande influência na análise dos padrões encontrados é a de *data warehousing*, juntamente com as técnicas associadas de OLAP (ZAÏANE *et al.*, 1998), já que os dados de utilização da Web possuem muito em comum com aqueles de um *data warehouse* (DYRESON, 1997, KIMBALL & MERZ, 2000): tais dados são sempre agregados ao final dos logs, os quais crescem rapidamente ao longo do tempo, impossibilitando a análise de todo ele, e levando à necessidade de sumarização para fins de desempenho. Há também requisitos de segurança, pois certas porções dos logs não devem ser vistas pelo analista.

Assim, KIMBALL & MERZ (2000) apresentam uma visão unificada do processo de mineração de utilização da Web focada essencialmente em *data warehousing*, à qual eles denominam "*data webhousing*". Além de abordarem temas comuns aos outros trabalhos de mineração de utilização, fazem-no de modo integrado, incluindo a utilização de técnicas e ferramentas OLAP, e o desenvolvimento de dois esquemas complementares de *data warehouse* em estrela para o armazenamento de dados minerados dos logs dos servidores Web. Num deles, é criada uma tabela fato para

representar cada acesso ou clique do visitante; no outro, a tabela fato representa as sessões.

3.3. Trabalhos relacionados

Em MOBASHER *et al.(*1996) e COOLEY *et al.* (1997), é apresentada a arquitetura genérica proposta para a mineração de utilização Web e implementada parcialmente no WebMiner. Nela, o processo de mineração era dividido originalmente em duas etapas: a primeira incluía os processos dependentes do domínio que transformarão os dados brutos em transações a serem mineradas: atividades de limpeza e integração dos dados brutos e a divisão das entradas dos logs em sessões e transações. A segunda etapa abarcava a aplicação de técnicas de mineração de dados na busca dos padrões de utilização. Posteriormente (COOLEY *et al.,* 1999, MOBASHER *et al.,* 2000), a segunda fase foi dividida em outras duas: mineração de regras e análise dos padrões.

O ambiente WebMiner provê uma linguagem de acesso pela qual se podem especificar consultas com critérios mais sofisticados do que a simples freqüência de acesso. COOLEY (2000), em sua tese de doutorado, consolida vários dos pontos apresentados em COOLEY et al. (1997, 1997a, 1999), mostrando o desenvolvimento do sistema WebSIFT (Web Site Information Filter) para testar algumas hipóteses relacionadas à mineração de dados. Em relação ao pré-processamento de dados, o trabalho conseguiu provar duas das hipóteses: a) é possível inferir, a partir de arquivos de log no formato ECLF, as páginas não registradas nos logs devido ao uso de cache nos clientes; e b) o tipo de utilização de uma página pode ser inferido a partir do tempo gasto pelo usuário na mesma. Por outro lado, o trabalho não conseguiu provar a hipótese de que os dados disponíveis em logs ECLF sejam suficientes para identificar precisamente as sessões dos usuários.

MANNILA & TOIVONEN (1996) tentaram descobrir padrões freqüentes a partir dos logs de acesso a servidores Web (MANNILA *et al.*, 1995). Seus trabalhos desenvolvem algoritmos de mineração sobre os dados e consideram que, após a remoção dos acessos a imagens, os logs dos servidores dão uma visão exata da utilização dos sites (a mesma consideração é encontrada em CHEN *et al.,* 1996).

Em YAN *et al.* (1996), os usuários do servidor são agrupados a partir da análise dos logs dos servidores. Os links mostrados aos usuários são selecionados de maneira dinâmica, a partir das páginas acessadas pelos demais usuários do grupo ao qual ele pertence.

AMIR *et al.* (1997), introduzem um método de agregação dos dados a serem minerados em busca de regras de associação. Eles agregam em uma estrutura de trie as transações de um conjunto de dados como seqüências de itens com uma ordem arbitrária, combinando seqüências com prefixos iguais. A regra A $\rightarrow$ B será válida se e somente se os dois eventos A e B aparecerem no mesmo sub-ramo da árvore. Nesse caso, porém, cada regra de associação só aparecerá uma vez na árvore, pois não há ordem entre elas.

O sistema SiteHelper (NGU *et al.,* 1997), a partir de informações extraídas dos logs, associadas a dicas explicitadas pelos próprios usuários, faz a recomendação das páginas do site. Aproxima-se bastante, portanto, das soluções baseadas em agentes inteligentes descritas no capítulo 2, tais como o WebWatcher.

A ferramenta PageGather, introduzida por PERKOWITZ & ETZIONI (1997, 1998) vale-se de uma metodologia de agrupamento para descobrir e reunir, em *clusters,* páginas que foram visitadas juntas, enfocando as páginas de potencial interesse para um grupo de usuários, sem levar em conta, porém o caminho que conduziu o usuário até essas páginas. Além disso, eles foram inovadores ao propor o conceito de sites adaptativos, que aprendem através dos padrões de utilização e com isso adaptam-se e aperfeiçoam-se automaticamente.

SCHECHTER *et al.* (1998) utilizam “perfis de caminhos” (“path profiles”) para gerar dinamicamente o conteúdo a ser acessado pelos usuários. Eles consideram o problema do cache de menor importância, devido ao enfoque dado ao conteúdo dinâmico do site.

O sistema WebLogMiner (ZAÏANE *et al.,* 1998) utiliza técnicas de OLAP juntamente com as de mineração de dados, armazenando em estruturas de cubos os dados extraídos dos logs de servidores Web, a fim de permitir que sejam feitas predições, classificações e análises de séries temporais a partir dos dados assim obtidos.

São extraídos resultados interessantes sobre o tráfego Web e a evolução do comportamento dos usuários ao longo do tempo, mas sem se deter na relação existente entre o comportamento dos usuários e a organização dos sites de acordo com tal comportamento. O sistema assenta-se sobre a ferramenta DBMiner, um sistema de mineração de dados baseado em *data warehouse*, implementado sobre um banco de dados relacional.

A arquitetura do WebLogMiner possui quatro etapas distintas:

1) pré-processamento dos dados, com a filtragem dos dados irrelevantes e a criação de um banco de dados relacional com os dados remanescentes;

2) construção de um cubo de dados com as dimensões existentes;

3) uso de técnicas OLAP – drill-down, roll-up, slice and dice – no cubo formado em 2;

4) finalmente, utilização de técnicas de mineração de dados no cubo para predizer, classificar, e descobrir relacionamentos interessantes.

O sistema Footprints (WEXELBLAT & MAES, 1999) grava os passos percorridos pelos visitantes do site, acumulando-os em caminhos percorridos freqüentemente, que poderão ser utilizados pelos futuros usuários.

O modelo adotado no sistema WUM (SPILIOPOULOU *et. al*, 1999, SPILIOPOULOU & FAULSTICH, 1998) é interessante em dois pontos: a) antecipa o fato de que os indicadores de "importância" vão muito além da simples freqüência dos acessos; b) possui um destacado ganho de desempenho sobre os mineradores convencionais, por processar seqüências agregadas ao invés de dados de log brutos e aplicar passos intermediários de otimização durante o processo.

No WUM, são aplicados os passos tradicionais de preparação de dados (COOLEY *et al.,* 1997) para, em seguida, combinarem-se os dados em um "log agregado". Os acessos a imagens são filtrados, assim como em outros sistemas. O sistema assume que dois acessos consecutivos a partir da mesma máquina em um determinado intervalo de tempo provêm do mesmo visitante, porém o uso de outro método não afetaria as demais etapas de preparação de dados.

No final da fase de preparação de dados, as transações são agrupadas em uma coleção de trilhas, onde cada trilha representa uma determinada seqüência de páginas que foi percorrida por um ou mais usuários. À quantidade de transações que acessaram uma trilha chama-se tráfego. Em seguida, constrói-se uma árvore agregada de trilhas, combinando-se as trilhas com prefixos iguais.

Um prefixo é qualquer subconjunto de páginas percorridas na trilha, a partir da sua primeira página. Cada nó da árvore possui uma indicação da quantidade de usuários que chegaram a ele, a partir do prefixo que o originou. A esta quantidade dá-se o nome de suporte do nó. O nó raiz é criado artificialmente, para possibilitar a agregação de todas as trilhas da árvore. Seu suporte é a quantidade total de percursos feitos na trilha.

O WUM irá se preocupar não com a coleção de trilhas, mas com a árvore agregada montada a partir delas, chamada de "log agregado" do servidor Web. Como cada trilha e cada prefixo em comum aparecem na árvore apenas uma vez, esse log será bastante reduzido em relação ao original, diminuindo as necessidades de

armazenamento e possibilitando melhor desempenho por parte dos algoritmos de mineração. À medida que novos dados forem adicionados ao log original, poderão ser utilizados para formar uma nova árvore agregada de menor tamanho, que será combinada com a árvore agregada maior.

Ao invés de fazer como SRIKANT & AGRAWAL (1996), o WUM implementa um novo mecanismo para combinar sub-ramos do log agregado e descobrir padrões de navegação. Um “padrão de navegação” (SPILIOPOULOU, 1999) corresponde a uma generalização da árvore agregada, sendo um grafo construído de acordo com um descritor ou *template*. Este, por sua vez, equivale a uma seqüência de identificadores e máscaras, cada identificador correspondendo a uma ocorrência de uma página. O padrão de navegação é construído descobrindo-se quais sub-ramos da árvore estão de acordo com o descritor e combinando seus prefixos comuns e os nós correspondentes aos identificadores do descritor. Os suportes destes novos nós equivalerão à soma dos suportes dos nós combinados. Os demais nós permanecem intactos.

O fato de serem mantidos no padrão os demais nós não identificados da máscara leva a uma complexidade e custo de processamento maiores que em outros mineradores de seqüências (SRIKANT & AGRAWAL, 1996). Contudo, eles são fundamentais para o analista ou desenvolvedor de site.

O WUM é implementado em Java, com uma interface gráfica para realizar consultas e ver seus resultados, podendo ser utilizado para descobrir padrões em seqüências de vários tipos, não apenas em seqüências de navegação em páginas Web. Diferentemente de vários outros sistemas que utilizam os dados brutos dos logs dos servidores Web, na abordagem do WUM, como os caminhos possuem uma ordenação, dois eventos A e B poderão aparecer em várias seqüências agregadas, em diversas posições, o que torna seus algoritmos mais complexos.

GAUL *et al*. (2000) procuram ir além dos resultados obtidos no WUM. Eles não se limitam, como o WUM, a encontrar seqüências generalizadas descritas por *templates*. Ao invés disso, desenvolveram, a partir do algoritmo *apriori* de AGRAWAL & SRIKANT (1994), um algoritmo genérico para a descoberta de **todas** as subseqüências possíveis dentro de uma lista de seqüências. A vantagem desta abordagem, apontam eles, é que, ao se fazer a recomendação de páginas com base em subseqüências, não se perde a ordem de acesso às páginas (como acontece quando se usam conjuntos de páginas), nem se fica preso ao comportamento local de navegação, mas sim a padrões mais amplos.

Os trabalhos de BORGES & LEVENE (1998, 1999, 2000) utilizam o modelo de navegação baseado em cadeias de Markov, já citado anteriormente. Passando pelas etapas de preparação comumente encontradas nos demais trabalhos, eles definem, entretanto, uma gramática probabilística de hipertexto (HPG – "*hypertext probabilistic grammar*"), a partir da qual pode ser gerada uma linguagem probabilística de hipertexto que representa as sessões dos usuários.

Uma HPG é uma gramática regular probabilística que possui um mapeamento um-para-um entre o conjunto de símbolos não-terminais e o conjunto de símbolos terminais, além de dois estados adicionais, S e F, que correspondem aos estados de início e fim de uma sessão. Uma página Web corresponde a um símbolo não-terminal, e um link corresponde a uma regra de produção da gramática.

A HPG, por corresponder a uma cadeia de Markov, pode ter calculada a sua entropia, que servirá como base para a avaliação da probabilidade de serem alcançadas certas páginas a partir de outras. Uma alta entropia significa um elevado grau de incerteza nas ações do usuário, ou seja, não se pode fazer muitas predições sobre o seu comportamento futuro, o que pode decorrer da falta de informações sobre seu comportamento. Baixa entropia, por outro lado, implica num elevado nível de

conhecimento sobre o comportamento do usuário, o que se reflete numa gramática com um pequeno número de regras curtas.

LARSEN *et al.* (2000) também adotam uma visão estocástica da navegação na busca de padrões de navegação. Eles tentam segmentar os usuários a partir do algoritmo GGM (*Generalizable Gaussian Mixture*), que procura generalizar os comportamentos de navegação de modo a obter um aprendizado supervisionado com base num modelo de distribuição gaussiano. Com isso, conseguem não só a segmentação do comportamento dos usuários, mas também a classificação das próprias páginas Web.

JOSHI & KRISHNAPURAM (2000) apresentam uma abordagem para a identificação automática das sessões, utilizando técnicas de agrupamento fuzzy. Eles definem sessão como uma seqüência temporalmente compacta de acessos de um usuário. Adotam uma medida de distância entre duas sessões que tenta capturar não só as similaridades entre as URLs mas também a estrutura do site. Finalmente, desenvolvem dois algoritmos fuzzy robustos (FCMdd e FCTMdd) para agrupar as sessões em *clusters* significativos. O trabalho é ampliado em KAMDAR & JOSHI (2000) para mostrar como podem ser desenvolvidos sites que se adaptem ao perfil do usuário, construindo dinamicamente páginas com os links de seu interesses, a partir dos *clusters* construídos pelo sistema.

ANDERSEN *et al.* (2000) apresentam os resultados de um projeto levado a cabo em uma empresa financeira dinamarquesa, em que se procurou, através da análise dos logs de utilização dos servidores Web, descobrir a eficácia dos *banners* mostrados nas páginas Web da empresa e identificar quais as seqüências de páginas que levam o usuário perder o interesse no site e deixá-lo (a estas seqüências eles chamam "*killer subsessions*").

Adotando uma visão centrada em *data warehousing*, como na proposta por KIMBALL & MERZ (2000), os autores utilizaram, contudo um modelo diferente para a

definição das tabelas fatos: ao invés de centradas em cliques ou em sessões, centraram-nas em subsessões, já que, para o objetivo perseguido, necessitavam identificar de maneira eficiente quais as subseqüências mais interessantes. Para eles, uma subseqüência é um subconjunto qualquer dos acessos de uma sessão de usuário (excetuados os subconjuntos com apenas um acesso). Assim, uma sessão possui várias subseqüências, sendo que subseqüências diferentes podem se sobrepor (uma subseqüência ou subsessão corresponde, portanto, a um episódio). Através desta modelagem, lhes foi possível a correta identificação das *"killer subsessions"* e uma análise precisa e rápida da eficiência dos banners.

TVEIT (2000) utiliza a programação lógica indutiva do sistema Progol para descobrir padrões de utilização na forma de regras de primeira ordem que representem as sessões dos usuários, com o intuito de melhorar o desempenho e a própria qualidade do site. CHEN *et al.* (1996) detêm-se apenas em caminhos dominantes estatisticamente, através da descoberta de regras de associação.

NANOPOULOS & MANOLOPOULOS (2000, 2001) seguem uma linha parecida à de GAUL *et al.* (2000), porém se afastam do algoritmo *apriori,* por considerarem-no insuficiente, já que não leva em conta a estrutura do site. Assim, desenvolvem um algoritmo próprio que encontra conjuntos de seqüências, mas utilizando, como um dos critérios de seleção, os links entre as páginas do site.

ANDERSON *et al.* (2001) propõem dois sistemas de personalização, Proteus e MinPath, para a customização de sites Web voltados para usuários de dispositivos móveis, PDAs, telefones celulares e pagers. Ambos os sistemas baseiam-se na mineração dos logs de acesso e dos conteúdos dos servidores Web para a construção de modelos que representem os seus usuários e que possibilitem a transformação do site de modo a maximizar a sua utilidade para o visitante.

