Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Programa de Pós-Graduação em Ciência da Computação

Web2DB - Uma Ferramenta para Construção de Representações Relacionais de Sítios da Web

Marcelo Dias Correa

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do grau de Mestre em Ciência da Computação.

Orientador: Alberto Henrique Frade Laender

Co-orientador: Paulo B. Góes

Belo Horizonte, março de 2008

"Life is what happens to you while you're busy making other plans."
(John Lennon 1940-1980)

Agradecimentos

É com muita satisfação que concluo esta dissertação de mestrado. E, é claro, chegar até aqui não foi nada fácil. Foram mais de dois anos de muito esforço para viabilizar o projeto. Definitivamente não teria chegado até aqui sem o apoio de diversas pessoas que não permitiram que eu me abatesse nos momentos mais difíceis deste período.

Primeiramente, gostaria de agredecer a Deus por me dar força para vencer mais essa etapa na minha vida. Ele foi a luz para o caminho que decidi trilhar. Agradeço aos meus pais que me deram formação necessária para vida e sempre priorizaram a boa educação dos filhos. Hoje temos mais uma prova do quanto isso valeu a pena. Agradeço também aos meus irmãos que souberam compreender meus momentos de necessária concentração durante esse tempo.

Agradeço à Priscila, que nesse período foi de namorada à noiva, pelo apoio e carinho, me confortando nos momentos de maior angústia e me incentivando incondicionalmente.

Faço um agradecimento especial aos colegas e amigos da ATAN que souberam compreender a minha necessidade de estar inserido em dois contextos bem diferentes e me deram a flexibilidade necessária para a execução das atividades do mestrado. Trabalhar e estudar é um desafio, mas o apoio deles me ajudou muito.

Por fim agradeço ao Prof. Alberto Laender que para mim é um exemplo de orientador. Fico muito feliz de ter tido a oportunidade de aprender com ele (e quanto). Ele é uma pessoa extremamente competente na sua função, realçando uma seriedade e dedicação inconfundúveis na atividade de formar os seus alunos. Seu trabalho me motivava a prosseguir e continuar aprendendo. Meus sinceros agradecimentos pelo apoio e pela paciência (principalmente nos momentos de ausência...).

Resumo

A crescente demanda por informação de qualidade, para análise e tomada de decisão, favorece o crescimento de ferramentas e métodos de automação do processo de extração e tratamento de dados da Web. O advento da Web trouxe consigo uma infindável quantidade de documentos e dados que se encontram difusos na Web. A centralização desses dados é de suma importância, pois reduz esforços na obtenção de dados de grandes repositórios, permitindo que esses esforços sejam dispendidos na análise e tomada de decisão, ou seja, retirar informação dos dados. Em muitos casos o interesse reside em uma forma efetiva de buscar informação ao invés de navegar por páginas da Web procurando dados de interesse, que muitas vezes não estão estruturados da melhor forma.

A motivação para este trabalho surgiu da necessidade de se criar um processo que permita a coleta de páginas contendo dados de interesse e efetue a extração desses dados a partir de uma representação relacional previamente criada pelo usuário. O banco de dados relacional gerado como resultado desse processo permite que dados contidos na Web possam ser analisados e manipulados de acordo com as necessidades de uma determinada aplicação.

Neste contexto foi desenvolvida a Web2DB, uma ferramenta que, a partir da modelagem de um sítio eletrônico da Web, permite o planejamento e execução da coleta das páginas e posteriormente a extração dos dados, armazenando-os em um banco de dados relacional. O usuário configura os tipos de página a serem coletados, os dados de interesse para a extração e a forma de carregamento dos dados no banco de dados. A ferramenta permite ainda a geração de visões para que os dados extraídos das páginas possam ser visualizados de forma mais aderente às necessidades dos usuários da ferramenta.

É utilizada uma estratégia de extração dos dados baseada em exemplos. O foco na participação do usuário, nas fases de mapeamento do processo como um todo, visa agregar valor com o conhecimento do negócio envolvido. O restante das atividades é feita de forma automática. Trata-se de uma nova abordagem prática para o problema de extração de dados da Web, quando o objetivo é a análise de uma grande massa de dados difusa em vários sítios eletrônicos na Web. A ferramenta permite a construção de representações relacionais de grandes sítios da Web e, por ser genérica, pode ser aplicada a qualquer sítio eletrônico que contemple os requsitos da ferramenta.

Abstract

The increasing demand for valuable information to be used in the analysis and decision-making processes favors the development of tools and methods that automate the extraction and treatment of web data. The rise in Web's popularity has given place for an enormous quantity of documents widely spread over the Web. The centralization of the data is important because it reduces the efforts on retrieving the useful information from the vast repositories, allowing the efforts to focus more on the analysis and decision-making processes rather than lower-level data-handling techniques. In many cases the interest resides in an effective way to search for information rather than visiting unstructured web pages hoping to find the right data.

The motivation for this work started from the need to create a process that would permit the collection of web pages containing the desired user data and the extraction of the data based on a relational representation previously configured. The resulting relational database could be analyzed and manipulated according to the needs of many applications.

In this context it was designed Web2DB, a tool that, giving a model for a web site, permits the configuration and execution of page data collection and then the extraction of the data to a database. The user can customize the types of pages to be collected, the extraction interest data and the way which the database will be populated. The tool also permits the generation of views so the extracted data can the visualized in the most convenient way.

The tool uses the example-based data extraction strategy. The user participation in the process-mapping phase is intended to aggregate value from the business model into the process. The following activities after the mapping phase are done automatically by the tool. This is a practical approach to the data extraction problem aiming the analysis of a vast diffuse data spread on web sites. The tool is suitable for relational representations of big web sites and, for being customizable, can be applied in most electronic sites that meet a list of requisites for the extraction.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contexto

Ao longo da história ocorreram diversas revoluções que modificaram e ampliaram os meios de comunicação como, por exemplo, o surgimento da máquina de impressão no século XVI, do telégrafo no século XIX e, no último século, do telefone, do rádio e da televisão. Mais recentemente esse processo culminou com o advento da Internet, que hoje eliminou as distâncias geográficas e colocou o mundo inteiro interconectado.

A Internet desencadeou uma valorização e uma dependência cada vez maior do conceito de informação. É importante destacar que o conceito de informação é diferente do conceito de dado. No caso deste último, trata-se apenas de um registro, sem valor ou significado para a compreensão humana. O grande desafio está em agregar valor ao dado e transformá-lo em informação. Com o advento da Internet, essa questão, que duas décadas atrás era emergente, hoje é cada vez mais importante no dia-a-dia. O uso da informação é hoje ponto chave na tomada de decisões e devido a isso é de suma importância garantir que o dado apresentado em páginas da Web possa ser usado como informação de forma confiável.

Desde o início da década de 90 até os dias atuais, o número de páginas dispostas na Web cresceu de forma muito rápida e isso levou naturalmente ao advento das máquinas de busca. Atualmente, as máquinas de buscas mais populares (como Google e Yahoo!) têm indexados em suas bases algo da ordem de dezenas de bilhões de páginas, o que faz com que a Web se torne o maior repositório público de informação que se tem notícia. No entanto, mesmo com o auxílio das máquinas de busca, o grande volume de informação existente trouxe algumas conseqüências, como, por exemplo, a dificuldade em centralizar

informação que está espalhada em diversas páginas, mesmo que seja sobre um mesmo domínio. Como o volume de páginas é muito grande, investimentos têm sido feitos no sentido de facilitar para o usuário a separação das informações úteis das não úteis aos objetivos de seu interesse. De fato, hoje em dia o uso de uma máquina de busca pode retornar uma série de páginas (e outros documentos) e um vasto volume de dados de difícil manipulação. O que o usuário quer na realidade é um acesso fácil a essa informação.

Outra questão emergente nesse contexto é a disposição e apresentação dessa informação. Os documentos são predominantemente descritos em HTML (*Hypertext Markup Language*), linguagem que foca na apresentação e não no conteúdo (Abiteboul *et al.*, 2000). Essa linguagem confere um grau de liberdade grande em relação à forma de apresentação das informações nas páginas. Assim, fica cada vez mais difícil projetar um programa que, de forma automática, consiga percorrer um conjunto de páginas, extrair seu conteúdo e tratar isso de forma centralizada, facilitando para o usuário a busca não somente por páginas, mas por uma informação específica de seu interesse, mas que estava difusa em diversos documentos.

Além do custo de implementação de programas com esse objetivo, um ponto importante a se destacar diz respeito aos esforços de manutenção e de execução desses programas. Primeiramente, o processo deve estar mapeado em um programa que permita a sua manutenção de forma fácil, pois a evolução das tecnologias é muito rápida e alterações nos sítios eletrônicos são freqüentes (como se depende da estrutura de apresentação das páginas, alterações ocorridas têm de ser refletidas nos processos de extração dos dados dessas páginas). A execução muitas vezes exige um esforço grande e algumas soluções existentes tratam muitas etapas do processo de forma manual. A automatização, além de reduzir esforços, reduz o índice de erros que são provocados pela intervenção humana no processo. Esses programas devem ser capazes de lidar com um ambiente que é cada vez mais dinâmico e com um volume de documentos cada vez maior.

Um fator que torna a extração mais difícil é a estrutura HTML que as páginas possuem, que podem apresentar pequenas variações, mesmo estando em um mesmo domínio, devido a particularidades de cada documento. Uma extração de dados eficiente deve ser capaz de perceber uma estrutura padrão do documento e mesmo assim ser possível, da forma mais automática possível, extrair os atributos de interesse.

É de suma importância orientar o usuário e definir os dados de interesse e converter

esses dados, previamente apresentados nas páginas da Web, em uma forma mais estruturada e centralizada, como, por exemplo, um documento XML ou um banco de dados.

O trabalho desenvolvido nesta dissertação se insere nesse contexto da crescente demanda por informação de qualidade, centralizada para realização de análises e tomada de decisão. Muitos trabalhos vêm sendo realizados no sentido de desenvolver técnicas e ferramentas para extração de dados da Web (Laender *et al.*, 2002a). Eles focam na automatização do processo e garantia de qualidade dos atributos extraídos dos documentos.

1.2 Motivação e Justificativa

Há uma grande variedade de aplicações que pode se beneficiar do processo de extração de dados de sítios eletrônicos. Como exemplo de aplicação dessas ferramentas de extração de dados da Web, podemos citar o comércio eletrônico, que hoje aflora pela Internet, com um volume de negócios cada vez maior. Um fator motivacional para este trabalho surge nesse contexto, onde a automação da extração de dados de leilões eletrônicos, por exemplo, facilitaria a busca por produtos, melhores preços e promoções e o estudo do comportamento de vendedores e compradores, enfim, boas oportunidades de negócio. Outra aplicação é a centralização em bibliotecas digitais das informações bibliográficas de artigos eletrônicos, difusos entre os mais diversos sítios de conferências, instituições acadêmicas, editoras, etc.

Ao facilitar a análise dos dados contidos em sítios de comércio eletrônico, como leilões, por exemplo, podemos impulsionar esse negócio, pois os usuários dessas ferramentas terão informações mais consolidadas para a suas análises. Dessa forma, esses usuários podem, centralizando essas informações, avaliar o comportamento dos leilões e com isso até propor melhorias no sítio (como, por exemplo, disposição das informações e configuração das páginas) ou nas etapas do processo de compra. Assim, poupa-se o trabalho desse usuário de comparar diversos documentos e decidir o que tem a ver com seu interesse de negócio ou não. Ou seja, é preciso dar a esse usuário segurança na análise dos dados. Além disso, permite a um analista ou pesquisador, interessado em estudar tendências, testar hipóteses de comportamento de vendedores e compradores.

Dentro desse contexto, viu-se que é fundamental criar um processo que permita a coleta de páginas contendo a informação de interesse e efetue a extração de dados contidos nessas páginas a partir de uma representação relacional previamente criada pelo usuário.

O banco de dados relacional construído como resultado desse processo permite que dados contidos na Web possam ser analisados e manipulados de acordo com as necessidades de uma determinada aplicação (Figura 1.1).

Figura 1.1: Visão geral do processo de coleta das páginas e extração de dados da Web

Enfim, a motivação principal para a realização do trabalho aqui apresentado é contribuir com essas aplicações, auxiliando na tomada de decisão por meio da extração de dados de qualidade e dentro dos objetivos de negócio dos usuários.

1.3 Objetivo do Trabalho

Este trabalho descreve uma ferramenta, denominada Web2DB, que sistematiza o processo descrito anteriormente, permitindo a construção de representações relacionais de grandes sítios da Web, como, por exemplo, sítios de leilões eletrônicos. Nesse tipo de aplicação, a análise dos dados gerados durante um leilão é extremamente importante para se avaliar o comportamento e as decisões dos compradores. Essa importância pode ser evidenciada pelos trabalhos de Bapna *et al.* (2000, 2001). A Web2DB permite que os dados estejam difundidos em várias páginas de um mesmo sítio, pois cabe a ela a coleta dessas páginas percorrendo os vários tipos de páginas interrelacionadas existentes nesse sítio.

O processo se completa com a disponibilização dos dados em um repositório. A Web2DB insere os dados extraídos em um banco de dados relacional modelado pelo próprio usuário por meio da própria ferramenta e orientado aos seus objetivos de negócio. Além disso, pode-se criar visões para os dados, facilitando a geração, por exemplo, de armazéns de dados (do inglês *data warehouses*), como pode ser visto em Inmon (1996).

A maioria dos sítios eletrônicos armazena os seus dados em um banco de dados e os apresenta mediante requisição do usuário. Fica difícil, na maioria das vezes, ter acesso

a muitas páginas simultaneamente e fazer análises mais detalhadas. A Web2DB busca obter esses dados por meio da coleta das páginas de interesse de forma automática. Com isso, os dados são transpostos para um banco de dados relacional em que se tem domínio e acesso, podendo até incluir dados de vários sítios eletrônicos em um mesmo banco de dados, para fazer análises e comparações entre o conteúdo desses sítios.

O objetivo deste trabalho foi, portanto, desenvolver a ferramenta Web2DB de modo que fosse possível, de forma genérica e automática, efetuar a coleta de páginas e extração de dados de sítios eletrônicos de um domínio específico e de interesse do usuário, abrangendo o maior número de aplicações possíveis.

1.4 Estrutura da Dissertação

A dissertação a seguir está organizada da seguinte forma. O Capítulo 2 apresenta uma revisão bibliográfica sobre técnicas, conceitos e ferramentas de extração de dados da Web que influenciaram a ferramenta desenvolvida. O Capítulo 3 traz o detalhamento do trabalho desenvolvido e o funcionamento da ferramenta. O Capítulo 4 apresenta os resultados práticos obtidos com o trabalho desenvolvido e uma avaliação da eficácia da ferramenta, tendo como base dois estudos de caso. Por fim, no Capítulo 5, são discutidas as conclusões e possibilidades de trabalhos futuros que podem estender a ferramenta aqui desenvolvida.

Capítulo 2

Extração de Dados da Web

Com o grande volume de documentos existentes hoje na Web há uma dificuldade em se obter o conteúdo das páginas e não apenas localizá-las em máquinas de busca tradicionais. Viu-se que era necessário superar as limitações das máquinas de busca, fornecendo conteúdo de qualidade, dentro dos interesses dos usuários. Essa situação fez surgir as ferramentas de extração de dados da Web.

Conforme já abordado anteriormente, alguns fatores complicadores influem na decisão da melhor técnica de extração de dados a ser adotada para um determinado contexto. A Figura 2.1 mostra um exemplo dos dados contidos em um sítio de leilão eletrônico[1]. Os dados estão contidos em vários documentos difusos pelo sítio, fazendo com que os usuários tenham que entrar em várias páginas para obter informações de interesse. Além disso, pode-se ver que os dados estão em vários tipos diferentes de páginas, que se interrelacionam.

Diversos trabalhos nessa área têm sido realizados objetivando analisar e desenvolver técnicas e ferramentas para a extração de dados de páginas da Web. Uma visão geral das abordagens e ferramentas de extração de dados da Web pode ser vista em Laender *et al.* (2002a). As próximas seções tratarão dos mais recentes e relevantes trabalhos realizados em relação a esse assunto e que serviram como subsídio para o trabalho desenvolvido nesta dissertação.

2.1 Abordagens para Extração de Dados

Os dados em documentos da Web são apresentados, de maneira geral, de forma semi-estruturada. Conforme já apresentado, a motivação do trabalho é permitir que esses

[1] http://www.ebay.com

Figura 2.1: Extração de dados da Web

dados sejam capturados e estruturados em um banco de dados centralizado de forma a ser facilmente manipulado. Isso permitirá ao usuário tomar as decisões quanto a um objetivo de negócio específico. A estrutura dos dados existe nos documentos, mas de maneira implícita, já que os dados aparecem misturados com marcadores HTML e outros componentes, que não são de interesse. Para recuperar essa estrutura é necessário fazer uma engenharia reversa no código HTML.

Para a extração dos dados tradicionamente se usa a abordagem de criação de programas extratores, ou *wrappers*, que mapeiam os dados existentes em páginas da Web e os extraem, armazenando-os de acordo com um formato previamente definido (por exemplo, XML). Os *wrappers* utilizam regras de extração, previamente definidas, que permitem localizar o contexto em que os dados estão contidos nos documentos a serem pesquisados.

O problema de geração de *wrappers* pode ser definido da seguinte forma (Laender *et al.*, 2002a): Dada uma página da Web (S) contendo um conjunto de objetos implícitos, determinar o mapeamento (W) necessário para se criar um repositório de dados (R), com os objetos de S. Este mapeamento W deve ser capaz de reconhecer e extrair dados de outra página S' similar a S. O termo similar é usado significando páginas providas pelo

mesmo sítio da Web em questão. Assim, o *wrapper* é um programa que executa esse mapeamento W. A Figura 2.2 ilustra esse processo.

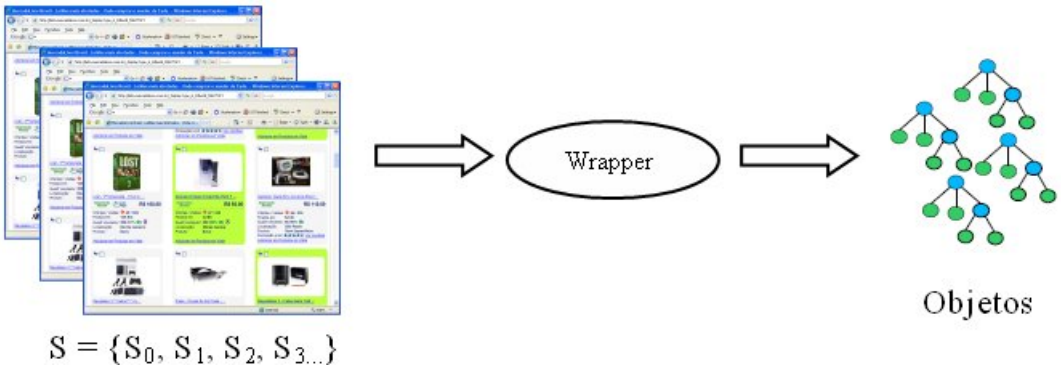

Figura 2.2: Atuação de um *wrapper*

Além disso, os *wrappers* devem ser precisos e robustos, ou seja, devem obter todos os dados corretamente e reconhecer as pequenas variações estruturais em sua apresentação, enfim, compreender o contexto em que o dado está inserido para a extração. Como já dito anteriormente, o importante é garantir a geração de um *wrapper* da forma mais automática possível mas, no entanto, sem perder flexibilidade. Em geral, os dois parâmetros (automação e flexibilidade) andam em direções opostas, devendo-se encontrar um ponto de equilíbrio entre eles (já que quanto maior a automação, menor o erro humano, o que gera resultados de maior confiabilidade).

De modo geral, a geração de *wrappers* pode ser realizada conforme três níveis de automação:

- Manual, que apresenta grande esforço de programação e é de difícil manutenção (grande incidência de erros);
- Semi-Automática, que requer a intervenção do usuário, seja, por exemplo, para alguma codificação ou especificação de exemplos;
- Automática, que utiliza mecanismos de inferência com base em um formato comum das páginas, mas requerem delimitadores fortes para os atributos, já que o processo é todo automatizado.

Ainda é possível, considerando o grau de automação, seguir a seguinte taxonomia para classificar as ferramentas de extração de dados da Web de acordo com as suas características e técnica utilizada (Laender *et al.*, 2002a):

- Orientadas a HTML: tomam como base a herança estrutural do código HTML. A página é convertida em uma árvore que reflete a hierarquia dos marcadores e a partir delas são geradas as regras de extração dos dados. As regras podem ser geradas de forma semi-automática (Liu *et al.*, 2000) ou automaticamente (Crescenzi *et al.*, 2001; Reis *et al.*, 2004). A ferramenta Lixto (Baumgartner *et al.*, 2001), que utiliza uma técnica de associação de filtros, também pode ser inserida nesta classificação.

- Baseadas em Processamento de Linguagem Natural: efetuam o aprendizado das regras de extração de dados a partir de documentos da análise do texto em linguagem natural. As regras de extração são derivadas analisando-se a relação entre as sentenças e considerando-se a sintaxe e semântica associados. São mais usadas em documentos contendo texto livre. Trabalhos como o de Califf and Mooney (1999) e Soderland (1999) se baseiam nessa abordagem.

- Por indução: são baseadas na utilização de técnicas de aprendizado de máquina que, a partir de um conjunto de exemplos de treinamento, permitem inferir regras de extração para os dados (mas que geralmente dependem de uma estrutura pré-determinada das páginas). Como exemplos de ferramentas que utilizam este método temos WIEN (Kushmerick, 2000), que depende de muitos exemplos e de uma estrutura pré-determinada das páginas e STALKER (Muslea *et al.*, 2001), que realiza uma extração hierárquica e possibilita tratar tipos aninhados.

- Assistidas: permitem ao usuário especificar visualmente os dados a serem extraídos de modo a gerar as regras de extração. Como exemplos temos as ferramentas DEByE (Laender *et al.*, 2002b), que gera as expressões por meio da seleção de exemplos, e NoDoSE (Adelberg, 1998), que utiliza decomposição hierárquica.

