On Designing a Market Monitoring Web Agent System

Simon Fong
Faculty of Science and Technology
University of Macau
SAR Macau, China
+853 3974473
ccfong@umac.mo

Yang Hang
Faculty of Science and Technology
University of Macau
SAR Macau, China
+853 66874953
ma76562@umac.mo

ABSTRACT
World-Wide-Web is a huge pool of valuable information for companies to know what their competitors are doing and what products and services they offer up-to-date. Companies can gather business intelligence from the Web for planning countermeasures strategies. Hence it is crucial to have the right tool to effectively gather such information from the Web. Many information retrieval and monitoring technologies have been developed. But they are more for generally tracking changes and downloading the whole websites for offline browsing. This paper is to shed some light on specifically the design of a Web monitoring system for gathering business information relevant to a company. The Watcher Agent is a server-based system that is built with two main parts, namely Price Watcher and Market Watcher. The system will assist company users in price information collection, news information filtering, and product ranking estimation, thus saving time and effort for them.

Categories and Subject Descriptors
H.3.5 [**Online Information Services**]: Web-based services

General Terms
Algorithms, Management, Design, Economics, Experimentation.

Keywords
Web-intelligence, Competitor-intelligence, web extraction, information collecting system.

1. INTRODUCTION
The Web has grown into a prevalent platform for disseminating information of all kinds, ranging from entertainment to financial news on a global scale. A significant amount of valuable information that can be publicly viewed on the Web is growing tremendously, as more and more companies commit to update their websites for showing their latest business information for publicity. From a business perspective, the Web is a huge pool of free information for companies to know what their competitors are doing and what products and services they offer up-to-date. By acquiring such information from the Web, the companies can gather business intelligence for planning countermeasures and making themselves more competitiveness. Hence it is crucial for a company to have the right tool to effectively gather such information from the Web. Many information retrieval and monitoring technologies have been proposed in the literature, such as OpenCQ [1], WebCQ [2], CONQUER [3] and Niagara [4]. There are many commercial products too in models of Application Service Provider such as the ones listed in Table 1.

Table 1 Web Monitoring Systems

Service/Product	*URL*
WatchThatPage	*http://www.watchthatpage.com*
Wisdomchange	*http://www.wisdomchange.com*
ChangeDetection	*http://www.changedetection.com*
ChangeDetect	*http://www.changedetect.com*
Track Engine	*http://www.trackengine.com*
WebsiteWatcher	*http://www.aignes.com*

They are generally content monitoring services that periodically watch Web pages and other Internet resources for keyword related content or changes. However they are more for generally tracking changes and downloading the whole websites for offline browsing.

In this research, we focus on a business environment in which knowing one's competitors is of crucial importance to the survival and growth of any business. Before the Internet became popular, it used to be an expensive process in obtaining business intelligence information quite often from printed publication and other media channels. Although nowadays much of the information is freely available from the Web, they are scattered and dynamic in nature. Like searching for a grain in an ocean, acquiring useful information from the competitors' Web sites as well as from other Web news portals is still a tedious and time-consuming task.

Many companies today opt to invest certain resources in collecting information about their competitors from the Web and other channels. It is a regular routine that they want to know what their competitors are doing, what products and services they offer and any news that concern about them. This is usually done by manual browsing, by the marketing personnel.

iiWAS2008, November 24–26, 2008, Linz, Austria.

From our experiences of dealing with commercial executives, the approach of acquiring business intelligence from the Web by manual browsing poses a number of problems, as follow:

1. Business websites especially that of the large international cooperates often have a large number of pages in the number of hundreds, which makes it very difficult for manual browsing without any automated assistance. The overwhelming amount is a major cause to human errors in selecting the correct information.

2. Different companies may organize the same information very differently, due to differences in culture and practices. One company may use one format and another may use a different style. The diverse and unstructured formats make manual browsing a daunting task for human users.

3. Beyond the competitors' websites, there are certainly other websites, forums, news portals feature news about the competitors and their products. Searching the World-Wide-Web for all possible sites that might have mentions of the competitors is an extremely tedious task if done manually.

4. The speed of updates on some information portal, such as stock market, headline news could be beyond that of a human task that consumes time in continuously searching, downloading, extracting, analyzing and archiving. The balance of completeness and timeliness of downloading online information has been discussed in [5]. This implies some automated process must be implemented to capture the right information over the Web at the right time intervals.