Recentemente, foi proposta uma aplicação XML que permite descrever os logs de um servidor Web: LOGML (PUNIN *et al.*, 2001). Ela estrutura o servidor Web como

um grafo Web, baseando-se, para isso, em XGMML, uma nova aplicação XML voltada à descrição de grafos (PUNIN & KRISHNAMOORTHY, 2001). Através de LOGML, pode-se obter um instantâneo do site Web à medida que o usuário visita as suas páginas e links. LOGML pode ainda ser utilizada para se prover metadados sobre o próprio servidor Web analisado.

3.4. Segurança

A segurança e a privacidade são fatores que sempre se interpõem quando se deseja realizar atividades de mineração de dados ou data warehousing. Ao se debruçar sobre os dados de um data warehouse, o analista pode encontrar não só padrões gerais mas também informações confidenciais sobre usuários individuais.

Na Web, isto fica ainda mais evidente, pois, ao navegar, o usuário está deixando o registro de seus hábitos e comportamentos, sejam eles de navegação, compras, preferências, em centenas, milhares de logs e bancos de dados espalhados por organizações as mais diversas.

Por isso, ao se realizar a mineração de utilização, deve-se sempre ter em mente o correto balanceamento entre as necessidades de informação e o direito à privacidade do usuário. Iniciativas legislativas como a *Directive on Data Protection* da União Européia procuram colocar um freio no afã inesgotável das empresas em registrar, analisar e utilizar como bem entenderem os dados dos seus clientes Web (WANG, 2000).

A Amazon.Com e a Doubleclick são dois exemplos de companhias que foram processadas por clientes sob a alegação de mau uso dos dados sobre eles coletados (WHITING, 2000). Outras companhias têm voluntariamente criado normas e limites internos para a coleta e utilização dos dados dos usuários, sempre lhes informando e

pedindo sua permissão para fazerem tais atividades – até mesmo enviar um *cookie* para a máquina do cliente.

4. MineraWeb: um ambiente para mineração de utilização Web

Neste capítulo, é apresentado o ambiente MineraWeb, voltado para a mineração de utilização da Web. São descritos os componentes e as etapas de mineração previstos pelo ambiente. Além disso, são mostrados como foram desenvolvidos alguns protótipos para validá-lo. Ao final, são mostradas duas aplicações que ilustram maneiras pelas quais ele pode ser usado como ferramenta para o auxílio à navegação e à construção de sites adaptativos.

4.1. Apresentação

Uma das dificuldades para a mineração de dados de utilização Web está no fato de os sistemas comerciais existentes, por sua natureza proprietária, serem fechados, com pouco espaço para configuração e ampliação de suas características por parte do administrador.

Por outro lado, os sistemas experimentais já propostos, abordados no capítulo anterior, sofrem quase sempre das mesmas limitações: foram desenvolvidos com vistas a um determinado tipo de experimento ou análise de um procedimento específico de mineração. Assim, os métodos que utilizam, os tipos de arquivos de entradas que analisam, os tipos de saídas e relatórios que oferecem são quase sempre fixos, pré-determinados.

Por exemplo, os trabalhos de TVEIT (2000) e LARSEN *et al.* (2000), são voltados principalmente para o desenvolvimento de modelos de navegação e técnicas

de mineração delimitadas. Portanto, ambos precisam se preocupar com as questões de menor importância envolvidas no processo de mineração de utilização, notadamente o pré-processamento dos dados, antes de se deterem no seu objeto de estudo, que é o estudo dos algoritmos que propõem

Portanto, o usuário está preso aos métodos de análise implementados e à estrutura mesma do sistema. O sistema WebMiner, como visto no capítulo anterior, é uma exceção, por fazer parte de uma proposta de arquitetura mais ampla, que engloba as várias fases do processo de mineração de utilização.

Ademais, os pesquisadores de mineração de dados que decidam enveredar pelos caminhos da mineração de utilização de dados da Web devem enfrentar um problema recorrente: todas as tarefas básicas de pré-processamento dos logs, limpeza, filtragem, identificação de usuários, sessões e transações, devem ser resolvidas, mesmo que se deseje apenas fazer o teste de um algoritmo específico de mineração de dados, ou de um método de consulta ou visualização de padrões.

Para preencher essa lacuna, propomos um ambiente modularizável, aberto e expansível para suportar todas as etapas da mineração de utilização Web, que integre diversas das propostas das várias ferramentas e sistemas abordados até aqui.

Por essas três características (modularizável, aberta e expansível), o ambiente proposto permite que se agreguem, a qualquer momento, novos métodos de leitura, filtragem e pré-processamento de dados, além de estar preparado para aceitar diferentes fontes de dados. Os dados do sistema estarão concentrados em uma base de dados relacional, acessível, portanto, por diversas linguagens de programação.

O ambiente não restringe necessariamente a linguagem de programação ou SGBD específico a ser utilizado, ou seja, seus módulos podem ser implementados de maneira independente no SGBD e nas linguagens que forem mais convenientes para o usuário.

Assim, os algoritmos usados para descoberta e análise de padrões poderão ser modificados, configurados e acrescentados sempre que necessário e na linguagem desejada. Podem também ser utilizadas ferramentas auxiliares de terceiros, geradores de relatórios, planilhas ou ferramentas OLAP para a extração de dados úteis ao usuário.

Pelo fato de estarem os dados armazenados em um SGBD relacional, poderão ser acessados, a qualquer tempo, a partir de páginas Web construídas dinamicamente. Assim, fica aberto o caminho para a construção de sites parcialmente adaptativos às necessidades e padrões de navegação dos usuários. Estes, por sua vez, podem ter seus dados cadastrados em perfis, os quais podem também ser utilizados pelo próprio sistema para refinar a descoberta e as análises de padrões de uso. A base de dados pode ainda incluir informações sobre as estruturas dos sites a serem analisados, também úteis para a análise de padrões.

A esse ambiente, juntamente com o toolbox ou conjunto básico de ferramentas e protótipos implementados, chamamos **MineraWeb** (**figura 1**). O MineraWeb é adequado para o pesquisador de mineração de utilização, não só por lhe oferecer um modelo comum para seus trabalhos, mas também por disponibilizar ferramentas que lhe permitirão concentrar-se no problema que estiver trabalhando, sem se preocupar com detalhes acessórios, tal qual, por exemplo, a tarefa de filtragem de logs.

O MineraWeb é, portanto, de grande valia para as pesquisas em mineração de utilização, por servir de plataforma comum de testes de novos métodos e ferramentas de apoio a serem aplicados na área, além de estudos comparativos de diferentes abordagens ou algoritmos de mineração.

Além desse enfoque nos pesquisadores de mineração de utilização, o MineraWeb pode também ser utilizado como ferramenta de apoio por um administrador ou projetista de sites Web que definir como deseja realizar suas análises

de utilização e o que pretende extrair em termos de conhecimento sobre o uso de seu próprio site ou mesmo de outros sites externos.

O MineraWeb será descrito com mais detalhes nas seções seguintes.

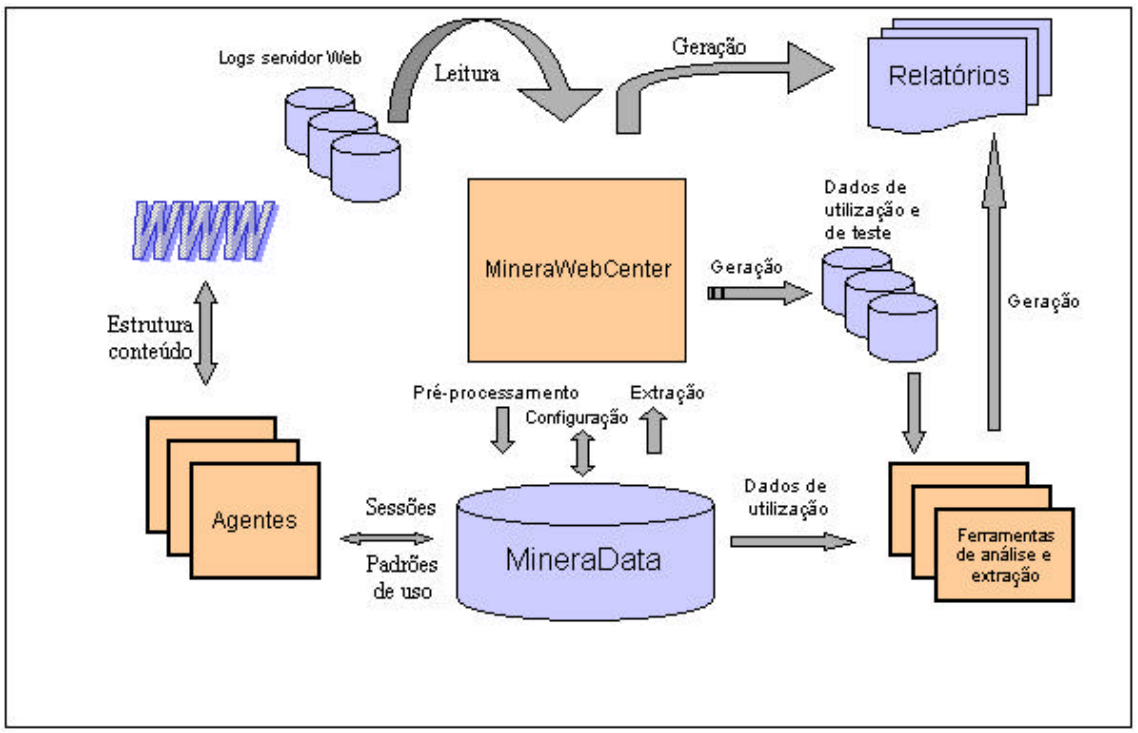


Figura 1 - Visão geral do ambiente MineraWeb.

4.2. As fases da mineração no MineraWeb

O MineraWeb prevê um repositório de dados central, (**MineraData**), um módulo principal para configuração, entrada e saída de dados e outras tarefas , incluindo a busca e análise de padrões (**MineraWebCenter**), além de outros módulos de busca e análise de padrões e módulos agentes que se conectem ao repositório e à Web, com múltiplas finalidades.

Com esses componentes, pode-se atender a todos os requisitos necessários ao processo de mineração de utilização, assim como prover meios para a utilização efetiva desses padrões, seja através da adaptação de sites ou de recomendação aos usuários, já que é possível o desenvolvimento fácil e rápido de páginas configuráveis a

partir dos próprios padrões de utilização descobertos e armazenados na própria base de dados. Com isso, um administrador Web pode projetar um site semi-adaptativo com diferentes níveis de configuração.

Pela classificação de COOLEY *et al.* (1997), o MineraWeb é um ambiente que se encaixa tanto na categoria de ferramentas para descoberta de padrões como na de ferramentas para análise de padrões. Portanto, é uma ferramenta (ou ambiente) mista.

As etapas de mineração propostas para o MineraWeb seguem aproximadamente aquelas definidas e seguidas por boa parte dos sistemas, sendo descritas nas próximas seções.

4.2.1. Integração e preparação de dados

No Mineraweb, as fontes de dados a partir das quais será feita a mineração de padrões incluem, naturalmente, os logs dos servidores Web. Porém, podem ser utilizados também dados trazidos diretamente por agentes (**figura 1**) que façam a interface com os usuários e registrem diretamente no repositório central os acessos. Neste caso, há uma grande vantagem: a identificação dos usuários e de suas sessões é enormemente facilitada, como visto no capítulo 2. Não serão, portanto, necessárias nem a filtragem, nem as etapas de identificação.

Os dados de acesso consolidados, já limpos e transformados, serão armazenados em tabelas em um banco de dados relacional, o MineraData, a partir de um modelo centrado nas sessões e nas visitas individuais (*page views*) dos usuários.

No caso dos dados provenientes dos logs de servidores Web, será utilizado o MineraWebCenter, que fará a filtragem das entradas de log indesejadas, para, em seguida, realizar a identificação de usuários, sessões e transações, através dos algoritmos disponíveis para o usuário.

Um ponto adicional a ser salientado é a possibilidade de serem gerados arquivos texto com formatos específicos (apenas as colunas desejadas) para o uso de ferramentas de terceiros, que estejam projetadas para fazer análises apenas em determinados formatos.

4.2.2. Descoberta de padrões

Após a fase inicial de integração das diferentes fontes de dados na base unificada, podem ser executados os procedimentos de descoberta de padrões, através de algoritmos específicos. No MineraWeb, isto pode ser feito tanto pelo MineraWebCenter quanto por ferramentas externas de terceiros.

Como realçado na seção anterior, as ferramentas de terceiros que estejam configuradas apenas para ler dados em formatos específicos (por exemplo, arquivos contendo sessões) podem se beneficiar da geração desses arquivos pelo MineraWebCenter. Mas essas ferramentas poderão também ser adaptadas para que acessem a base MineraData, no caso de ser possível tal adaptação.

4.2.3. Análise de padrões

Após a descoberta dos padrões, estes serão armazenados na própria base de dados, de maneira que possam ser analisados pelos administradores, utilizando o próprio MineraWebCenter ou ferramentas adequadas, tais como os cubos gerados pelo Analysis Services da Microsoft.

4.2.4. Aplicação dos padrões

Uma etapa importante no ambiente MineraWeb é aquela que vem **após** as etapas convencionalmente aceitas de mineração de utilização. Os padrões descobertos nas etapas anteriores poderão ser utilizados por agentes, ou mesmo

diretamente por páginas Web criadas dinamicamente, para, com isso, haver uma retroalimentação para o usuário, de onde tudo se originou.

Essa retroalimentação se dá a partir da criação de páginas ou sites que se adaptem aos visitantes. Ainda que muitos não considerem ser essa propriamente uma etapa da mineração de utilização, acreditamos que a sua inclusão no processo como um todo oferece uma visão mais clara até mesmo dos seus objetivos, sem nunca perder de vista o usuário, que é o foco último de todas as análises.

4.3. MineraData

A base de dados é o ponto central e espinha dorsal do ambiente MineraWeb. Nela, são armazenados de maneira unificada todos os dados utilizados pelos demais módulos.

A preocupação maior foi se projetar um modelo de dados lógico amplo e flexível o bastante para permitir o armazenamento dos diversos tipos de dados extraídos dos logs de servidores Web e outras fontes, além de outros tipos de dados a serem utilizados nas outras etapas da mineração de utilização.