- Baseadas em Ontologias: baseiam-se na ontologia construída manualmente por uma pessoa experiente no assunto de interesse para, em função disto, extrair os dados automaticamente. O processo de extração não somente ocorre de forma automática mas também é adaptável a outras fontes de dados de um mesmo domínio. Um trabalho representativo nesse contexto pode ser visto em Embley *et al.* (1999).

A geração manual, como já dito, requer esforço de implementação e manutenção e pode se tornar inviável quando se trata de grande volume de documentos e dados. O

método por indução se mostra bastante interessante pois mescla intervenção humana, uma vez que requer o fornecimento de exemplos, e automação, que utiliza os exemplos de treinamento para inferir as regras de extração dos dados. Segundo Kushmerick (2000), o método de indução deve se preocupar em fornecer uma cobertura a um maior número de sítios eletrônicos possíveis. Para atender a uma maior variabilidade nas páginas, faz-se necessário um maior número de exemplos e estes em maior número podem afetar a performance. Assim, é importante realizar um *trade-off* com o custo necessário para fornecer amostras dos documentos e o custo de performance necessário para a indução a partir dos exemplos, buscando um ponto ótimo nesse contexto.

É importante, além de se preocupar com o grau de automação, fornecer uma interface de fácil utilização pelo usuário. Os *wrappers* assistidos focam na geração visual dos extratores. Arantes *et al.* (2001) propõem uma interface gráfica para tratamento de dados da Web através de visões das páginas de interesse. Uma vez realizada a extração (nesse trabalho em questão é utilizada a DEByE), essas visões mapeiam os atributos extraídos e permitem que os dados sejam analisados com facilidade. Pode-se, inclusive utilizar mais de uma fonte de dados para, por exemplo, confrontar informações de sítios eletrônicos diferentes, mas que tratam de conteúdos semelhantes. Além disso, a ferramenta permite também tratar da atualização dos dados extraídos definindo-se políticas para isso (*pooling*, *pushing* e *on demand*). Como atualmente a dinâmica dos dados na Web é alta, essa técnica é de fundamental importância para um extrator de dados da Web. Como será visto mais adiante, as técnicas desse trabalho serão aplicadas conceitualmente na ferramenta desenvolvida nesta dissertação. Outros trabalhos de destaque no contexto de *wrappers* assistidos podem ser citados, como o Lixto (Baumgartner *et al.*, 2001) e a DEByE (Laender *et al.*, 2002b) que serão detalhados mais adiante.

Em resumo, a geração de *wrappers* pode variar de inteiramente manual, passando por abordagens semi-automáticas, até inteiramente automática. Quanto mais automática for a técnica utilizada, mais dependente da estrutura HTML do documento será a ferramenta. O grau de flexibilidade em relação ao tipo de documento considerado vai ser maior à medida que se permite maior intervenção do usuário (por exemplo, ferramentas inteiramente manuais permitirão tratar arquivos de texto). O projeto de uma ferramenta de extração de dados da Web deve considerar o objetivo a que ela se destinará, diante da avaliação da técnica a ser usada e do grau de automação, além de outros requisitos que

serão listados a seguir.

De uma maneira geral, um *wrapper* analisa a estrutura de um documento HTML e gera expressões que representem os dados a serem extraídos nesse documento. A seguir acessa as páginas onde serão extraídos os dados e aplica essas expressões, buscando o padrão previamente definido, visto que as páginas apresentam estruturas semalhantes. Partindo desse princípio, alguns trabalhos focaram na análise dos documentos HTML como uma árvore, dispondo os marcadores HTML nos nodos, conforme mostrado na Figura 2.3.

Como as estruturas de árvore possuem operações específicas que podem ser aplicadas a elas, pode-se tirar proveito disso na geração do *wrapper*, como proposto por Zhai and Liu (2005). Reis *et al.* (2004) também se utilizaram da análise da estrutura de árvore para implementar um algoritmo para extração automática de notícias da Web. Esse algoritmo se baseia no conceito de distância de edição em árvore (custo associado ao conjunto mínimo de operações necessárias para transformar uma árvore A em uma outra árvore, B). Esses trabalhos ainda detalham outros conceitos importantes para serem usados na manipulação das árvores.

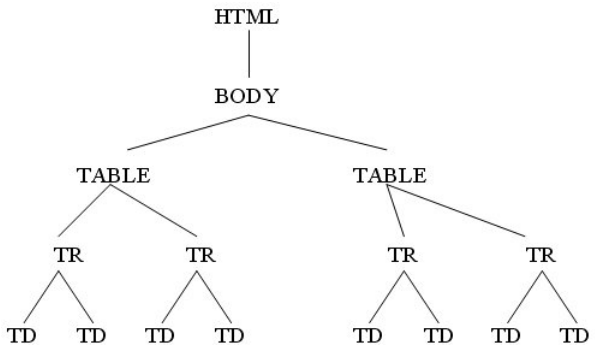

Figura 2.3: Árvore HTML

Muitas vezes as ferramentas de extração de dados da Web desenvolvem linguagens próprias, especializadas, para geração dos *wrappers*, no sentido de realizar o mapeamento da estrutura a ser extraída como uma implementação de um algoritmo. Wood and Ow (2005), por exemplo, estenderam a linguagem SQL para acessar páginas da Web e obter os dados de interesse como se fossem consultas a um banco de dados. A sintaxe é semelhante à da linguagem SQL tradicional e os comandos são executados em cima do conteúdo HTML do documento em questão. É uma abordagem no sentido de convergir as tecnologias, ou seja, uma pessoa acostumada a lidar com bancos de dados conseguirá extrair dados da Web

de maneira bastante simples. O resultado da consulta é uma tabela cujos dados podem ser facilmente inseridos em um banco de dados e utilizados por outras aplicações para gestão do conhecimento, por exemplo. A ferramenta TSIMMIS (Hammer and Garcia-Molina, 1997) também utiliza essa abordagem baseada em linguagens especiais para geração de *wrappers*.

Pode-se ver então que existem diversas abordagens para geração de *wrappers* e, dependendo da necessidade e dos objetivos desejados, alguns requisitos que serão detalhados mais adiante devem ser considerados para avaliação de qual a melhor abordagem a ser utilizada.

Outro ponto importante a ser levantado é que muitos sítios eletrônicos, mesmo que sobre um mesmo domínio, apresentam diferentes tipos de página, que são interrelacionadas. Os dados a serem extraídos podem, por ventura, estar espalhados entre os diversos tipos de página de um mesmo sítio eletrônico. Assim é importante que uma ferramenta de extração de dados seja apta a tratar esse relacionamento entre as páginas, feitos através de *hyperlinks*, e extrair os dados sob essas condições, ou seja, dados apresentados em documentos diferentes que se interrelacionam. Isso cria uma dificuldade a mais no processo, uma vez que analisar o conteúdo dos *hyperlinks* e conseguir obtê-los de forma automática é uma tarefa complexa, devido às várias formas que um *hyperlink* pode estar disposto no documento (Westerveld *et al.*, 2001).

Liu *et al.* (2004) desenvolveram uma ferramenta que permite a modelagem dos dados através de visões lógicas dos sítios da Web. Os dados são extraídos percorrendo as múltiplas páginas existentes. Ela apresenta uma interface visual que permite realizar a modelagem do sítio eletrônico e da extração dos dados. Como já mencionado, Arantes *et al.* (2001) deram contribuições nesse sentido, pois desenvolveram uma ferramenta que permite o mapeamento de visões das páginas e a extração dos dados de domínios distintos mas que apresentam dados semelhantes. Com isso, consegue-se agrupar dados de diferentes páginas mas que representem os mesmos objetos (ou estruturas), por meio de uma interface amigável com o usuário.

Assim, um requisito importante em uma ferramenta de extração de dados da Web é permitir que as entidades envolvidas sejam modeladas independente de onde as informações estão no sítio eletrônico. Esse requisito guiou de forma acentuada a modelagem conceitual da ferramenta Web2DB.

Além disso, alguns trabalhos focam, além da extração dos dados da Web, na geração de objetos, classes, enfim, entidades com um maior valor agregado ao entendimento do usuário. Esses objetos estão definidos implicitamente nas páginas da Web como, por exemplo, produtos, lojas, categorias, pessoas, vendedores, etc. (Nie *et al.*, 2007). Em alguns domínios específicos pode-se, com isso, ter ferramentas de buscas mais poderosas e precisas.

Para que se possa avaliar as diversas abordagens para extração de dados da Web existentes, deve-se utilizar alguns critérios que definem em um dado contexto qual a melhor abordagem a ser utilizada. Sahuguet and Azavant (2001) em seu trabalho destacaram importantes aspectos na concepção, desenvolvimento e avaliação de técnicas e ferramentas para extração de dados da Web, dentre os quais podemos citar:

- Automação: a tarefa de busca de informação de interesse geralmente necessita de uma navegação extensa e quanto maior o volume de dados mais difícil de se manipular os dados. Algumas ferramentas necessitam de escrita de código e apresentam alterantivas para contornar essa situação. Quanto maior a automação maior a facilidade de se lidar com grande volumes de dados;

- Usabilidade: a iteração com o usuário é fundamental no processo, assim como a iteração com outras aplicações. Aspectos que influenciam na usabilidade podem ser a existência de interfaces gráficas, uso de poucos cliques no processo de extração, fácil visualização dos resultados, etc.;

- Complexidade das estruturas: conforme dito anteriormente, a maior parte dos dados na Web está, de forma implícita, representando uma estrutura complexa. É importante que as ferramentas permitam estruturar esses dados e lidar com eles no processo de extração;

- Agregação de valor: deve-se comunicar com várias fontes de dados e permitir agregar mais valor aos dados, estendendo sua aplicação para vários tipos de sítios da Web, o que geralmente vai significar diferentes estruturas de documentos. Eventualmente até mesmo fontes não-HTML poderiam ser suportadas dependendo da aplicação que se quer.

Analisando os aspectos descritos acima, as ferramentas desenvolvidas lidarão com

uma série de requisitos que devem ser considerados na busca pela melhor relação custo-benefício que tornarão a ferramenta mais eficiente. Sahuguet and Azavant (2001) também listam alguns desses pontos. São eles:

- Autonomia: não se deve assumir muitas premissas em relação ao conteúdo das páginas. Os dados devem ser acessados na forma como são apresentados, sem se preocupar com detalhes específicos de um sítio eletrônico;
- Robustez: a ferramenta deve estar preparada para execução em diversos tipos de ambientes;
- Evolução e adaptação: ponto dos mais importantes, as ferramentas devem ser fáceis de serem modificadas para suportar a rápida e constante evolução da Web. Além de estar apta a extrair os dados mesmo diante da ocorrência de modificações nas páginas (ou pelo menos demandar o mínimo de alteração), a ferramenta também deve estar apta a extrair os dados de outra fonte dentro de um mesmo domínio (Golgher *et al.*, 2001);
- Escalabilidade: a divisão das tarefas de extração e coleta de páginas deve permitir tratar um volume crescente de dados.

De uma forma macro, os trabalhos que têm sido feitos têm o intuito de obter os dados que são apresentados e que originam de um banco de dados do qual não se tem acesso e que deve ser regerado por meio da extração dos dados das páginas disponíveis no sítio. De posse desse banco de dados, diversas análises podem ser feitas. Com isso o dado passa a ter um potencial maior de aproveitamento, agregando mais valor ao processo posterior de análise desses dados. Muitos trabalhos focam então, na associação de páginas da Web a modelos conceituais de banco de dados (Embley *et al.*, 1999; Mansuri and Sarawagi, 2006; Wang and Lochovsky, 2003).

O objetivo maior do processo de geração de *wrappers* é possibilitar uma integração dos dados das páginas da Web com outras aplicações, diferentes daquelas às quais os dados estão associados. Existe uma série de fatores e parâmetros que influenciam na decisão da abordagem com que serão gerados os extratores de dados. Em função deles surgem diversas ferramentas para aplicar as técnicas existentes. O avanço da tecnologia contribui para a constante e rápida evolução dessas ferramentas. A próxima seção mostrará algumas

das principais ferramentas existentes e que foram importantes conceitualmente para a elaboração do trabalho desenvolvido nesta dissertação.

2.2 Descrição de Algumas Ferramentas de Extração de Dados

Com a difusão das técnicas de extração de dados da Web, surgiram diversas ferramentas. Cada uma delas tem uma característica específica, e, em função do objetivo de aplicação que se quer, o usuário pode ter vantagens e desvantagens de usar uma ferramenta em detrimento de outra. Aliás, a proporção com que essas ferramentas vêm surgindo cada vez mais demanda uma análise sobre elas, seus recursos e aplicações.

Para o desenvolvimento da ferrameta Web2DB, algumas ferramentas existentes foram estudadas para que as suas melhores características fossem incorporadas à nova ferramenta. Assim, procurou-se criar uma ferramenta genérica de extração de dados da Web que, a partir da modelagem do usuário, pudesse extrair dados de documentos de um dado sítio, armazenando-os em um banco de dados relacional. Serão apresentadas agora as principais ferramentas estudadas e algumas de suas características, que posteriormente embasaram o desenvolvimento da ferramenta Web2DB.

O RoadRunner (Crescenzi *et al.*, 2001) é uma ferramenta que promove a geração automática dos *wrappers*. Todo o processo de extração de dados é automatizado. É feita uma compraração de duas instâncias de um documento HTML e gerado o *wrapper* baseado nas semelhanças e nas diferenças encontradas nesses documentos. Assim, não há qualquer intervenção do usuário no processo como, por exemplo, no apontamento de exemplos. Tudo é realizado de forma automática, já que a ferramenta percebe a estrutura da página e gera as regras de extração correspondentes.

Já a ferramenta XWRAP (Liu *et al.*, 2000) trata a geração de *wrappers* de forma semi-automática. Há uma maior iteração com o usuário, que participa do processo por meio de várias etapas. Em cada uma delas a ferramenta apresenta os componentes que são utilizados. O usuário visualiza o resultado da extração e pode dar um *feedback* para a ferramenta, refinando a estratégia para aumentar a eficácia da extração. Ao final é gerado o código do *wrapper*. Essa ferramenta permite isolar blocos de código gerado fazendo com que a tarefa de construção do *wrapper* seja estendida a várias páginas da Web e não somente uma página específica, possibilitando o reaproveitamento do *wrapper* em mais

de um domínio. Ela se utiliza de exemplos fornecidos pelo usuário em uma interface gráfica com alto grau de usabilidade. Tanto XWRAP quanto RoadRunner se baseiam na árvore que reflete a estrutura HTML das páginas de interesse e as regras de extração são aplicadas a essa árvore. W4F (Sahuguet and Azavant, 2001) é uma outra ferramenta que usa essa abordagem. O usuário define os dados, o caminho de acesso e de carregamento e as regras de mapeamento para extração dos dados, usando uma linguagem específica para esse fim.

Em relação à geração de *wrappers* por indução tem-se, como exemplo, a ferramenta STALKER (Knoblock *et al.*, 2000). Ela recebe como entrada um conjunto de exemplos fornecidos pelo usuário e a descrição da estrutura das páginas. Os *wrappers* vão sendo gerados de forma iterativa até que trate o maior número possível de variações dos exemplos, tratando, de forma automática, as diferenças existenes nas páginas. Os dados extraídos são estrturados em formato XML. A maior contribuição desse trabalho é a utilização de aprendizado de máquina para estender o *wrapper* inicial gerado por meio de indução. Isso permitiu maior automação do processo e a extração de dados com maior precisão.

A ferramenta NoDoSE (*Northwestern Document Structure Extractor*) também gera *wrappers* de forma semi-automática. O usuário participa do processo, realizando a modelagem de forma iterativa por meio de uma interface gráfica amigável. Além de páginas HTML, a ferramenta também permite extrair dados de documentos textuais (Adelberg, 1998).

Um trabalho de geração assistida de *wrappers* foi desenvolvido por Baumgartner *et al.* (2001). A técnica envolvida no Lixto permite ainda criar regras e condições envolvendo a extração, resultando na geração sob a atuação de um filtro especificado pelo usuário. Ele pode deternimar o padrão (condição) que deseja encontrar antes de um determinado atributo, no próprio atributo e também o intervalo de valores possíveis. Isso permite a geração de um *wrapper* especializado, que apresenta os dados extraídos em um documento XML. A ferramenta Lixto é importante neste estudo pois, assim como a Web2DB, permite o tratamento de múltiplos tipos de páginas que se interrelacionam por meio de *hyperlinks* no processo de extração.

A DEByE (*Data Extraction By Example*) é uma ferramenta que apresenta uma interface gráfica para que o usuário forneça como entrada um conjunto de exemplos dos dados a serem extraídos. A partir desses exemplos são geradas as regras de extração

dos dados presentes nas páginas similares às utilizadas para especificação dos exemplos. Essas regras, denominados *object extraction patterns* (OEP), determinam a estrutura que envolve os dados a serem extraídos e são usadas como orientação para o algoritmo que percorre as páginas e extrai os dados, utilizando uma extratégia *bottom-up* (Laender *et al.*, 2002b). Uma inovação trazida pela DEByE foi a utilização de tabelas aninhadas para definir objetos de estrutura complexa (Laender *et al.*, 2000).

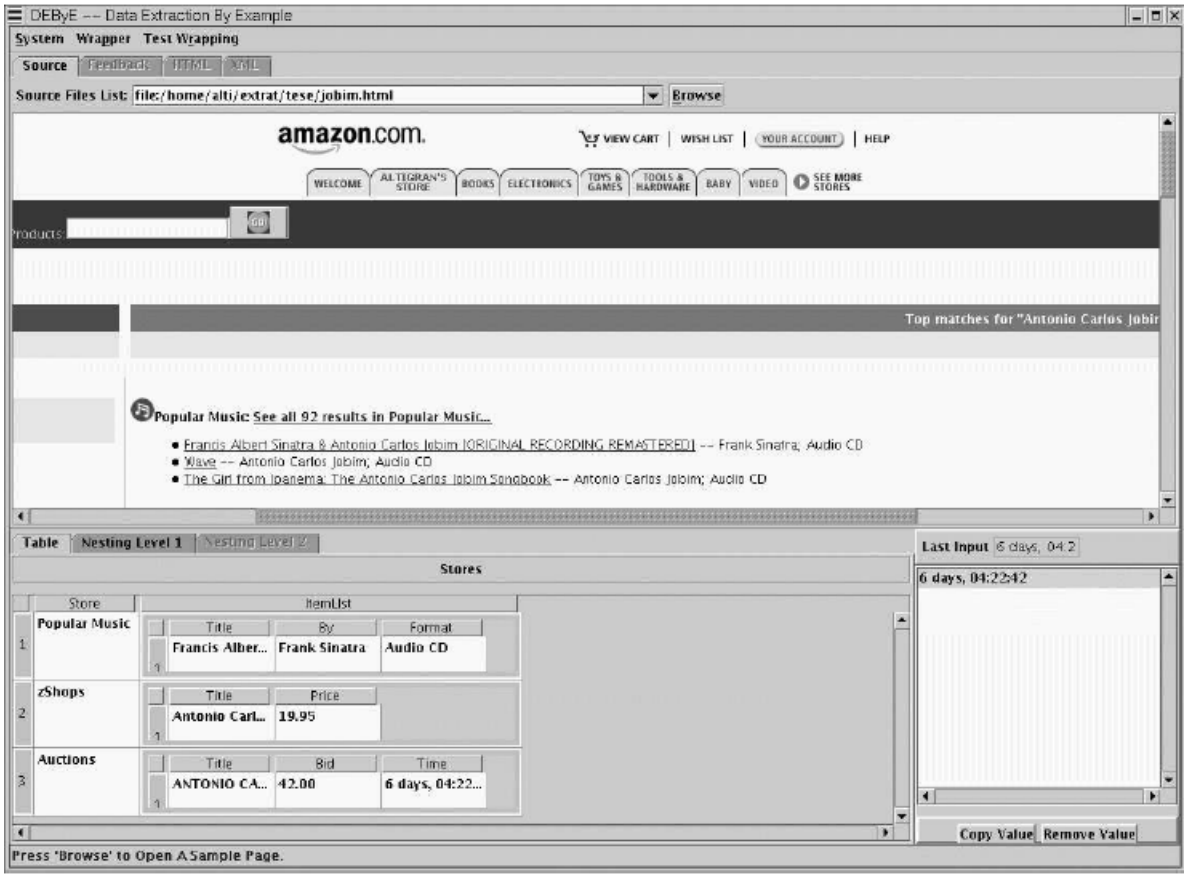

Figura 2.4: Interface gráfica da DEByE

Enfim, a ferramenta utiliza a percepção que o usuário tem dos objetos (classes) presentes nos documentos (que não estão estruturados dessa forma na apresentação das páginas). Os experimentos descritos em Laender *et al.* (2002b) e Ribeiro-Neto *et al.* (1999) mostraram um elevado percentual de extração dos dados com poucos exemplos utilizados. A DEByE também foi importante no processo de concepção da abordagem a ser utilizada pela Web2DB. A Figura 2.4 ilustra a interface gráfica da DEByE e a criação de uma tabela aninhada que modela a estrutura dos dados encontrados nas páginas da *amazon.com* onde três tipos de item (objetos) são considerados: *Popular Music*, *eShops*

e *Auctions*.

Muitas outras ferramentas foram desenvolvidas e outras continuam surgindo a cada dia (Ashraf and Alhajj, 2007; Li, 2007; Zhai *et al.*, 2007). Foi realizado uma análise preliminar de algumas delas, como mostrado acima, como forma de avaliar os principais conceitos utilizados no desenvolvimento da Web2DB. Além disso, uma outra ferramenta, DESANA (Sá Júnior *et al.*, 2006), foi utilizada como base da arquitetura adotada para o desenvolvimento da Web2DB. A próxima seção é dedicada a apresentar os detalhes referentes à DESANA.