The amount of information within a site and new sites is growing at a phenomenal rate. Monitoring such information can no longer be easily done manually.

2. OUR PROPSED SOLUTION

The objective of this paper is on the design of an automated market monitoring Web agent system, namely Market Watcher Agent, for gathering business information relevant to a company in an automated approach.

The Watcher Agent proposed in this paper is an autonomous software program that "spies" on the competitors' prices and news information over the web. The watcher agent consists of mainly two parts, namely, market watcher and price watcher. In this paper, market watcher is described in detail while detailed description on price watcher can be found in [6]. The Market Watcher is an information-collecting tool, which assists the users to monitor the specified Web sites, e.g. competitors' web sites, and to locate the relevant information automatically. The market watcher has two sub-components, namely market monitor and market explorer. The Market Monitor works as an information filter. The objectives of the Market Monitor is to find the news updating on competitors' web sites and to find news articles from some established news portals for particular products, for example, CNN and BBC. The Market Explorer, on the other hand, is an information provider, which is able to get the most updated information worldwide. As the Internet is extremely dynamic, it will never be enough to get news from a fixed number of Web sites, i.e., the competitors' official web sites and certain news portals. With the help from various types of the Internet search engines, worldwide information can be collected by passing users' queries to the search engines and retrieving the top matches. At least two kinds of information can be discovered with the search records from such popular search engines. One is the information about a particular product that cannot be easily found from the competitors' official web sites or news portals: for example, user's feedback or technical web site's product review. The other is the product ranking, or how prominent your product can be reached by the Internet users from search engines. For example, if the query "inkjet printer" is given to Google search engine, manufacturers in the top 20 matches will be Epson, Kodak and HP.

In summary, Table 2 lists the information which constitutes to business intelligence, and from where over the Web such information could be obtained.

Table 2 Summary of Web information to be monitored

Where can be found	*Web information as business*
Competitors' websites	*Competitors price information*
	Competitors product
	Competitors company
News portals	*Product news*
	Company news
Popular search engines	*Product news*
	Company news
	Search engine rankings

The rest of the paper is organized as follows. In Section 3, the system architecture is described in detail. Operational processes for market monitor and market explorer are presented in Section 4 and 5 respectively. Finally we conclude our work in Section 6.

3. SYSTEM ARCHITECTURE

As shown in Figure 1, in the Information Retrieval Layer, the URL Retrieval Engine takes two parameters as input, the URL and downloading level. The URL Retrieval Engine issues requests to the corresponding Web server and retrieves Web pages. The Web pages are then stored as data files. On the other hand, the matches from the Internet search engines are also retrieved in retrieval layer. The search queries are from the Market Watcher.

The Compilation Layer is the core of the Watcher Agent, which includes the Price Watcher and Market Watcher. The Price Watcher takes the data files and the list of product names. It then detects the matched product names, and extracts price respectively from the Web pages. The Market Watcher is made up of two sub-components, namely, Market Monitor and Market Explorer. The Market Monitor monitors the Web pages for interesting updates and news information as an information filter. The Market Monitor can be set to work repeatedly on either daily or weekly basis. However, there could possibly be overwhelming amount of news on the Web sites. Hence it provides an instant events section. A small module named Instant News Watcher will be designed as a part of the Market

Watcher Agent. The Instant News Watcher can monitor the instant news section from few important Web sites on an hourly base or even every few minutes upon scheduled.