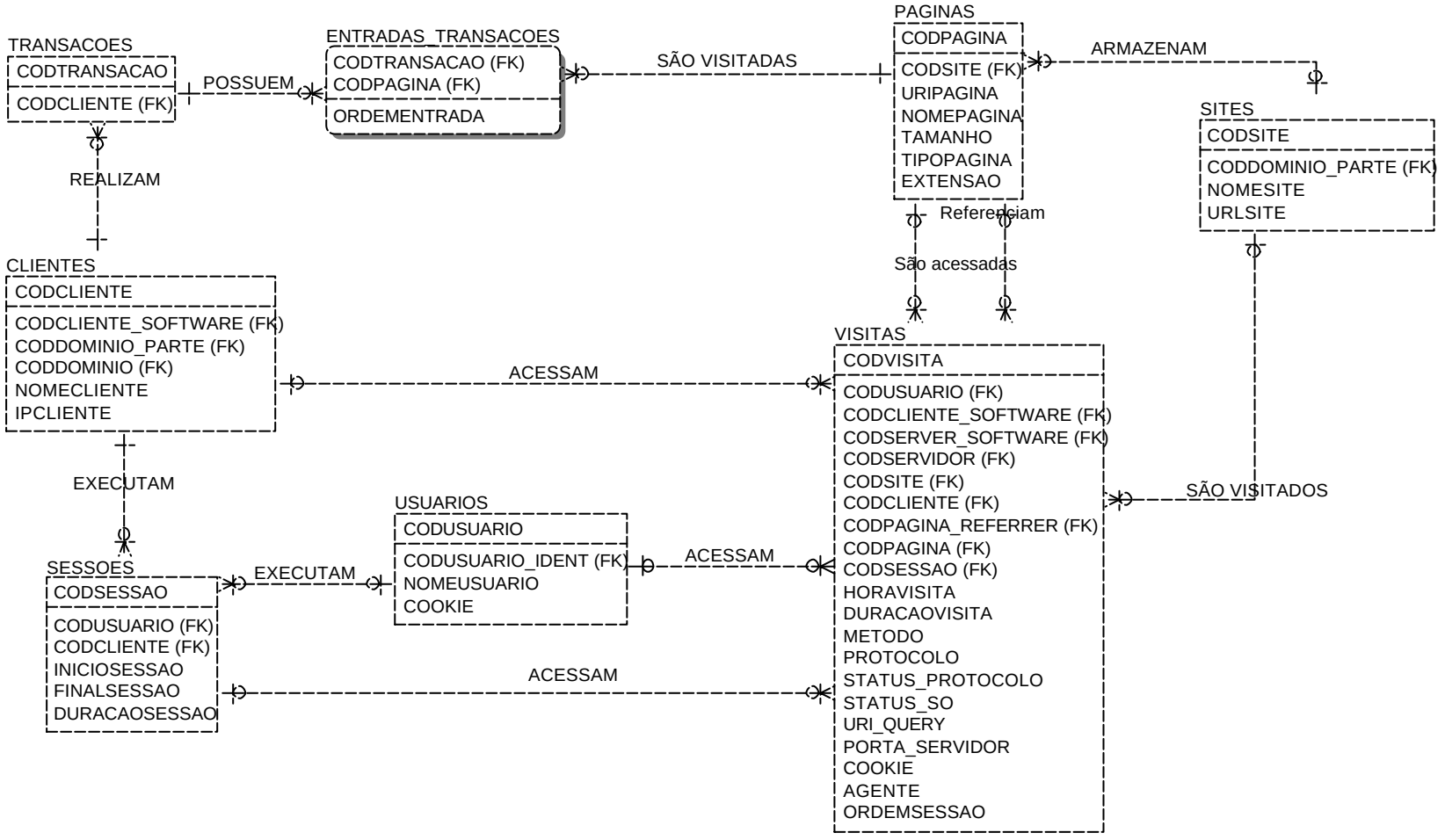
Como o ambiente em si não pretende estar preso a qualquer SGBD, o enfoque maior foi dado ao desenvolvimento do modelo de dados lógico. Porém, naturalmente, na implementação concreta, haverá sempre de se optar por um gerenciador de BD específico para o modelo físico correspondente, que poderá ser o SGBD mais adequado ou disponível a cada realidade.

Na implementação do protótipo, a modelagem de dados foi realizada utilizando-se a ferramenta CASE Platinum ERWin/ERX 3.52. O projeto físico, por sua vez, com todas as tabelas e domínios definidos para o MineraData, foi implementado no MS SQLServer 2000, um dos SGBD's mais populares do mercado.

Adicionalmente, experimentou-se também a geração de uma base de dados Oracle 8i a fim de validar a questão da portabilidade do ambiente.

A **figura 2** mostra as tabelas envolvidas na entrada de dados. Como o MineraWeb prevê a sua utilização para a administração de vários sites, a tabela SITES guarda as informações sobre cada um deles. Através do MineraWebCenter, poderão ser cadastradas as informações sobre eles. Tanto a tabela de VISITAS quanto a de PAGINAS são ligadas à tabela de sites. O modelo prevê algumas redundâncias a fim de reduzir as expectativas de desempenho à medida que a base cresça, diminuindo assim o número de junções necessárias em alguns casos.

Figura 2 - MINERADATA - Modelo para entrada de dados



A tabela VISITAS é central no modelo de dados. Ela registra cada clique ou *page view* realizado pelos usuários. Nela, estão armazenadas as informações básicas sobre uma visita ou acesso a determinada página. Há uma correspondência quase literal entre algumas das colunas da tabela VISITAS e os dados presentes em uma entrada de log de servidor Web (**tabela 4**).

Assim, nesta tabela são encontradas colunas que representam os códigos de retorno dos protocolos utilizados e do sistema operacional, os métodos, protocolos, cookies e agentes usados pelos clientes, a porta do servidor que disponibilizou a página, o horário da visita. Por isso mesmo, todas essas colunas serão inseridas sem maiores problemas durante o pré-processamento.

Por sua vez, outras colunas da tabela VISITAS só poderão ser preenchidas a partir de um certo trabalho de investigação. Por exemplo, todas as colunas que são chaves estrangeiras, especialmente as que referenciam a página acessada, o cliente que fez o acesso e o site em questão, só poderão ser preenchidas descobrindo-se, de antemão, quais as chaves primárias nas tabelas correspondentes, o que é tarefa da ferramenta que estiver realizando a entrada de dados.

As colunas DURACAOVISITA, ORDEMSESSAO e CODSESSAO, por outro lado, estão intimamente ligadas ao processo de identificação de sessões. Caso a entrada de dados seja feita a partir de logs de servidores Web, elas somente serão preenchidas quando a identificação for realizada. No caso da entrada de dados feita diretamente por agentes, não há essa limitação, já que o agente mantém um controle sobre as sessões.

Note-se que a tabela VISITAS é ligada duplamente à tabela PAGINAS, representando a página que foi acessada e a página a partir da qual partiu o acesso (*referrer).*

A tabela CLIENTES representa as máquinas a partir das quais são feitos os acessos. Contém portanto, uma identificação baseada tanto no nome qualificado do domínio quanto no número IP das máquinas.

A tabela USUARIOS, por sua vez, armazenará os usuários que executam os acessos e as sessões. Aqui, não há preocupação ainda de haver uma identificação rigorosa do usuário, com dados demográficos e similares. O enfoque maior é apenas na identificação menos formal do usuário, para as etapas de identificação de sessões e transações.

Pelos mesmo motivos de desempenho já apontados anteriormente, a tabela VISITAS liga-se tanto à tabela CLIENTES quanto às tabelas USUARIOS e SESSOES. A redundância pode poupar uma série de junções desnecessárias, especialmente na fase de identificação de sessões e transações.

As tabelas SESSOES e TRANSACOES serão preenchidas durante as etapas de identificação, no caso de dados obtidos a partir de arquivos de logs. Os agentes, ao fazerem a inserção de seus dados, já disponibilizarão informações sobre as sessões, e, eventualmente, sobre as transações.

Na tabela de sessões, são definidas colunas para a hora de início e fim da sessão e sua duração. Para se encontrar as referências acessadas em uma sessão, há que se fazer a junção desta tabela com a tabela VISITAS. As visitas são ordenadas, dentro da mesma sessão, pela coluna ORDEMSESSAO em VISITAS.

Para as transações, como não são importantes as informações detalhadas sobre cada acesso, pois o que mais interessa nesse caso é a agregação dos acessos s páginas, foi definida uma tabela ENTRADA_TRANSACOES, que exprime diretamente a ligação muitos-para-muitos existente entre TRANSACOES e PAGINAS.

4.4. MineraWebCenter

4.4.1. Configuração e pré-processamento

O MineraWebCenter é o módulo principal do sistema MineraWeb. É um programa desenvolvido em Borland C++ Builder 4, que tem como funções principais a definição dos diversos parâmetros de configuração do sistema, a carga e filtragem iniciais dos dados de log, extração de dados para geração de arquivos de logs customizados, geração de dados de teste.

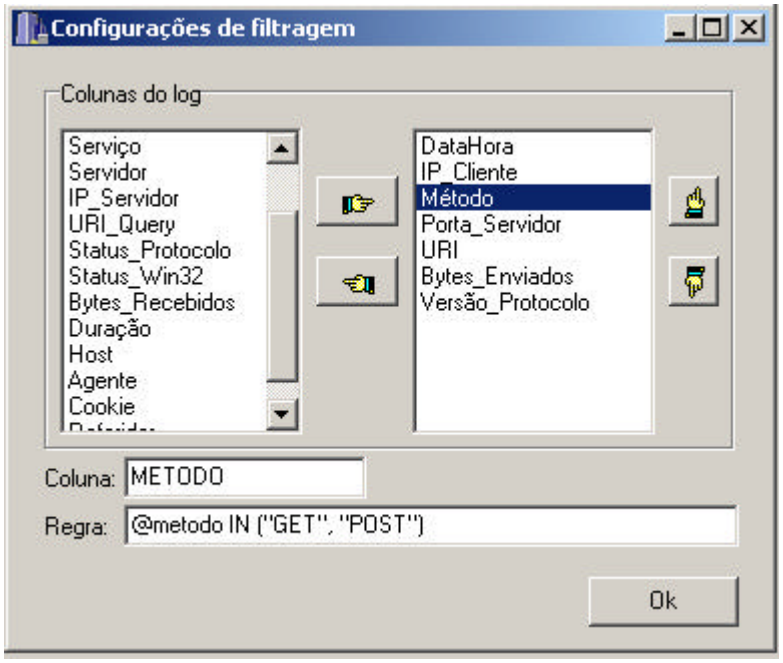


Figura 3 - MineraWebCenter: Tela de configuração da leitura de arquivos de log

O MineraWebCenter realiza a leitura e o pré-processamento de logs de servidores Web de uma maneira bastante flexível para o usuário. Dada a profusão de padrões de logs, e as distintas necessidades de filtragem por parte de cada sistema, o programa permite que se configurem não só quais serão as estruturas dos logs a serem lidos, como também quais os dados que deverão ser lidos, a partir de regras de filtragem.

Os logs de servidores Web, como visto no capítulo anterior, são geralmente armazenados utilizando os padrões **Common Log Format (tabela 2),** do NCSA, ou **Extended Log Format (tabela 3),** do W3C. Além disso, muitos servidores Web podem gravar dados em formato proprietário, como é o caso do IIS (Internet Information Server) da Microsoft, que pode armazenar os dados de utilização em formato binário ou em fontes de dados ODBC.

Para fazer a configuração de um determinado arquivo de entrada, o MineraWebCenter disponibiliza ao usuário uma lista dos possíveis campos ou colunas encontrados comumente nos logs Web (**figura 3**). O usuário terá então que selecionar quais desses campos estão presentes no log de entrada, e em que ordem. Para campos do tipo data, deverá especificar também o formato utilizado no arquivo.

Cada campo selecionado pelo usuário poderá ter a sua própria regra de filtragem, regra essa que será especificada de forma semelhante àquela pela qual se define uma regra de integridade de coluna (*column integrity constraint*) na criação de uma tabela em SQL.

Assim, cada coluna pode ter uma expressão relacional estilo SQL, incluindo os conectivos lógicos AND, OR e NOT, os operadores =, <>, >, <, >=, <= e as cláusulas LIKE, IN e BETWEEN. Da mesma forma que em uma regra de integridade de coluna, não podem, contudo, ser feitas, em uma regra de uma dada coluna, referências a outras colunas.

Nas regras, os nomes dos campos deverão ser sempre precedidos de '@', para que possam ser identificados pelo MineraWebCenter.

Através de alguns exemplos, procuraremos deixar claro como pode ser utilizada essa característica do MineraWebCenter para facilitar o processo de filtragem dos dados dos logs de servidores Web.

- @metodo IN ("GET", "POST")

Neste exemplo, foi especificada uma regra de filtragem para o campo “METODO” do log. Só deverão ser consideradas as entradas dos logs cujo valor do campo método seja GET ou POST.

- @datahora >= ‘1/1/2002’ AND @datahora < ‘2/1/2002’

Da mesma forma, poderia ser definida uma regra para a coluna “datahora” estabelecendo as entradas selecionadas log serão apenas aquelas com datas entre ‘1/1/2001’ e ‘`1/1/2002’, inclusive.

- @extensao NOT IN (‘GIF’, ‘JPG’, ‘BMP’)

Esta regra estabelece quais são as extensões de arquivos que devem ser ignoradas durante a leitura de um log de servidor Web. A coluna EXTENSAO, apesar de não estar definida na **tabela 4**, pode ser passada como parâmetro para a entrada de dados. O próprio MineraWebCenter fará o parsing das entradas, separando a extensão do resto do nome do arquivo.

- UPPER (@agente) LIKE ‘MOZILLA%’

Esta regra define que uma entrada de log só será aceita para inserção se o campo AGENTE for iniciado com a palavra Mozilla, não importando se em maiúsculas ou minúsculas. É uma mostra da flexibilidade desse esquema, pois podem também ser utilizadas e combinadas as funções disponíveis pelo SGBD.

Nesse ponto, deve-se tomar cuidado, entretanto, com a questão da portabilidade. Regras que tenham sido definidas utilizando-se as funções do SQL Server, por exemplo, podem se tornar inválidas caso transportadas para uma base Oracle. Por isso, é interessante que o administrador atenha-se às expressões suportadas pelo ANSI SQL, no caso de utilizar várias SGBDs diferentes.

Tabela 2: Exemplo de entradas de log em Common Log Format geradas pelo MS IIS

Cliente	Data e hora	Porta	Método	Página acessada	Protocolo	Status HTTP	Tamanho da Página
a.xyz.com.br	26/11/2001 16:14	8050	GET	/Default.htm	HTTP/1.1	304	1630
a.xyz.com.br	26/11/2001 16:14	8050	GET	/beto4.JPG	HTTP/1.1	304	5040
a.xyz.com.br	26/11/2001 16:15	8050	GET	/adriana/	HTTP/1.1	302	331
a.xyz.com.br	26/11/2001 16:15	8050	GET	/adriana/	HTTP/1.1	200	2217
a.xyz.com.br	26/11/2001 16:18	8050	GET	/adriana/ORGULHO.MUS	HTTP/1.1	200	1114
a.xyz.com.br	26/11/2001 16:25	8050	GET	/scripts/iserver.dll?p=100	HTTP/1.1	200	3040
a.xyz.com.br	26/11/2001 16:25	8050	GET	/scripts/iserver.dll?p=110	HTTP/1.1	200	5404

Tabela 3: Exemplo de entradas de log em Extended Log Format geradas pelo MS IIS

Data e hora	Cliente	Servidor	Porta	Método	Página	Query	Proto-colo	Status HTTP	Agente	Página referidora	Tamanho da Página
26/11/2001 16:14	a.xyz.com.br	www.xyz.com.br	8050	GET	/Default.htm		HTTP/1.1	304	Mozilla/4.51+en +(WinNT;+I)		1630
26/11/2001 16:14	a.xyz.com.br	www.xyz.com.br	8050	GET	/beto4.JPG		HTTP/1.1	304	Mozilla/4.51+en +(WinNT;+I)	http://ab.c.com.br/ Default.htm	5040
26/11/2001 16:15	a.xyz.com.br	www.xyz.com.br	8050	GET	/adriana/		HTTP/1.1	302	Mozilla/4.51+en +(WinNT;+I)	http://ab.c.com.br/ Default.htm	331
26/11/2001 16:15	a.xyz.com.br	www.xyz.com.br	8050	GET	/adriana/		HTTP/1.1	200	Mozilla/4.51+en +(WinNT;+I)	http://xy.z.com.br/ Default.htm	2217
26/11/2001 16:18	a.xyz.com.br	www.xyz.com.br	8050	GET	/adriana/ORGULHO.MUS		HTTP/1.1	200	Mozilla/4.51+en +(WinNT;+I)	http://k.lm.com.br/ adriana/	1114
26/11/2001 16:25	a.xyz.com.br	www.xyz.com.br	8050	GET	/scripts/iserver.dll	p=100	HTTP/1.1	200	Mozilla/4.51+en +(WinNT;+I)		3040
26/11/2001 16:25	a.xyz.com.br	www.xyz.com.br	8050	GET	/scripts/iserver.dll	p=110	HTTP/1.1	200	Mozilla/4.51+en +(WinNT;+I)		5404

Tabela 4: Colunas possíveis num arquivo de log lido pelo MineraWebCenter

DataHora	Timestamp da visita
IP_Cliente	Nome ou número IP da máquina cliente de onde partiu a visita
Usuario	Usuário que acessou a página, quando for o caso. Servidores que realizam autenticação podem gravar este dado.
Servidor	Nome do servidor Web
IP_Servidor	IP do servidor Web
Porta_Servidor	Porta utilizada no acesso ao servidor Web (normalmente, 80 para http)
Método	Método de acesso utilizado no acesso (mais comumente, GET ou POST)
URI	Universal Resource Identifier: nome padrão usado para a identificação de recursos na Internet.
URI_Query	Parâmetros passados no acesso à página.
Status_Protocolo	Código numérico retornado pelo protocolo HTTP
Status_Win32	Código de retorno do sistema operacional. De menor importância na mineração.
Bytes_Enviados	Número de bytes enviados do servidor ao cliente. Normalmente, corresponde ao tamanho do arquivo lido.
Bytes_Recebidos	Número de bytes recebidos pelo servidor do cliente.
Duracao	Duração da requisição. Corresponde ao tempo gasto pelo servidor Web para atender à requisição, não tendo nada a ver com a duração de referência, o tempo gasto pelo usuário na página.
Versão_Protocolo	Versão do protocolo utilizado no acesso. Por exemplo: HTTP/1.1
Host	Nome ou número IP do servidor Web, mais a porta de acesso: www.xyz.com.br:80
Agente	Nome do agente. Ex: Mozilla/4.51, para navegadores Netscape 4.51
Cookie	Nome do cookie, quando for o caso
Referidor	URL completa da página que referenciou a página atual. Por exemplo, http://www.xyz.com.br/default.html

Para tornar mais fácil o trabalho do administrador, o MineraWebCenter já possui uma lista pré-definida com os principais tipos de arquivos que são normalmente filtrados na mineração de utilização. A esta lista poderão ser incluídos outros tipos de arquivos a serem filtrados ou removidos aqueles tipos que sejam interessantes para o usuário e que, por isso, devam aparecer nos dados consolidados.