2.3 DESANA

Conforme mencionado na seção anterior, na abordagem DEByE o contexto dos dados a serem extraídos é determinado a partir dos exemplos fornecidos pelo usuário. Além disso, o usuário especifica a estrutura alvo dos dados a serem extraídos por meio de uma tabela aninhada. Existem estratégias específicas para que, dado um conjunto de *avp - attribute value pairs* (associação do atributo com o seu valor) de exemplo, a ferramenta obtenha a expressão regular que conterá o padrão para extração dos atributos das páginas. A DEByE utiliza a estratégia *bottom-up*. Nela, o atributo é localizado de forma atômica, ao invés de fazê-lo a partir do objeto. Por exemplo, em uma página com a lista de livros à venda, inicialmente o algoritmo identificaria os atributos, como título, autores, editor e preço, para em seguida agrupá-los em uma tupla que define um livro.

Silva (2002) propõe um algoritmo, denominado *Hot Cycles*, que reconhece a estrutura dos dados sem a necessidade de especificar exemplos através de tabelas aninhadas. A partir dos dados extraídos das páginas (por exemplo, utilizando um *wrapper* gerado através da especificação de exemplos), o algoritmo *Hot Cycles* mapeia esses dados nas classes em que estão envolvidos, construindo objetos complexos automaticamente, apenas reconhecendo o contexto desses dados na estrutura da página HTML. Uma vez fornecidos os exemplos e extraídos os dados, o algoritmo identifica os objetos, agrupando os dados automaticamente.

Para isso, o algoritmo *Hot Cycles* constrói um grafo, denominado grafo adjacente, onde os vértices representam os atributos obtidos pelo padrão de extração mapeados nos exemplos fornecidos. Os arcos que conectam um par de vértices A_i e A_j representam o número de vezes que um atributo A_i precede o atributo A_j. O algoritmo trabalha itera-

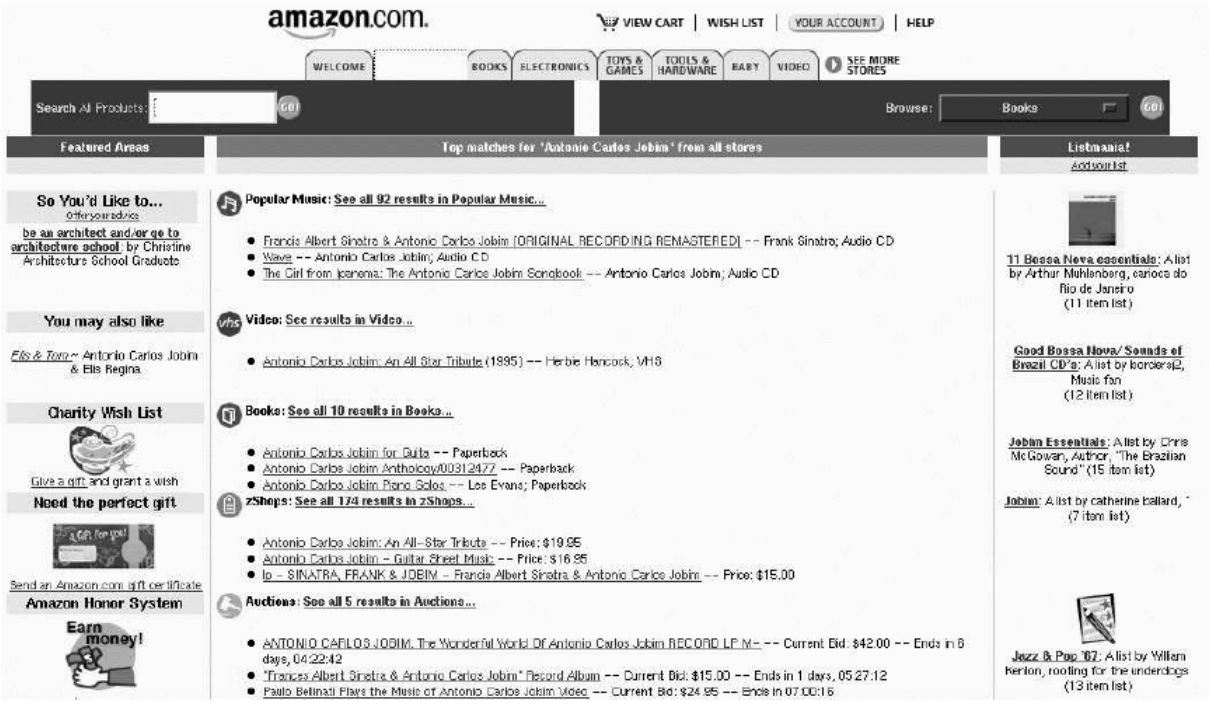

Figura 2.5: Exemplo de página

tivamente visando construir a cada etapa estruturas mais complexas, localizando ciclos nos grafos a partir dos arcos com maior valor. A cada ciclo encontrado, os atributos são agregados e representados por uma tupla. Em uma segunda etapa o algoritmo procura por laços (*loops*), ou seja, nós apontando para eles mesmos, para que seja construída uma lista de atributos. O algoritmo *Hot Cycles* utiliza a estratégia *bottom-up* para a construção dos objetos derivados dos dados extraídos. O algoritmo apresenta como saída uma lista de objetos que contém os dados agrupados nas estruturas complexas identificadas a partir dos dados extraídos (e armazenados em um documento XML com um formato pré-determinado).

A Figura 2.6 ilustra a seqüência de iterações para identificar objetos complexos pelo algoritmo *Hot Cycles*. O exemplo considera os atributos *StoreName*, *Item*, *Bid*, *Time*, *Author* e *BookType* de uma loja virtual (amazon.com) representada na Figura 2.5.

Como pode ser visto na Figura 2.6, ao final da execução do algoritmo é obtido um conjunto de objetos definidos pelo tipo (*StoreName*,[(*Item*, {*Author*}, *BookType*), (*Item*, *Bid*, *Time*)]), conforme modelo proposto por Laender *et al.* (2000). No caso, por exemplo, um dos atributos pode ser um variante, ora contendo os atributos *Item*, *Author* e *BookType*, ora contendo os atributos *Item*, *Bid* e *Time*. O atributo *Author*, quando

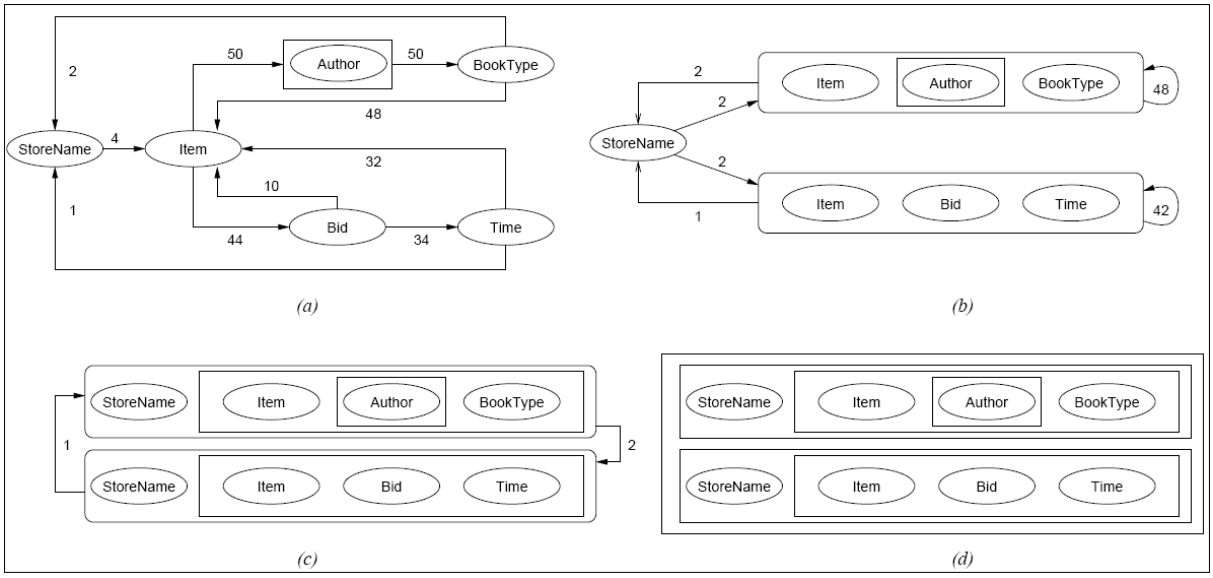

Figura 2.6: Exemplo da execução do algoritmo *Hot Cycles*

presente, o faz na forma de um conjunto de autores, para uma mesma tupla de objetos. Para este caso o documento XML de saída apresentará uma lista de valores para *Author*, como parte da tupla que compõe o segundo atributo. É gerado também um documento XML com as regras de extração dos atributos e ao final, após aplicação dessas regras pelo *wrapper*, é gerado um documento XML com os dados extraídos refletindo a estrutura identificada. Um exemplo desse documento XML de saída com os dados extraídos pode ser visto na Figura 2.7.

O algoritmo *Hot Cycles* tem importante aplicação na definição automática de estruturas complexas, geralmente implícitas nas páginas, facilitando o trabalho de especificação dos objetos pelo usuário. Uma vez mapeado os atributos, tem-se a estrutura sugerida pelo algoritmo, embora esta possa não ter obrigatoriamente a mesma estrutura imaginada pelo usuário. O resultado é um documento XML que contém as tuplas e listas de atributos que foram extraídos das páginas de interesse que devem ter a mesma estrutura que as páginas fornecidas para os exemplos dos atributos.

O algortimo *Hot Cycles* foi implementado por Sá Júnior *et al.* (2006) em uma ferramenta denominada DESANA. A DESANA apresenta a arquitetura descrita na Figura 2.8. Uma interface gráfica visual permite que o usuário defina os objetos envolvidos e os atributos contidos nesses objetos e forneça os exemplos para a geração dos *wrappers*. Após a seleção dos exemplos e geração dos *wrappers*, a ferramenta extrai os dados das páginas, que têm que estar previamente coletadas e armazenadas juntamente com as páginas de

```
<?xml version = "1.0" encoding = "iso-8859-1"?>
<OBJECTS>
  <TUPLE ipos="11514" type="Store">
    <ATOM ipos="11514" type="StoreName">
        <VALUE fpos="11527" ipos="11514"><![CDATA[Popular Music]]></VALUE>
    </ATOM>
    <LIST ipos="11819" type="ItemList">
      <TUPLE ipos="11819" type="ItemList">
        <ATOM ipos="11819" type="Item">
          <VALUE fpos="11900" ipos="11819"><![CDATA[Francis Albert ...]]]></VALUE>
        </ATOM>
        <ATOM ipos="11908" type="By">
          <VALUE fpos="11921" ipos="11908"><![CDATA[Frank Sinatra]]></VALUE>
        </ATOM>
        <ATOM ipos="11923" type="Format">
          <VALUE fpos="11931" ipos="11923"><![CDATA[Audio CD]]></VALUE>
        </ATOM>
      </TUPLE>
      <TUPLE ipos="12171" type="ItemList">
        ...
      </TUPLE>
    </LIST>
  </TUPLE>
  ...
</OBJECTS>
```

Figura 2.7: Exemplo de saída do algoritmo *Hot Cycles*

exemplos fornecidas. Nessa etapa é gerado um documento XML com os dados extraídos das páginas. Esta fase segue a estratégia *bottom-up* utilizada na DEByE. De posse desse documento XML, o algoritmo *Hot Cycles* é executado e, após definir a estrutura dos dados a partir dos exemplos fornecidos, agrupa os dados extraídos segundo a estrutura inferida a partir dos exemplos especificados pelo usuário na interface gráfica. O usuário pode ainda salvar as definições dos atributos e dos exemplos e visualizar as páginas dentro da própria ferramenta.

Será mostrado mais adiante que a ferramenta da DESANA responsável por extrair e agrupar os dados nas estruturas será utilizada como componente arquitetural mais importante da Web2DB. Os métodos responsáveis pela extração dos dados e execução do algoritmo *Hot Cycles*, juntamente com a interface gráfica desenvolvida para esta dissertação, vão definir um processo diferenciado de coleta de páginas, modelagem, extração e carregamento de dados da Web em um banco de dados específico. Essa nova ferramenta tem como intuito estender a aplicação da DESANA para novos contextos como descrito na próxima seção.

2.4 Aplicações

Uma aplicação típica que requer o uso de uma ferramenta de extração de dados da Web é a de análise de sítios de leilões eletrônicos. Informações importantes, como média de

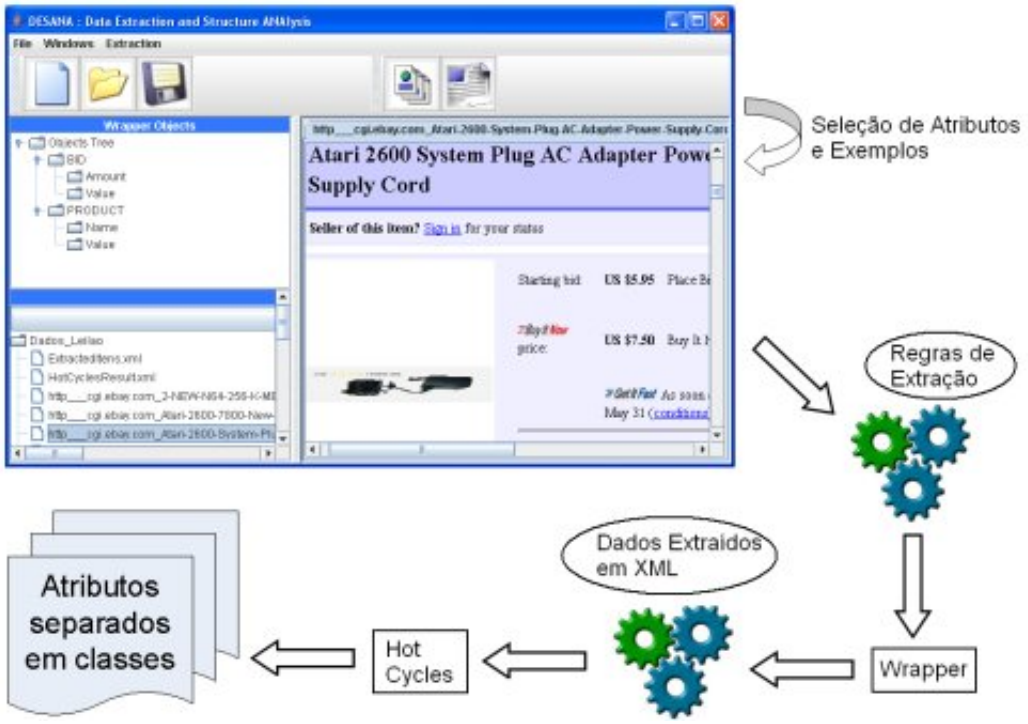

Figura 2.8: Arquitetura da DESANA

valores negociados, número de lances por leilão, enfim, informações que permitem analisar as decisões feitas pelos usuários dos leilões eletrônicos, podem ser obtidas por meio dos dados disponíveis nas páginas dos leilões eletrônicos. Por essa razão, torna-se importante a existência de ferramentas que ajudem nesse objetivo. O foco dos trabalhos realizados é observar o comportamento dos leilões pela Internet e os fatores que podem influenciar as decisões dos usuários desses sites (Bakos, 1997; Bapna *et al.*, 2000, 2001, 2004). Esses trabalhos trazem resultados de análise de dados históricos, experimentos, simulações e leilões *on-line*, possibilitando uma análise de estratégias empregadas nos leilões, entre outras coisas. Embora seja uma iniciativa importante no sentido de fazer uma vasta análise no comportamento desse ramo de comércio eletrônico, é necessário evoluir no sentido de uma maior automatização do processo de aquisição e análise dos dados bem como no uso de grandes massas de dados. Além disso, a dinâmica desse processo oferece desafios, como, por exemplo, as rápidas mudanças que ocorrem nos leilões de um dia para o outro. Ou seja, ainda há lacunas que precisam ser preenchidas e novas ferramentas podem surgir nesse contexto.

Além dos leilões eletrônicos, outro domínio importante onde as ferramentas de extração de dados da Web são úteis é o de publicações científicas. Hoje em dia exis-

tem diversos sítios de conferências, instituições acadêmicas e editoras com dados sobre diferentes tipos de publicação, autores e instituições, por exemplo. Esses dados estão geralmente difusos em vários sítios eletrônicos, embora todos tratem de entidades (objetos) semelhantes. É importante então permitir um fácil acesso a esses dados, centralizando-os em um banco de dados único para se ter uma visão mais ampla e não apenas de uma única conferência ou instituição específica, por exemplo. Uma ferramenta aplicada a esse contexto deve permitir a integração de várias fontes de dados em um único banco de dados.

Além do comércio eletrônico e das publicações científicas, outras aplicações para as ferramentas de extração de dados da Web podem ser facilmente encontradas, como, por exemplo, as que envolvem notícias na Web, informações de competições esportivas, dados das bolsas de valores, etc. Deve-se destacar que hoje há um volume cada vez maior de dados disponíveis na Web de modo que ferramentas analíticas que permitem processar esses dados são cada vez mais necessárias.

O desenvolvimento de ferramentas como meio de extrair esses dados da Web e povoar um banco de dados permite a análise do negócio envolvido. Um ambiente de armazéns de dados, por exemplo, fornece armazenamento, funções e respostas a consultas que ultrapassam a capacidade de bancos de dados tradicionais. Um armazém de dados, assim como um de banco de dados, envolve uma coleção de dados e um sistema que permita o armazenamento e o tratamento desses dados. A diferença é que bancos de dados tradicionais são voltados para aplicações transacionais, enquanto os armazéns de dados são essencialmente direcionados para aplicações de apoio à tomada de decisão, sendo otimizados para recuperação de dados (Elmasri and Navathe, 2002).

Um armazém de dados pode ser definido como uma coleção de dados orientados ao assunto, integrados, não-voláteis e variantes no tempo, para fornecer apoio a decisões gerenciais (Inmon, 1996). Fornecem com isso dados para análise complexa, descoberta de conhecimento e tomada de decisão. Os armazéns de dados inserem-se nesse contexto de extração de dados da Web, pois se bem alimentados disponibilizam estrutura e funções especiais para a análise dos dados, transformando-os em informação. Em relação aos sítios de leilões eletrônicos, os dados envolvidos se adequam à utilização de armazéns de dados para a gestão e auxílio à tomada de decisão, pois o usuário teria acesso a dados de diversos leilões (andamento, evolução dos lances, etc.). Os armazéns de dados são adequados para

esse contexto. As ferramentas de extração de dados da Web devem apresentar os dados extraídos, seja por exemplo em um banco de dados relacional ou documentos XML, de forma a facilitar a importação dos dados para um armazém de dados.

Existem várias aplicações que fazem uso de armazéns de dados como as aplicações OLAP (*on-line analytical processing*). Essas aplicações possibilitam uma visão dos dados em várias dimensões e operações específicas para a sua manipulação.

2.5 Contexto da Web2DB

Conforme visto nas seções anteriores, existem diversas técnicas e estratégias para extração de dados da Web e muitas ferramentas foram e continuam sendo desenvolvidas para essa finalidade. O trabalho realizado nesta dissertação possibilitou o projeto e desenvolvimento de uma ferramenta, denominada Web2DB, que será detalhada no próximo capítulo. Procurou-se inovar no processo de extração de dados da Web, com o foco na disponibilização dos dados de forma mais fácil de serem analisados.

O trabalho desenvolvido procurou utilizar as melhores abordagens e conceitos analisados neste capítulo para construção de uma ferramenta que possa ser aplicada em um contexto genérico, podendo ser usada em qualquer tipo de aplicação, como será avaliado mais adiante nesta dissertação. Para tanto, ela foi testada em dois ambientes distintos, permitindo automatizar a extração de dados tanto de sítios de leilões eletrônicos quanto de sítios de publicações científicas, preenchendo possíveis lacunas nos trabalhos de análise desses dados.

Capítulo 3

Ferramenta Desenvolvida

O trabalho realizado e detalhado nesta dissertação trata do desenvolvimento da ferramenta Web2DB. A ferramenta utilizou técnicas de extração de dados da Web implementadas pela DESANA e visa a coleta de páginas e extração dos dados das páginas coletadas, de acordo com a orientação fornecida pelo usuário. Ela foi concebida levando em consideração os requisitos de avaliação de ferramentas de extração de dados da Web, discutidos anteriormente.

A metodologia de trabalho foi baseada inicialmente no estudo de ferramentas existentes e análise de pontos importantes para o desenvolvimento de ferramentas de extração de dados da Web, bem como técnicas e abordagens mais utilizadas, conforme apresentado anteriormente. A ferramenta foi desenvolvida de forma a facilitar as seguintes tarefas:

- Modelar os dados a serem extraídos;
- Extrair os dados de páginas da Web de forma fácil e automática;
- Coletar as páginas de interesse de forma automática;
- Exportar os dados extraídos para um banco de dados para posterior análise;
- Facilitar a interação com o usuário, que poderá modelar todo o processo de acordo com os seus interesses.

Em cima desses macro-objetivos da ferramenta foi desenhada a arquitetura da solução e as funções a serem implementadas. Por fim, fez-se o desenvolvimento da ferramenta e os resultados a serem apresentados posteriormente permitiram validar o trabalho desenvolvido.

As próximas seções irão detalhar a ferramenta Web2DB, descrevendo a sua arquitetura, componentes, funções e forma de utilização.

3.1 Visão Geral da Ferramenta Web2DB

A Web2DB tem a função principal de obter os dados de um sítio eletrônico importando-os para um banco de dados. Para tanto é necessário identificar nas páginas em questão as entidades envolvidas, que normalmente estão implícitas na estrutura HTML de apresentação dos dados. Dessa forma é importante a atuação humana nesse processo para definir os dados de interesse e associá-los a um destino.

