

Using Open Web APIs in Teaching Web Mining

HSINCHUN CHEN¹, XIN LI¹, MICHAEL CHAU², YI-JEN HO¹, CHUNJU TSENG¹

¹ The University of Arizona

² The University of Hong Kong

With the advent of the World Wide Web, many business applications that utilize data mining and text mining techniques to extract useful business information on the Web have emerged from Web searching to Web mining. It is important for students to acquire knowledge and hands-on experience in Web mining during their education in computer science or information systems curricula. In this paper, we report our experience in using open Web APIs that have been made available by major Internet companies (e.g., Google, Amazon, and eBay) to teach Web mining in classrooms. After reviewing related background in Web mining, open Web APIs, and Web mining education, we describe the details of the class project and provide sample codes and implementation. Four sample Web mining systems developed by students using selected open Web APIs are also presented. Overall, the class project achieved its objectives and students acquired valuable experience in leveraging the power of the APIs to build important and interesting Web mining applications.

Categories and Subject Descriptors: D.2 [Software Engineering]; H.3.1 [Content Analysis and Indexing]; H.3.3 [Information Search]; H.3.5 [Online Information Services]; H.5.4 [Hypertext/Hypermedia]; K.3.2 [Computer and Information Science Education].

General Terms: Design

Additional Key Words and Phrases: Web mining, Web computing, education, open Web APIs

1. INTRODUCTION

The World Wide Web has become an indispensable part of many business organizations. A large number of companies have used the Web as a source of knowledge, a door to their customers, a medium for advertisements and marketing, and a vehicle for reengineering their business processes. In order to effectively utilize the power of the Web, information technology (IT) professionals need to have sufficient knowledge and experience in various Web technologies and applications. Traditional education in computer science and information systems curricula may address these areas to some extent, but is definitely not sufficient. In recent years, new courses in Internet- and Web-related topics have been offered in many universities around the world to better equip students with such knowledge. These include basic courses such as Internet networking, Internet application development, Web search engines, and so forth. More advanced

This study was supported in part by NSF National Science Digital Library Program; "Intelligent Collection Services for and about Educators and Students: Logging, Spidering, Analysis and Visualization," DUE-0121741.

Authors' addresses: H. Chen, X. Li, Y. Ho, C. Tseng, Department of Management Information Systems The University of Arizona, Tucson, Arizona 85721, Email: xinli@email.arizona.edu; M. Chau, School of Business The University of Hong Kong, Pokfulam, Hong Kong.

Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 2001 ACM 1073-0516/01/0300-0034 \$5.00

topics, such as Web mining, are also becoming increasingly important. Web mining, defined as the use of data mining, text mining, and information retrieval techniques to extract useful patterns and knowledge from the Web [Etzioni, 1996; Chen & Chau, 2004], has been frequently used in real world applications, such as business intelligence [Chau et al., 2002], Website design [Fang et al., 2006], and customer opinion analysis [Liu et al., 2005]. It is imperative for students to acquire knowledge and hands-on experience in Web mining applications.

However, building a Web mining application or a Web services application from scratch is not an easy task that every student could complete in a semester. One way of allowing students to acquire the practical training needed while keeping the amount of work reasonable is through the use of existing resources or software packages [Chau et al., 2003]. Many large companies such as Google, Microsoft, Amazon, and eBay are opening access to their search services and data through open Application Programming Interfaces (APIs). These services allow organizations to use the Web more effectively by sharing applications and data. They also allow developers to build applications more easily by having access to such shared resources. In education, these APIs provide an ideal playground for students to gain some practical skills in Web mining techniques.

In this paper, we report our experience designing a Web mining class project based on open Web APIs for students in a graduate-level course at the University of Arizona. In a group project for the class, students were required to design and implement a Web mining application using the open Web APIs provided by Google, Amazon, and eBay. The rest of the paper is organized as follows. In Section 2, we review the background for this paper, including Web mining research, open Web APIs, and education in software development and Web mining. In Section 3, we discuss the objectives and design of our class project. Some sample source codes are also given for illustration. Section 4 presents four sample systems that were developed by the students in the project for different application domains. Lastly, in Section 5 we discuss our conclusions and outline some suggestions for future research.

2. BACKGROUND

In this section, we first provide an overview of the research in the area of Web mining and its applications in solving business problems.

2.1 Web Mining

Since the advent of the Internet, many studies have investigated the possibility of extracting knowledge and patterns from the Web because it is publicly available and contains a rich set of resources. Many Web mining techniques are adopted from data

mining, text mining, and information retrieval research [Etzioni, 1996; Chen & Chau, 2004]. Most of these studies aimed to discover resources, patterns, and knowledge from the Web and Web-related data (such as Web usage data or Web server logs).

Web mining research can be classified into three categories: Web content mining, Web structure mining, and Web usage mining [Kosala & Blockeel, 2000]. Web content mining refers to the discovery of useful information from Web contents, including text, images, audio, video, etc. One example of Web content mining research is resource discovery from the Web [Cho et al., 1998; Chakrabarti et al., 1999; Chau & Chen, 2003], including research on spiders and crawlers. Web document categorization and clustering are also important topics in Web content mining research. These are often used for post-retrieval analysis of search engine results [Zamir & Etzioni, 1999; Chen et al., 2003]. Information extraction from Web pages also falls into the category of Web content mining [Hurst, 2001].

Web structure mining studies the model underlying the link structures of the Web. Such models have been widely used to infer important information about Web pages. Web structure mining has been largely influenced by research in social network analysis and citation analysis (bibliometrics). Citations (linkages) among Web pages are usually indicators of high relevance or good quality. We use the term in-links to indicate the hyperlinks pointing to a page and the term out-links to indicate the hyperlinks found in a page. Usually, the larger the number of in-links, the better a page is considered to be. The rationale is that a page referenced by more people is likely to be more important than a page that is seldom referenced. In addition, it is reasonable to give a link from an authoritative source (such as Yahoo) a higher weight than a link from an unimportant personal homepage. Web structure mining has been used for search engine result ranking and other Web applications, with PageRank [Brin & Page, 1998] and HITS [Kleinberg, 1998] being the most widely used.