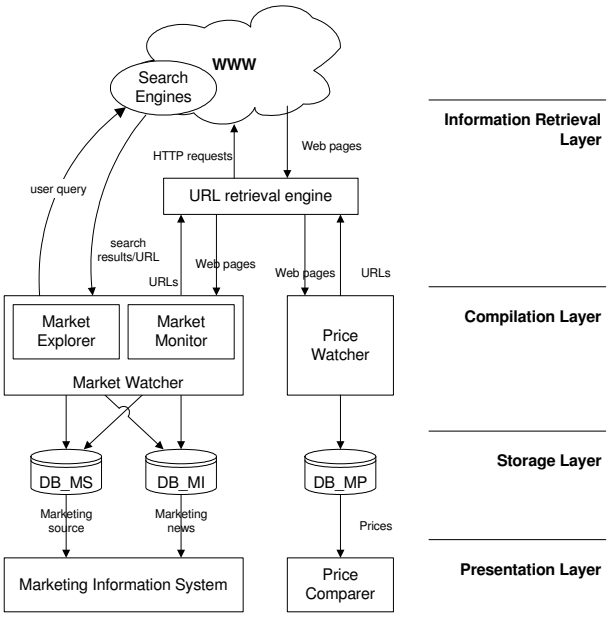

Figure 1 Watcher Agent System Architecture

The Market Explorer is a both retrieval and formatting tool, which passes the user queries to a number of popular search engines available on the Internet and collects the matches from each of the search engines.

The Storage Layer is the local database, which stores price information collected by Price Watcher, the marketing news from Market Watcher, and the searching matches from the Market Explorer.

The Presentation Layer consists of a set of Web pages generated from the local database upon request. Part of the Web interface works for the price comparison of various Web sites on any particular product. For the Market Monitor, a set of market news reports will be generated directly from the database. The instant news report can also be generated upon user's request. The local search engines that directly work on the database are also a part of the presentation layer. These search engines can help user to locate the relevant information from the local database. Since the data is ready to use, local searching will save users' time substantially. For instance, three local search engines can be developed for Market Watcher Database, Price Database, and Market Explorer search history respectively. Search engine for the Market Watcher Database will help user search for the news reports about the query. Search engine for the Price Watcher Database will be able to locate the price of a particular product easily. Users may find relevant information from the search history with the help of a local search engine rather than make a search session over the Internet.

4. MARKET MONITOR

4.1 Information Extraction

One important operation of market monitor is to extract information from competitors' web sites. Given a list of pre-defined competitors' web sites, information about new product release or similar in other area should be extracted using a full or semi-automatic HTML wrapper [7]. A HTML wrapper is a kind of software that extracts a certain paragraph or a section from HTML pages based on the HTML tags, formatting or structural information. HTML wrappers normally make use of rule-based leaner or machine learning techniques to learn how to extract the desired information accurately based on the given sample pages. With such a HTML wrapper, market monitor scans the web sites (given by the users) based on the schedule setting and extracts the detailed news information.

4.2 Information Filtering

Another information resource for the market monitor is from the news portals. Different from the pre-defined the competitor's web sites, majority of the news articles from a new website, say CNN or BBC, are normally not relevant to competitors. Taking an Information filtering approach is necessary to filter out the unrelated information. Filtering is the operational mode in which the queries remain relatively static while new documents come into the system (and leave).

In this filtering task, a user profile describing the user's preferences is constructed. Such a profile is then compared with the incoming documents in an attempt to determine those that might be of interest to this particular user profile. Hence, the filtering approach can be used to select news articles broadcast every day or the newly uploaded Web pages from specific Web sites. One of the difficulties to design such a filtering system is on how to construct the user profile that truly reflects the user's preferences.

Typically, the filtering task simply indicates to the user the documents that might be of interest to him. The task of determining which ones are really relevant is fully reserved to the user. The documents ranking generated by the system may or may not be presented to the user. However an internal ranking is normally computed to determine potential relevancy of documents. For example, the documents with a higher ranking than the predefined threshold may be selected.

4.3 User's Profile Construction

Two approaches, which are introduced in [8], can be described as words "static" and "dynamic". The static approach is to simply take a set of keywords from user to construct the user's profile. The keywords then can be directly compared with the documents arriving at the system. A similar way is used by a number of Web sites like Hotmail where a set of choices are listed to be selected by users. The choices may be personal interests such as sports news, music, or computer news. The result of the selection will be used to construct a simple user profile. The dynamic approach works exactly the same as the static one in the beginning. A set of keywords is required from user to construct an initial profile. As new documents arrive, the system uses the initial profile to select the documents of potential interest and present to user. The user will then go through the recommended documents, select the relative ones

and pass this information back to the system with a feedback process. The system uses this feedback information to adjust the user profile so that it reflects the new preferences just declared. The dynamic approach keeps catching up with the user's feedback and adjusts the profile to be as close to the user's preferences as possible. This is how the dynamic approach will be employed in the Market Watcher Agent.