A configuração completa de carga de um site pode ser gravada na base de dados e reutilizada futuramente.

Na implementação do protótipo, a filtragem propriamente dita foi realizada por uma procedure armazenada no SGBD, o que torna mais simplificada a programação das regras de filtragem, já que estas têm a mesma estrutura de uma regra de integridade SQL.

As regras serão inseridas diretamente no código da procedure, por meio de macro-substituição. Com isso, o código do MineraWebCenter ficou mais enxuto, não sendo necessárias trabalhosas operações de conversão e formatação de dados em C, pois os dados lidos em cada coluna serão apenas repassados diretamente ao SGBD para que este realize a filtragem. O parsing limitou-se, portanto, à formatação das datas e à descoberta das extensões.

Como a implementação foi feita em SQLServer, a *stored procedure* foi escrita em Transact-SQL. Assim, a macro-substituição das regras no corpo da procedure foi feita utilizando-se a procedure interna "sp_executesql", que permite que sejam executados trechos inteiros de código construídos dinamicamente.

A procedure de entrada de dados "**insere_entrada**" recebe os parâmetros de uma entrada de log a ser filtrada e inserida, na forma de pares {($valor_1$, $regra_1$), ($valor_2$, $regra_2$),…($valor_n$, $regra_n$)}, fazendo, para cada par, a verificação do valor contra a regra correspondente (caso ela tenha sido definida). Se todos os valores supridos para uma entrada de log obedecerem às suas respectivas regras, ela será inserida na base de dados.

O seguinte trecho de código ilustra como é realizada a substituição dinâmica do código na procedure "**insere_entrada**". O exemplo mostra o teste feito na regra para o campo METODO. Testa-se se a regra associada ao campo está vazia. Caso negativo, ela é acrescentada ao código dinâmico de validação.

```
if @regra_metodo is not null
  set @sqlstring = @sqlstring +
                'if not ( ' + @regra_metodo + ')    ' +
                   '   set @result = 0                '
```

Para cada par (VALOR, REGRA), é repetido um trecho de código semelhante, até que, no final, a variável **@sqlstring** terá todas as regras construídas e prontas para serem testadas.

Só então será executada a consulta dinâmica, chamando-se **sp_sqlexecute**:

```
execute sp_executesql @sqlstring,
                N'@result bit OUTPUT,
                  @DataHora datetime = NULL,
                  @IP_Cliente tp_IP = NULL,
                      ...
                  @Metodo tp_metodo = NULL,
                      ...
                @result OUTPUT,
                      @p_DataHora,
                      @p_IP_Cliente,
                      ...
                        @p_Metodo,
                      ...
```

Assim, são definidos os valores a serem substituídos dinamicamente em cada regra, correspondentes aos parâmetros passados para "insere_entrada" (@p_datahora, etc.). A variável **@result** será utilizada como retorno, para avaliar se todas as regras foram bem-sucedidas e, conseqüentemente, se a entrada de log poderá ser inserida na base.

Na inserção dos dados, a procedure deverá ter o cuidado de localizar corretamente em quais tabelas (VISITAS, PAGINAS, etc.) eles deverão ser colocados. Além disso, deve fazer o controle de se uma determinada página já foi inserida.

O MineraWebCenter permite também que o conjunto de configurações de leitura especificadas pelo usuário seja gravado na base de dados, para que possa ser recuperado, modificado e reutilizado futuramente.

4.4.2. Exportação de dados e geração de dados de teste

Da mesma maneira que o MineraWebCenter faz a leitura e filtragem dos dados brutos dos logs de servidor Web, armazenando-os na base de dados central, ele permite, de maneira inversa, que sejam gerados arquivos customizados a partir desta

base de dados. Estes arquivos podem ser necessários para ferramentas de mineração e análise que utilizem dados armazenados em formatos específicos.

Portanto, se os dados sobre os acessos dos usuários, suas sessões e transações estão armazenados na MineraData, podem ser facilmente lidos na base e exportados para arquivos no formato desejado pelo administrador. A configuração desses formatos de gravação é feita de maneira bastante semelhante à configuração da leitura de dados: são especificados quais os campos desejados para os arquivos de saída, e, eventualmente, quais os critérios de filtragem das colunas exportadas.

Assim como no processo de entrada de dados, também aqui na exportação o trabalho pesado de filtragem é feito por procedures armazenadas no servidor de banco de dados. A procedure usada como base para essa extração de dados é "**extrai_entrada**", sendo bastante similar á '**insere_entrada**". Ao MineraWebCenter cabem apenas a configuração das regras, as chamadas à procedure e a gravação propriamente dita dos registros nos arquivos de saída. Também aqui as configurações podem ser gravadas na base para posterior modificação e uso.

Uma outra opção oferecida pelo MineraWebCenter é a geração de dados de teste. Muitas vezes o desenvolvedor de ferramentas de mineração de utilização necessita de dados de teste que possam ser utilizados, por exemplo, para a comparação de diferentes abordagens de mineração. Não é comum, porém, encontrarem-se ferramentas direcionadas especificamente para a geração deste tipo de dados.

O MineraWebCenter dá ao desenvolvedor esta possibilidade, permitindo-lhe especificar quais os tipos de dados de teste que deseja, e com que tipos de distribuição eles deverão ser gerados.

4.4.3. Identificação de sessões e transações

O MineraWebCenter é também o local a partir de onde o usuário ou administrador do site Web executa os procedimentos de identificação de usuários, sessões e transações. O sistema pode ser modularizado para implementar um ou mais tipos de algoritmos para cada uma destas etapas.

Na implementação do protótipo, os algoritmos foram desenvolvidos, mais uma vez, como procedures armazenadas no servidor. Além das procedures escritas em Transact-SQL, desenvolvidas no MS SQLServer 2000, implementou-se também uma procedure de identificação de sessões em PL/SQL, numa base Oracle, criada para assegurar que o sistema pode ser desenvolvido em diferentes plataformas.

Para a identificação de sessões, implementou-se o algoritmo de janelas de tempo, com o valor default de 25 minutos para a identificação das quebras de sessões, como proposto por CATLEDGE & PITKOW (1995). Porém, tal valor pode ser reconfigurado no próprio MineraWebCenter, antes de se dar início ao processo de identificação. O algoritmo permite também que se defina um valor para a duração máxima de toda a sessão, conforme proposto por SPILIOPOULOU et al. (1998, 1999).

A procedure "**identifica_sessoes_tempo**" é responsável por esse processo. Ela recebe como parâmetros o código do site, a data/hora de início e fim, definindo o intervalo que será varrido, e os parâmetros opcionais de time-out para quebra de sessão: INTERVALO_VISITAS, que define o tempo máximo entre duas visitas para que elas sejam consideradas pertencentes à mesma sessão, e DURACAO_TOTAL, o tempo máximo de duração de uma sessão.

Esses parâmetros podem ser definidos na tela do MineraWebCenter que executará a chamada a "**identifica_sessoes_tempo**".

A procedure basicamente percorre todas as entradas de deste site, para este intervalo, e vai inserindo-as numa tabela auxiliar, VISITAS_AUX, criando uma nova

sessão para cada cliente, à medida que eles apareçam. As seções em aberto vão sendo mantidas na tabela SESSOES_AUX. As estruturas das tabelas VISITAS_AUX e SESSOES_AUX são cópias das tabelas originais.

Cada nova página acessada pelo cliente é acrescentada à sessão atual, na tabela auxiliar, sendo calculado o tempo entre a página atual e a anterior (valor a ser armazenado mais tarde na coluna DURACAOVISITA da tabela VISITAS). Sempre que o algoritmo detectar que esta duração ultrapassou o limiar permitido entre duas visitas sucessivas, a sessão será dada como encerrada. Todas as suas entradas serão então copiadas de VISITAS_AUX para VISITAS, assim como a linha correspondente à própria sessão, que será movida de SESSOES_AUX para SESSOES.

A partir de então, uma nova entrada para aquele usuário marcará o início de outra sessão. A sessão também se considera encerrada se a sua duração total exceder o parâmetro DURACAO_TOTAL.

O MineraWebCenter possui também um módulo de classificação das páginas Web. O administrador poderá, através dele, especificar quais os tipos de página Web de um determinado site a ser analisado. A classificação utilizada é uma modificação daquela proposta em COOLEY *et al. (*1999): as páginas podem ser classificadas em páginas de cabeçalho, de conteúdo e de navegação.

Consideramos que as páginas ditas “pessoais” são apenas um caso específico das demais, sendo, portanto, desnecessária a sua inclusão, bem como as páginas de “look-up”, pois estas podem ser consideradas ora páginas de navegação, ora páginas de conteúdo. A coluna TIPOPAGINA foi colocada na tabela PAGINAS para prever essa classificação.

Caso o administrador não proceda à classificação manual, ela pode ser feita automaticamente pela chamada da procedure CLASSIFICA_PAGINAS, a partir do MineraWebCenter. A classificação automática, no entanto, apenas diferencia entre

páginas de conteúdo e de navegação, a partir dos tempos médios de referência de cada páginas. Páginas acima de um certo limiar de tempo são consideradas de conteúdo. As demais serão classificadas como de navegação ou auxiliares.

Para a identificação de transações, foi implementado um algoritmo que procura por transações de conteúdo, utilizando um limiar de tempo. Para isso, é de grande utilidade a classificação das páginas citada anteriormente. O algoritmo também está implementado na forma de *stored procedure* ("**identifica_transacoes_tempo**"). Seus parâmetros são o código do site analisado, o início e o fim do período no qual serão considerados os acessos, e um valor de limiar de tempo da transação.

Basicamente, ele procura, dentro das sessões, pelas páginas de conteúdo, ignorando as páginas de navegação, mas utilizando o limiar de tempo. Sempre que este for ultrapassado, a transação é finalizada. Todas as páginas de conteúdo acessadas dentro deste período são consideradas como partes da transação.

4.4.4. Busca de padrões

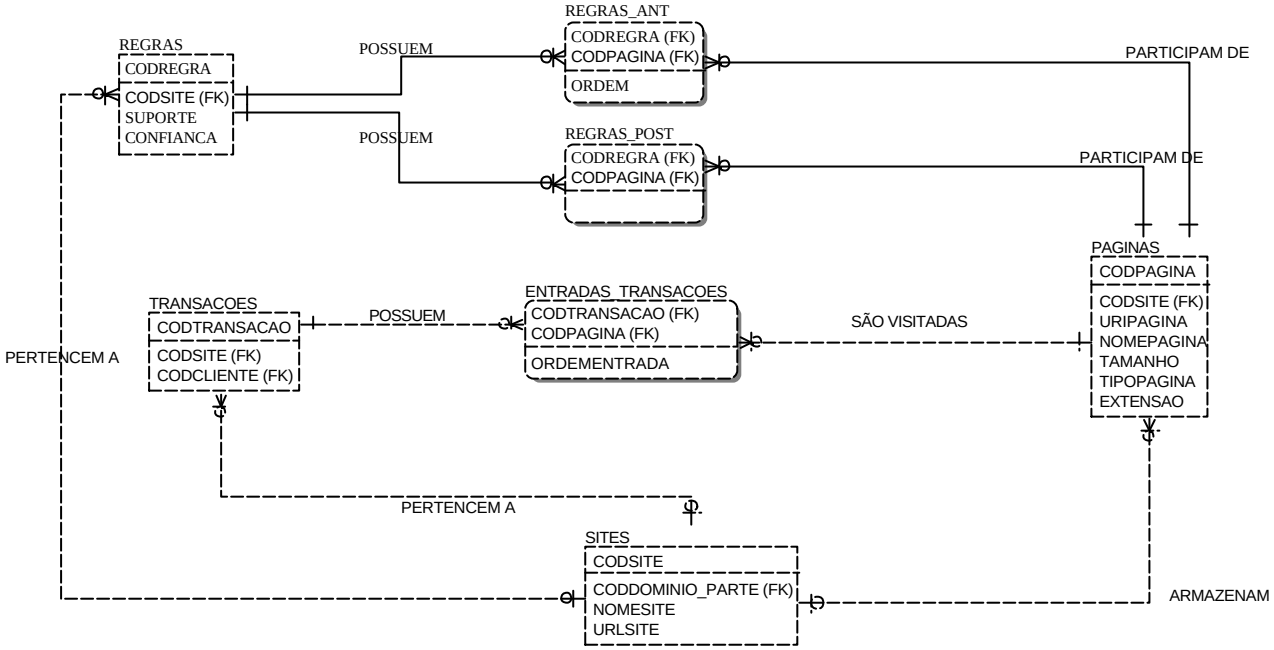
Outro aspecto do MineraWebCenter é que a ele podem ser incorporados algoritmos e módulos para a descoberta de padrões. Para ilustrar esta finalidade, foi implementado um algoritmo simplificado para a descoberta de regras de associação entre as transações identificadas anteriormente. O algoritmo recebe como parâmetros um valor de suporte e outro de confiança, além do código do site, tendo sido também implementado como uma procedure armazenada em Transact-SQL: "**identifica_regras_1**".

As regras são obtidas no formato $A \rightarrow B$, significando que o acesso à página A é acompanhado do acesso à página B, dentro de uma mesma transação. Todas essas regras são então armazenadas na própria base de dados, onde poderão ser

consultadas seja pelo MineraWebCenter, seja por ferramentas geradoras de relatório ou de páginas Web dinâmicas.