Web usage mining focuses on using data mining techniques to analyze search logs or other activity logs to find interesting patterns. A Web server log contains information about every visit to the pages hosted on the server, such as files requested, user's IP address, and timestamp. By performing analysis on Web usage log data, Web mining systems can discover knowledge about a system's usage characteristics and the users' interests. Such knowledge has various applications, such as personalization and collaboration in Web-based systems, marketing, Website design, Website evaluation, and decision support [Armstrong et al., 1995; Chen & Cooper, 2001; Marchionini, 2002; Fang et al., 2006].

2.2 Open Web APIs

The Web no longer contains merely static pages and contents. Powered by modern database technologies, network technologies, and computational ability developments, many Websites provide sophisticated services to customers. By viewing the service provided by a Website as an application [Baresi et al., 2000], the different Website functionalities can be considered different modules of the application. The concept of Web APIs enables direct access to these modules from the client side or a third party's Website. In that case, a browser is not necessary for using the services provided by the Websites. The functions provided by a Website can be integrated into a third party's Website or software package, which is transparent to the end users. Currently, more than 200 Websites have published Web APIs for access to their services [ProgrammableWeb, 2006]. The idea is to leverage third party efforts on value-adding services and GUIs, so that the Website can focus on and enlarge the usage of its core service.

Many of the open Web APIs are constructed based on the Web services architecture. Some of them have been wrapped into libraries written in Java, .NET, JavaScript, etc., which hide the detailed Web services protocols from the developers and make it easier for the developers to use. Web services are a technique developed to support inter-platform function calling. Before the emergence of Web services, several similar techniques were used in different contexts, such as Remote Procedure Call (RPC), Common Object Request Broker Architecture (CORBA), Component Object Model (COM), Distributed Component Object Model (DCOM), Remote Method Invocation (RMI), and Messaging [Nandigam et al., 2005]. In comparison with its predecessors, Web services are based on XML (Extensible Markup Language) and HTTP (Hypertext Transfer Protocol), which provide better platform and language independent interoperability. Both the requests to the Web application and the responses from the Web application are packaged into XML format and transferred using HTTP through the Internet [Roy & Ramanujan, 2001]. The implementation details of the server side are hidden from the users/third party developers, but the interfaces are publicly available. The users/third party developers have flexibility to select the client side development platform and language. The structured input/output information in XML format can be easily presented in user interfaces developed by the third party. The structured data acquired from the Web application can also be used for further data mining analysis.

The companies that open Web APIs to the public belong to several different categories, such as Web search (e.g., Google and Yahoo), chat and messaging (e.g., MSN and AOL), geographic map (e.g., NASA, Google Maps, Yahoo Map, and Microsoft

MapPoint), e-commerce (e.g., Amazon, eBay, and Paypal), shipping (e.g., FedEx and UPS), and others (e.g., BBC and Skype). In the following we describe the three sets of open Web APIs that are most relevant to the current paper, namely Amazon, eBay, and Google.

2.2.1 Amazon. As a leader in e-commerce, Amazon Web Services offers a variety of Web services that allow developers to build businesses based on Amazon's data. Such Web APIs include access to Amazon's product data and e-commerce functionality (Amazon E-Commerce Service) and access to Amazon's sales history data (Amazon Historical Pricing). Amazon also provides Web Services for access of storage (Amazon S3), queue (Amazon Simple Queue Service), and Web search (Alexa Web Search Platform) [Amazon, 2006].

Amazon E-Commerce Service is based on SOAP (Simple Object Access Protocol) and WSDL (Web Service Description Language) standards [Curbera et al., 2002]. In addition, Amazon provides a REST protocol for access. With this set of APIs, developers can access many features of the Amazon E-Commerce functionality. It provides detailed product attributes, images, pricing information, and customer reviews for virtually all products across every product category in the Amazon.com product catalog. Using this set of open Web APIs, developers can perform complex search functions on the available products in Amazon's virtual market. Amazon E-Commerce Service also provides interface to access the customers' wish lists. With access to the products, reviews, wish lists, and so forth, developers can easily build e-commerce Websites to market their own products and perform transactions through Amazon. It is also possible to use this information to create value-added applications [Zadel & Fujinaga, 2004]. To use Amazon E-Commerce Service, developers need to create a free Amazon Web Service account. The usage of the service is restricted to one call per second per IP address.

The Amazon Historical Pricing service gives developers programmatic access to the actual sales data for books, music, videos, and DVDs (as sold by third party sellers on Amazon.com) since 2002. The data includes the average, minimum, maximum, and median prices for the specified items over the given date range(s). Sellers can use Amazon Historical Pricing to make informed decisions on pricing and purchasing. The Amazon Historical Pricing Web service requires a monthly subscription fee.

2.2.2 eBay. Using the eBay Web Services API, developers can create Web-based applications to conduct business with the eBay Platform [Mueller, 2004]. The API can access the data on eBay.com and Half.com. Developers can perform functions such as sales management, item search, and user account management [eBay, 2006].

The eBay Web Services API is also based on the SOAP and WSDL standards. The eBay developer center provides wrappers of the SOAP APIs in Java, .NET, and PHP. To use the eBay API, developers need to join the eBay developer program (free). Depending on the type of membership, a developer can make either 10,000 free API calls per month or 1.5M free API calls per day. eBay provides a testing environment — sandbox.ebay.com — to developers. The testing environment employs the same mechanism as the production system. Developers can add or remove products on this system just for program testing purpose. Thus the program development process will not affect real users' usage.

2.2.3 Google. Google provides several kinds of open Web APIs for accessing their advertisement service (AdWords API), blog service (Blogger Atom API), map service (Google Maps API), and the Web search service (Google Search API). The most widely used are Google Search API and Google Maps API [Google, 2006].

Google Search API are implemented as a Web service using SOAP and WSDL standards. Google provides a Java library which wraps the SOAP APIs. The APIs enable developers to query all the Web pages on Google's server. It also allows developers to access the cached Web pages on Google's server and get suggested spellings for incorrect spellings of the search keywords. With this set of APIs, it is very easy for a developer to create a search engine interface connecting to Google and add more functions [Hoong & Buyya, 2004]. To access the Google Search API service, a developer needs to create a Google account and obtain a license key. The license key must be included with each query submitted. Each account and license key entitles the user to 1,000 automated queries per day. Currently the Google Search API is still a beta service.

Google Maps API is a JavaScript API which can embed Google Maps in Web pages. It is designed mainly for data representation purposes. The Google Maps API works in a browser interface. To use this API, an API key should be obtained from Google. According to the terms, the Websites (or part of the Websites) that use Google Maps API should be accessible to the consumers without charge; it cannot be used in enterprise applications.