Since the Market Watcher is designed to be categorization supported, one user profile is constructed for each category. The user profile or category profile in this case, is the keywords given by both the system users. The keywords are called domain keywords for that category.

The dynamic category profile is achieved with a user feedback session. For each Web page recommended by the Market Watcher Agent, there are three choices:

(1) ***worth reading*** - user agreed with the recommended Web page,

(2) ***no comments*** - user did not give any choice, and

(3) ***don't waste my time*** - user disagreed with the recommended Web page, for the user to feedback..

Among the total number of users willing to give feedback, the feedback value is derived with the following formula:

$$\text{feedback value} = \frac{\text{User Agreed - User disagreed}}{\text{Total number of user feedback}}$$

The total number of feedbacks received must be greater than the predefined threshold to make the feedback value valid. In case of the feedback value is high enough (i.e. greater than the threshold), the top matched keyword and the top index term from the Web page will be added to the category profile. The top matched keyword refers to the one from the Web page and has the most number of matches with any of the keywords from user's query. The top index term is the most frequently used index term from the Web page. As a result, the feedback from users is represented by the weight increase of the corresponding index terms in the user query.

4.4 News article filtering based on document similarity

The input for the Market Watcher is the category profile and a set of Web pages. The output is the information indicating which pages are relevant to the user's requirement (Figure 2).

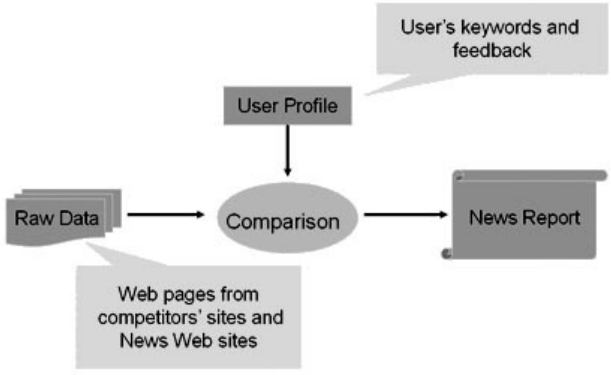

Figure 2 Information filtering process for Market Monitor

The information is stored in the local database and news report can therefore be generated. The information includes the Web page title, updated time, URL, similarity level, and a summary. The summary will be the first one or two sentences of the page as most of the news writers give a summary in the beginning of the news article. The similarity is calculated with the following formula:

$$Similarity = Query \text{ Vector} \times \text{Document Vector}$$

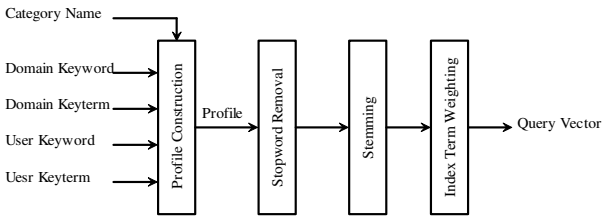

Figure 3 Query Vector Process

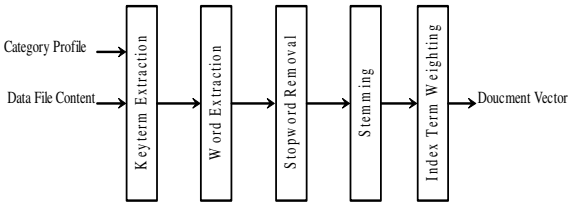

Figure 4 Document Vector Process

Seen from Figure 3, the Query Vector is calculated by the process that can be divided into four steps. Firstly, after obtaining the domain's and user's raw data (Keyword and Keyterm) of a certain category name, system constructs those information into a profile, which is used to run the stopword removal computing in the next step. When the redundancy is obliterated, a new data list is stemmed. In terms of this list, at last, system will give an index term weighting to the data and creates the Query Vector.

Compared with the Query Vector Process, Document Vector Process is a little more complex (Figure 4). Instead of the Profiles Construction process, the input of Document Vector Process is extracted directly from WWW. Also, after running the redundancy removing and index term weighting process the Document Vector is created.