Por exemplo, para um sítio de leilão eletrônico[1], as páginas de um leilão contêm dados como: valores, lances, descrição do item, foto, forma de pagamento, prazo, entre outros (Figura 3.1). Para uma análise mais profunda de vários leilões de uma categoria específica, por exemplo, seria interessante que os dados estivessem centralizados em um banco de dados. Na verdade eles estão dispostos em um banco de dados único, mas sem acesso disponível, o que apenas pode ser feito pelas páginas. Para evitar o acesso a várias páginas, faz-se necessário um processo que colete essas páginas e extraia os dados, importando-os para um banco de dados que se tenha acesso, para que se possa então processá-los de forma apropriada. A Web2DB atua no cerne dessa questão, trazendo para esse banco de dados os dados que estão acessíveis apenas via acionamento das páginas HTML.

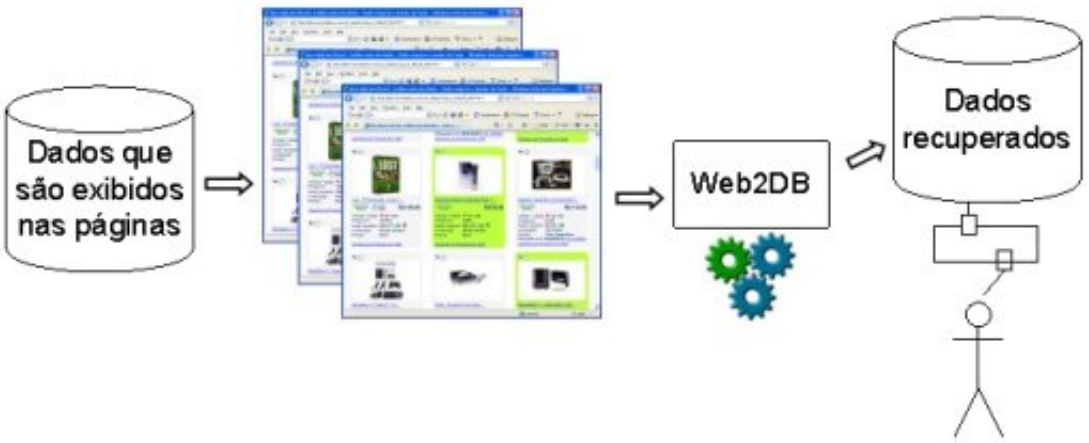

Figura 3.1: Atuação da Web2DB

[1] http://www.ebay.com

A Figura 3.2 apresenta o esquema de um banco de dados hipotético de leilões, de acordo com a notação da ferramenta DBDesigner [2]. Esse esquema será utilizado posteriormente para ilustrar o comportamento da ferramenta a cada etapa do processo. Ele é composto pelas seguintes tabelas:

- AUCTION: tabela principal, que contém os dados do leilão como número de lances, período, localização, entre outros. Se relaciona com as tabelas PRODUCT, SELLER e BIDS por meio de atributos que definem chaves estrangeiras.
- SELLER: tabela que contém os dados de vendedores como pontuação, nome, entre outros. Cada leilão envolve um único vendedor.
- PRODUCT: tabela que contém os dados do produto como descrição, categoria, entre outros. Cada leilão refere-se apenas a um único produto.
- BIDS: tabela que contém os dados dos lances executados no leilão como data e hora, nome do comprador, pontuação, valor, entre outros. Cada leilão envolve uma lista de 0 a N lances.

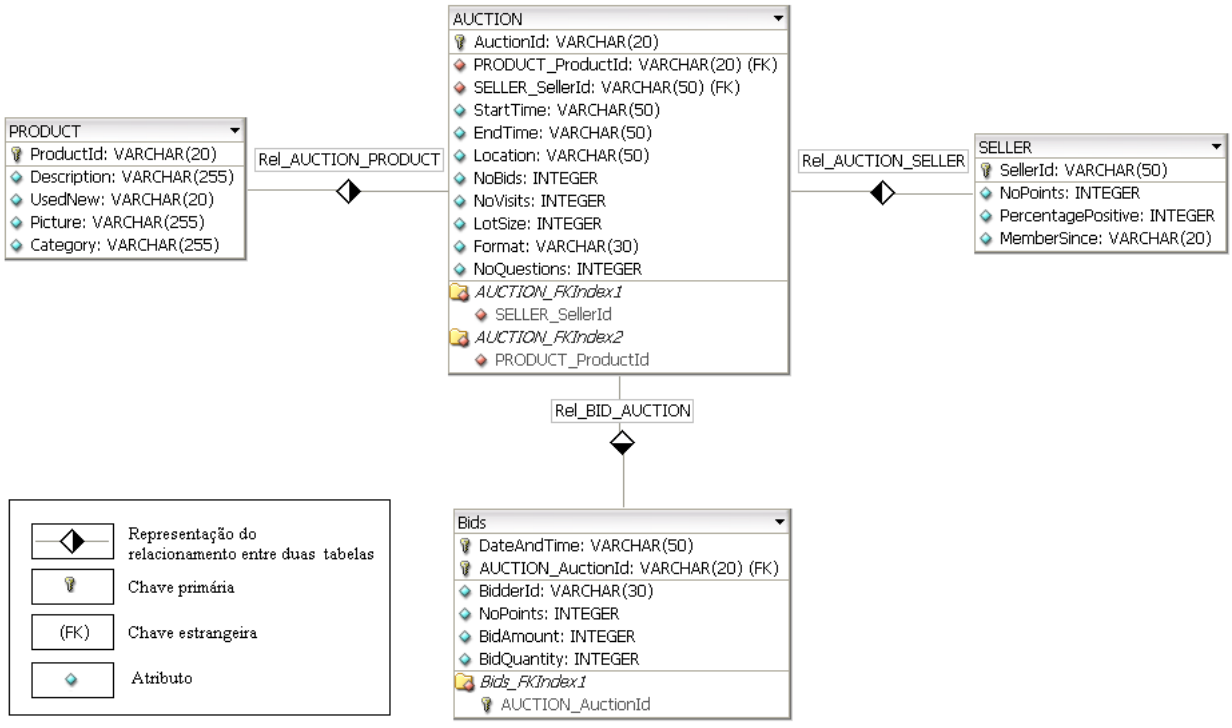

Figura 3.2: Exemplo de um esquema de banco de dados para dados de leilões eletrônicos

[2]http://fabforce.net

A Web2DB foi desenvolvida em linguagem Java e apresenta os seguintes componentes em sua arquitetura:

- Camada de apresentação: responsável pela interface visual, disponilizando as funções por meio das telas em um formato de *wizard*, para facilitar a sua utilização;
- Camada de aplicação: essa camada é responsável pela extração dos atributos e agrupamento dos mesmos nas entidades que serão inseridas posteriormente no banco de dados. Esta camada é representada pela API[3] da ferramenta DESANA que gera o extrator de dados a partir de exemplos fornecidos pelo usuário;
- Repositório: os dados do projeto[4] que o usuário modela é salvo em formato XML para que, a qualquer instante, ele possa interromper o processo e reiniciá-lo posteriormente.

O processo é sistematizado nas seguintes etapas: 1) Modelagem do Banco de Dados; 2) Geração do Plano de Coleta das Páginas; 3) Coleta das Páginas; 4) Mapeamento dos Dados a serem Extraídos; 5) Extração dos Dados e 6) Inserção dos Dados no Banco de Dados. A Figura 3.3 ilustra o processo de utilização da Web2DB.

Como pode ser visto, a atuação do usuário é fundamental para modelar os objetivos de negócio envolvidos na extração. Ele atua na especificação do esquema do banco de dados, na especificação do plano de coleta das páginas e no mapeamento dos dados extraídos das páginas coletadas para o banco de dados. Ao final, pode ainda gerar visões para melhor análise dos dados extraídos. Tudo é realizado por meio de uma interface gráfica amigável. Uma vez que o usuário atuou com o conhecimento do negócio, especificando as informações que representam o seu interesse numa dada extração, a ferramenta é capaz de realizar o restante do processo de forma automática. Existe um repositório fonte de informações (páginas da Web), a partir do qual os dados são extraídos e exportados para um banco de dados final. As próximas seções detalham cada uma das etapas sistematizadas na Web2DB.

[3] *Application Programming Interface*: interface que uma aplicação, sistema operacional ou biblioteca provê para suportar requisições feitas por outros programas. Permite encapsulamento de lógica para que a mesma seja reaproveitada.

[4] Projeto: o termo projeto é utilizado neste contexto para representar os dados de entrada do usuário na ferramenta Web2DB, usados para configurar toda a modelagem necessária para que a ferramenta seja executada. Esses dados incluem aqueles necessários para realizar a coleta das páginas e extração dos dados, exemplos fornecidos, esquema do banco de dados, entre outros.

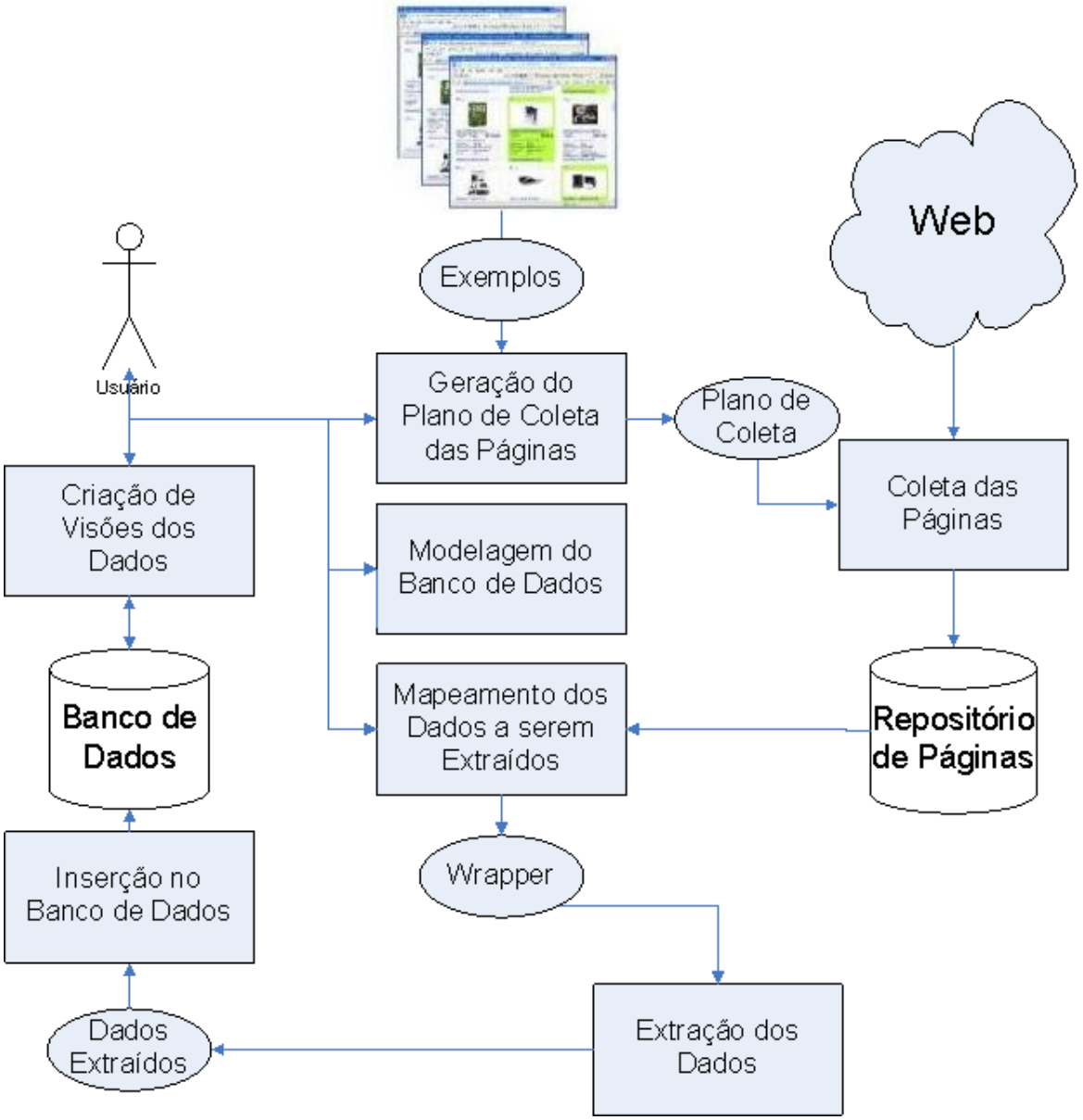

Figura 3.3: Processo de utilização da Web2DB

3.2 Modelagem do Banco de Dados

A primeira etapa trata da modelagem do banco de dados que receberá os dados extraídos das páginas. Inicialmente o usuário fornece um nome para o projeto, ou seja, o diretório onde serão geradas todas as saídas da ferramenta e onde também serão armazenadas as páginas coletadas para extração, observações para algum detalhe que seja importante registrar e por fim o nome das tabelas que serão carregadas no banco de dados. A Figura 3.4 mostra a tela que é utilizada nessa etapa.

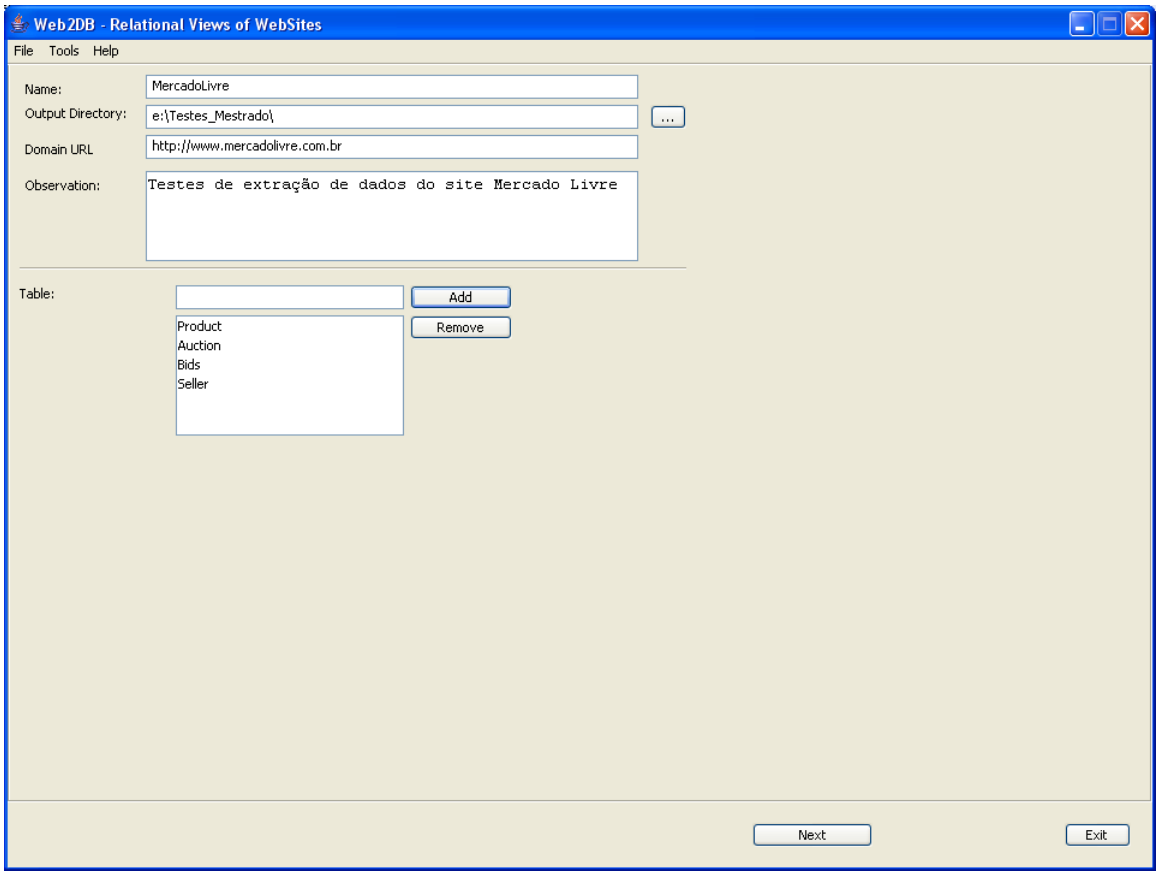

Figura 3.4: Web2DB - Dados gerais iniciais

Feito isso, o usuário avança para a modelagem do banco de dados em si. Nesse momento, as tabelas definidas são apresentadas para que se possa adicionar os atributos, que serão alvo da extração com seus respectivos tipos. Os tipos podem ser, por exemplo, *string*, número e data. Além disso, é informado também o relacionamento entre as tabelas por meio de chaves primárias e estrangeiras (Elmasri and Navathe, 2002). Embora a Web2DB não apresente nenhuma restrição de carregamento de dados quanto ao relacionamento entre as tabelas, é importante conceitualmente a utilização desses parâmetros

para a correta criação dos atributos.

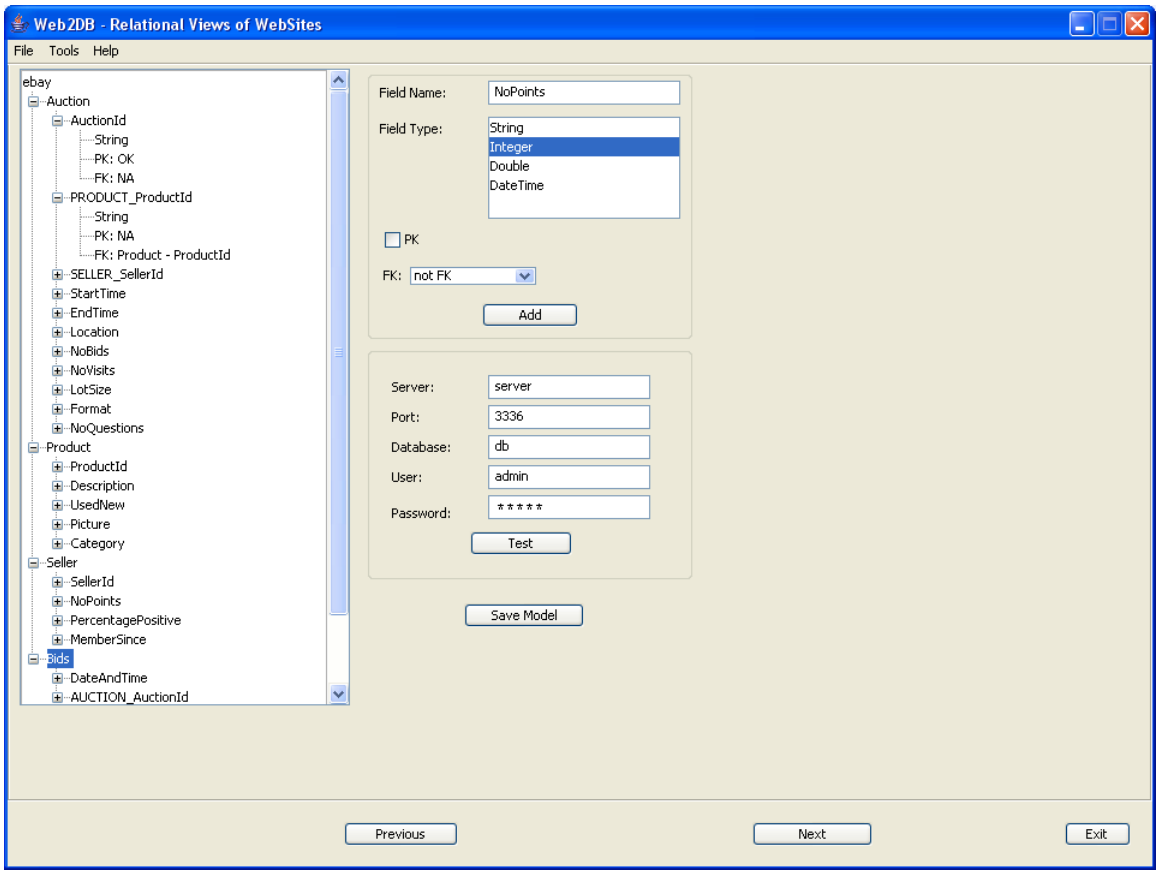

Figura 3.5: Web2DB - Modelagem do Banco de Dados

Por fim, é preciso também informar os dados necessários para se efetuar a conexão com o banco de dados (como, por exemplo, usuário, senha, servidor etc.) para posterior inserção dos dados. É possível também verificar se a conexão com o banco de dados está ativa. A Figura 3.5 mostra a tela em que essa etapa é realizada.

Nesse momento o esquema do banco é salvo em formato XML (Figura 3.6) e será utilizado pela ferramenta Web2DB nas etapas seguintes, uma vez que os atributos do banco de dados serão mapeados nas páginas. Essa etapa é importante pelo fato de ser o momento em que o usuário analisa mais detalhadamente os dados de interesse, dentro dos objetivos de análise que ele deseja posteriormente. É baseado nessa modelagem que todas as etapas seguintes ocorrem, pois o foco da ferramenta é alimentar o banco de dados correspondente com os dados da Web.

```
- <ModelDW>
    <ModelName>Mercado_Livre</ModelName>
    <Observation>Teste de extracao de dados do site Mercado Livre</Observation>
    <DOMAIN_URL>http://www.mercadolivre.com.br</DOMAIN_URL>
  - <CONNECTION_DATA>
      <SERVER>server</SERVER>
      <PORT>3336</PORT>
      <DATABASE>db</DATABASE>
      <USER>admin</USER>
    </CONNECTION_DATA>
  - <Tables>
    - <Table>
        <Name>Auction</Name>
      - <Fields>
        - <Field>
            <Name>AuctionId</Name>
            <Type>String</Type>
            <PrimaryKey>OK</PrimaryKey>
            <ForeignKey>NA</ForeignKey>
          </Field>
        - <Field>
            <Name>PRODUCT_ProductId</Name>
            <Type>String</Type>
            <PrimaryKey>NA</PrimaryKey>
            <ForeignKey>Product - ProductId</ForeignKey>
          </Field>
          ...
  </Tables>
</ModelDW>
```

Figura 3.6: Web2DB - Documento XML resultante da modelagem do banco de dados

3.3 Geração do Plano de Coleta das Páginas

Uma vez definida a estrutura do banco de dados que irá conter as dados extraídos é necessário obter a fonte dos dados de interesse, que são as páginas dos sítios eletrônicos. A etapa de coleta de páginas representa o processo de formação do repositório fonte das informações de interesse para extração. Inicialmente o usuário necessita modelar a coleta das páginas para depois realizá-la.