2.3 Teaching Web Mining and Software Development

Programming and software development courses have been taught in most computer science and information systems curricula as lectures, often with lab discussions. In recent years, due to the rapid development of the Web, more techniques related to Internet and Web applications have been introduced in such classes [McDowell, 2005]. Web mining techniques and data mining techniques were began to be introduced in more

courses [Kalathur, 2006; Menczer, 2006]. The courses on Web mining [Davison, 2005; Davulcu, 2002; Xie, 2006] covered the topics on Web search, Web usage mining, text mining, information extraction, and link analysis. Such training educates students on how to extract information from the Web and discover knowledge from such information. To help students understand the topics better and assess their performance, individual programming or written assignments are widely used in these classes.

Group projects or group programming exercises have become more popular in many of the courses related to programming and software development. It has been suggested that such group activities promote cooperative learning and a positive experience among students [Dutt, 1994; McConnell, 1996]. It also has been shown that group programming exercises used in an introductory programming course improve problem-solving abilities and increase knowledge and interpersonal skills [Granger & Lippert, 1999; Poindexter, 2003]. Similarly, Harris [1995] suggests that group projects may provide opportunities for students to improve their written and oral skills. Several researchers also stress the importance of allowing students to work on projects that can be applied to real-world problems, and thus gain valuable hands-on experience [Fox, 2002].

Some courses that discussed Web mining-related topics used Web search engine as a topic for group projects [Chau et al., 2003]. For example, Davison's course required students to implement a search engine for the Lehigh University Website [Davison, 2004]. Such a project provides students with empirical understanding of spidering, parsing, indexing, and link analyzing techniques. While creating a search engine, students would view the Web as linked Web pages. Applying Web service/Web API-related topics in Web mining-related group projects enables students to view the Web as a set of applications. Instead of just focusing on the details of the programming techniques, using open Web APIs to develop Web mining projects lets students practice how to integrate different existing Web application components to build their own value-added applications. When compared with the free-text data retrieved using search engines, the data retrieved through open Web APIs is usually structured data, which is much easier to analyze using data mining techniques.

3. A CLASS PROJECT COMBINING WEB MINING AND OPEN WEB APIS

3.1 Objectives

Given the advantages of using open Web APIs and the vast amount of Web APIs provided by different Websites, we found few course projects that consolidated open Web APIs within Web mining projects. We believe that it is important for students to acquire some classroom knowledge and experiences in both areas. In this paper, we

report on a class project that we designed and developed in 2005. Details of the project are discussed in the following subsection.

3.2 Web Mining Using Google, eBay, and Amazon APIs

In the class project each group of students was required to create a Web business with a complete Website and business functionalities for specific customers, using one or more of the three open Web APIs (Amazon, eBay, and Google) together with other data mining techniques learned in the class. Given the comprehensive functions provided by the three Web APIs, the main challenge of the project was to integrate these components and design attractive system features. Students were encouraged to integrate open source data mining components, such as WEKA [Witten & Frank, 2005], to analyze the data retrieved through the Web APIs.

Each team consisted of three members who would participate in the design, coding, implementation, and analysis of the prototype. Each team member was required to participate in all aspects of the project. They could use their own machines as a server or share a server provided by the university's open computing laboratory. Lab sessions were provided to familiarize students with Web API programming. A teaching assistant was also provided to assist the students with programming problems. The students were given three and a half months to finish the project and were required to submit a project proposal after the first month of class. At the end of the semester, each group would present their work, demonstrate the system and submit a final report. The projects were graded based on system functionalities, novelty, and business feasibility.

```
import com.google.soap.search.*;
public class GoogleMain {
    public static void main(String[] args) {
        GoogleSearch gSearch = new GoogleSearch();
        gSearch.setKey("[your key here]");
        try{
            byte[] cachedContent = gSearch.doGetCachedPage("http://ai.arizona.edu");
            System.out.println(new String(cachedContent)); //Print cached page
            gSearch.setQueryString("Web Services");
            GoogleSearchResult searchResult = gSearch.doSearch();
            System.out.println(searchResult); //Print search result
            String suggestion = gSearch.doSpellingSuggestion("Webapi");
            System.out.println(suggestion); //Print spelling suggestion
        }catch(Exception ex){
            ex.printStackTrace();
        }
    }
}
```

Fig. 1. Sample Java code for using the Google Search API.

In the class lab sessions, two simple scenarios of using the Google Search API and the Amazon E-Commerce Service were introduced¹. The Google Search API example was implemented using the Java library provided. It directly called three methods to implement simple keyword search, cached page retrieval, and keyword suggestion functions (Figure 1). The Amazon E-Commerce Service example was implemented based on the REST standard using the Jakarta httpclient library. The program obtained the content of a specific wish list (in XML format) and displayed the content in text format according to a pre-designed XSL format (Figure 2). Step-by-step tutorials were given in the lab sessions to make sure that the students understood these two different implementation methods and were able to use them in their projects.

```
import java.io.IOException;
import org.apache.commons.httpclient.DefaultHttpMethodRetryHandler;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.HttpException;
import org.apache.commons.httpclient.HttpStatus;
import org.apache.commons.httpclient.methods.GetMethod;
import org.apache.commons.httpclient.params.HttpMethodParams;
public class HttpClientMain {
    public static void main(String[] args)
    {
        HttpClient client = new HttpClient();
        GetMethod method = new GetMethod("http://webservices.amazon.com/onca/xml?
Service=AWSECommerceService&SubscriptionId=[yourid]&Operation=ListLookup&Li
stType=WishList&ListId=1Y3TY8UXS2N6O&ResponseGroup=ListItems,ListInfo&Sty
le=http://www.u.arizona.edu/~chunju/text.xsl&ContentType=text/plain");
        method.getParams().setParameter(HttpMethodParams.RETRY_HANDLER,new
DefaultHttpMethodRetryHandler(3, false));
        try {
            int statusCode = client.executeMethod(method);
            if (statusCode != HttpStatus.SC_OK) {
                System.err.println("Method failed: " + method.getStatusLine()); //Error happens
            }
            byte[] responseBody = method.getResponseBody();
            System.out.println(new String(responseBody)); //Print the retrieved wishlist
        } catch (HttpException e) {
            System.err.println("Fatal protocol violation: " + e.getMessage());
        } catch (IOException e) {
            System.err.println("Fatal transport error: " + e.getMessage());
        } finally {
            method.releaseConnection(); //Close the connection
        }
    }
}
```

Fig. 2. Sample Java code for using the Amazon Web Service.