As regras descobertas serão guardadas nas tabelas REGRAS, REGRAS_ANT e REGRAS_POST (**figura 4**). A tabela REGRAS mantém as informações sobre as métricas de suporte e confiança de cada regra descoberta. Em REGRAS_ANT, ficam armazenados os antecedentes de cada regras. Cada antecedente pode ter uma ou mais páginas, bem como pode ser ordenado (ainda que o algoritmo implementado no protótipo só considere antecedentes compostos por apenas uma página). Em REGRAS_POST, são armazenadas as páginas referenciadas na segunda parte das regras.

Figura 4 – Modelo de dados para a descoberta de regras de associação



4.5. Outras ferramentas de busca, análise e visualização

O ambiente MineraWeb prevê que a busca de padrões possa ser realizada não só pelo MineraWebCenter, como também por ferramentas externas incorporadas ao toolbox. Em ambos os casos, os dados utilizados poderão ser tanto aqueles armazenados na base de dados, quanto os arquivos customizados gerados pelo MineraWebCenter.

Além disso, sempre que for necessário, o modelo de dados pode ser modificado e ampliado para suportar o armazenamento de dados auxiliares ou dos resultados dos procedimentos executados pelas ferramentas e módulos de busca de padrões. Exemplo disso são as regras de associação geradas pelo MineraWebCenter, que são inseridas em tabelas da própria base de dados.

Para a visualização das informações de utilização obtidas, foi testada também a implementação de cubos OLAP que podem ser manipuladas pelo administrador. Para tanto, foram utilizadas as ferramentas que compõem o pacote Analysis Services do MS SQLServer 2000.

Por exemplo, foi criado um cubo tendo como tabela fato VISITAS, num esquema *snow-flake.* As dimensões agregadas foram os domínios, máquinas clientes e as páginas (**figuras 2 e 6**). A dimensão dos domínios é hierárquica, representando desde as máquinas até os domínios de mais alto nível (*org, com, edu, br,* etc.). As dimensões numéricas desse cubo permitem analisar não só as freqüências de acessos às páginas, como também a duração dos acessos. O Analysis Services Manager permite não só a definição do cubo, dos fatos e suas dimensões, como também a manipulação do mesmo.

4.6. MineraCrawler

Este módulo foi desenvolvido com o propósito de varrer um determinado site, percorrendo todos os seus links e gravando a sua estrutura na base de dados do MineraWeb. Assim, sempre que o administrador precisar carregar a estrutura de um site, poderá recorrer ao MineraCrawler, que pode ser executado a partir do próprio MineraWebCenter.

Como toda ferramenta desse tipo (“crawlers”, “spiders”), o MineraCrawler funciona em background, acessando o servidor Web onde se encontra o seu “objetivo” tal como se fora um navegador, porém deixando uma “assinatura” que permite a fácil identificação dos acessos por ele realizados.

A configuração do MineraCrawler fica armazenada na base de dados MineraData, sendo feita através de opções do MineraWebCenter. Ao fazer a leitura das páginas de um site, a configuração dirá ao software qual o nível de profundidade desejado na busca e quais os tipos de páginas que devem ser ignorados (as extensões dos arquivos).

A implementação do MineraCrawler foi feita em Delphi 5, utilizando uma biblioteca pública de funções e componentes de acesso ao protocolo HTTP. O programa, a partir da página inicial indicada na configuração, faz uma varredura do tipo *breadth-first,* prosseguindo até o nível configurado.

4.7. MineraRedirect

O MineraRedirect é uma ferramenta que faz o redirecionamento das páginas visitadas em um site. No ambiente MineraWeb, corresponde a um agente. A função do MineraRedirect é sugerir aos usuários as páginas que eles podem visitar a partir da página atual.

O MineraRedirect também pode utilizar as regras mineradas para mostrar ao usuário quais as páginas que costumam ser acessadas a partir do ponto onde ele estiver. Oferece também, como opções de visualização, a listagem das transações e seqüências mais comuns iniciadas numa determinada página, assim como as quantidades de acesso e tempos de referência médios associados a cada página. Finalmente, o MineraRedirect pode mostrar e sugerir páginas escolhidas aleatoriamente dentro da base de dados, caso o usuário assim o deseje.

O MineraRedirect é uma aplicação escrita em Delphi 5, a partir da mesma biblioteca de funções e componentes utilizada no MineraWebCrawler. Sua função é interceptar todas os acessos de um usuário às páginas que ele visita durante sua sessão, fazendo com que estas páginas sejam reformatadas antes de serem enviada ao navegador do cliente. Essas páginas podem estar em qualquer site Web, pois o MineraRedirect será responsável por buscá-la para o usuário.

A reformatação fará, essencialmente, com que todos os links da página visitada sejam modificados de maneira tal que as URLs apontem sempre para o próprio redirecionador, com os links originais transformando-se em simples parâmetros dos links modificados. Assim, o usuário estará sempre navegando dentro do próprio site do redirecionador, ainda que esteja acessando páginas de diversos locais.

A página construída pelo MineraRedirect possui dois quadros (“frames”) particionados horizontalmente (**figura 5**). O frame superior, de menor tamanho, é uma espécie de cabeçalho, servindo para mostrar ao usuário informações gerais de identificação, mas, principalmente, as informações que possam ser de interesse para sua navegação, como, por exemplo, quais os links mais acessados por outros usuários a partir da página atual. Este tipo de informação é obtido dinamicamente a partir da própria base de dados do MineraWeb.

O frame inferior, por sua vez, mostrará apenas a página original que está sendo visitada, porém esta será apresentada ao visitante na versão modificada, com os links

reformatados e apontando para o próprio site do redirecionador, deixando assim o usuário numa espécie de “gaiola virtual”.

O MineraRedirect, é, portanto, uma espécie de agente auxiliar da navegação,que aproveita as informações armazenadas na base de dados do MineraWeb para fornecer ao usuário dicas úteis para sua própria navegação. Estas informações podem facilitar a navegação feita com fins específicos, quando o usuário estiver interessado, por exemplo, em encontrar páginas relacionadas a um determinado tema, ou páginas acessadas por um determinado grupo de pessoas.

O modelo de dados do MineraWeb prevê também que sejam guardadas informações demográficas sobre os usuários (**figura 6**). Com isso, a ferramenta pode também se classificar, nesse aspecto, como auxiliar da navegação colaborativa. E caso o usuário esteja interessado na navegação serendípica, pode visitar apenas os links aleatórios recomendados pelo sistema.

Todavia, além de ser uma ferramenta de apoio à navegação, o MineraRedirect faz também a coleta de dados de utilização, armazenando-os diretamente na base de dados do sistema. Os dados por ele coletados apresentam uma grande vantagem em relação àqueles obtidos dos logs de servidores Web pelo MineraWebCenter: já possuem a identificação de usuários e sessões, pois a navegação é feita a partir do registro explícito dos usuários.

O MineraRedirect, portanto, gravará na base não somente os dados que são obtidos normalmente em arquivos de logs, mas também os dados sobre os usuários e suas sessões. Pode-se também modificá-lo de maneira a tentar identificar e incluir dados sobre a transações dos usuários.

Além disso, o MineraRedirect pode ser configurado para determinar quais os tipos de arquivos cujos acessos deverão ser logados. Portanto, para os dados obtidos

a partir da ferramenta, será desnecessário que se proceda, posteriormente, às etapas de limpeza e filtragem, identificação de usuários, sessões e transações.

Na versão implementada no protótipo, o MineraRedirect foi construído como uma aplicação ISAPI, uma biblioteca de ligação dinâmica (DLL) a ser carregada automaticamente pelo servidor Web (foi utilizado o MS IIS - Internet Information Services,sob Windows 2000) quando o redirecionador for acionado. Entretanto, para os servidores Web que não sejam compatíveis com o padrão ISAPI, a ferramenta pode ser fácil e rapidamente adaptada para se transformar em um programa CGI, compatível com os mais diversos servidores do mercado.

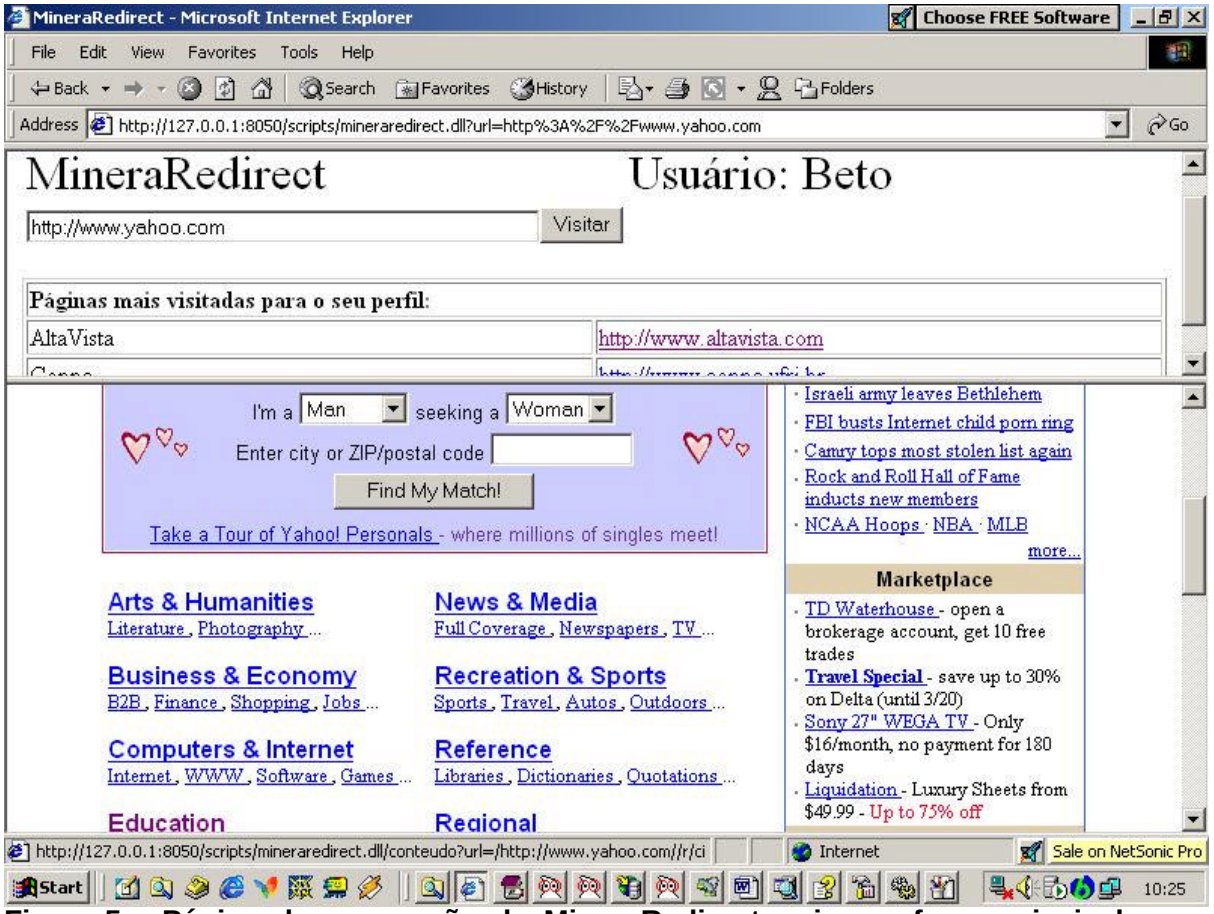


Figura 5 - Página de navegação do MineraRedirect: acima, o frame principal; abaixo, o frame de conteúdo.

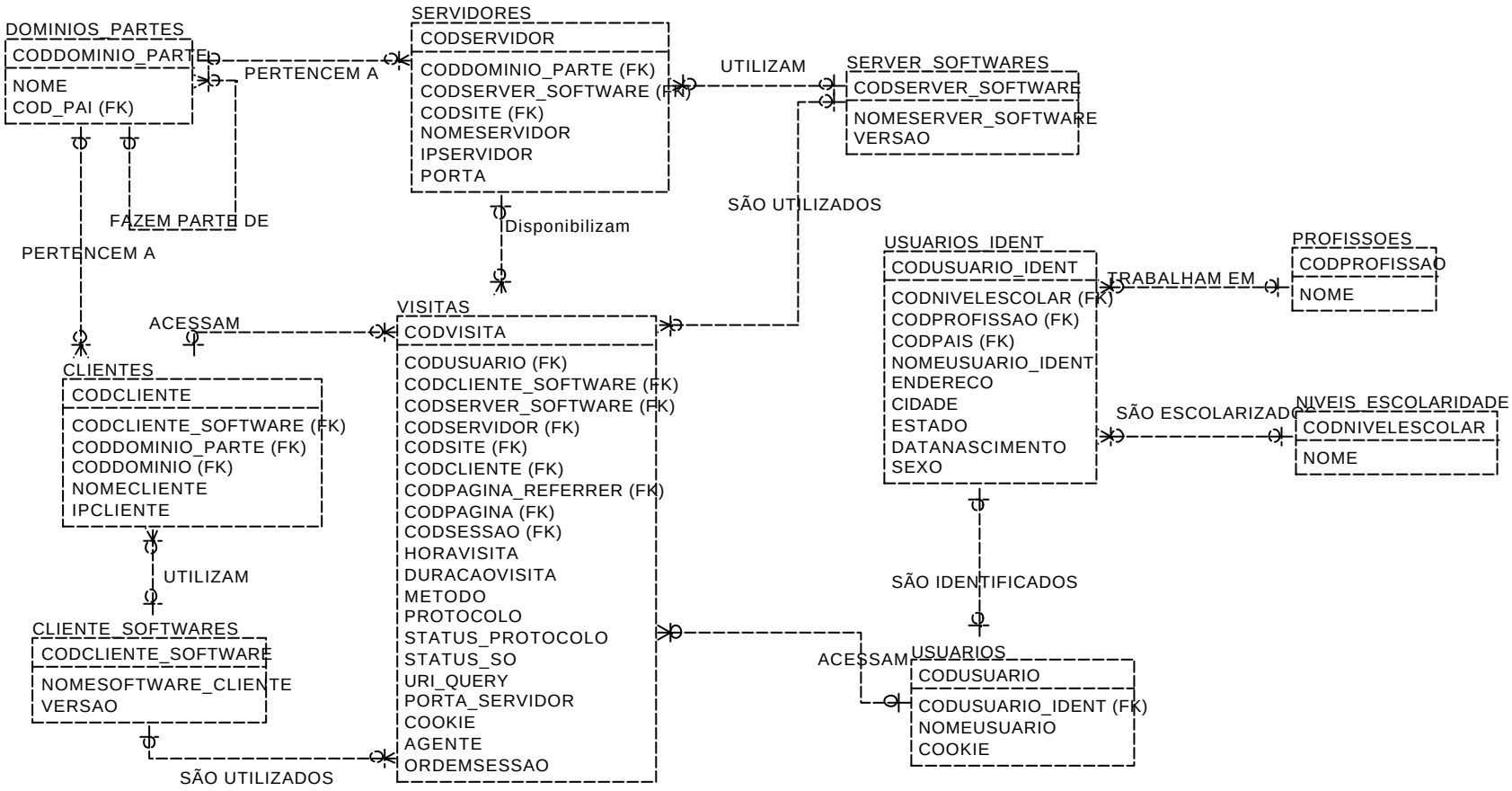
A utilização da linguagem Delphi e de seus componentes proporciona portabilidade na medida que a migração para o sistema Linux, por exemplo, pode ser feita sem maiores problemas com o uso do Kylix.

Quando o usuário, com o seu navegador, conecta-se ao servidor onde está o MineraRedirect, e acessa a DLL, verá uma tela inicial de login e cadastro. Caso não seja cadastrado, preencherá os seus dados e passará a fazer parte da base de dados do MineraWeb. A partir daí, pode navegar pela tela do MineraRedirect, podendo eventualmente selecionar as opções de auxílio que preferir, no frame superior.