Essa modelagem é feita por meio da geração de um plano de coleta das páginas. Como já havia sido mencionado, é muito comum os dados estarem difundidos entre vários tipos de página, mesmo que em um mesmo sítio eletrônico. Cada tipo de página tem uma estrutura HTML própria e estão interrelacionadas por meio de *hyperlinks*. No entanto, existe normalmente uma lógica natural de navegação pelas páginas e a Web2DB utiliza isso para automatizar o processo de percorrer todos os *hyperlinks* e coletar as páginas que são armazenadas em um repositório.

O usuário informa para a Web2DB como ela deve percorrer as páginas e quais coletar, criando assim um coletor especializado que obtém apenas páginas específicas,

segundo um plano de coleta previamente elaborado.

A geração do plano de coleta ocorre da seguinte maneira. O usuário abre uma página inicial e a partir dela informa *hyperlinks* que devem ser acionados e o tipo de página que os mesmos acessam. Essa página inicial pode estar salva em disco ou ser acessada *on-line*. A Web2DB possui uma interface gráfica que apresenta o conteúdo das páginas de modo que dentro da própria ferramenta o sítio eletrônico pode ser visualizado a partir de sua URL. Páginas já acessadas via URL são salvas no repositório e podem ser acessadas diretamente a partir dele. Enfim, o usuário deve selecionar os *hyperlinks* que interligam as páginas para cada tipo de página existente no contexto do sítio eletrônico.

O processo de mapeamento dos *hyperlinks* e dos tipos de página existentes deve ser repetido ao menos uma vez para cada tipo de página que se deseja coletar. Ao fim se obtém um mapeamento entre tipos de página e os atributos que as interligam. O plano de coleta gerado é salvo para posteriormente ser executado para se percorrer o caminhamento dos *hyperlinks*, extrair as páginas e salvá-las no repositório.

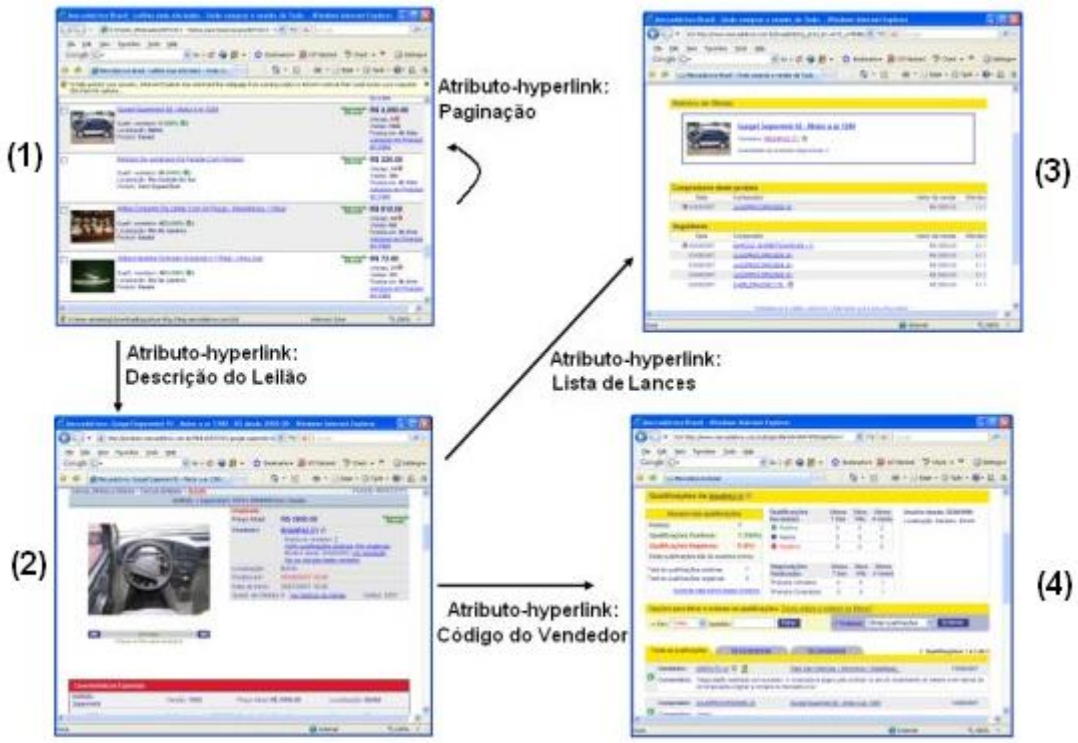

Figura 3.7: Exemplo seqüência para geração de um plano de coleta das páginas

A Figura 3.7 ilustra um exemplo de geração do plano de coleta das páginas para um sítio de leilão eletrônico. O primeiro tipo de página contém uma lista de leilões

disponíveis no momento (1). Cada item dessa lista possui um *hyperlink* que se conecta a um outro tipo de página que contém os dados do leilão em questão (2). Nesse novo tipo de página, ainda podem ser acessados *hyperlinks* para visualizar outros tipos de página, como a lista de lances efetuados no leilão (3) ou dados dos vendedores e compradores (4). A especificação dos *hyperlinks* e a seqüência em que essa ação é feita fornece para a ferramenta a seqüência do caminhamento que deve ser feito pelas páginas, para que todas sejam acessadas e coletadas na etapa de coleta que vem a seguir.

A Web2DB possui uma interface gráfica que facilita a execução do processo de geração do plano de coleta que, ao final, irá tornar totalmente automática a etapa de coleta das páginas que contêm os dados a serem extraídos. A Figura 3.8 ilustra como o usuário especifica os *hyperlinks* e a conexão entre os vários tipos de página de interesse.

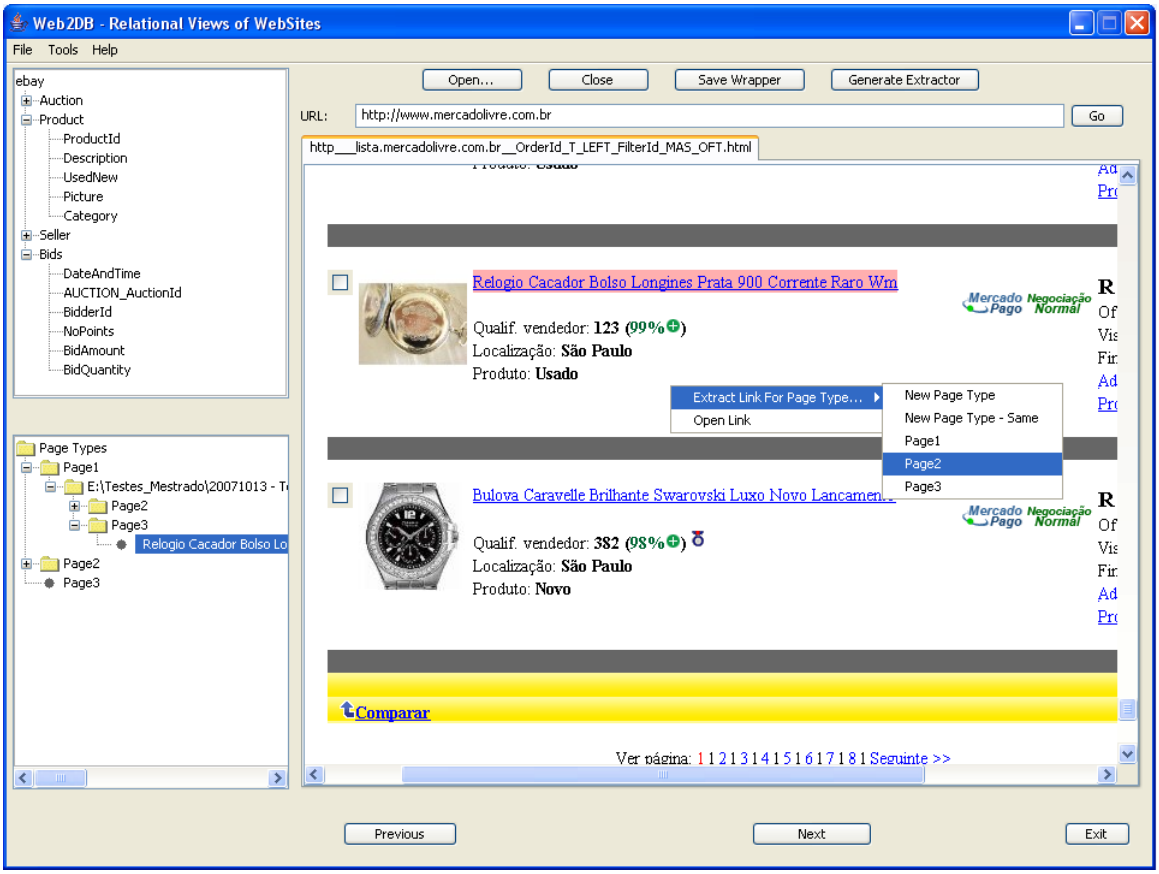

Figura 3.8: Web2DB - Especificação de *hyperlinks*

Nesse ponto é interessante destacar a lógica envolvida nesse processo de geração do plano de coleta das páginas. Ao selecionar um *hyperlink* que identifica a conexão com outro tipo de página, é utilizada técnica de extração de atributos baseadas em exemplos para gerar expressões regulares que extraem os *hyperlinks* e permitem a coleta automati-

camente. No caso, os *hyperlinks* são tratados como atributos e a extração desses atributos permite determinar o padrão desses *hyperlinks* que devem ser acessados e extraídos pelo coletor. Assim, com poucos exemplos, pode-se, por exemplo, extrair uma lista enorme de leilões, já que geralmente há um padrão para os *hyperlinks* que acessam as páginas de leilões a partir dessa lista.

A Web2DB utiliza esses exemplos de *hyperlinks* como atributos para gerar um *wrapper*. Cada tipo de página possui um *wrapper* que instrui o extrator a coletar os *hyperlinks* das páginas que serão armazenadas pelo agente de coleta da Web2DB.

A partir do plano de coleta é gerado um coletor que é responsável por implementar o caminhamento entre as páginas e utilizar os *wrappers* dessa etapa para coletar as páginas de interesse. Todo o processo de caminhamento é feito com base no plano de coleta que é representado como um documento XML (ver Figura 3.9). Esse plano especifica os atributos a serem extraídos (*hyperlinks*), as expressões regulares que definem o padrão do *hyperlink* e os tipos de página envolvidos. A próxima seção irá detalhar o funcionamento do coletor.

```
- <COLLECTOR>
  - <PAGE_TYPE id="Page1">
    - <PATTERN to="Page2">
        <![CDATA[ </td>\s*<td class="[^"]*">\s*<div class="[^"]*">\s*(?:[\s]|(?:</?img\b
        [^<>]*>))*?(?:<a href="[^"]*">)?\s*([^<>]+)\s*</a>\s*</div>\s*  ]]>
      </PATTERN>
    </PAGE_TYPE>
  - <PAGE_TYPE id="Page2">
    - <PATTERN to="Page3">
        <![CDATA[ <td width="[^"]*" colspan="[^"]*">\s*<span id="[^"]*">\s*<a href="[^"]
        *">\s*([\p{L}\d\s]+)\s*</a>\s*</span>\s*</td>\s*  ]]>
      </PATTERN>
    - <PATTERN to="Page4">
        <![CDATA[ <img src="[^"]*" width="[^"]*" border="[^"]*">\s*\(\s*<a href="[^"]*">\s*
        ([\d]+)\s*</a>\s*<img align="[^"]*" border="[^"]*" height="[^"]*" width="[^"]*" alt="[^"]*"
      </PATTERN>
    </PAGE_TYPE>
    <PAGE_TYPE id="Page3" />
    <PAGE_TYPE id="Page4" />
  </COLLECTOR>
```

Figura 3.9: Plano de coleta das páginas

3.4 Coleta das Páginas

Após a geração do plano de coleta das páginas, a Web2DB gera um coletor que, a partir de um tipo de página inicial e da modelagem realizada, caminha pelas páginas, coletando-as e salvando-as em disco. A forma como é feita a coleta na Web2DB permite

que a extração dos dados da Web, foco principal do estudo, possa ser feita mesmo para os casos em que os atributos estejam difusos em várias páginas de um mesmo sítio eletrônico. Muitas soluções assumem os dados sempre em uma mesma página e situações como essa exigem um esforço maior na extração. Na Web2DB esse esforço é minimizado pelas etapas de modelagem do banco de dados e geração do plano de coleta.

A partir do plano de coleta é gerado um *wrapper* que usa os *hyperlinks* como atributos a serem extraídos. Além da regra de extração para os *hyperlinks* utilizados como exemplo, o *wrapper* gerado nesta etapa armazena também as informações do tipo de página que contém o *hyperlink* e o tipo de página que ele acessa. Isso permite fazer o caminhamento e armazenar as páginas extraídas nos devidos diretórios, para facilitar posteriormente a extração dos dados das páginas coletadas. O utilização de técnicas de extração de dados baseadas em exemplos para implementar a coleta de páginas permitiu a utilização da API da DESANA nesse momento, possibilitando o reaproveitamento das funções que ela implementa.

Para o exemplo da Figura 3.7 temos os seguintes tipos de página:

- Tipo 1: Lista de leilões;
- Tipo 2: Dados dos leilões;
- Tipo 3: Lista de lances;
- Tipo 4: Dados dos compradores;
- Tipo 5: Dados dos vendedores.

Figura 3.10: Exemplo de caminhamento entre as páginas

O relacionamento entre as páginas por meio dos *hyperlinks* permite que seja montada uma árvore para representar a estrutura do sítio eletrônico, ou pelo menos do grupo de páginas de interesse. Para o exemplo acima mencionado essa estrutura seria semelhante à da Figura 3.10. Assim, dado o tipo inicial de página e para cada página desse tipo salva em disco, o algoritmo de coleta de páginas aplica o extrator correspondente, obtendo os *hyperlinks* para caminhar para o próximo tipo de página (tipo 2). A cada novo tipo de página o processo é repetido, salvando a páginas que são coletadas em disco, utilizando o *wrapper* gerado a partir do plano de coleta para obter os *hyperlinks* a serem acessados. Quando se acessa o último nó da árvore, não há regra de extração definida, então o processo termina e, recursivamente, retorna ao tipo de página anterior, repetindo o processo para novos *hyperlinks* que ainda não tenham sido percorridos. A Figura 3.11 apresenta o algoritmo que realiza essa operação de coleta das páginas baseado nos *hyperlinks* selecionados como atributos a serem extraídos.

```
1   ExtracaoDePaginas(Dir: Diretorio Inicial; M: Plano de Coleta)
2           // nesta primeira etapa obtem-se os links da página inicial de coleta
3           // e percorre-se a lista desses links que é retornada
4           P = ObterPadraoDeExtracaoDoLinkInicial(M);
5           Para cada arquivo em Dir faça
6                   Lista Links = AplicaPadraoEExtraiOsLinks(P);
7                   Para cada elemento de Links faça
8                           Coleta(Links(i), TipoDePaginaAAcessar, M)
9                   fim para
10          fim para
11  fim
12
13  Coleta(Link: Hyperlink, T: TipoDePagina, M: ModelagemColeta)
14          // para cada link é obtido os tipos de páginas que ele acessa pelo
15          // plano de coleta (links que precisam ser coletados)
16          Lista L = ObtemListaDeTiposDePaginasQueAcessa(T, M);
17          AcessarHyperlink(Link);
18          ArmazenarPagina(T);
19          // são obtidos os links para cada tipo de página
20          Para cada elemento em L faça
21                  Lista P = ObterPadraoDeExtracaoDoLink(M);
22                  Para cada elemento em P faça
23                          Lista ProximosLinks = AplicaPadraoEExtraiOsLinks(P);
24                          Para cada elemento de ProximosLinks faça
25  // recursivamente avança no caminhamento até percorrer todas as páginas de um link
26  // inicial quando o processo é repetido até o fim da lista de links da página inicial
27                                  Coleta(ProximosLinks(i), L(i), M)
28                          fim para
29                  fim para
30          fim para
31  fim
```

Figura 3.11: Algoritmo de coleta das páginas

O coletor pode ser executado diretamente a partir da própria Web2DB que exibe um acompanhamento do processo de coleta das páginas com o *status* do processo para cada tipo de página. Também há a possibilidade de executar a coleta em um processo paralelo. No caso, é compilada uma classe Java em tempo de execução, que implementa o algoritmo da coleta e é executada em paralelo, podendo o usuário continuar utilizando a ferramenta durante esse período.

A Figura 3.12 mostra a tela em que é acionada a função de coleta de páginas. É necessário informar o diretório onde são localizadas as páginas do tipo inicial, onde o caminhamento começa e se será executado como um processo paralelo ou não.

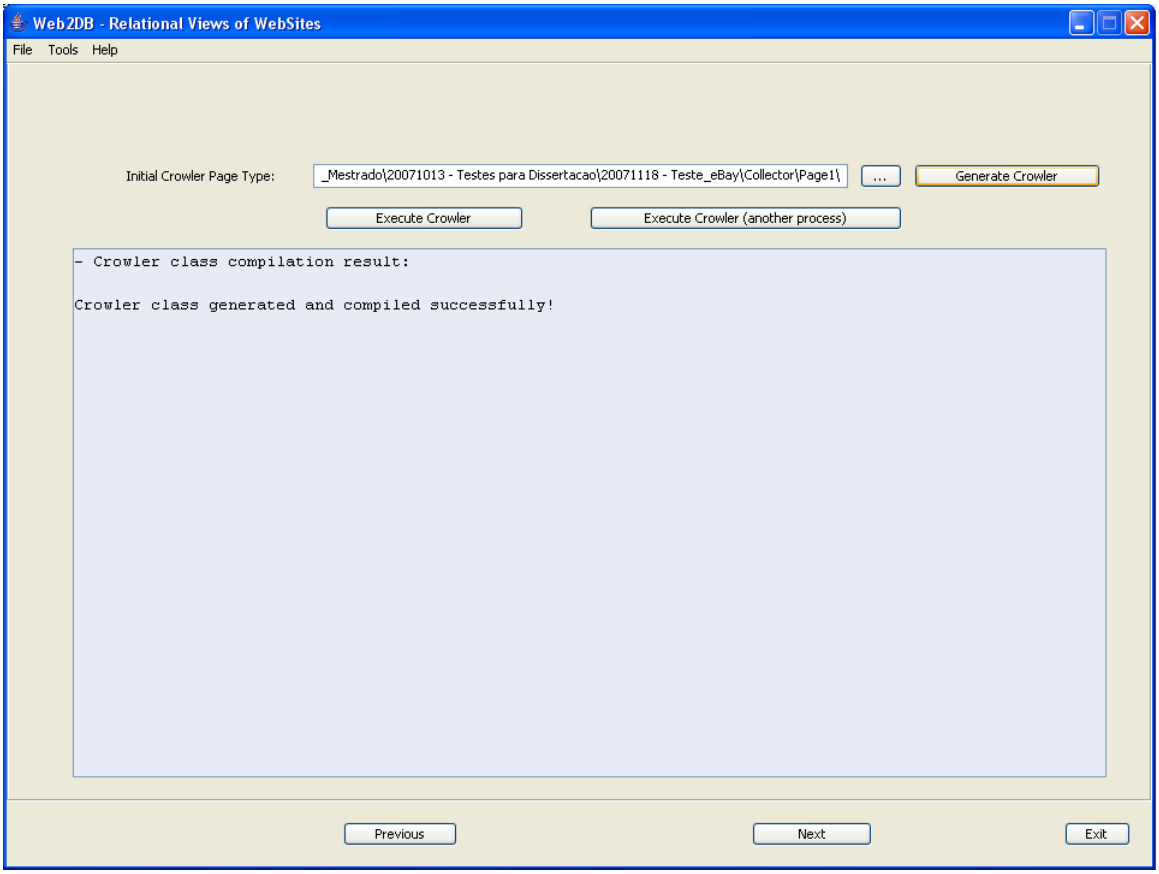

Figura 3.12: Web2DB - Agente de coleta de páginas

3.5 Mapeamento dos Dados a Serem Extraídos

Nesse momento, o usuário já tem todas as páginas com os dados de interesse, ou seja, a origem das informações. Resta então informar a localização de cada atributo do banco de dados nas respectivas páginas. Vale destacar que a Web2DB possibilita

a extração de dados localizados de múltiplos tipos de página que se inter-relacionam, com a possibilidade de coletar automaticamente essas páginas, como foi visto nas seções anteriores. O plano de coleta considera cada tipo de página separadamente, para que nessa etapa ocorra o mapeamento dos dados contidos em cada tipo de página para as tabelas do banco de dados.