¹ Sample codes and tutorial are available online in our course Website at:
http://ai.arizona.edu/hchen/chencourse/Website/WebAPI_project.htm

After the lab sessions, the students experimented with these APIs. They then designed their value-added business based on extensive team discussions. They could choose to include data mining algorithms and packages in their design. Finally, the students needed to integrate the Web APIs into their Website to perform selected business operations. Depending on the system functionality, many systems may need to store data in relational databases, such as Oracle, Microsoft SQL Server, or MySQL. They could also acquire the data in real-time to get the most up-to-date customer or product information.

4. STUDENT PROJECTS

At the end of the semester, each project group gave a demonstration of their system. Overall, we found the projects well designed and interesting. The students applied what they learned from the class as well as demonstrating their creativity and teamwork in the project. By incorporating the services and data provided by Amazon, Google, and eBay with the latest data mining and visualization technologies, the students developed many innovative business models.

In order to illustrate what the students achieved in the project, four selected examples, namely Wishsky, Tucson Book Xchange, Cellphone Intelligent Auctioning (CIA), and SciBubble, are discussed in this section.

4.1 Wishsky

Wishsky is an integrated “wish list” management system which enables customers to monitor the news and ongoing sales/auctions related to the products in their wish lists. Wishsky also recommends alternative products and related products to the customers.

Wishsky was implemented on Tomcat using Java Server Page (JSP). As shown in Figure 3, the system architecture has three layers. The bottom layer of the system is the database layer. Wishsky uses a local Microsoft SQL Server which stores the customer information, wish list data, and product recommendation information through Java Database Connectivity (JDBC). Wishsky can retrieve relevant product data through the Web APIs from Amazon, eBay and Google.

The middle layer of the system is the logic control layer, which contains four modules (written in Java). The API Query Module sends requests to and gets results from eBay, Google, and Amazon. It encapsulates the three sets of Web APIs and provides an interface for the upper layer to access the three data repositories. The Database Query Module encapsulates the communication with the backend database via JDBC. The Data Mining Module integrates the Apriori algorithm of the WEKA package [Witten & Frank, 2000; 2005], which provides product recommendations according to the wish lists in the

system. The three modules are coordinated by a Control Module, which also controls the data and logic flows between the user interface and database layers.

The top layer of the system is the presentation layer. HTML, JSP, and JavaScript are used to generate friendly and intuitive user interfaces.

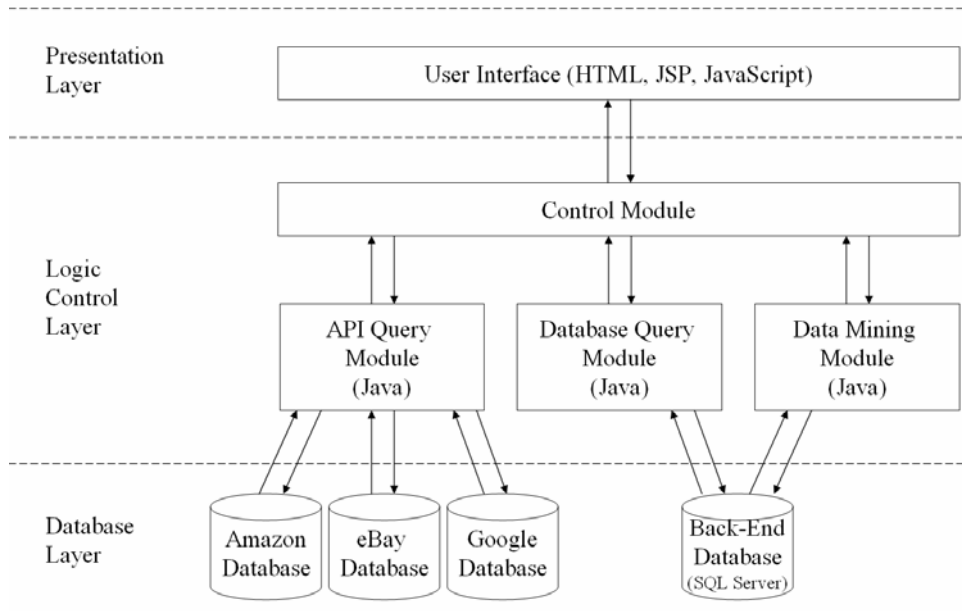


Fig. 3. Wishsky system architecture.



Fig. 4. Wishsky Website.



Fig. 5. The page for personalizing a wish list.



Fig. 6. Wishsky displays a user's wish list, the item news, and the recommendations.

Figures 4 to 6 show a sample scenario of a user session. The Wishsky homepage features the most popular items in the Wishsky system according to users' wish lists. In addition to the item descriptions (e.g., "Star Wars" and "Episode III"), the right frame shows news related to those popular items, which was retrieved using the Google Search API (Figure 4). After registration, the user can create a Wishsky wish list by importing and modifying his/her Amazon wish list (Figure 5). After the wish list is set up, the system will personalize each user's view to include the wish list, item-related news, and

product recommendations (Figure 6). The user can choose among different algorithms to obtain product recommendations or search available auctions of their wish list items on eBay.

4.2 Tucson Book Xchange

Tucson Book Xchange is an online book exchange system. The system supports online book selling and buying, mainly for Tucson residents. It also enables users to search for books on Amazon and eBay Websites by integrating their Web APIs. One unique feature of the system is that it visualizes the actual location of the books. By incorporating Google Maps API, the system is able to provide map images and driving directions according to buyer's and seller's home addresses in their profiles. Figure 7 provides us with such an example: after user "aydink" reserved a book, the map image in the right frame locates the user and the book, and the directions and address information are shown in the left frame. The Tucson Book Xchange system architecture is similar to the Wishsky system. However, at the presentation layer, JavaScript and AJAX (Asynchronous JavaScript and XML) technology are used to incorporate the Google Maps features.