Os desenhos tanto da página principal como dos dois frames do MineraRedirect são construídos a partir de *templates* armazenados no servidor Web. Com isso, são facilmente modificáveis para atender às necessidades e preferências do projetista do site.

Para a realização de testes do próprio MineraRedirect, o MineraWebCenter foi dotado também de uma opção que permite a carga e a visualização do código fonte de uma página Web, ao lado da versão do código modificado pelo mesmo algoritmo de formatação usado pelo MineraRedirect.

Figura 6 – Modelo de dados com informações demográficas e dimensões para os domínios



Com o uso desta opção, mais a possibilidade de alteração dos templates, torna-se mais fácil para um desenvolvedor entender e modificar os procedimentos e algoritmos de formatação das páginas redirecionadas utilizados pelo MineraRedirect.

Algumas considerações tiveram que ser feitas na implementação do MineraRedirect, para atender a diversas peculiaridades do protocolo HTTP (FIELDING *et al*., 1997). Ao carregar uma página, o programa deve sempre verificar cuidadosamente quais os códigos de retorno enviados pelos servidores Web. Por exemplo, para a classe de códigos de retorno 300, que indicam que a página foi movida para uma outra localização, o sistema precisa fazer um novo acesso, para não ser enganado e mostrar a página errada ao usuário.

Um outro aspecto foi o tratamento de páginas que também possuam frames. Utilizar frames dentro de frames é algo quase sempre problemático, dada a grande quantidade de variações possíveis na construção de diferentes modelos de páginas desse tipo. Muitos desenvolvedores de sites, por exemplo, ao construírem suas páginas, já prevendo que elas possam ser sub-enquadradas dentro de outras páginas que possuam frames (justamente o que faz o redirecionador), tentam impedi-lo usando scripts que detectem se a página está ou não sofrendo tal tentativa de sub-enquadramento.

Para tentar contornar esses scripts, o MineraRedirect faz um parsing dos códigos de JavaScripts em busca dos padrões mais comuns utilizados para fazer esta verificação, removendo-os quando encontrar. Caso isso não fosse feito, as páginas em questão simplesmente seriam abertas sobrepondo-se aos frames do MineraRedirect, ou seja, seriam criadas como páginas sem frames.

Por sua vez, o próprio MineraRedirect vale-se desta estratégia para forçar a carga das páginas sempre a partir do seu próprio enquadramento. O trecho de código a seguir ilustra como o programa insere um JavaScript no início do corpo da página Web, para forçar o seu carregamento a partir do frame desejado.

```
// Insere script para recarregar o frame principal
if (htmlParser.Tag.Tag = tagBody) Then
begin
  Texto_out.Add('<script type="text/javascript" language="JavaScript"> ');
  Texto_out.Add('<!--                                                  ');
  Texto_out.Add('top.frames[0].location.href="' + URL_PRINCIPAL + RequestURL +'"; ');
  Texto_out.Add('//--></script>                                        ');
end;
```

4.8. Adaptação de páginas

No ambiente MineraWeb, como os dados estão armazenados em um repositório central, torna-se possível, a partir da implementação de páginas dinâmicas que acessem esse banco de dados, construir sites que reajam de maneira automática ou semi-automática a determinados usuários.

O próprio módulo MineraRedirect é um exemplo de como podem ser construídas páginas adaptáveis – nesse caso, com um objetivo bem específico: mostrar ao usuário as páginas mais “interessantes” sob determinados pontos de vista.

Outros métodos, contudo, podem incluir o desenvolvimento de páginas ASP que acessem a base do MineraWeb e aproveitem os dados sobre os padrões de utilização lá armazenados para construir a página segundo certas condições.

Uma outra possibilidade é que se definam, na base, determinadas páginas que possuam eventuais “reservas” ou “substitutas”. Sempre que uma dessas páginas estiver sendo acessada de maneira indesejável (por exemplo, tendo uma freqüência de acesso abaixo de um determinado patamar, a página “titular” pode ser substituída pela reserva).

Enquanto muitos usuários, ao acessarem a Web, sabem exatamente O QUE desejam, ainda que não saibam ONDE, muitos outros, em determinados momentos, vêem-se frente a frente com a pergunta: “O QUE ACESSO AGORA?”. Melhor desligar o computador, desconectar da rede, ou ainda há coisas interessantes para ver hoje? Dado que a quantidade de informações é tão grande, muitas vezes pode-se chegar a tal situação paradoxal: não se saber nem mesmo o que acessar, já que tudo se perde num grande mar de informações indiferenciadas.

Na Internet, os sites que ajudam a resolver a pergunta “ONDE encontrar algo?” são comuns e bastante populares, é enorme a quantidade de engenhos de busca dos mais diferentes tipos, desde aqueles que agrupam os assuntos por categoria, como o YAHOO! (2002), até aqueles que armazenam uma grande quantidade de páginas e fazem buscas por palavras-chaves, dos quais o GOOGLE (2002) é uma das melhores expressões.

Para se responder à pergunta “O QUE ACESSO AGORA?”, há muitos sites que reúnem usuários baseados em suas preferências, mas de maneira bastante específica. Por exemplo, o YAHOO! (2002), oferece o YahooGroups (antigo eGroups, que foi absorvido pelo Yahoo!), que implementa listas de correspondência, reunindo os usuários em grupos afins em torno de um determinado tema, havendo a troca de mensagens eletrônicas direcionadas para aquele tema.

Os grandes portais, geralmente comerciais, procuram reunir, em um só local, informações de diversos tipos para um grande número de usuários, funcionando como uma versão virtual e moderna de uma grande biblioteca / central de notícias. Muitas vezes associados a grandes Provedores de Acesso Internet, os portais tentam fazer com que os usuários de certa forma os “adotem”, da mesma maneira como fazem com o seu jornal diário ou revista semanal.

Para cativar os visitantes, os portais, gratuitos ou não, procuram lançar mão de uma série de recursos: num extremo, estão os pesados investimentos no fornecimento

de informação (tanto através de estrutura própria, como é caso dos portais de grandes grupos de comunicação, quanto pela contratação e terceirização de serviços de terceiros e Provedores de Informação os mais diversos, como a Reuters, por exemplo); no outro extremo, está a personalização dos serviços oferecidos, de modo a tornar não só agradável, como também útil a passagem do usuário pelo site. O Yahoo!, por sua vez, oferece um serviço de personalização, o MyYahoo!, um exemplo muito bem acabado da implementação desse conceito.

Observando-se diferentes usuários, pode-se notar uma tendência a procurar informações de diferentes espécies, em diferentes momentos. Um executivo pode ter o hábito diário de acessar as páginas da bolsa de valores, um advogado talvez não saia de casa sem antes ler a home-page de um jornal diário, um adolescente acessa diariamente a página da sua revista de jogos preferida. No fim-de-semana, entretanto, talvez o executivo prefira consultar a home-page da FIFA, pois é aficcionado em futebol, e talvez o advogado queira acessar a home-page de um grupo de escritores amadores, para ver as últimas poesias enviadas. Por sua vez, o adolescente pode desejar ver a classificação do campeonato mundial de skate que está sendo realizado neste fim-de-semana no Taiti.

Estes diferentes perfis podem ser agrupados, por similaridade, a perfis de outros usuários, formando uma rede de padrões, em que cada usuário pode estar ligado a diversos outros que tenham algo em comum, em termos de preferências pessoais, ao fazer seus acessos Web. Um usuário pode, dessa forma, ser classificado em diversos grupos, de acordo com seus hábitos e desejos, podendo, ainda, esta sua classificação variar no tempo.

Pessoas de um determinado grupo poderiam então querer compartilhar entre si suas páginas mais visitadas, de modo a conseguir informações de outras fontes sobre aqueles assuntos que lhes interessam. Além disso, em determinados momentos,

pode-se querer saber o que as pessoas de um determinado perfil estão acessando com maior freqüência.

Um sistema que disponibilize tal tipo de informação de modo fácil e claro é, certamente, um bom lugar para se ter respondida a questão “O QUE ACESSO AGORA?”. Tal tipo de sistema pode se constituir em um ponto de convergência de usuários ávidos não só por encontrar novas fontes de informações, uma alternativa aos engenhos de busca, como também por encontrar sugestões sobre o que acessar na Web.

O ambiente MineraWeb oferece uma base para a criação de tal sistema, a ser disponibilizado na própria Web, como uma página em que os usuários sejam cadastrados e, a partir daí, tenham as suas visitas à Web registradas e seus perfis atualizados e classificados em grupos de similaridade. A partir daí, os usuários podem fazer consultas à base de dados, seja por grupos, seja por temas, obtendo então listas de páginas de acesso freqüente, em um período temporal qualquer.

O MineraRedirect já oferece a possibilidade de cadastrar os usuários e agrupá-los segundo características comuns, além de fornecer listas de quais páginas são mais acessadas em determinadas condições, a partir da página atualmente visitada pelo usuário.

4.9. Validação

Com vistas à validação da hipótese de que o ambiente proposto adequa-se a pesquisadores que desejem testar métodos de mineração, sem levar em conta as tarefas secundárias associadas à mineração de utilização, foi realizada uma validação do ambiente.

O MineraWeb foi disponibilizado para um pesquisador, com o objetivo deste implementar no ambiente um método de mineração diferente do implementado no toolbox.

Para a validação, foi utilizado um conjunto de dados retirado do “The Internet Traffic Archive” (ITA, 2002), mantido pelo ACM SIGCOMM. Foi selecionado o conjunto EPA-HTTP (BOTTOMLEY, 1995), que contém o registro de todos os acessos realizados no dia 29/08/1995 ao servidor WWW EPA, localizado em Research Triangle Park, EUA (**tabela 5**).

O log contém um total de 47.748 registros, dos quais 46.015 utilizam o método GET, 1.622 usam o método POST, 107 correspondem a métodos HEAD e 6 são entradas com métodos inválidos. O formato do arquivo corresponde aproximadamente ao Common Log Format. Não é informado qual o servidor Web utilizado.

Inicialmente, foi cadastrado na base de dados o site EPA-http, para que as páginas lidas fossem associadas corretamente a ele. Utilizando-se o MineraWebCenter, procedeu-se à leitura e filtragem dos dados. Foram especificadas as seguintes regras de filtragem:

@extensão NOT IN (‘JPG’, ‘BMP’, ‘GIF’, ‘ZIP’)

@uri NOT LIKE ‘%waisgate/bin%’

Pela primeira regra, foram eliminados os indesejáveis arquivos de imagens e arquivos compactados. Pela segunda regra, foram eliminadas as chamadas a um programa CGI, também indesejáveis para o processo. Ao final da leitura, foram incluídas 23.865 entradas de log no banco de dados. O restante foi filtrado pelas regras.

Em seguida, no MineraWebCenter, foram executados os algoritmos de identificação de usuários e sessões. O trabalho envolvido foi, tão somente, disparar os dois procedimentos em seqüência.

Finalmente, foi implementado pelo pesquisador um algoritmo alternativo de descoberta de regras de associação, utilizando-se do modelo de dados subjacente. O algoritmo foi o mesmo implementado no toolbox, porém, ao invés de utilizar uma stored procedure em Transact-SQL, foi programado um módulo em C++ Builder, acessando o mesmo banco de dados.

4.10. Comparação com os trabalhos relacionados

O MineraWeb integra várias das propostas encontradas em outros trabalhos relacionados. A tabela 6 mostra uma comparação na qual são mostradas os principais trabalhos ou produtos, juntamente com o MineraWeb. As colunas descrevem características que podem ou não estar presentes em cada um deles.

A coluna “Arquitetura modular” indica se o trabalho abrange ou não uma plataforma de alcance, modularizável e expansível. O WebMiner foi um dos pioneiros nesta área. A modularização dos componentes e das etapas de mineração de utilização (COOLEY *et al.*, 1999) é uma das principais vantagens do MineraWeb. Além disso, o ambiente permite a incorporação de novas características que venham a surgir futuramente na área.

Na coluna “Linguagem de consulta”, indica-se se o produto possui ou não uma linguagem de consulta para facilitar a busca e análise dos resultados, o que não é o caso do MineraWeb. É o caso, porém, do WUM e sua linguagem, MINT.

A coluna “OLAP, data warehousing” indica se o trabalho propõe o uso de ferramentas OLAP ou estruturas de data warehousing para o armazenamento e análise dos dados. O MineraWeb grava os dados de navegação em uma base de dados relacional, onde poderão ser analisados com técnicas de *data warehousing* e uso de ferramentas OLAP (ZAÏANE *et al.,* 1998, KIMBALL & MERZ, 2000)

A coluna “Modelo estocástico” mostra se o produto utiliza um modelo de navegação baseado em cadeias de Markov e fundamentação estatística. São poucos os trabalhos que o fazem.

A coluna “Sites adaptativos” indica se os trabalhos que são, em alguma medida, voltados para o desenvolvimento de sites que utilizam os padrões minerados para se auto-adaptar. O MineraWeb, como visto, possui todas as características para isso.

Da mesma forma, o MineraWeb possui todas as características necessárias à implementação de suporte à recomendação de páginas, como indicado na coluna seguinte. O MineraRedirect é um agente auxiliar de navegação voltado para esse fim (JOACHIMS *et al.*, 1997, ARMSTRONG *et al.*, 1995).

A última coluna mostra características específicas ou particulares de cada solução. Como visto, o MineraWeb possui, além das características em comum com outros trabalhos, algumas particularidades. Uma das mais interessantes é a possibilidade de carga e filtragem customizadas, com a definição de regras de filtragem de um modo semelhante às regras de integridade SQL. Além disso, a utilização de crawler para obter a estrutura dos sites como sugerido em PIROLLI *et al* (1996).

Tabela 5: Trecho do log do EPA-http

Cliente	Data e hora *	Método	Página lida	Protocolo	Status	Bytes enviados
141.243.1.172	29:23:53:25	GET	/Software.html	HTTP/1.0	200	1497
Query2.lycos.cs.cmu.edu	29:23:53:36	GET	/Consumer.html	HTTP/1.0	200	1325
tanuki.twics.com	29:23:53:53	GET	/News.html	HTTP/1.0	200	1014
Wpbfl2-45.gate.net	29:23:54:15	GET	/	HTTP/1.0	200	4889
Wpbfl2-45.gate.net	29:23:54:16	GET	/icons/circle_logo_small.gif	HTTP/1.0	200	2624
Wpbfl2-45.gate.net	29:23:54:18	GET	/logos/small_gopher.gif	HTTP/1.0	200	935
Wpbfl2-45.gate.net	29:23:57:53	GET	/cgi-bin/waisgate?port=210&ip_address=earth1.epa.gov	HTTP/1.0	200	2431
140.112.68.165	29:23:54:19	GET	/logos/us-flag.gif	HTTP/1.0	200	2788
Wpbfl2-45.gate.net	29:23:54:19	GET	/logos/small_ftp.gif	HTTP/1.0	200	124
Wpbfl2-45.gate.net	29:23:54:19	GET	/icons/book.gif	HTTP/1.0	200	156
Wpbfl2-45.gate.net	29:23:54:19	GET	/logos/us-flag.gif	HTTP/1.0	200	2788
tanuki.twics.com	29:23:54:19	GET	/docs/OSWRCRA/general/hotline	HTTP/1.0	302	-
Wpbfl2-45.gate.net	29:23:54:20	GET	/icons/ok2-0.gif	HTTP/1.0	200	231
tanuki.twics.com	29:23:54:25	GET	/OSWRCRA/general/hotline/	HTTP/1.0	200	991
tanuki.twics.com	29:23:54:37	GET	/docs/OSWRCRA/general/hotline/95report	HTTP/1.0	302	-
Wpbfl2-45.gate.net	29:23:54:37	GET	/docs/browner/adminbio.html	HTTP/1.0	200	4217
tanuki.twics.com	29:23:54:40	GET	/OSWRCRA/general/hotline/95report/	HTTP/1.0	200	1250
wpbfl2-45.gate.net	29:23:55:01	GET	/docs/browner/cbpress.gif	HTTP/1.0	200	51661
dd15-032.compuserve.com	29:23:55:21	GET	/Access/chapter1/s2-4.html	HTTP/1.0	200	4602
tanuki.twics.com	29:23:55:23	GET	/docs/OSWRCRA/general/hotline/95report/05_95mhr.txt.html	HTTP/1.0	200	56431

* O prefixo “29” indica o dia 29/08/1995. O prefixo “30” indica o dia 30/08/1995.