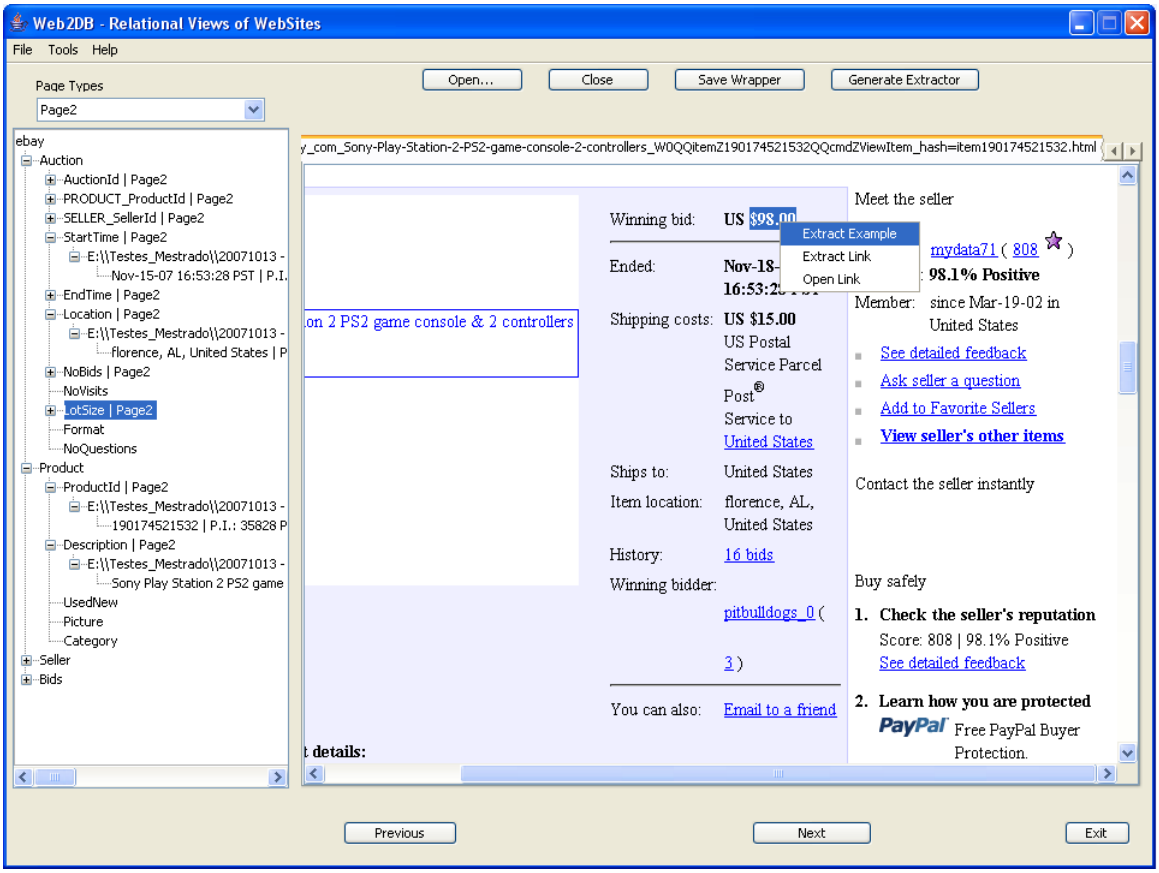

Figura 3.13: Web2DB - Mapeamento dos dados a serem extraídos

Existe uma premissa na Web2DB de que os atributos de uma tabela estão todos concentrados em um único tipo de página, mas pode-se ter várias tabelas extraídas de vários tipos distintos de página. É possível, como será mostrado a seguir, utilizar mecanismos na Web2DB para tratar os casos em que os atributos de uma tabela seja provenientes de mais de um tipo de página. A Figura 3.13 apresenta a tela da Web2DB onde é feito o mapeamento dos dados das páginas para os atributos do banco de dados.

Como pode ser observado, o usuário seleciona o tipo de página em questão, o atributo do banco de dados e o valor desse atributo na página. A página é aberta dentro da própria ferramenta, a partir do diretório em que foram salvas as páginas do tipo selecionado. Um menu de contexto permite atribuir ao valor selecionado um exemplo

para o atributo.

```
- <WRAPPER name="Mercado Livre Wrapper">
    <MODEL_FILE>E:\Testes_Mestrado\Mercado_Livre.xml</MODEL_FILE>
    <PageType>Page2</PageType>
  - <ATTRIBUTE id="Auction.PRODUCT_ProductId">
    - <FILE name="E:\\Testes_Mestrado\\Page3\\Pagina_Coletada1.xhtml">
        <SAMPLE rendstart="1662" rendend="1674" ipos="22978" fpos="22989">120125830658</SAMPLE>
        <SAMPLE rendstart="1662" rendend="1674" ipos="22978" fpos="22989">120125830658</SAMPLE>
      </FILE>
    - <FILE name="E:\\Testes_Mestrado\\Page3\\Pagina_Coletada2.xhtml">
        <SAMPLE rendstart="1681" rendend="1693" ipos="22977" fpos="22988">200113995839</SAMPLE>
        <SAMPLE rendstart="1681" rendend="1693" ipos="22977" fpos="22988">200113995839</SAMPLE>
      </FILE>
    </ATTRIBUTE>
  - <ATTRIBUTE id="Auction.Location">
    - <FILE name="E:\\Testes_Mestrado\\Page3\\Pagina_Coletada1.xhtml">
        <SAMPLE rendstart="2072" rendend="2126" ipos="29402" fpos="29455">Minas Gerais</SAMPLE>
        <SAMPLE rendstart="2072" rendend="2126" ipos="29402" fpos="29455">Minas Gerais</SAMPLE>
      </FILE>
    - <FILE name="E:\\Testes_Mestrado\\Page3\\Pagina_Coletada2.xhtml">
        <SAMPLE rendstart="2072" rendend="2102" ipos="28545" fpos="28573">Rio de Janeiro</SAMPLE>
        <SAMPLE rendstart="2072" rendend="2102" ipos="28545" fpos="28573">Rio de Janeiro</SAMPLE>
      </FILE>
    </ATTRIBUTE>
  - <ATTRIBUTE id="Auction.NoBids">
    - <FILE name="E:\\Testes_Mestrado\\Page3\\Pagina_Coletada1.xhtml">
        <SAMPLE rendstart="1867" rendend="1871" ipos="28399" fpos="28402">7</SAMPLE>
        <SAMPLE rendstart="1867" rendend="1871" ipos="28399" fpos="28402">7</SAMPLE>
      </FILE>
    - <FILE name="E:\\Testes_Mestrado\\Page3\\Pagina_Coletada2.xhtml">
        <SAMPLE rendstart="1847" rendend="1852" ipos="27344" fpos="27348">15</SAMPLE>
        <SAMPLE rendstart="1847" rendend="1852" ipos="27344" fpos="27348">15</SAMPLE>
      </FILE>
    </ATTRIBUTE>
    ...
</WRAPPER>
```

Figura 3.14: Web2DB - Documento XML com exemplos fornecidos para geração do *wrapper*

Quando um exemplo é criado para um atributo, a posição desse valor no arquivo HTML é armazenada para que, junto com os demais exemplos a serem fornecidos, possa ser gerado o *wrapper* correspondente. A Figura 3.14 mostra a saída gerada para os exemplos fornecidos e que serão utilizados para gerar o *wrapper* em seqüência. Esta saída se apresenta em um formato semelhante ao da DEByE (Laender *et al.*, 2002b). É gerado um *wrapper* para cada tipo de página selecionado. Assim, o documento XML contém a posição dos exemplos fornecidos nas páginas, para que possam ser geradas as expressões regulares correspondentes que serão usadas pelo *wrapper* para extração dos dados de todas as páginas coletadas.

O usuário pode a qualquer momento salvar o estágio parcial do *wrapper* e continuar o projeto do ponto em que parou, alterando os exemplos fornecidos, se for o caso. Nessa etapa é importante que o usuário verifique e conheça as varições da disposição de cada

atributo nas páginas. Assim, os exemplos devem ser fornecidos para abranger o maior número de situações possíveis. Por exemplo, se um atributo apresenta ao lado do seu valor algum ícone ilustrativo que aparece de acordo com uma determinada situação, os exemplos devem ser fornecidos para as duas situações. Isso evita que o extrator de dados obtenha apenas os dados das páginas para um tipo de situação, ignorando as demais. A amostra deve ser significativa para cobrir esse ponto e o volume de páginas usadas como fonte de dados para extração grande o suficiente para conter todas as situações possíveis da forma de apresentação de cada atributo nas páginas.

3.6 Extração dos Dados

Gerado o extrator, resta, portanto, extrair os dados das páginas coletadas segundo a modelagem realizada. Todo o processo de extração é feito automaticamente ao clicar no botão *Generate Extractor*. Esta ação executa o método da API da DESANA que encapsula o extrator. Esse extrator irá gerar as expressões regulares para a extração de dados em função dos exemplos fornecidos (ver Figura 3.14). Aindá há a separação por tipo de página nesse momento. Cada tipo de página possui um extrator com as respectivas regras de extração dos atributos. A regra é aplicada em cada atributo e os dados extraídos são armazenados em formato XML. Esses dados ainda não estão estruturados de maneira a formar as tuplas que serão inseridas no banco de dados (Figura 3.15).

Para concluir essa etapa, é executado o algoritmo *Hot Cycles* implementado pela biblioteca DESANA. Ele é executado para cada tipo de página cadastrada, identificando, no contexto dos dados extraídos e os tipos de objeto (classes) envolvidas. O objetivo nesse momento é agrupar os dados extraídos de modo a formar as tuplas que comporão as tabelas do banco de dados. Ao final do agrupamento as tuplas são unificadas em um documento XML que passa a conter todas as tuplas que serão posteriormente inseridas no banco de dados. A Figura 3.16 mostra o resultado final da extração de dados para um caso hipotético de sítios de leilões eletrônicos.

A Web2DB tem o objetivo de ser aplicada a sítios eletrônicos que apresentam um padrão uniforme de apresentação dos dados. São geralmente páginas geradas automaticamente e que possuem um banco de dados associado, mas que o acesso aos dados se dá apenas por meio dessas páginas. Como o algoritmo *Hot Cycles* necessita identificar um contexto para o agrupamento dos dados, se não houver um padrão nas páginas, a sua

execução, e até mesmo a extração em si, será falha. A Web2DB foi concebida para esse tipo de sítio eletrônico, hoje em dia muito comum.

```
- <ATOMS>
  - <ATTRIBUTE name="Auction.PRODUCT_ProductId" numOfAtt="25">
    - <VALUE source="E:\Testes_Mestrado\Page2\Pagina_Coletada1.xhtml" ipos="22978" fpos="22989">
        <![CDATA[ 120125830658 ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page2\Pagina_Coletada2.xhtml" ipos="23502" fpos="23513">
        <![CDATA[ 300115544250 ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page2\Pagina_Coletada3.xhtml" ipos="23496" fpos="23507">
        <![CDATA[ 300115544205 ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page2\Pagina_Coletada4.xhtml" ipos="22990" fpos="23001">
        <![CDATA[ 140122336065 ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page2\Pagina_Coletada5.xhtml" ipos="23279" fpos="23290">
        <![CDATA[ 180123585108 ]]>
      </VALUE>
      ...
  </ATTRIBUTE>
  - <ATTRIBUTE name="Auction.Location" numOfAtt="25">
    - <VALUE source="E:\Testes_Mestrado\Page3\Pagina_Coletada1.xhtml" ipos="29402" fpos="29455">
        <![CDATA[ Minas Gerais ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page3\Pagina_Coletada2.xhtml" ipos="30103" fpos="30135">
        <![CDATA[ Bahia ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page3\Pagina_Coletada3.xhtml" ipos="30079" fpos="30111">
        <![CDATA[ Rio de Janeiro ]]>
      </VALUE>
    - <VALUE source="E:\Testes_Mestrado\Page3\Pagina_Coletada4.xhtml" ipos="28571" fpos="28591">
        <![CDATA[ Para ]]>
      </VALUE>
      ...
  </ATTRIBUTE>
  ...
</ATOMS>
```

Figura 3.15: Web2DB - Resultado da extração de dados

É importante destacar a contribuição da biblioteca DESANA nessa etapa, pois ela fornece métodos eficientes para que os dados sejam extraídos e estruturados segundo a modelagem desejada, facilitando o agrupamento automático (por meio do algoritmo *Hot Cycles*) dos dados de interesse do usuário. Isso permitiu o uso da DESANA para atuar em um requisito importante da ferramenta Web2DB: permitir a extração de dados de múltiplos tipos de página. Esse requisito é a maior contribuição da ferramenta Web2DB, o usuário especifica a modelagem da estrutura das páginas, permitindo que a coleta das páginas e a extração dos dados sejam realizadas automaticamente, mesmo quando os dados não estejam em páginas de um mesmo tipo.

Os dados, que estão dispostos em documentos diferentes, são agrupados em tipos de entidade (classes) com a utilização dos métodos da DESANA. Isso permite ao usuário ter maior domínio sobre os dados extraídos e facilita a análise e inserção dos mesmos em

um banco de dados.

Com a conclusão dessa etapa, os dados extraídos, que antes só seriam acessados por meio de navegação entre as páginas, estão todos concentrados em um documento XML único. Resta agora inserir esses dados no banco de dados que foi modelado no início do processo. Trata-se da etapa final do processo de extração de dados, descrita a seguir.

Figura 3.16: Web2DB - Resultado final da extração de dados

3.7 Inserção de Dados no Banco de Dados

No momento em que se chega a essa etapa o usuário já está de posse de todos os dados extraídos em formato XML, agrupados em tuplas pelo algoritmo *Hot Cycles*. Resta agora inserir esses dados no banco de dados. A primeira ação a ser feita é confirmar os dados de acesso ao banco de dados, que foram previamente preenchidos, durante a etapa de modelagem do banco de dados.

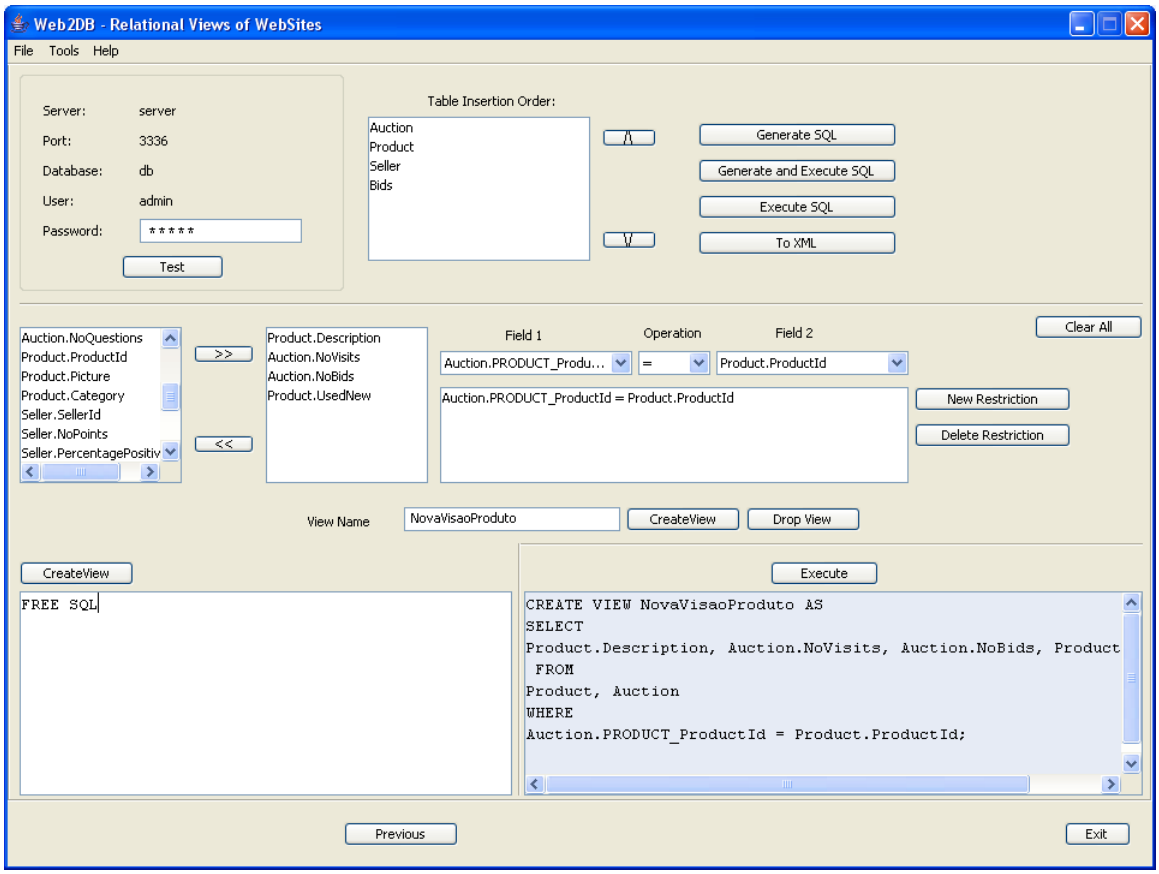

Figura 3.17: Web2DB - Inserção de dados no banco de dados

É realizada uma conversão do documento XML com os dados extraídos em uma seqüência de comandos da linguagem SQL, para a inserção das tuplas no banco de dados. Os tipos de objeto armazenados no documento XML correspondem às tabelas do banco de dados e os atributos às colunas dessas tabelas. A ferramenta Web2DB extrai os valores do documento XML e automaticamente gera o comando SQL de inserção das tuplas no banco de dados. Ao final, o usuário executa esses comandos por meio da ação de um botão na tela, quando a ferramenta se conecta ao banco de dados e insere os dados. Ainda é possível gerar um arquivo em formato XML de forma mais amigável do que a saída do

algoritmo *Hot Cycles*, que permite que um arquivo em formato XML possa ser usado também como repositório de dados.

A Figura 3.17 ilustra a tela em que essa etapa é realizada e a Figura 3.18 mostra um exemplo do documento XML gerado e do comando SQL para inserção dos dados no banco de dados. Essa etapa conclui o processo, alimentando e construindo um banco de dados que visa reproduzir aquele que gerou as páginas HTML do sítio eletrônico. Com os dados no banco de dados pode-se fazer análises que, a partir de acessos aos sítios diretamente, somente seriam possíveis a partir de um processo bastante trabalhoso e totalmente manual. Como se pode ver, depende-se da modelagem do banco de dados feita pelo usuário, mas uma vez feita essa modelagem, pode-se desenvolver algum sistema para tratar os dados extraídos e gerar relatórios para a análise que se objetiva fazer com esses dados.

```
- <DATA model="Mercado Livre">
  - <OBJECT name="Auction">
      <ATTRIBUTE name="PRODUCT_ProductId">120125830658</ATTRIBUTE>
      <ATTRIBUTE name="Location">Minas Gerais</ATTRIBUTE>
      <ATTRIBUTE name="NoBids">7</ATTRIBUTE>
    </OBJECT>
  </DATA>

INSERT INTO
Product(AuctionId, PRODUCT_ProductId, SELLER_SellerId, StartTime, EndTime,
Location, NoBids, LotSize, Format, NoQuestions)
VALUES
('120125830658', '120125830658', 'Seller021', '10-08-2007', '10-09-2007',
'Minas Gerais', 7, 54, 1, '', 11);
```

Figura 3.18: Comando SQL gerado para a inserção de dados

3.8 Criação de Visões

Ao final da etapa de inserção de dados o usuário possuriá o banco de dados populado com os dados extraídos das páginas. A ferramenta Web2DB permite, ainda, criar a partir do banco de dados visões, facilitando a geração de comandos na liguagem SQL para esse fim. Ou seja, a Web2DB agiliza e facilita a criação dessas visões (já que poderiam ser criadas diretamente a partir do banco de dados, sem a ferramenta), pois tem acesso a toda a configuração do esquema do banco de dados e de conexão com o mesmo.

Conforme pode ser visto na Figura 3.17, na parte inferior da tela é apresentada uma lista com todas as colunas de todas as tabelas do banco de dados. O usuário seleciona as colunas que farão parte da visão a ser criada. Em seguida é necessário informar as condições (restrições) que serão consideradas para gerar a visão. Nesse momento são realizadas operações no banco de dados como junções de tabelas e seleção de valores. Pode-

se dar um nome para a visão, para salvar o arquivo com o comando SQL correspondente, caso o usuário queira executá-lo posteriormente. Pode-se também executar os comandos diretamente da ferramenta ou usar o comando SQL gerado diretamente sobre o banco de dados. Para o caso de visões que demandem operações mais complexas, não suportadas pela interface gráfica da Web2DB, há um campo na tela que permite ao usuário executar consultas diretamente no banco de dados.

Por meio das facilidades providas para a criação de visões, é possível, inclusive, contornar uma restrição da ferramenta. Conforme dito anteriormente, a Web2DB possui uma restrição que diz respeito ao fato de que é preciso que todos os dados correspondentes aos atributos de uma tabela do banco de dados estejam disponíveis em um único tipo de página. É possível, pela modelagem, alimentar duas tabelas com dados de uma única página mas não o contrário. Para casos como esse, se os dados de dois tipos de página estiverem relaciondas a uma única tabela, basta considerar duas tabelas separadas e, posteriormente, criar uma visão que unifique as duas tabelas. Para isso, é necessário que as duas tabelas possuam um atributo em comum que possa ser usado para unificar as duas tabelas por meio de uma operação de junção. Dessa forma, o usuário visualizará no banco de dados seus dados como se fosse uma única tabela.

A Figura 3.19 mostra duas tabelas distintas e a tabela resultante da visão criada unificando parcialmente as suas tuplas. Note que ambas as tabelas possuem em comum o atributo *ProductId* que é chave primária da tabela *PRODUCT*.

No caso de duas tabelas estarem inseridas em um mesmo contexto, mas eventualmente apresentadas em tipos de páginas diferentes, o usuário deve ter atenção ao especificar uma chave primária comum às duas tabelas para que posteriormente possa ser criada uma visão que realize a junção das tabelas. Dessa forma, o usuário poderá manipular os dados de modo centralizado, tornando transparente o fato desses dados estarem armazenados em duas tabelas diferentes.