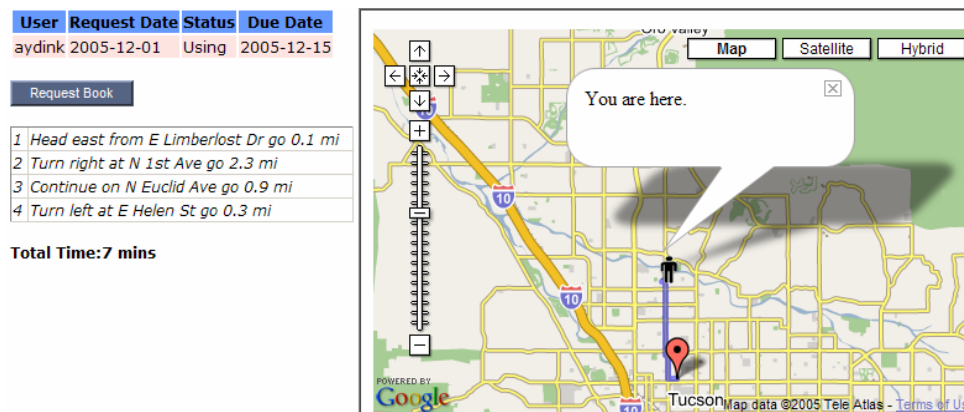


Fig. 7. The user interface of Google Maps API integration in Tucson Book Xchange.

4.3 Cellphone Intelligent Auctioning

Cellphone Intelligent Auctioning (CIA) is a cell phone auction history analysis system which provides value-added services to eBay cell phone buyers and sellers. The CIA group used ASP.NET framework to implement their application. The group developed a standalone application in C# to extract eBay item information and transaction information, which were stored in a backend MySQL database (Figure 8). Based on these transaction data, CIA performs analysis on how product price and quantity associate with four product aspects, namely time, brand, seller, and location. The results are presented in tables and graphs which provide users with a better idea of the overall trend of the eBay

cell phone market. For example, Figure 9 shows the time trend analysis of the average cell phone price for a period of two weeks. The analysis clearly shows the weekend effect, when the bidding prices of items are slightly lower than the average prices. Such analysis may help buyers and sellers make better auction decisions.

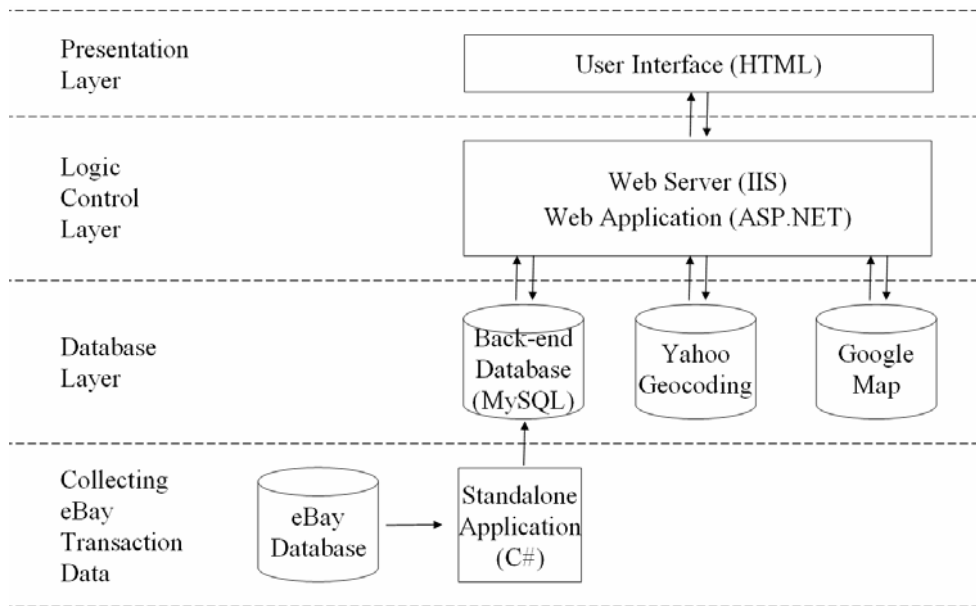


Fig. 8. CIA system architecture.

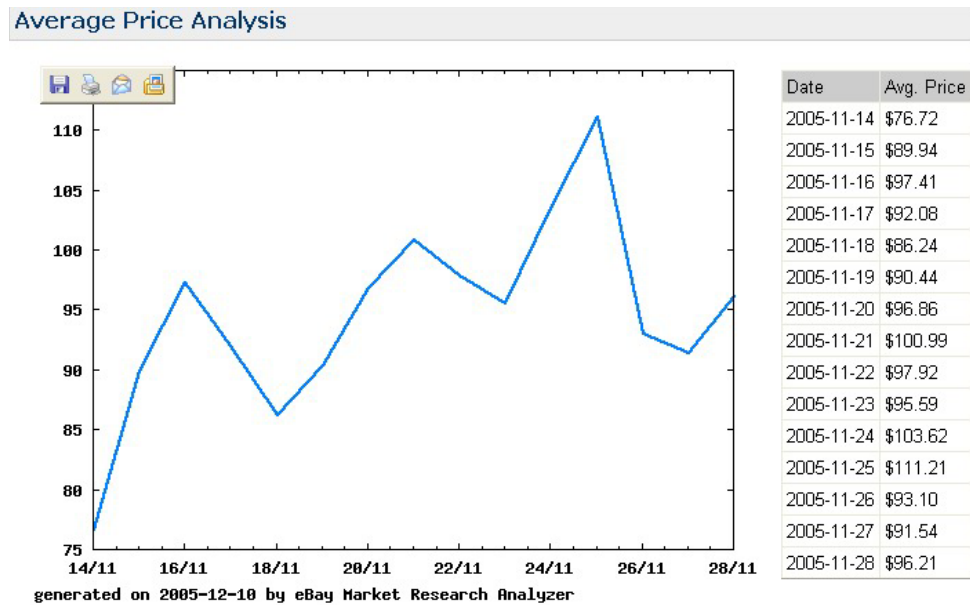


Fig. 9. The time trend analysis on average cell phone price.

To apply geographic analysis on the eBay auction transactions (retrieved using eBay API), the group combined several APIs together. After extracting location information from transactions, the graphical coordinates (latitude and longitude) information is

retrieved from Yahoo through the Yahoo Geocoding API and fed into the Google Maps API to visualize the locations on a map (Figure 10).