The Term Weight $w_{i,j}$ of index term k_i in document d_j is:

$$w_{i,j} = \frac{freq_{i,j}}{\sqrt{\sum_{i=0}^{t} (freq_{i,j})^2}}$$

The Term Weight $w_{i,p}$ of index term k_i in query q_j is:

$$w_{i,q} = \frac{freq_{i,q}}{\sqrt{\sum_{i=0}^{t} (freq_{i,q})^2}}$$

Thus, the Similarity calculation formula is:

$$sim(d_j, q) = \sum_{i=0}^{t} w_{i,j} \times w_{i,q}$$

4.5 Instant News Watcher

The Instant News Watcher is developed to retrieve the most updated news from homepages of some Web sites where there are instant event sections. The inputs are category profiles and homepages, and the output will be the updated news extracted.

The Instant News Watcher is a special case of the Market Watcher where more frequent monitoring is required. However, the output is the news instead of the summary of the entire page. The entire structure and most of the contents of the homepage of one Web site will not be updated on daily basis. The frequently updated part is the instant news section or instant event section. Therefore the document-ranking algorithm to calculate the similarity level of the profile and entire Web page cannot be applied to the Instant News Watcher.

In the Instant News Watcher design, the content of the Web page is divided into small sections based on its internal structure with help of the Semi-Data Tree Model that has been used in [9, 10]. Each time, one small section, e.g., one paragraph, one table cell, or one list item, is compared to the category profile. Since the input data is not so much different from the entire Web page, the similarity level given by the document-ranking algorithm will be relative low. For this reason, it's hard to set any default threshold. Therefore, the exact pattern matching is good enough in this case. If one section matches any of the keywords or key-terms from the category profile, the section is considered to be relevant and will then be saved to the database. The duplicated sections will be detected before storing to the database in order to save space.

5. Market Explorer

Market Explorer is a part of the Market Information System. The objective of the Market Explorer is to assist users to locate the required information with a number of popular search engines available such as AltaVista, Lycos, Excite, Yahoo, Catcha and InforSeek. Other than news from the competitors' official web sites and popular news portals, information about product review or user feedback cannot be easily extracted from the market monitor.

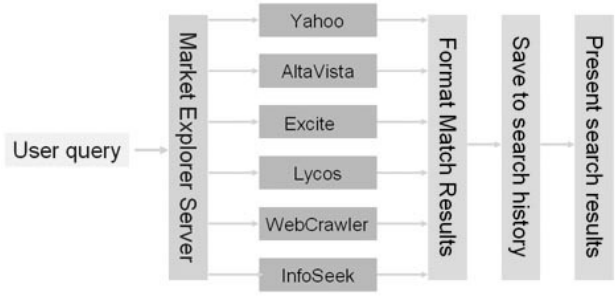

Figure 5 Operational block diagram of Market Explorer.

With the help of such engines (Figure 5), these kinds of information can be located with the user keywords queries. Similar to market monitor, two functions have been incorporated into the market explorer.

5.1 Information Extraction

To extract information from the popular search engines, simple keywords queries need to be derived in advance. The keywords can be product names, competitors' names, product brands or a combination of these. From the top matches of the search engines, say top 100, relevant information can be readily extracted. Each record that has been successfully extracted will be associated with a time stamp and stored in the local database for further analysis.

As the matching records from search engines are dynamically generated from databases, the search resultant web pages are normally in the similar format in terms of HTML structures. A HTML wrapper for each search engine is therefore necessary to extract the match records in the search resultant page. (Figure 6)

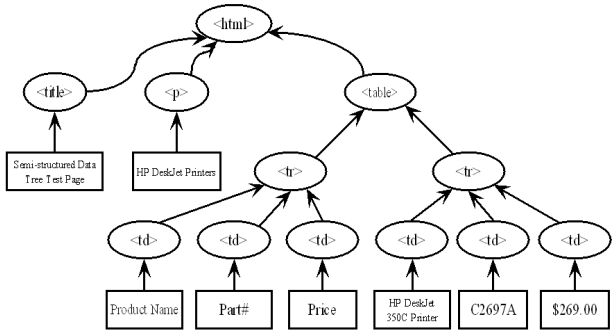

Figure 6 The SDT and search steps through HTML files

Similar to market monitor, the search queries in market explorer will be relatively static and a scheduled information extraction process needs to be conducted frequently, say once a day or once a week. Furthermore, a simple local search engine should be developed in order to navigate the extracted information easily as in [11].