A geração de visões torna a ferramenta Web2DB ainda mais flexível, fazendo com que o processo de extração de dados possa ocorrer de forma bem simples, pois ao final pode-se criar uma estrutura para visualizar os dados que se criadas desde o início poderiam tornar o processo de extração e caminhamento (coleta) pelas páginas muito complexo. Assim, qualquer tipo de modelagem de banco de dados pode ser utilizada, pois é possível depois adaptar a estrutura dos dados para serem utilizados, por exemplo, em ambientes

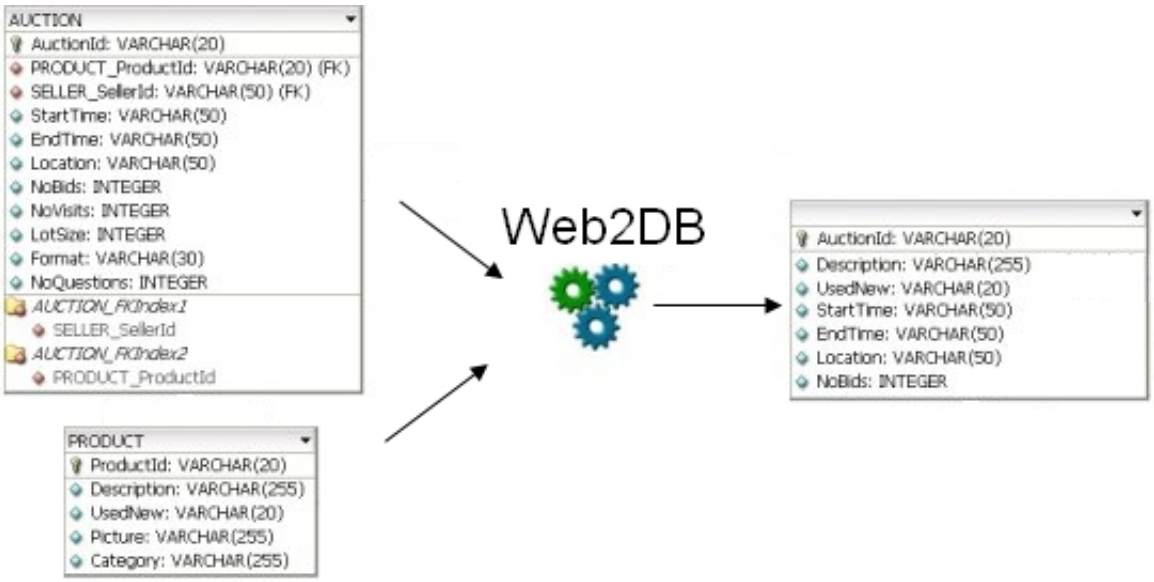

Figura 3.19: Web2DB - Exemplo de geração de uma visão

de armazéns de dados e para a geração de relatórios e gráficos de análise dos dados, sem que isso precise ser considerado no início do processo.

Capítulo 4

Avaliação da Ferramenta

Este capítulo tem como objetivo apresentar os resultados obtidos com a ferramenta Web2DB em aplicações distintas, de modo a validar o seu funcionamento e demonstrar sua eficácia. Numa análise aprofundada dos resultados obtidos, visa-se não somente verificar os pontos positivos da ferramenta mas também pontos de melhorias e limitações que poderiam ser tratados para aumentar o ganho na sua utlização.

As próximas seções descrevem as aplicações consideradas, os testes realizados e uma análise dos resultados obtidos.

4.1 Aplicações

A execução de testes práticos com a ferramenta Web2DB ocorreu em dois contextos distintos:

- Leilões eletrônicos: o uso da Web2DB para esse tipo de aplicação permite analisar o comportamento dos leilões e de seus usuários. Leilões eletrônicos são muito comuns hoje em dia, estando entre as principais formas de comércio eletrônico na Web. Fornecer subsídios para avaliação dessas aplicações possibilita a tomada de decisões mais adequadas, bem como a busca por melhores preços e negócios.

- Publicações científicas: o foco nesse contexto é a criação de um repositório de dados sobre artigos científicos de um determinado assunto.

A idéia central é mostrar que a ferramenta pode atuar em aplicações completamente diferentes, demonstrando a sua generalidade. Tratando dois casos distintos pode-se comprovar que a ferramenta é versátil, podendo ser inserida em qualquer contexto de sítios eletrônicos para extração de dados de suas páginas de forma automática e carregar

esses dados em um banco de dados para posterior análise. Além disso, a diversificação não só torna a ferramenta mais genérica como também auxilia na sua própria evolução, dado que cada sítio tem suas particularidades tecnológicas que a ferramenta deve estar apta a tratar, dentro das restrições que apresenta. Mais ainda, ao analisar os resultados obtidos pretende-se demonstrar a eficácia da ferramenta nas etapas de coleta de páginas e extração de dados.

4.2 Metodologia de Avaliação

A validação da ferramenta Web2DB seguiu uma metodologia que focou tanto na utilização da ferramenta quanto na avaliação da qualidade dos resultados gerados. A metodologia dos testes envolveu três etapas. A primeira delas foi a seleção de aplicações a serem utilizadas para validar a ferramenta. Feito isso, a segunda etapa consistiu em utilizar, para cada aplicação selecionada, todas as funções da ferramenta apresentadas no Capítulo 3. Nesse momento, um esquema do banco de dados da aplicação foi representado na Web2DB, os tipos de página de interesse foram modelados e a coleta das páginas realizada, com o conseqüente mapeamento e extração dos respectivos dados.

Por fim, foi feita a análise da qualidade da coleta das páginas e dos dados extraídos das mesmas. Para essa avaliação foram utilizadas medidas comuns no campo de recuperação de informação: precisão e revocação (Baeza-Yates and Ribeiro-Neto, 1999). Essas medidas permitem avaliar, quantitativamente, se os dados foram corretamente extraídos e se estes são relevantes dentro do domínio da aplicação desejada. Quanto à coleta das páginas, a precisão e revocação foi calculada para os tipos de página considerados. Já na extração dos dados, a precisão e a revocação foram calculadas para os atributos mais significativos no contexto. As tabelas com os dados de precisão e revocação são apresentadas para dar subsídio à avaliação que é feita dos resultados obtidos. Os pontos de divergência quanto ao resultado esperado são identificados objetivando uma melhoria e evolução da ferramenta Web2DB.

4.3 Resultados Obtidos

4.3.1 Sítio de Leilões Eletrônicos

Os leilões eletrônicos têm ocupado parcela significativa no comércio eletrônico na Web. A cada dia mais pessoas utilizam esse tipo de serviço. No entanto, com o volume cada vez maior de leilões e usuários, torna-se interessante a utilização de uma ferramenta para analisar a evolução dos leilões e auxiliar usuários na tomada de decisão sobre os melhores produtos e preços a serem negociados. A Web2DB auxilia na centralização desses dados de interesse.

O esquema do banco de dados utilizado para os testes realizados é apresentado na Figura 3.2. O teste realizado nesse contexto foi feito no sítio de leilões eletrônico ebay.com em novembro de 2007 e envolveu duas etapas. A primeira etapa se refere à obtenção dos leilões de uma determinada categoria de produtos (no caso *vídeo-games*) para análise diária das transações realizadas nas últimas 24 horas. É utilizada uma página com a lista de leilões, ordenados pelos mais próximos de término. Inicialmente a primeira página dessa lista é usada como exemplo para a geração do plano de coleta das páginas. Como essas listas são paginadas, o que se faz é utilizar o *hyperlink* que relaciona as várias páginas da lista de leilões como atributo para interrelacionar as páginas na coleta das mesmas. A Figura 4.1 ilustra a primeira etapa do processo do teste realizado.

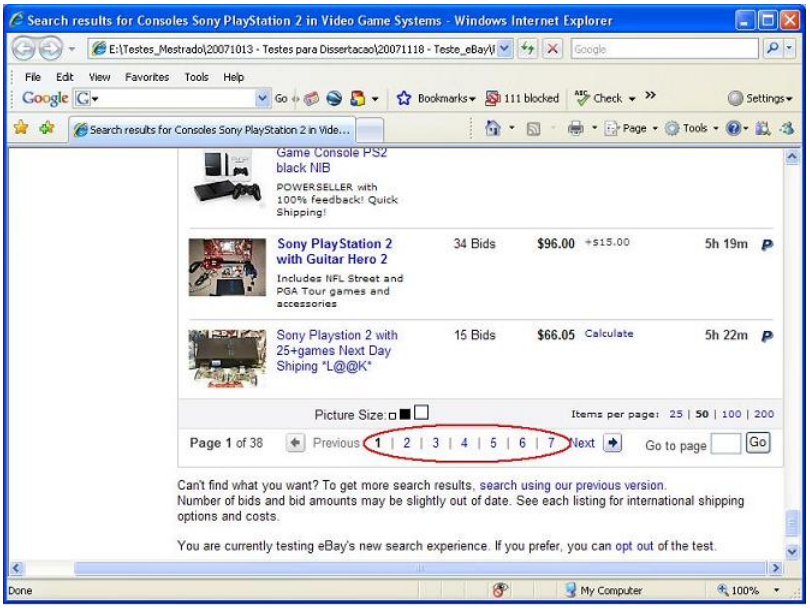

Figura 4.1: Obtenção da lista de leilões

Feita a coleta das páginas com as listas que disponibilizam acesso aos leilões que têm prazo de término menor que 24 horas, o processo é interrompido por 24 horas, período necessário para ocorrer o término dos leilões das páginas coletadas e a consolidação de todos os lances e do ganhador.

A Web2DB não possui um mecanismo de parada (interrupção) da coleta. Nesse caso, todas as páginas com os leilões atuais dessa categoria serão coletados. O usuário pode eliminar as páginas de excesso manualmente ou então criar uma visão ao final que exclua os leilões fora da data de interesse.

Passadas 24 horas, é realizada a segunda etapa. Inicialmente, é concluída a modelagem da coleta, desta vez, usando como entrada as páginas com a lista de leilões coletadas no dia anterior. A seguir, é modelado o caminhamento da lista para a página dos leilões e destas para as páginas de listagem dos lances e dos dados dos vendedores / compradores. Concluída essa modelagem, é feita uma nova coleta de páginas, a partir das quais os dados de interesse serão extraídos. Finalmente, de posse dos resultados da extração, são gerados os comandos SQL para inserção desses dados no banco de dados.

Essa aplicação é importante porque os leilões após finalizados não ficam disponíveis para acesso via navegação pelo sítio. Neste caso o *hyperlink* permanece acessível, mas não é intuitivo o acesso. Com a ferramenta, os dados dos leilões podem ser facilmente acessados e analisados mesmo após o seu término.

Os testes realizados consideraram um universo de páginas contendo:

- 9 páginas com lista de leilões;
- 450 páginas com dados dos leilões;
- 450 páginas com dados dos vendedores dos leilões;
- até 450 páginas com listas de lances efetuados e dados dos compradores (em alguns casos o leilão se encerra sem nenhum lance efetuado e não há limite para número de compradores que podem dar lance em um leilão).

Foram coletadas 85% das páginas que contém dados dos leilões e a partir destas foram coletadas todas as páginas de lista de lances e de dados de vendedores / compradores. O motivo da não-coleta da totalidade das páginas de leilões se deveu ao fato de que, durante a primeira etapa, o *hyperlink*, que é utilizado como atributo para definir o in-

terrelacionamento entre as páginas e é um dos parâmetros do plano de coleta das mesmas, estava definido como sendo a descrição do produto. Este, por sua vez, apresentava, em alguns casos, caracteres especiais que não são tratados pela ferramenta. Assim, os leilões cujas descrições apresentavam caracteres especiais não puderam ter suas páginas coletadas, uma vez que a ferramenta não conseguiu identificar o *hyperlink* dentro do padrão gerado para a extração.

A Tabela 4.1 apresenta o desempenho da ferramenta em relação às páginas coletadas, usando medidas comuns no campo de recuperação de informação: precisão e revocação (Baeza-Yates and Ribeiro-Neto, 1999). Como podemos ver, a precisão foi de 100% para todos os tipos de página e apenas para um tipo de página não se obteve 100% de revocação.

Tabela 4.1: Desempenho da Web2DB em um sítio eletrônico de leilões - coleta das páginas

Tipo de Pág.	Precisão	Revocação
Lista de Leilões	100%	100%
Dados dos Leilões	100%	85%
Dados dos Vendedores / Compradores	100%	100%
Lista de Lances	100%	100%

A partir da coleta realizada, os dados foram mapeados para um banco de dados semelhante ao já apresentado na Figura 3.2. Os dados foram extraídos e agrupados de acordo com os tipos de objeto reconhecidos para inserção no banco de dados via *script* de comandos SQL. A Tabela 4.2 apresenta a eficácia da ferramenta na etapa de extração dos dados, considerando os principais atributos envolvidos nessa extração. Foram consideradadas duas medidas de revocação: uma em função dos dados existentes nas páginas coletadas, consideradas como o universo de documentos existentes, e outra considerando o universo global de documentos existentes no sítio correspondente. Essa segunda medida é afetada pela revocação das páginas coletadas.

Do total de páginas relevantes coletadas, 97% dos dados de interesse foram extraídos corretamente. Além disso, tipos de objeto que foram mapeados em tabelas do banco de dados (PRODUCT, AUCTION, BID e SELLER) foram devidamente identificados, com os dados envolvidos corretamente agrupados em 100% dos casos, considerando os dados extraídos. O que se viu com esse teste foi o correto funcionamento da ferramenta, tanto na coleta quando na extração dos dados, já que quase a totalidade das páginas foi

Tabela 4.2: Desempenho da Web2DB em um sítio eletrônico de leilões - extração de dados

Atributos principais	Qtde Coleta	Extração	Precisão	Rev. Local	Rev. Global
Descrição do Produto	380	370	100%	97%	82,5%
Id do Leilão	380	370	100%	97%	82,5%
Data Término	380	370	100%	97%	82,5%
Localização	380	370	100%	97%	82,5%
Número de Lances	380	370	100%	97%	82,5%
Nome do Vendendor	380	369	100%	97%	82,5%

coletada, o mesmo ocorrendo com a extração dos dados das mesmas.

Dificuldades encontradas

Apesar dos resultados obtidos com a coleta das páginas e extração dos dados nos testes realizados, algumas dificuldades foram encontradas que evidenciam a necessidade de melhorias na ferramenta.

A primeira das dificuldades foi a performance apresentada pela ferramenta. Na etapa de extração, o algoritmo *Hot Cycles* da DESANA, mesmo executado em uma máquina com 1GB de memória RAM, apresentou problemas de performance e de consumo excessivo (e em alguns casos de *memory leak*) de memória para um volume grande de páginas e atributos (por exemplo, 400 páginas com cerca de 10 atributos em cada). Devido a isso, em alguns momentos a extração teve de ser executada em etapas, com grupos reduzidos de páginas coletadas a cada vez.

Outra questão importante foi a disposição dos dados nas páginas. Nos casos em que a página apresentava uma lista em que cada linha, juntamente com valores externos à lista (constantes na página), representava uma tupla a ser inserida no banco de dados, o algoritmo *Hot Cycles* não fez o correto agrupamento, reduzindo o volume de dados efetivamente inseridos no banco de dados.

4.3.2 Sítios de Publicações Científicas

Uma outra aplicação utilizada para validar a ferramenta Web2DB foi no contexto de sítios de publicações científicas. Foram selecionados para validação da ferramenta Web2DB os sítios dos seguintes periódicos: Computational & Applied Mathematics[1],

[1]http://www.scielo.br/scielo.php/script_sci_serial/pid_0101-8205/lng_en/nrm_iso

Journal of the Operational Research Society[2] e Empirical Software Engineering[3].

O objetivo foi a coleta dos dados dos artigos, em função do ano, mês e volume de publicação. Dados como nome dos autores, título, resumo e *hyperlink* para *download* do artigo foram mapeados em um banco de dados. Foi necessário definir cada sítio como um projeto separado na Web2DB, mesmo que o banco de dados destino dos dados seja o mesmo, pois cada um possui uma forma particular de interrelacionar suas páginas e agupar os conteúdos, que influencia diretamente na geração do plano de coleta.

Teste 1 - Computational & Applied Mathematics

O primeiro teste, feito no domínio do periódico Computational & Applied Mathematics, apresenta a seguinte disposição e volume das páginas a serem acessadas:

- Página com a lista de volumes do periódico;
- Páginas de cada volume contendo a lista de aritgos (total de 12 páginas);
- Páginas com a descrição dos artigos (total de 99 páginas).

Foi feita a modelagem da coleta em função dos tipos de página identificados acima com o objetivo de alcançar as páginas com os dados dos artigos (nome dos autores, título, resumo, volume, ano e mês de publicação, páginas, entre outros). A Tabela 4.3 apresenta o desempenho da ferramenta em relação às páginas coletadas.

Tabela 4.3: Desempenho da Web2DB no sítio eletrônico do periódico Computational & Applied Mathematics - coleta das páginas

Tipo de Página	Precisão	Revocação
Volume do periódico	100%	100%
Descrição dos Artigos	100%	92%

As 91 páginas coletadas com os dados dos artigos continham um total de 106 artigos. Desse total, o algoritmo *Hot Cycles* identificou 104 (98%). No entanto, em alguns casos os artigos não estavam completos (com todos os seus atributos). Isso se deveu ao fato de que alguns atributos não foram extraídos. A Tabela 4.4 mostra como foi o resultado da extração dos atributos em função das páginas efetivamente coletadas pela Web2DB.

[2]http://www.palgrave-journals.com/jors/archive/index.html?showyears=
[3]http://www.springerlink.com/content/100262/

Tabela 4.4: Desempenho da Web2DB no sítio eletrônico do periódico Computational & Applied Mathematics - extração de dados

Atributos principais	Qtde Coleta	Extração	Precisão	Rev. Local	Rev. Global
Título do Artigo	106	92	100%	87%	80%
Lista de Autores	106	89	100%	84%	77,3%
Título do periódico	106	102	100%	96%	88,3%

Pode-se ver que algumas páginas não foram coletadas, assim como alguns dos atributos. Uma análise desses casos mostra que a causa desse problema foi a não uniformidade de apresentação de algumas páginas. Em alguns casos a ausência de algum atributo que não foi destacado nos exemplos foi verificada e também algumas divergências no contexto da página (código HTML), que necessitaria do fornecimento de mais exemplos. O objetivo do teste foi verificar como seria o comportamento da ferramenta com poucos exemplos (no caso foram feitos três exemplos para cada atributo - número significativamente pequeno para o total de dados existentes). Um refinamento com o objetivo de concluir 100% de extração pode ser alcançado identificando-se os casos pontuais não extraídos e utilizando-os como exemplos em um novo processo de extração (realimentação).

Teste 2 - Journal of the Operational Research Society

O teste no sítio do periódico Journal of the Operational Research Society, apresenta a seguinte disposição das páginas a serem acessadas:

- 1 página com a lista de volumes;
- 237 páginas com os dados dos artigos de cada volume.

Do total de 238 páginas, conforme disposto acima, todas foram devidamente coletadas pela ferramenta (100% de revocação), embora algumas páginas adicionais não relevantes tenham sido coletadas, apresentando, portanto, uma precisão de 96.3%. Da mesma forma que o teste anterior, foram mapeados os dados dos artigos para a extração nas páginas coletadas. A Tabela 4.5 apresenta o desempenho da coleta das páginas.

A performance para este grupo de páginas foi semelhante à do teste anterior com ressalva à maior dificuldade do algoritmo *Hot Cycles* em agrupar corretamente as tuplas pela forma como está estruturado o código HTML da página. Para alguns atributos, como o nome dos autores, o algoritmo não identificou corretamente o contexto na página

Tabela 4.5: Desempenho da Web2DB no sítio eletrônico do periódico Journal of the Operational Research Society - coleta das páginas

Tipo de Página	Precisão	Revocação
Volume do periódico	100%	100%
Descrição dos Artigos	96,3%	100%

de forma a agrupar corretamente esses atributos. Isso ocasionou a perda de alguns dos dados de interesse no processo de extração. No entanto, essa questão é inerente à API da DESANA, de modo que devem ser verificados dois pontos: tratar esses casos de forma a evoluir a biblioteca permitindo a sua utilização mais amplamente ou então caracterizar a abrangência dos tipos de página que podem ser tratados com essa biblioteca. No caso, o que julgamos mais interessante seria usar esses casos de exceção para contribuir para a sua evolução e com isso permitir a utilização mais ampla da bilbioteca, que passaria a tratar um maior número de casos e com maior eficiência.

Um ponto interessante desse teste foi que, como o sítio desse periódico é mais antigo, as páginas apresentam muitas diferenças tecnológicas em sua construção. À medida que o tempo foi passando as páginas foram evoluindo. Entre o primeiro e o último ano a forma de apresentação dos dados diverge, o que dificultou a geração das expressões regulares. Um número maior de exemplos tiveram que ser fornecidos e a extração executada em etapas, visto que em alguns casos a extrutura HTML era completamente diferente entre um grupo de páginas e outro, ainda que sob o mesmo domínio. Por fim, outro fator relevante foi que para este sítio, a lista de autores não pôde ser extraída, pois a ferramenta não conseguiu indentificar o contexto desse atributo para geração da expressão regular responsável pela extração. Esse fatores fizeram com que o resultado da extração fosse inferior ao apresentado no teste anterior. Os testes neste sítio apresentaram também problemas com relação a performance, assim como os testes nos sítios de leilões eletrônicos. O motivo foi o mesmo, a existência de um volume grande de dados e páginas para extração dos dados. Devido a isso, o processo de extração neste caso também precisou ser feito em etapas.

A Tabela 4.6 apresenta os resultados dos dados extraídos para os principais atributos envolvidos. Nesse caso, como a revocação da coleta das páginas foi 100% não é apresenta a revocação acumulada, já que trata do mesmo valor. No entanto, como não

houve precisão de 100% será inserida nessa tabela um coluna destacando a precisão acumulada, ou seja, levando em conta o domínio real de documentos da aplicação na Web (e que na extração dos dados pode ser afetado pela precisão da coleta das páginas).

Tabela 4.6: Desempenho da Web2DB no sítio eletrônico do periódico Journal of the Operational Research Society - extração de dados

Atributos principais	Qtde Coleta	Extração	Precisão	Rev. Local	Rev. Global
Título do Artigo	3600	2209	100%	96,3%	61%
Lista de Autores	3600	0	0%	0%	0%
Título do periódico	3600	2209	100%	96,3%	61%

Teste 3 - Empirical Software Engineering

O teste feito no sítio do periódico Empirical Software Engineering apresenta a seguinte disposição das páginas a serem acessadas:

- 1 página com a lista de volumes;
- 27 páginas com a lista de artigos de cada volume;
- 158 páginas com a descrição dos artigos.

A Tabela 4.7 mostra os resultados obtidos para a coleta das páginas neste sítio.

Tabela 4.7: Desempenho da Web2DB no sítio eletrônico do periódico Empirical Software Engineering - coleta das páginas

Tipo de Pág.	Precisão	Revocação
Lista de Artigos	100%	100%
Descrição dos Artigos	100%	98,1%

Para este sítio eletrônico a coleta obteve um desempenho positivo, apresentando um percentual relativamente pequeno de páginas não coletadas, mas com precisão de 100% na coleta de todos os tipos de página. A Tabela 4.8 apresenta o desempenho da extração dos dados para os atributos de maior interesse envolvidos.