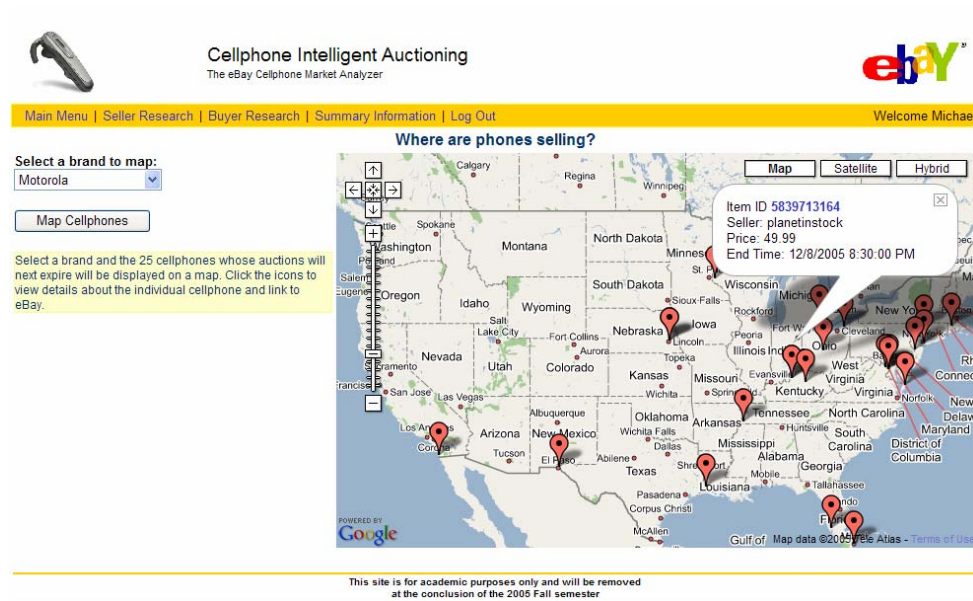


Fig. 10. The geographical analysis on cell phone auction transactions.

4.4 SciBubble

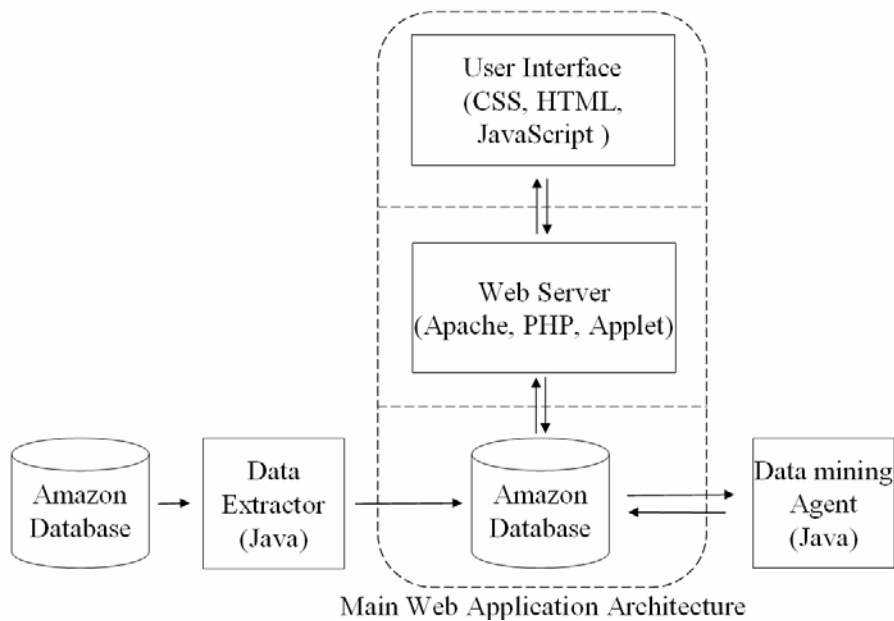


Fig. 11. SciBubble system architecture.

SciBubble is an Amazon science fiction book portal which features distinct visualization that helps customers find their books of interest. The science fiction book data, including book details and customer reviews, were retrieved from Amazon through the Amazon E-

Commerce Service API and were loaded into a MySQL database (Figure 11). The similarity between each pair of books was pre-calculated according to such information as publisher, rating, publication year, ranking, and category. An algorithm was designed to visualize similar books which are related to a given book search query. In this algorithm, each book is represented as a bubble and the similarities between books are represented by the distances and angles between the bubbles (Figure 12).

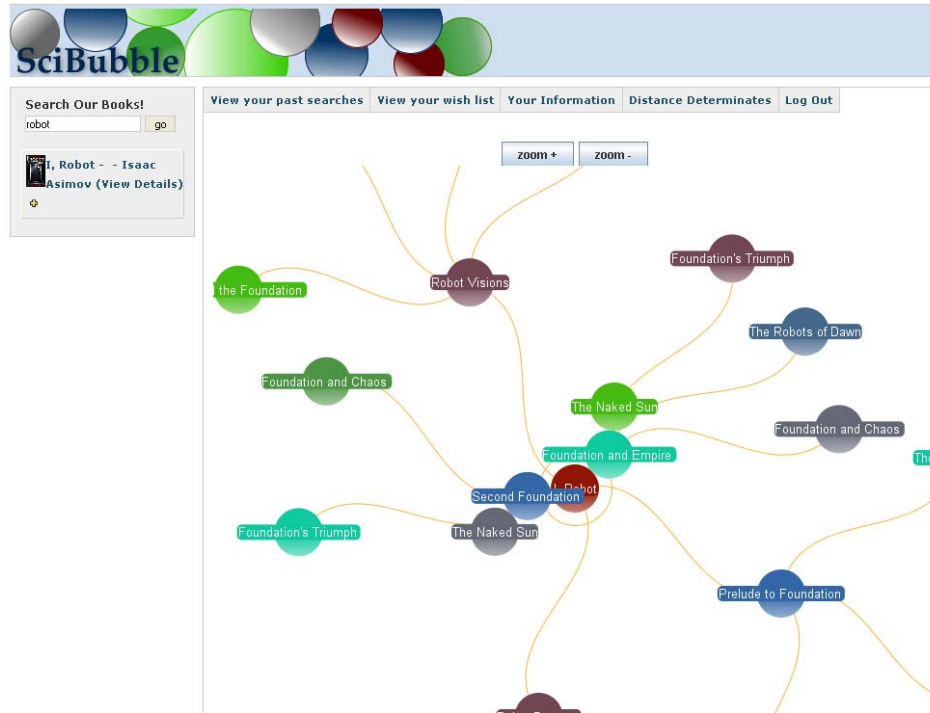


Fig. 12. The visualization interface of SciBubble.