Tabela 6: Comparativo de soluções

	Arquitetura modular	Linguagem de consulta	OLAP, data warehousing	Modelo estocástico	Sites adaptativos	Recomendação de páginas	Características específicas
Amir							Uso de tries
Andersen et al.			X				Killer subsessions eficácia de banners
Anderson et al.					X		Proteus, MinPath
Borges & Levene				X			HPG
FootPrints						X	
Gaul et al.							Subseqüências genéricas
Joshi & Krishnapuram							Agrupamento Fuzzy, FCMdd, FCTMdd
Larsen et al.				X			GGM
MINERAWEB	**X**		**X**		**X**	**X**	**Carga e filtragem customizáveis Stored procedures MineraCrawler MIneraRedirect Modelo de dados**
Nanopoulos & Manolopoulos							Subseqüências genéricas
PageGather					X	X	Clustering
Schechter et al.						X	“Path profiles”
SiteHelper						X	
Tveit							Progol
WebLogMiner	X		X				
WebMiner	X	X					
WebSIFT	X	X					
WUM		X					Interestingness, log agregado, Java

5. Conclusão

A mineração de utilização Web é ferramenta fundamental na construção, planejamento e manutenção de sites Web. Os padrões de navegação descobertos por ela são de grande importância como subsídios para a tomada de decisões ao se projetar um site e, mais tarde, ao se realizar a evolução do mesmo. A mineração de utilização apóia-se, principalmente, na análise dos dados brutos gerados nos logs dos servidores Web. Entretanto, esses logs possuem uma série de limitações, decorrentes, entre outras coisas, do próprio fato de ser o HTTP um protocolo sem estado, não guardando informações sobre as sessões.

Para contornar esses problemas, deve ser feito o pré-processamento dos dados brutos, que permita a limpeza dos registros indesejados e a identificação dos usuários e sessões, com uma eventual agregação dos dados em unidades mais semânticas, como as transações. No entanto, ainda assim, alguns questionam a validade dos logs como uma maneira confiável de se analisar a utilização de um site.

Neste trabalho, foram levantados os principais métodos usados nas diversas etapas da mineração de utilização de dados Web, as formas de se fazer o pré-processamento dos dados e a identificação dos usuários, sessões e transações, assim como as estratégias usadas para a busca e análise de padrões de navegação.

Foi proposto um ambiente genérico para mineração de utilização que procura contornar algumas das limitações identificadas nas diversas ferramentas. Assim, o ambiente aqui proposto tem como características ser aberto, modularizável e expansível. Com isso, é uma plataforma adequada para o desenvolvimento de estudos que testem e implementem novos métodos para cada uma das etapas da mineração de utilização. Além disso, oferece uma base adequada para a criação de páginas e

sites que reajam automaticamente aos padrões de utilização dos seus visitantes. O ambiente propõe um modelo de dados para comportar as várias necessidades que surgem em cada etapa da mineração, podendo o mesmo ser ampliado ou adaptado de acordo com as necessidades do administrador.

Foi implementado um toolbox, formado por protótipos de alguns aspectos do ambiente, notadamente os procedimentos de configuração, pré-processamento e conversão de dados, identificação de usuários, sessões e transações (MineraWebCenter), além de ferramentas auxiliares para o apoio à navegação e coleta de dados de utilização e estrutura Web (MineraDirect e MineraCrawler).

A possibilidade de geração de dados de testes é um outro aspecto interessante para o pesquisador de mineração de utilização, pois pode facilitar o estudo e a comparação dos métodos de mineração. O mesmo pode ser dito da geração de arquivos customizados a partir dos dados armazenados no repositório central.

Foi implementado um algoritmo para a busca de regras de associação, bem como criados ainda alguns protótipos de cubos OLAP que podem ser usados para a visualização de dados minerados, através do Analysis Services. Foi sugerida uma maneira de implementar páginas que, acessando a base de dados, possam reagir e adaptar-se aos usuários que estejam visitando o site.

A utilização por terceiras das ferramentas de carga, configuração e identificação de usuários, sessões e transações, aliada à implementação de um módulo de descoberta de padrões que pôde acessar a base de dados comum do MineraWeb, mostrou que o ambiente é efetivamente adequado como plataforma para testes de novos métodos de mineração de utilização.

Como propostas futuras, ficam em aberto a implementação de outros métodos de mineração e o aperfeiçoamento daqueles já desenvolvidos no toolbox, com a possível ampliação do modelo de dados do MineraData. O aperfeiçoamento constante

das técnicas de pré-processamento também é fundamental. Seria interessante que fossem implementadas também, no próprio código do MineraWebCenter, as rotinas de carga, filtragem e identificação, avaliando-se a diferença de desempenho em relação às rotinas implementadas aqui como stored procedures.

Um outro ponto a ressaltar como proposta futura é a definição detalhada de interfaces de acesso ao ambiente, através do qual o desenvolvedor de ferramentas ou módulos possa acessar de maneira uniforme o banco de dados, tanto quando estiver fazendo a gravação quanto a leitura de informações do MineraData.

Efetivamente, essa foi uma das dificuldades encontradas na validação, já que o módulo de busca de regras de associação implementado diretamente como um aplicativo C++ teve que utilizar seus próprios métodos de acesso aos dados. O caminho inicial mais adequado para atingir este fim é provavelmente a padronização dos métodos internos de acesso utilizados pelo MineraRedirect.

Como assinalado por COOLEY (2000), a pesquisa em mineração de utilização ainda tem muito a ser acrescentado em termos de incremento de vários de seus aspectos, inclusive das técnicas de integração e limpeza dos dados brutos, bem como de identificação de transações.

Para o aperfeiçoamento do processo de descoberta de padrões, podem ser utilizadas e adaptadas as técnicas já existentes de mineração de dados, ou podem ser desenvolvidas novas técnicas específicas. Muitos dos algoritmos precisam ser melhorados tanto em termos de eficiência quanto de eficácia, tanto os algoritmos de identificação de usuários, sessões e transações como também aqueles utilizados no reconhecimento de padrões.

Um outro ponto importante a ser estudado e aprofundado é a própria natureza distribuída dos logs de utilização. Nos trabalhos aqui descritos, bem como no próprio ambiente proposto, é sempre assumido que os dados a serem analisados estão

armazenados no mesmo local. Pode-se, entretanto, partir para o desenvolvimento de modelos e algoritmos que levem em conta esta característica distribuída (SAYAL, 2001).

A ampliação da mineração de utilização para novos domínios, como a área de dispositivos móveis (ANDERSON *et al,* 2001) e a integração cada vez maior de XML à mineração de dados prometem também ser campos bastante férteis para esta área tão promissora.

KOHAVI (2001) salienta que a área de comércio eletrônico (*e-commerce*) é hoje o principal domínio que se apresenta para a mineração de dados de uma forma geral, e, mais especificamente, para a mineração de utilização da Web, chegando a qualificá-lo de "*killer domain*". O comércio eletrônico promete, sim, ser a "fronteira final", mas não definitiva, para a mineração de utilização da Web, da mesma forma que, um dia, as aplicações de mineração de dados e data warehousing foram alçadas aos píncaros da fama ao serem aplicadas nos mais variados ramos do comércio "real" ("*bricks & mortars")*.

6. Referências Bibliográficas

AGRAWAL, R., SRIKANT, S., 1994, “Fast algorithms for mining association rules”, **In: Proceedings of the 20th VLDB Conference**, pp. 487-499, Santiago, Chile, Sep.

ALTAVISTA, 2002, **AltaVista Home-page**, url: http://www.altavista.com.

AMIR, A., FELDMAN, R., KASHI, R., 1997, “A new and versatile method for association generation”, **Information Systems**, v.2, pp. 333-347.

ANALOG, 2002, **Analog Logfile Analysis Home-page**, url: http://www.analog.cx.

ANDERSEN, J., GIVERSEN, A., JENSEN, A.H., et al., 2000, “Analyzing Clickstreams Using Subsessions”, In: **Proceedings of the ACM Third International Workshop on Data Warehousing and OLAP (DOLAP00)**, pp. 25-32, Washington, DC, USA, Nov**.**

ANDERSON, C. R., DOMINGOS, P., WELD, D. S., 2001, "Personalizing Web Sites for Mobile Users", In: **Proceedings of the 10th International WWW** Conference, pp. 565-575, Hong Kong, China, May.

ARMSTRONG, R., FREITAG, D., JOACHIMS, T. et al., 1995, “Webwatcher: A learning apprentice for the World Wide Web”, In: **Papers of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous**, **Distributed Environments**, pp. 6-12.

BERRY, A., LINOFF, G., 1997, **Data Mining Techniques: For Marketing, Sales, and Customer Support**, 1 ed., New York, USA, Wiley Computer Publishing.

BORGES, J., LEVENE, M., 1998, “Mining Association Rules in Hypertext Databases”, In: **Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD98)**, pp. 149-153, Menlo Park, CA, USA, Aug.

BORGES, J., LEVENE, M., 1999, “Data mining of users navigation patterns, In: **LNCS - Lecture Notes in Computer Science, Web Usage Analysis and User Profiling**, v. 1836, pp. 92-111, Springer-Verlag.

BORGES, J., LEVENE, M., 2000, “A Heuristic to Capture Longer User Web Navigation Patterns”, In: **Proceedings of the 1st International Conference on Electronic Commerce and Web Technologies**, pp. 155-164, Greenwich, UK, Sep.

BOTTOMLEY, L, 1995, **EPA-HTTP dataset,** url: http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html.

BRITANNICA, 2002, **Enciclopédia Britannica On-line**, url: http://www.britannica.com. 2002.

BUNEMAN, P., DAVIDSON, S., HILLEBRAND, G., et al., 1996, "A query language and optimization techniques for unstructured data", In: **Proceedings of ACM-SIGMOD International Conference on Management of Data**, pp. 505-516, Montreal, Canada, June.

CATLEDGE, L., PITKOW, J., 1995, “Characterizing Browsing Strategies on the World Wide Web”, In: **Proceedings of the 3rd International WWW Conference**, Darmstad, Germany, Apr.

CHEN, M., PARK, J. S., YU, P. S., 1996, “Data Mining for Path Traversal Patterns in a Web Environment”, In: **Proceedings of the 16th Conference on Distributed Computing Systems**, pp. 385-392, Baltimore, Maryland, USA, May.

CHEN, M., PARK, J.S., YU, P.S., 1998, “Efficient data mining for path traversal patterns”, **IEEE Transactions on Knowledge and Data Engineering**, v.10, n. 2 (Mar), pp. 209-221.

CHERKASSKY, V., MULIER, F., 1998, **Learning From Data – Concepts, Theory and Methods,** 1 ed., New York, USA, John Wiley & Sons, Inc.

CODD, E. F., 1970, A Relational Model of Data for Large Shared Data Banks, **Communications of the ACM**, v. 13, n. 6 (Jun), pp. 377-387, url: http://www.acm.org/classics/nov95/toc.html.

CODD, E. F., CODD, S. B., SALLEY, C. T., 1993, **Providing OLAP to User-analysts: an IT Mandate**, white paper, E.F.Codd Associates, url: http://www.hyperion.com/solutions/whitepapers.cfm.

COOLEY, R., 2000, **Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data**, Ph.D. thesis, University of Minnesota, Minneapolis, USA.

COOLEY, R. MOBASHER, B., SRIVASTAVA, J., 1997, “Web Mining: Information and Pattern Discovery on the World Wide Web” (1997), In: **Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI97)**, Newport Beach, CA, USA, Nov.

COOLEY, R., MOBASHER, B., SRIVASTAVA, J., 1997a, “Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns”, In: **Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97),** Newport Beach, CA, USA, Nov.

COOLEY, R. MOBASHER, B., SRIVASTAVA, J., 1999, "Data Preparation for Mining World Wide Web Browsing Patterns", **Knowledge and Information Systems** v.1, n.1 (Jan), pp. 5-32.

COVE, J. F., WALSH, B. C., 1988, "Online text retrieval via browsing", **Information Processing and Management**, v.24, n.1, pp. 31-37.

CUSTOMERCENTRICS, 2002, **CustomerCentrics Home-page,** url: http://www.netgen.com.

DEOGUN, J. S. , RAGHAVAN, V. V., SARKAR, A., et al., 1997, "Data mining: Research trends, challenges, and applications", In: Lin, T. Y., Cercone, N. (eds), **Roughs Sets and Data Mining: Analysis of Imprecise Data**, pp. 9-45, Boston, MA, USA, Kluwer Academic Publishers.

DIX, A. MANCINI, R., 1997, "Specifying history and backtracking mechanisms", In: Palanque, P., Paterno, F. (eds), **Formal Methods in Human Computer Interaction**, 1 ed., Londres, UK, Springer-Verlag.

DOORENBOS, R. B., ETZIONI, O., WELD. D. S., 1997, "A scalable comparison shopping agent for the World Wide Web", In: **Proceedings of the 1st International Conference on Autonomous Agents (AGENTS'97)**, pp. 39-48, New York, NY, USA.

DYRESON, C., 1997, "Using an Incomplete Data Cube as a Summary Data Sieve", **Bulletin of the IEEE Technical Committee on Data Engineering**, Mar, pp. 19-26.

FAYYAD, U., M., PIATETSKY-SHAPIRO, G., SMYTH, P., 1996, "From data mining to knowledge discovery: An overview", In: Fayyad, U. M.,, Piatetsky-Shapiro, G.,

Smyth, P., Uthurusamy, R. (eds), **Advances in Knowledge Discovery and Data Mining**, pp. 1-36, Menlo Park, CA, USA, AAAI Press.

FIELDING, R., IRVINE, U. C., GETTYS, J. et al., 1997, **Hypertext Transfer Protocol - HTTP/1.1**, Request for Comments 2616, url: http://www.w3.org/Protocols/History.html

FOSS, A., WANG, W., ZAÏANE, O.R., 2001, "A Non-Parametric Approach to Web Log Analysis", In: **Proceedings of Workshop on Web Mining in First International SIAM Conference on Data Mining (SDM2001)**, pp. 41-50, Chicago, IL, USA, Apr.

FULLER, R., DE GRAAFF, J. J., 1996, 'Measuring User Motivation from Server Log File", Microsoft Usability Group, url: http://www.microsoft.com/usability/webconf/fuller/fuller.htm.

GAUL, W., SCHMIDT-THIEME, L., 2000, "Mining web navigation path fragments", **Worskhop on Web Mining for E-Commerce - Challenges and Opportunities, in WEBKDD'2000**, Boston, MA, USA, Aug, url: http://robotics.stanford.edu/~ronnyk/WEBKDD2000/papers.

GOOGLE, 2002, **Google Home-page**, url: http://www.google.com.

HALLAM-BAKER, P.M., BEHLENDORF, B., 1996, **Extended Log File Format,** W3C Working Draft WD-session-id-960323, url: http://www.w3.org/pub/WWW/TR/WD-logfile.html.

HALLAM-BAKER, P.M., CONNOLLY, D., 1996a, **Session Identification URI**, W3C Working Draft WD-session-id-960221**,** url: http://www.w3.org/pub/WWW/TR/WD-session-id.html

HAN, J., CAI, Y., CERCONE, N., 1993, "Data-driven Discovery of Quantitative Rules in Relational Databases", **IEEE Transactions on Knowledge and Data Engineering**, v. 5, pp. 29-40.