5.2 Product Ranking

With the popularity of the Internet, the World Wide Web has become one of the important information sources for many users. Browsing and search are the two major information access methods. When user wants to find more information about a particular product, a search with the popular search engine is necessary for him/her. Therefore, get to know how easy their products can be accessed from the search engines is important to managers. In market explorer, we have come out with a way of product ranking.

Given a product name, for example inkjet printer, all top N matches from each popular search engine e in the defined search engine list E will be retrieved. For each retrieved match record m, the URL, denoted by m.url, the order in the returned search page, m.o can be easily extracted. For any given manufacture's URL, denoted by p.url, the rank of the manufacture is defined as:

$$p.rank = \sum_{e \in E} (N - m.o) \text{ where } m.url.domain = p.url.domain$$

With such a rank, how easy the competitor's web pages can be reached from the search engines is clear.

6. CONCLUSION

In this paper, we proposed an autonomous software agent called Market Watcher that collects competitors' product prices, news and monitors their updates on the Web. Market Watcher is built as a market research tool for the users at the back-end. They both run autonomously as to relieve monitoring tasks over websites and search engines otherwise to be tediously carried out by human. The collected intelligence information is usually supplied to marketing managers for business decision making. So far this project has implemented up to the monitoring functions. It is envisaged that Market Watcher can be scaled up to include data-mining functions on competitors' information and automatic reporting as well, in the near future. We are in the progress of integrating Market Watcher into a full business intelligence infrastructure

7. ACKNOWLEDGMENTS

The authors are grateful that this work is supported by the Research Council, University of Macau.

8. REFERENCES

[1] L. Liu, C. Pu, and W. Tang, 1999. “Continual queries for internet scale event-driven information delivery”, Knowledge and Data Engineering, 11(4):610–628.

[2] L. Liu, C. Pu, and W. Tang, 2000. “WebCQ: Detecting and delivering information changes on the web”, In Proceedings of International Conference on Information and Knowledge Management, November.

[3] L. Liu, C. Pu, W. Tang, and W. Han, 1999. “CONQUER: A continual query system for update monitoring in the WWW”, International Journal of Computer Systems, Science and Engineering.

[4] J. Naughton, D. DeWitt, D. Maier, A. Aboulnaga, J. Chen, L. Galanis, J. Kang, R. Krishnamurthy, Q. Luo, N. Prakash, R. Ramamurthy, J. Shanmugasundaram, F. Tian, K. Tufte, E. Viglas, Y. Wang, C. Zhang, B. Jackson, A. Gupta, and R. Chen, 2001. “The Niagara internet query system”, IEEE Data Engineering Bulletin, 24(2):27–33.

[5] S. Pandey, K. Dhamdhere, C. Olston, 2004. “WIC: A General-Purpose Algorithm for Monitoring Web Information Sources”, Proceedings of the 30th VLDB Conference, Toronto, Canada.

[6] S. Fong, A. Sun, and K.-K. Wong, 2001. “Price Watcher Agent for E-Commerce”, in Proc. of the 2nd Asia-Pacific Conf. on Intelligent Agent Technology (IAT-2001), pp. 294--299, Maebashi City, Japan, Oct..

[7] S. J. Lim and Y. K. Ng, 1999. “An automated approach for retrieving hierarchical data from HTML tables”, In Proc. of the 8th Inter. Conf. on Information and Knowledge Management, pages 466-474.

[8] B. Y. Ricardo and R. N. Berthier, 1999. Modern Information Retrieval, ACM Press, New York.

[9] S. Chawathe, A. Rajaraman, H. Garcia-Molina, and J. Widom, 1996. “Change Detection in Hierarchically Structured Information”, Proc. of the ACM SIGMOD Int. Conf. on Management of Data, pp.493-504, Montreal, Quebec, June.

[10] S. Chawathe, S. Abiteboul, J. Widom, 1998. “Representing and Querying Changes in Semistructured Data”, Proc. of the Int. Conf. on Data Engineering, pp.4-13, Orlando, Florida, February.

[11] F. Douglis, T. Ball, Y. Chen, and E. Koutsofios, 1998. “The AT&T Internet Difference Engine: Tracking and Viewing Changes on the Web”, World Wide Wide, Vol.1, Issue 1, pp.27-44, Baltzer Science Publishers.