4.5 Observations of Student Projects

Most groups successfully achieved their objectives. The student projects covered a wide range of different business models and utilized selected Web APIs to implement different functionalities. The students used not only the three APIs suggested by the instructor (Amazon E-Commerce Service, eBay API, and Google Search API), but also other APIs, such as Yahoo Geocoding API and Google Maps API. A summary of selected projects and Web APIs is shown in Table I. Among these Web APIs, the Amazon E-Commerce Service was most frequently used. The Amazon E-Commerce Service provides a lot of structured information related to products, customer behaviors, and business transactions. The special data characteristics attracted most of the students.

Table I. Summary of selected student projects

<i>Project name</i>	<i>Business model</i>	<i>Web APIs</i>	<i>Innovations</i>
Cellphone Intelligent Auctioning	Cell phone sales analysis	eBay API, Yahoo Geocoding API, Google Maps API	Statistical analysis
GiftChannel	Wish list management	Amazon E-Commerce Service	Product recommendation
MusicBox	Music album sales/news portal	Google Search API, Amazon E-Commerce Service	Product recommendation and visualization
PriceSmart	Market analysis /Sales management	Amazon E-Commerce Service	Data analysis (SQL server 2005)
SciBubble	Science fiction book portal	Amazon E-Commerce Service	Visualization
Tucson Book Xchange	Local book flea market	Amazon E-Commerce Service, eBay API, Google Maps API	Driving directions
Wishsky	Wish list management	Amazon E-Commerce Service, eBay API, Google Search API	Product recommendation (WEKA)

The students designed several interesting business models which are value-added services for E-commerce Websites. One such business model is to provide sales history analysis and market analysis to the sellers/buyers for their better business decisions. The second type of business model is to provide product recommendations and personalized news to customers. This business model aims to improve product sales by providing the products/news that the customers may be interested in. Another popular business model is providing geographical information to aid business transactions.

In addition to using Web APIs, most groups incorporated data mining techniques or statistical analysis in their systems. Some students integrated software packages, such as WEKA and MS SQL Server; while others implemented selected data mining algorithms from scratch. Different visualization techniques were used in the projects. Some projects provide dynamic charts in the data analysis; others implemented network visualization modules to visualize the complex relations between products. Table II summarizes the online resources (e.g., Web APIs and data mining/visualization toolkits) used by the students.

The students have different backgrounds. Not every student was familiar with programming and Web application development. However, we observed that they were able to form balanced teams and collaborated closely in the project, making the projects successful. Most of the projects had innovative business models and most students did a

good job delivering their ideas in the final presentations and project reports. Through solving the problems encountered in the projects, students were able to improve their programming and system integration skills. As in other group projects, we observed that students benefited from team projects by learning from each other and improved their abilities to work in a team.

On the other hand, we also observed some weaknesses in the projects that need to be improved. For example, few projects considered the issues related to system performance and scalability. Some database designs may cause a long query delay for large datasets. Some of the programs need to be optimized in order to improve the system performance. Another shortcoming of the projects is that the students were not required to work with real clients. This made some of the projects less practical for real-world operations. In the future, it would be a good idea to require students to work with real-world clients and businesses.

Table II. Online resources used in the projects

<i>Type</i>	<i>Name</i>	<i>URL</i>
Web APIs	Amazon Web Services	http://www.amazon.com/webservices/
	Google APIs	http://code.google.com/apis.html
	eBay API	http://developer.ebay.com/
	Yahoo APIs	http://developer.yahoo.com/
Data analysis packages	WEKA	http://www.cs.waikato.ac.nz/ml/weka/
	Yale	http://sourceforge.net/projects/yale/
	MS SQL Server 2005	http://www.microsoft.com/sql/
	IBM DB2 intelligent miner	http://www.ibm.com/software/data/iminer/
Network visualization toolkits	JUNG	http://jung.sourceforge.net/
	Graphviz	http://www.graphviz.org/

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we report on our experience designing a class project for students in a graduate course to use open Web APIs for developing Web mining applications. Overall, the results are encouraging. Through these Web mining projects, we observe that most students were able to build innovative Web applications within a short period of time; which would be impossible if the students had to develop the systems from scratch. We observe that students developed the abilities to create interesting business models and integrate necessary system components to implement them. In addition to the three suggested APIs from Google, eBay, and Amazon, students also successfully identified and incorporated other Web APIs and data mining/visualization tools in the projects.

Most students expressed that they had learned a lot during the project. They commented that the system implementation experience was much better than what they thought they could achieve at the beginning of the semester. In the future, we plan to encourage students to identify and experiment with other Web-based open source software. With the rapid growth of Web-based open source software, it would be interesting to study how students utilize such software in learning Web-related topics and developing e-commerce business ideas.

ACKNOWLEDGMENTS

We thank Google, Amazon, eBay, Yahoo, and WEKA for making their APIs, software, and data available to the public. We also thank the students who took our course and participated in developing the Web mining applications.