HUBERMAN, B., PIROLLI, P., PITKOW, J., LUKOSE, R., 1998, "Strong regularities in World Wide Web surfing", **Science**, v. 280, n. 5360 (Apr), pp. 95-97.

INMON, W. H., 1997, **Como Construir o Data Warehouse**, 2 ed., Editora Campus.

ITA, 2002, **Internet Traffic Archive,** url: http://ita.ee.lbl.gov.

JOACHIMS, T., FREITAG, D., MITCHELL, T., 1997. "Webwatcher: A tour guide for the world wide web", In: **Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI97)**, pp. 770-775, Nagoya, Japan, Aug.

JOSHI, A., KRISHNAPURAM, R., 2000, "On Mining Web Access Logs", In: **Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'2000)**, pp. 63-69, Dallas, TX, USA, May.

KAMDAR, T., JOSHI, A, 2000, **On Creating Adaptive Web Servers Using Weblog Mining**, In: Technical Report CS-TR-00-05, CSEE Department, University of Maryland, Baltimore, USA, MD.

KATO, H., NAKAYAMA, T., YAMANE, Y., 2000, "Navigation analysis tool based on the correlation between contents distribution and access patterns", In: **Worskhop on Web Mining for E-Commerce - Challenges and Opportunities, in WEBKDD'2000**, Boston, MA, USA, Aug, url: http://robotics.stanford.edu/~ronnyk/WEBKDD2000/papers.

KIMBALL, R.,1996, **The Data Warehouse Toolkit,** 1 ed., New York, USA, John Wiley & Sons.

KIMBALL, R., 1997, A Dimensional Model Manifesto, **DBMS Magazine**, v. 10, n. 8 (Aug), url: http://www.dbmsmag.com/9708d15.html.

KIMBALL, R., MERZ, R., 2000, **The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse,** 1 ed., New York, USA, John Wiley & Sons.

KOBAYASHI, M., TAKEDA, K.M, 2000, "Information retrieval on the Web", ACM Computing Surveys, v. 32, n. 2 (Jun), pp. 144-173.

KOHAVI, R., 2001, "Mining E-Commerce Data: The Good, the Bad, and the Ugly", In: **Proceedings of 7th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD01),** pp. 8-13, San Francisco, CA, Aug.

KOSALA, R., BLOCKHEEL, H., 2000, "Web Mining Research: A Survey", SIGKDD Explorations, v. 2, n. 1 (Jul), pp. 1-15.

LARSEN, J., HANSEN, L. K., SZYMKOWIAK, A., et al., 2000, "Webmining: Learning from the World Wide Web", In: Proceedings of Nonlinear Methods and Data Mining Conference, pp. 106-125, Rome, Italy, Sep.

LEVENE, M., BORGES, J., 2001, "Zipf's Law for Web Surfers", **Knowledge and Information Systems: an International Journal**, v. 3, n. 1 (Feb), pp. 120-129.

LEVENE, M., LOIZOU, G, 1999, **Computing the entropy of user navigation in the web**, In: Research Note RN/99/42, Department of Computer Science, University College London, London, UK.

LEVENE, M., LOIZOU, G., 2001, "Web Interaction and the Navigation Problem", **Encyclopedia of Microcomputers**, v. 28, n. 7, url: http://www.navigationzone.com/library.

LI, T., 2001, **Web-Document Prediction And Presending Using Association Rule Sequential Classifiers**, M.Sc. dissertation, School of Computing Science, Simon Fraser University, Vancouver, BC, Canada.

LIEBERMAN, H., 1995, "Letizia: An agent that assists Web browsing", In: **Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI95)**, pp. 924-929, Montreal, Canada, Aug.

LUOTONEN, A., NIELSEN, H. F., BERNERS-LEE, T., 1995, **The Common Logfile Format**. http://www.w3.org/Daemon/User/Config/Logging.html, 1995.

MAEDCHE, A., STAAB, S., STOJANOVIC, N., et al., 2001, "SEmantic PortAL - The SEAL approach", In: Fensel, D. , Hendler, J., Lieberman, H., Wahlster, W. (eds.), **Creating the Semantic Web,** MIT Press, Cambridge, MA, USA.

MANNILA, H., 1997, "Methods and problems in data mining", In: **Proceedings of the 6th International Conference on Database Theory (ICDT'97)**, LNCS - Lecture Notes in Computer Science, v. 1186, Springer-Verlag, pp. 41-55, Delphi, Greece, Jan.

MANNILA, H., TOIVONEN, H., 1996, "Discovering Generalized Episodes Using Minimal Ocurrences", In: **Proceedings of 2nd ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD96)**, pp. 146-151, Portland, OR, USA, Aug.

MANNILA, H., TOIVONEN, H., VERKAMO, A.I., 1995, "Discovering frequent episodes in sequences", In: **Proceedings of the 1st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD95)**, pp. 210-215, Montreal, Aug.

MENDELZON, A., MIHAILA, G., MILO, T., 1996, "Querying the world wide web", In: **Proceedings of the 4th Conference on Parallel and Distributed Information Systems**, Miami, Florida, USA, Dec.

MITCHELL, T. M., 1997, **Machine Learning**, 1 ed., Boston, MA, USA, McGraw-Hill.

MLADENIC, D., 1999, "Text learning and related intelligent agents: a survey", **IEEE Intelligent Systems,** v. 14, n. 4 (Jul), pp. 44-54.

MOBASHER, B., JAIN, N., HAN, E., SRIVASTAVA, J., 1996, **Web Mining: Pattern Discovery from World Wide Web Transactions**, Technical Report TR-96050, Dep. of Computer Science, University of Minnesota, Minneapolis, USA.

MOBASHER, B., DAI, H, LUO, T., et al., 2000, "Combining web usage and content mining for more effective personalization", In: **Proceedings of the International Conference on E-Commerce and Web Technologies (ECWeb),** LNCS - Lecture Notes in Computer Science, v. 1875, Springer-Verlag, pp. 165-176, London, Sep.

NANOPOULOS, A., MANOLOPOULOS, Y., 2000, "Finding Generalized Path Patterns for Web Log Data Mining", In: **Proceedings of the East-European Conference on Advanced Databases and Information Systems (ADBIS'00)**, pp. 215-228, Prague, Czech Republic, Sep.

NANOPOULOS, A., MANOLOPOULOS, Y., 2001, "Mining Patterns from Graph Traversals", **Data and Knowledge Engineering**, v. 37, n.3 (Apr), pp. 243-266.

NETSCAPE, 1999, **Persistent client State HTTP Cookies Preliminary Specification**, white paper, Netscape Communications Corp., url: http://home.netscape.com/newsref/std/cookie_spec.html.

NGU D. S. W., WU, X, 1997, "SiteHelper: A Localized Agent that Helps Incremental Exploration of the World Wide Web", **Computer Networks and ISDN Systems,** v. 29, n. 8, pp. 1249-1255.

NIELSEN, J., 1990, **Hypertext and Hypermedia,** 1 ed., Boston, MA, USA, Academic Press.

PERKOWITZ, M., ETZIONI, O., 1997, "Adaptive web sites: an AI challenge", In: **Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI97),** pp. 16-23, Nagoya, Japan, Aug.

PERKOWITZ, M., ETZIONI, O., 1998, "Adaptive web sites: Automatically synthesizing web pages", In: **Proceedings of the 15th National Conference on Artificial Intelligence**, pp. 727-732, Madison, WI, USA, Jul.

PERKOWITZ, M., ETZIONI, O., 1999, "Adaptive web sites: Conceptual cluster mining", In: **Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI99)**, pp. 264-269, Stockholm, Sweden, Aug.

PIATESTKY-SHAPIRO, G., 2000, "Knowledge Discovery in Databases: 10 years after", **SIGKDD Explorations**, v. 1, n. 2 (Jan), pp. 59-61.

PIATESKI-SHAPIRO, G., MATHEUS, C.J., 1994, "The Interestingness of Deviations", In: **AAAI94 Workshop on Knowledge Discovery in Databases**, pp. 25-36.

PIROLLI, P., PITKOW, J., RAO, R., 1996, "Silk from a Sow's Ear: Extracting Usable Structures from the Web", In: **Proceedings of the ACM SIGCHI'96 Conference on Human Factors in Computing Systems,** pp. 118-125, Vancouver, BC, Canada, Apr.

PITKOW J., 1997, "In Search of Reliable Usage Data on the WWW", **In: Proceedings of the 6th International WWW Conference**, pp. 451-463, Santa Clara, CA, USA.

PITKOW, J., BHARAT, K., 1994, “Webviz: a Tool for World-Wide-Web Access Log Analysis”, **In: Proceedings of the 1st International WWW Conference,** Geneve, Switzerland, May.

PUNIN, J., KRISHNAMOORTHY, 2001, **XGMML 1.0 Draft Specification (eXtensible Graph Markup and Modeling Language)**, url: http://www.cs.rpi.edu/~puninj/XGMML/draft-xgmml.html

PUNIN, J., KRISHNAMOORTHY, M., ZAKI, M. J., 2001, "LOGML - Log Markup Language for Web Usage Mining", **WebKDD'2001 Workshop in conjunction with the ACM-SIGKDD 2001**, San Francisco, CA, Aug

QUINLAN, J. R., 1986, Induction of decision trees, **Machine Learning**, v. 1, n. 1., pp. 81-106.

QUINLAN, J. R., 1993, **C4.5: Programs for machine learning,** San Mateo, CA, USA, Morgan Kaufmann.

SAVASERE, A., OMIECINSKI, E., NAVATHE, S., 1995, “An efficient algorithm for mining association rules in large databases”, **In: Proceedings of the 21th VLDB Conference**, pp. 432-443, Zurich, Switzerland, Sep.

SANE 2002, **Analyzing web site traffic**, Sane Solutions white paper, url: http://www.sane.com/products/NetTracker/whitepapers.html

SAYAL, M., SCHEUERMANN, P., 2001, "Distributed Web Log Mining Using Maximal Large Itemsets", **Knowledge and Information Systems** v. 3, n. 4, pp. 389-404

SCHECHTER, S., KRISHNAN, M., SMITH, M. D., 1998, "Using path profiles to predict HTTP requests", **Computer Networks and ISDN Systems**, v. 30, n. 1, pp. 457-467.

SHAHABI, C., ZARKESH, A. M., ADIBI, J., SHAH, V., 1997, "Knowledge discovery from users web page navigation", In: **Proceedings of the International Workshop on Research Issues in Data Engineering IEEE (RIDE'97)**, pp. 20-31, Birmingham, UK, Apr.

SHAHABI, C., BANAEI-KASHANI, F., FARUQUE, J., 2001, A Reliable, Efficient, and Scalable System for Web Usage Data Acquisition, In: **Proceedings of the WebKDD'2001 Workshop in conjunction with the ACM-SIGKDD 2001**, San Francisco, CA, Aug.

SPERTUS, E., 1998, **ParaSite: Mining the Structural Information on the World-Wide Web,** Ph.D. thesis, Department of EECS, MIT, Cambridge, MA, USA.

SPILIOPOULOU, M., 1999, "The laborious way from data mining to web mining", **International Journal of Comp. Sys., Sci. & Eng., Special Issue on "Semantics of the Web"**, v. 14 (Mar), pp. 113-126.

SPILIOPOULOU, M., 2000, "Web usage mining for Web site evaluation", **Communications of the ACM**, v. 43 , n. 8 (Aug), pp. 127-134.

SPILIOPOULOU, M., FAULSTICH, L. C., 1998, "WUM: A Web Utilization Miner", In: **Proceedings of the EDBT Workshop, WebDB98, LNCS - Lecture Notes in Computer Science**, v. 1590, Springer-Verlag, Valencia, Spain.

SPILIOPOULOU, M., FAULSTICH, L.C., WINKLER, K., 1999, "A Data Miner analyzing the Navigational Behaviour of Web Users", In: **Proceedings of Workshop on Machine Learning in User Modeling of the ACAI'99***,* Creta, Grécia, Jul.

SRIKANT, R., AGRAWAL, R., 1996, “Mining sequential patterns: generalizations and performance improvements”, In: **Proceedings of the 5th International Conference on Extending Database Technology (EDBT’96)** , pp. 3-17, Avignon, France, Mar.

SU, Z., YANG, Q., ZHANG, H., et al., 2001, “Correlation-based Document Clustering using Web Logs”, In: **Proceedings of the 34nd IEEE Hawai’i International Conference on System Sciences (HICSS-34)**, Hawaii, USA, Jan.

TAUSCHER, L., GREENBERG, S., 1997, “Revisitation Patterns in World Wide Web Navigation”, In: **Proceedings of the ACM SIGCHI'97 Conference on Human Factors in Computing Systems**, pp. 399-406, Atlanta, Georgia, USA, Mar.

TVEIT, A., 2000, **Web Usage Mining with Inductive Logic Programming,** url: http://citeseer.nj.nec.com/tveit00web.html.

W3C, 1999, **W3C Web Characterization Activity**, url: http://www.w3.org/WCA.

W3C, 2001, **URIs, URLs, and URNs: Clarifications and Recommendations 1.0,** Report from the joint W3C/IETF URI Planning Interest Group, W3C Note 21 September 2001, url: http://www.w3.org/TR/uri-clarification.

W3C, 2002, **Resource Description Framework (RDF)**, url: http://www.w3.org/RDF.

WANG, Y., 2000, “Web Mining and Knowledge Discovery of Usage Patterns”, In: **Proceedings of the 1st International Web-age Information Management Conference (WAIM'2000)**, pp. 227-232, Shanghai, China, Jun.

WEBTRENDS, 2002, **WebTrends Home-page**, url: http://www.webtrends.com

WEXELBLAT, A., MAES, P., 1999, “Footprints: History-Rich Tools for Information Foraging”, In: **Proceedings of the ACM SIGCHI'99 Conference on Human Factors in Computing Systems,** pp. 270-277, Pittsburgh, PA, USA, May.

WHITING, R., 2000, “Mind Your Business: Companies rethink their privacy policies as public concern grows”, **InformationWeek,** n. 776, pp. 22, url: http://www.informationweek.com.

YAHOO!, 2002, **Yahoo! Home-page**, url: http://www.yahoo.com.

YAN, T. W., JACOBSEN, M., GARCIA-MOLINA, H. et al., 1996, “From User Access Patterns to Dynamic Hypertext Linking”, **In: Proceedings of the 5th International WWW Conference**, pp. 1007-1014, Paris, France, May.

ZAÏANE, O., R., 1999, **Resource and Knowledge Discovery from the Internet and Multimedia Repositories**, Ph.D. thesis, School of Computing Science, Simon Fraser University, Vancouver, BC, Canada. url: http://www.cs.aue.auc.dk/datamining/papers/osmarzaianephd.pdf

ZAÏANE, O.R., 2000, “Web Mining: Concepts, Practices and Research”, **Conference Tutorial Notes, XIV Brazilian Symposium on Databases (SBBD'2000)**, pp. 410-474, João Pessoa, Paraíba, Brasil, Oct.

ZAÏANE, O. R., 2001, **Web Usage Mining for a Better Web-Based Learning Environment**, Technical Report TR01-05, Department of Computing Science, University of Alberta, Edmonton, Canada.

ZAÏANE, O. R., XIN, M., HAN, J., 1998, “Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs”, In: **Proceedings of Advances in Digital Libraries Conference (ADL'98)**, pp. 19-29, Santa Barbara, CA, USA, Apr.

ZAKI, M. J., LESH, N, OGIHARA, M, 1998, “PLANMINE: Sequence Mining for Plan Failures”, In: **Proceedings of 4th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD98)**, pp. 369-374, New York, USA, Aug.