Além do percentual de dados que não foram extraídos (12%), cerca de 40% não foram corretamente agrupados nos objetos mapeados no banco de dados e apareceram de forma fragmentada na saída do algoritmo de extração dos dados, o que dificultou a inserção dos dados no banco de dados.

Tabela 4.8: Desempenho da Web2DB no sítio eletrônico do periódico Empirical Software Engineering - extração de dados

Atributos principais	Qtde Coleta	Extração	Precisão	Rev. Local	Rev. Global
Título do Artigo	155	136	100%	88%	86,3%
Lista de Autores	155	136	100%	88%	86,3%
Título do periódico	155	136	100%	88%	86,3%

Apesar das dificuldades encontradas e pontos de melhoria identificados para a ferramenta Web2DB, os testes realizados verificaram uma importante aplicação para a Web2DB: permitir que sítios completamente diferentes possam ser usados como fonte de dados para um mesmo destino, já que o banco de dados com os dados dos artigos pode ser o mesmo, se considerados os mesmos atributos. Isso viabiliza a centralização das informações que estão não somente difusas em um domínio, mas apresentadas em vários domínios distintos.

Além disso, vimos que a Web2DB funciona bem em contextos mais controlados, devido à questão dos exemplos. Sítios sem padronização na apresentação das informações comprometem a qualidade da coleta das páginas e da extração dos dados.

4.4 Análise Consolidada

Os testes feitos foram extremamente positivos para evoluir a ferramenta no sentido de torná-la genérica, pois a cada teste surgia um novo desafio que era resultado de alguma especialização da ferramenta que precisava ser eliminada.

A partir das aplicações consideradas para a validação da ferramenta Web2DB, obteve-se um resultado significativo, pois nos dois casos mostrou-se ser possível automatizar o processo de modelagem, coleta de páginas e extração dos dados de sítios eletrônicos de forma genérica.

Além dos problemas encontrados e destacados anteriormente, vale ainda destacar dois pontos importantes na definição do uso da Web2DB para uma determinada aplicação. O primeiro deles é a questão tecnológica, pois a extração dos dados (e com isso a coleta das páginas) é feita a partir dos exemplos fornecidos pelo usuário com a análise do código HTML para encontrar o padrão da extração. Em vista disso, é fundamental que as páginas do sítio considerado sejam padronizadas, sem uso de *scripts*, de modo que o seu conteúdo

seja apresentado utilizando-se marcadores HTML padronizados.

O outro ponto se refere à necessidade de se fornecer um número suficiente de exemplos, que é um ponto chave para a extração dos dados. Como já dito antes, essa característica da ferramenta traz ganhos no processo de extração mas requer que o usuário conheça bem o domínio da sua aplicação, pois pequenas diferenças na apresentação dos atributos demanda o fornecimento de exemplos suficientes para abranger todos os casos. Por exemplo, uma lista de lances apresenta o nome da pessoa seguida de sua pontuação no sítio. Em alguns casos o nome é seguido de figuras que indicam a qualidade do comprador. Como essa figura altera a estrutura HTML, o usuário tem que fornecer os dois exemplos sob risco de extrair os dados apenas em uma das condições.

Além de coletar as páginas, conforme o plano de coleta modelado pelo usuário, a Web2DB fez uso do algoritmo *Hot Cycles* para automaticamente agrupar os dados extraídos, que originalmente estão dispostos em arquivos XML de difícil compreensão. Esses dados são então inseridos em um banco de dados, onde podem ser mais facilmente tratados.

Os testes com sítios de leilões eletrônicos apresentou um bom desempenho no que diz respeito à qualidade da coleta e da extração dos dados a serem inseridos no banco de dados. Em contrapartida, mostrou que a ferramenta apresenta uma performance ruim quando o volume de páginas e de dados a serem analisados aumenta consideravelmente. Isso fez com que, nesse caso, a extração fosse feita em etapas, pois com todos os dados de uma só vez o processo não era concluído. Além disso, percebeu-se que a ferramenta não estava tratando casos em que os dados eram apresentados em forma de lista, onde cada linha da lista representasse uma tupla do banco de dados e alguns atributos dessas tuplas eram exibidos na página fora dessa lista. A forma de agrupamento dos dados em casos como esse é mais complexa e diferente, e precisa ser trata pela ferramenta para obter resultados ainda melhores.

Já os testes com os sítios de publicações científicas evidenciaram a utilidade da ferramenta Web2DB para coletar as páginas e extrair os dados de sítios eletrônicos completamente distintos, mas que envolvem os mesmos tipos de dados, permitindo centralizá-los em um único banco de dados para análise. Apesar disso, o desempenho da ferramenta Web2DB se demonstrou inferior do que nos testes com os sítios de leilões eletrônicos. Os sítios nesses casos eram, de uma manera geral, menos padronizados e dificultaram o pro-

cesso de extração e identificação dos objetos envolvidos e em alguns casos o fornecimento de um número maior de exemplos não foi suficiente para melhora dos resultados.

Com relação à performance, o trabalho não focou em uma análise detalhada das causas dos problemas de performance para a extração de dados e execução do algoritmo *hot cycles* envolvendo um número grande de atributos e páginas. O algoritmo *hot cycles* tem complexidade linear, o que contribui para a ocorrência dessas situações, mas apenas uma análise aprofundada da questão poderá identificar limitações e/ou pontos de melhoria para correção deste problema. Assim, essa análise pode ser feita posteriormente no sentido de evoluir a ferramenta Web2DB.

Todos os testes realizados foram focados na avaliação dos resultados obtidos na coleta das páginas e na extração dos atributos, como forma de avaliar se o método proposto com a ferramenta Web2DB apresenta resultados de qualidade no que diz respeito à precisão e revocação dos dados. No entanto, vale destacar ainda que é válido realizar testes com usuários potenciais da ferramenta Web2DB. Esse tipo de teste pode ser feito posteriormente. Participando ativamente de todas as etapas do processo, um usuário potencial da ferramenta pode avaliar a usabilidade dela em uma aplicação prática, por exemplo. Assim, fecha-se o ciclo, pois é analisada a utilidade da ferramenta em um contexto prático e a eficácia quanto aos resultados obtidos, confome apresentado nesse capítulo.

Os testes, como já dito, permitiram determinar a eficácia da ferramenta Web2DB, além de levantar pontos de melhoria que precisam ser tratados futuramente para que ela tenha uma aproveitamento ainda maior. No entanto, os testes realizados permitiram verificar o cumprimento dos objetivos iniciais determinados na etapa de concepção da ferramenta Web2DB.

Capítulo 5

Conclusões

5.1 Revisão do Trabalho

No trabalho realizado propusemos e desenvolvemos uma ferramenta de coleta de páginas da Web, extração de dados dessas páginas e carregamento de um banco de dados com os dados extraídos. Todo esse processo é feito da forma mais automática possível, ao mesmo tempo que torna a interferência do usuário um aspecto importante, já que este, ao invés de atuar em atividades que pouco agregam, passa a atuar modelando o processo e usando o seu conhecimento e experiência da aplicação em questão. Todo o restante das atividades fica automatizada, reduzindo os esforços para a extração de dados da Web.

Iniciamos o projeto estudando as ferramentas e técnicas existentes para extração de dados da Web. Diante da análise feita e dos resultados obtidos decidiu-se por usar a API da ferramenta DESANA (Sá Júnior *et al.*, 2006) como biblioteca extratora dos dados. Foram definidos os objetivos a serem alcançados com a ferramenta, que orientaram a implementação realizada. Decidiu-se em dar ao usuário o papel chave no processo, fazendo a modelagem do repositório de dados, da coleta das páginas e da extração dos dados. Assim, ele é envolvido no que agrega mais valor: usar o seu conhecimento da aplicação a ser considerada. As tarefas decorrentes disso foram automatizadas.

O que se pode ver é que os objetivos inicialmente traçados para o projeto foram atingidos. Implementamos a ferramenta Web2DB com uma interface amigável ao usuário, orientada em etapas e com vários recursos de usabilidade. Concluído o desenvolvimento, a ferramenta foi posta à prova em vários contextos de sítios da Web e os resultados obtidos permitiram verificar que a ferramenta sistematizou um processo que em muitos casos, além de dispender muito tempo, é muito propício a erros, de modo que, se esses erros não

forem minimizados, a análise dos dados coletados perde valor. A Web2DB é genérica, ou seja, pode ser aplicada em vários contextos distintos.

Os resultados mostraram boa precisão e revocação da ferramenta Web2DB nas tarefas de coleta de páginas e extração de dados. Algumas limitações da ferramenta justificaram o fato de alguns dos valores de precisão e revocação estarem abaixo de 100% embora ainda em níveis razoáveis. É importante destacar que os testes realizados permitiram além de validar o funcionamento da ferramenta, comprovar também a sua eficácia. Os resultados mostraram a facilidade de se efetuar toda a modelagem para a extração dos dados e a criação de visões para facilitar a visualização posterior dos dados extraídos e exportados para um banco de dados. Além disso, permitiu também identificar uma aplicação para a ferramenta Web2DB muito útil no contexto de análise e extração de dados da Web: permitir que sítios completamente diferentes possam ser usados como fonte de dados para um mesmo destino, já que pode-se usar um mesmo banco de dados quando os dados envolvidos são os mesmos (mesmos atributos). Isso garante a centralização dos dados que estão não somente difusos em um domínio, mas em vários domínios distintos, mas que podem ser analisados em um único contexto.

Mas o que se pode concluir é que a ferramenta Web2DB ainda apresenta algumas limitações e requer ajustes que permitam a sua evolução e amadurecimento. Os resultados obtidos aqui validaram a relevância da ferramenta We2DB e o processo que ela se propõe a sistematizar. No entanto, alguns pontos de melhorias foram detectados a partir das dificuldades encontradas nos testes realizados. Os ajustes desses pontos visam aumentar a qualidade dos resultados obtidos com a execução da ferramenta em situações práticas reais. Vimos que ela funciona bem em vários casos diferentes, mas algumas limitações precisam ser eliminadas para que a ferramenta possa acompanhar a volatilidade, a dinâmica atual da tecnologia dos sítios eletrônicos da Web e as demandas por informação de qualidade. A próxima seção discute os pontos mais importantes que demandarão trabalhos futuros, no sentido de evoluir e amadurecer a ferramenta desenvolvida.

5.2 Trabalhos Futuros

O trabalho apresentado nesta dissertação teve seus objetivos alcançados considerando o escopo desejado inicialmente para o projeto. No entanto, muitas oportunidades surgem deste trabalho inicial realizado, pontos de evolução da ferramenta que podem

originar trabalhos futuros com o objetivo de agregar ainda mais valor à Web2DB.

O primeiro ponto é a questão da coleta. É uma etapa importante e que hoje é feita de forma mais generalizada e simples. É possível tornar a modelagem da coleta mais dinâmica e com isso abranger um número maior de aplicações. A modelagem poderia, por exemplo, permitir a adição de condições de parada da coleta. Por exemplo: o usuário pode ter interesse em avaliar, em uma determinada categoria de leilões, aqueles que apresentam valor maior do que R$ 1.000,00 ou que iniciaram em um determinado período; ou ainda, que possuem um limite inferior de número de lances.

Outro ponto diz respeito à extração dos dados. Pode haver a possibilidade de o usuário determinar regras para a inserção dos dados. Atualmente todos os dados encontrados nas páginas coletadas, em função dos exemplos fornecidos, são extraídos incondicionalmente. Com isso os dados poderiam, por exemplo, ser transformados antes da inserção (converter um formato de data, concatenar *strings*, aplicar fórmulas, etc.). Isso permitiria à ferramenta Web2DB ser usada como um ferramenta de ETL (Sweiger *et al.*, 2002). As ferramentas de ETL (*Extraction, Transformation and Loading*) são muito demandadas hoje em dia no contexto de integração de sistemas. Elas promovem a extração dos dados (de vários repositórios de origem), a transformação de acordo com regras definidas pelo usuário e a carga em um banco de dados de destino (Correa, 2004). No caso, a Web2DB estaria realizando o processo ETL considerando como fonte de dados de origem os sítios eletrônicos da Web, promovendo uma integração da Web com outros sistemas.

Tanto na coleta quanto na extração é possível exibir ao usuário estágios intermediários para que ele possa validar a qualidade dos exemplos fornecidos. Atualmente o usuário só tem como validar isso ao final do processo, o que em alguns casos pode gerar retrabalho. É importante essa interação, pois em alguns casos é necessário fornecer mais exemplos e uma visualização intermediária (por atributo, por exemplo) pode auxiliar a definir a etapa de extração com maior rapidez e assertividade.

Além desses itens de ajustes na ferramenta Web2DB, um ponto dos mais importantes diz respeito ao aspecto tecnológico da ferramenta. A ferramenta é dependente do código HTML para a definição das regras de extração a partir dos exemplos fornecidos. No entanto é intensa a evolução da arquitetura das páginas Web e alguns detalhes como o uso de *scripts* por exemplo dificultam a extração. Em alguns casos dois atributos estão em contextos diferentes mas podem ser coletados como um único atributo. Seria interessante

realizar uma análise na API da DESANA e verificar os pontos onde o código pode ser ajustado para que um maior número de tipos de página possam ser tratadas pela API, conforme discutido na análise dos resultados obtidos com os testes realizados.

Pode ser estudada alguma forma de otimização para melhora da performance, pois com um grande volume de páginas a ferramenta consome muita memória e tempo de processamento. Uma estratégia que poderia ser adotada é distribuir a tarefa de coleta das páginas e de extração dos dados. Por exemplo, para cada tipo de página a ser coletada ou ter dados extraídos poderia-se criar uma *thread* ou um processo distribuído para a execução dessas tarefas. Ao final, um processo centralizaria os resultados da coleta / extração.

Todas essas propostas dão subsídio para a continuidade deste trabalho de tranformação da Web2DB em uma ferramenta integradora completa, com diversas funções que permitam extrair os dados e carregá-los no banco de dados destino (que inclusive poderia ser mais de um) de forma automática, ampliando o espaço de aplicações que requeiram a extração de dados da Web.

Referências Bibliográficas

Abiteboul, S., Buneman, P., and Suciu, D. (2000). *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann.

Adelberg, B. (1998). NoDoSE: A tool for semi-automatically extracting structured and semistructured data from text documents. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 283–294, Seattle, WA, USA.

Arantes, A. R., Laender, A. H. F., Golgher, P. B., and Silva, A. S. (2001). Managing Web Data through Views. In *Proceedings of the Second International Conference on Electronic Commerce and Web Technologies*, pages 154–165, Munique, Alemanha.

Ashraf, F. and Alhajj, R. (2007). ClusTex: Information Extraction from HTML Pages. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pages 355–360, Niagara Falls, ON, Canadá.

Baeza-Yates, R. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. Addison-Wesley, Harlow, England.

Bakos, J. Y. (1997). Reducing Buyer Search Cost: Implications for Eletronic Marktplaces. *Management Science*, **43**(12), 1676–1962.

Bapna, R., Goes, P., and Gupta, A. (2000). A Theoretical and Empirical Investigation of Multi-Item On-Line Auctions. *Information Technology and Management*, **1**(1), 1–23.

Bapna, R., Goes, P., and Gupta, A. (2001). Insights and Analyses of Online Auctions. *Communications of the ACM*, **44**(11), 42–50.

Bapna, R., Goes, P., Gopal, R., and Marsden, J. R. (2004). Moving from Data-Constrained to Data-Enabled Research: The CIDRIS Experience in Collecting, Valida-

ting and Anlyzing Large Scale E-Commerce Data. Technical report, Dept. of Operations and Information Management, UConn School of Business, Connecticut, U.S.A.

Baumgartner, R., Flesca, S., and Gottlob, G. (2001). Visual Web Information Extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 119–128, Roma, Itália.

Califf, M. E. and Mooney, R. J. (1999). Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of the Sixteenth National Conference on Artificial Inteligence and Eleventh Conference on Innovative Applications of Artificial Inteligence*, pages 328–334, Orlando, FL, USA.

Correa, M. D. (2004). Mapping-Tool: Desenvolvimento de solução ETL (Extraction, Transformation and Loading) para a integração de sistemas de informação. Monografia de Conclusão de Graduação (Curso de Engenharia de Controle e Automação), Universidade Federal de Minas Gerais.

Crescenzi, V., Mecca, G., and Merialdo, P. (2001). RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of 27th International Conference on Very Large Data Bases*, pages 109–118, Roma, Itália.

Elmasri, R. and Navathe, S. (2002). *Sistema de Banco de Dados - Fundamentos e Aplicações*. LTC, Rio de Janeiro.

Embley, D. W., Campbell, C. M., Jiang, Y. S., and Liddle, S. W. (1999). Conceptual-model-based Data Extraction from Multiple-record Web Pages. *Data Knowledge Engineering*, **31**(3), 227–251.

Golgher, P. B., Silva, A. S., Laender, A. H. F., and Ribeiro-Neto, B. A. (2001). Bootstrapping for Example-Based Data Extraction. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 371–378, Atlanta, GA, USA.

Hammer, J. and Garcia-Molina, J. M. H. (1997). Semistructured Data: The Tsimmis Experience. In *Proceedings of the First East-European Symposium on Advances in Databases and Information Systems*, pages 1–8, St. Petersburg, Rússia.

Inmon, W. H. (1996). *Building the Data Warehouse*. John Wiley, New York.

Knoblock, C. A., Lerman, K., Minton, S., and Muslea, I. (2000). Accurately and Reliably Extracting Data from the Web: A Machine Learning Approach. *ACM SIGMOD Record*, **23**(4), 33–41.

Kushmerick, N. (2000). Wrapper Induction: Efficiency and expressiviness. *Aritficial Inteligence Journal*, **118**(1-2), 15–68.

Laender, A. H. F., Ribeiro-Neto, B. A., Silva, A. S., and Silva, E. S. (2000). Representing Web Data as Complex Objects. In *Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, pages 216–228, Londres, Inglaterra.

Laender, A. H. F., Ribeiro-Neto, B. A., Silva, A. S., and Teixeira, J. S. (2002a). A Brief Survey of Web Data Extraction Tools. *ACM SIGMOD Record*, **31**(2), 84–93.

Laender, A. H. F., Ribeiro-Neto, B. A., and Silva, A. S. (2002b). DEByE - Data Extraction By Example. *Data Knowledge Engineering*, **40**(2), 121–154.

Li, Y. (2007). The XML-based Information Extraction on Data-intensive Page. In *Proceedings of the Network and Parallel Computing Workshops*, pages 1027–1030, Los Alamitos, CA, USA.

Liu, L., Pu, C., and Han, W. (2000). XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources. In *Proceedings of the 16th International Conference on Data Engineering*, pages 611–621, Washington, DC, USA.

Liu, Z., Ng, W. K., Lim, E., and Li, F. (2004). Towards building logical views of websites. *Data Knowledge Engineering.*, **49**(2), 197–222.

Mansuri, I. R. and Sarawagi, S. (2006). Integrating Unstructured Data into Relational Databases. In *Proceedings of the 22nd International Conference on Data Engineering*, page 29, Atlanta, GA, USA.

Muslea, I., Mintom, S., and Knoblock, C. (2001). Hierarchical wrapper indiction for semistructured information sources. *Autonomous Agents and Multi-Agent Sustems*, **4**(1-2), 93–114.

Nie, Z., Ma, Y., Shi, S., Wen, J., and Ma, W. (2007). Web object retrieval. In *Proceedings of the 16th International Conference on World Wide Web*, pages 81–90, Baff, Alberta, Canadá.

Reis, D. C., Golgher, P. B., Laender, A. H. F., and Silva, A. S. (2004). Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International Conference on World Wide Web*, pages 502–511, New York, NY, USA.

Ribeiro-Neto, B. A., Laender, A. H. F., and Silva, A. S. (1999). Extracting Semi-Structured Data Through Examples. In *Proceeding of the Eighth International Conference on Information and Knowledge Management*, pages 94–101, Kansas City, MO, USA.

Sá Júnior, S. A. L. F., Oliveira, D. P., and Silva, A. S. (2006). Uma ferramenta para extração de dados da Web considerando contextos fracos. In *Simpósio Brasileiro de Banco de Dados / Sessão de Demos*, pages 25–30, Florianópolis, SC, Brasil.

Sahuguet, A. and Azavant, F. (2001). Building intelligent web applications using lightweight wrappers. *Data Knowledge Engineering*, **36**(3), 283–316.

Silva, A. S. (2002). Estratégias Baseadas em Exemplos para Extração de Dados Semi-Estruturados da Web. Tese de Doutorado, Dept. de Ciência da Computação da Universidade Federal de Minas Gerais.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, **34**(1-3), 233–272.

Sweiger, M., Madsen, M., Langston, J., and Lombard, H. (2002). *Clickstream Data Warehousing*. John Wiley.

Wang, J. and Lochovsky, F. H. (2003). Data extraction and label assignment for web databases. In *Proceedings of the 12th international conference on World Wide Web*, pages 187–196, Budapeste, Hungria. ACM Press.

Westerveld, T., Kraaij, W., and Hiemstra, D. (2001). Retrieving Web Pages using Content, Links, URLs and Anchors. In *Notebook of 10th Text Retrieval Conference* , pages 663–672, Gaithersburg, MD, USA.

Wood, C. A. and Ow, T. T. (2005). WEBVIEW: an SQL extension for joining corporate data to data derived from the web. *Communications of the ACM*, **48**(9), 99–104.

Zhai, C., Chang, K. C., and Chuang, S. (2007). Collaborative Wrapping: A Turbo Framework for Web Data Extraction. In *Proceedings of the IEEE 23rd International Conference on Data Engineering*, pages 1261–1262, Istambul, Turquia.

Zhai, Y. and Liu, B. (2005). Web data extraction based on partial tree alignment. In *Proceedings of the 14th International Conference on World Wide Web*, pages 76–85, Chiba, Japão.