REFERENCES

- Amazon. 2006. Amazon Web Services Website. <http://www.amazon.com/webservices/>
- ARMSTRONG, R., FREITAG, D., JOACHIMS, T. AND MITCHELL, T. 1995. WebWatcher: A Learning Apprentice for the World Wide Web. In *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Mar 1995.
- BARESI, L., GARZOTTO, F., PAOLINI, P. 2000. From Web Sites to Web Applications: New Issues for Conceptual Modeling. In *Conceptual Modeling for E-Business and the Web: ER 2000 Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling*, Salt Lake City, Utah, USA, p. 89.
- BRIN, S. AND PAGE, L. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th WWW Conference*, Brisbane, Australia, Apr 1998.
- CHAKRABARTI, S., VAN DEN BERG, M., AND DOM, B. 1999. Focused Crawling: A New Approach to Topic-specific Web Resource Discovery. In *Proceedings of the 8th International World Wide Web Conference*, Toronto, Canada, May 1999.
- CHAU, M. AND CHEN, H. 2003. Comparison of Three Vertical Search Spiders. *IEEE Computer* 36(5), 56-62.
- CHAU, M., HUANG, Z., CHEN, H. 2003. Teaching Key Topics in Computer Science and Information Systems through a Web Search Engine Project. *ACM Journal of Educational Resources in Computing* 3, 1-14.
- CHAU, S., SHIU, B., CHAN, I., AND CHEN, H. (forthcoming) Redips: Backlink Search and Analysis on the Web for Business Intelligence. *Journal of the American Society for Information Science and Technology*, accepted for publication.
- CHEN, H. AND CHAU, M. 2004. Web Mining: Machine Learning for Web Applications. *Annual Review of Information Science and Technology* 38, 289-329.
- CHEN, H., LALLY, A. M., ZHU, B. AND CHAU, M. 2003. HelpfulMed: Intelligent Searching for Medical Information over the Internet. *Journal of the American Society for Information Science and Technology* 54(7), 683-694.
- CHEN, H. M. AND COOPER, M. D. 2001. "Using Clustering Techniques to Detect Usage Patterns in a Web-Based Information System," *Journal of the American Society for Information Science and Technology* 52(11), 888-904.
- CHO, J., GARCIA-MOLINA, H., PAGE, L. 1998. Efficient Crawling through URL Ordering. In *Proceedings of the 7th WWW Conference*, Brisbane, Australia, Apr 1998.
- CURBERA, F., DUFTLER, M., KHALAF, R., NAGY, W., MUKHI, N., WEERAWARANA, S. 2002. Unraveling the Web services Web - An introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing* 6, 86-93.
- DAVISON, B.D. 2004. Course Website of CSE345/445 WWW Search Engines <http://www.cse.lehigh.edu/~brian/course/searchengines/>.
- DAVISON, B.D. 2005. Course Website of CSE 450: Web Mining Seminar <http://www.cse.lehigh.edu/~brian/course/webmining/>.
- DAVULCU, H. 2002. Course Website of CSE 591: Semantic Web Mining http://www.public.asu.edu/~hdavulcu/CSE591_Semantic_Web_Mining.html.
- DUTT, J. 1994. Cooperative learning approach to teaching an introductory programming course. In *Proceedings of the Ninth International Academy for Information Management*, Las Vegas, NV.
- eBay. 2006. eBay developers program <http://developer.ebay.com>.
- ETZIONI, O. 1996. The World Wide Web: Quagmire or Gold Mine. *Communications of the ACM* 39(11), 65-68.

- FANG, X., CHAU, M., HU, P. J., YANG, Z., SHENG, O. R. L. 2006. Web Mining-Based Objective Metrics for Measuring Website Navigability,” in *Proceedings of the International Conference on Information Systems*, Milwaukee, Wisconsin, USA, December 2006.
- FOX, T.L. 2002. A case analysis of real-world systems development experiences of CIS students. *Journal of Information Systems Education* 13, 343-350.
- Google. 2006. Google APIs. <http://code.google.com/apis.html>.
- GRANGER, M., LIPPERT, S. 1999. Peer learning across the undergraduate information systems curriculum. *Journal of Computers in Mathematics and Science Teaching* 18, 267-285.
- HARRIS, A.L. 1995. Developing the systems project course. *Journal of Information Systems Education* 6, 192-197.
- HOONG, D.C., BUYYA, R. 2004. Guided Google: A Meta Search Engine and its Implementation using the Google Distributed Web Services. *International Journal of Computers and Applications* 26(1).
- HURST, M. 2001. Layout and language: Challenges for Table Understanding on the Web. In *Proceedings of the 1st International Workshop on Web Document Analysis*, Seattle, WA, USA, September 2001, pp. 27-30.
- KALATHUR, S. 2006. Course Website of CS 699: Principles of Data Mining http://people.bu.edu/kalathur/courses/cs699_06_fall.htm.
- KLEINBERG, J. 1998. Authoritative Sources in a Hyperlinked Environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, California, USA, Jan 1998, pp. 668-677.
- KOSALA, R. AND BLOCCKEEL, H. 2000. Web Mining Research: A Survey. *ACM SIGKDD Explorations* 2(1), 1-15.
- LIU, B., HU, M., AND CHENG, J. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web, In *Proceedings of the WWW 2005 Conference*, May 10-14, 2005, Chiba, Japan, 342-351.
- MARCHIONINI, G. 2002. Co-Evolution of User and Organizational Interfaces: A Longitudinal Case Study of WWW Dissemination of National Statistics, *Journal of the American Society for Information Science and Technology* 53(14), 1192-1209.
- MCCONNELL, J., 1996. Active learning and its use in computer science. In *Proceedings of the SIGCSE/SIGCUE conference on Integrating Technology into Computer Science Education*, Barcelona, Spain.
- MCDOWELL, L. 2005. Course Website of IT350: Web & Internet Programming <http://www.cs.usna.edu/~lmcadowel/courses/it350/F05/CoursePolicy.htm>.
- MENCZER, F. 2006. Course Website of CSCI B659: Topics in Artificial Intelligence <http://informatics.indiana.edu/fil/Class/b659/>.
- MUELLER, J.P. 2004. *Mining eBay Web Services: Building Applications with the eBay API*. SYBEX Inc. Alameda, CA, USA.
- NANDIGAM, J., GUDIVADA, V.N., KALAVALA, M. 2005. Semantic Web services. *Journal of Computing Sciences in Colleges* 21, 50 - 63.
- POINDEXTER, S. 2003. Assessing active alternatives for teaching programming. *Journal of Information Technology Education* 2, 257-266.
- Programmableweb. 2006. <http://www.programmableweb.com/apis>.
- ROY, J., RAMANUJAN, A. 2001. Understanding Web services. *IEEE IT Professional* 3, 69-73.
- SCHONFELD, E. 2005. The Great Giveaway. *Business* 2.0, April 2005, 80-86.
- WITTEN, H. I. AND FRANK, E. 2005. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann Publishing, San Francisco, CA, 2005.
- WITTEN, H. I. AND FRANK, E. 2000. Nuts and bolts: Machine learning algorithms in Java. *Practical Machine Learning Tools and Techniques with Java Implementations*, 296-297. Available at <http://www.cs.waikato.ac.nz/ml/weka/>
- XIE, Y. 2006. Course Website of CS 8990: Web Search and Mining <http://science.kennesaw.edu/~yxie2/CS8990W/CS8990W.htm>.
- ZADEL, M., FUJINAGA, I. 2004. Web Services for Music Information Retrieval. In *Proceedings of the 5th International Conference on Music Information Retrieval*.
- ZAMIR, O. AND ETZIONI, O. 1999. Grouper: A Dynamic Clustering Interface to Web Search Results. In *Proceedings of the 8th World Wide Web Conference*, Toronto, May 1999.