Implementation of Price Watcher and URLs Updater Agents for E-commerce based Mobile Phone Shop

Htwe Htwe Yi, Nwe Nwe
Computer University (Hpa-An), Myanmar
htwehtweyeemlm@gmail.com

Abstract

This system is designed to help the users for retrieving the product and price information of the competitors' website and monitor updated information in the competitors' website for the most frequently interested product items of phone store web site. There are intelligence information agents in the system, namely, Price Watcher Agent, URLs Updater Agent, and Online Phone Store Agent. Price Watcher Agent is used to retrieve competitors' product prices over the World Wide Web, for the purpose of price comparison at an e-commerce retail shop. It collects competitors' price information. Price Watcher Agent is used Keyword-Patten-Sample based mining algorithm (KPS) to extract price values from other competitors' web sites. Competitors' URLs Updater Agent is the information collection tool that assists the shopper to spy on the competitors' website address (URLs) for the product information of competitors' web pages. Online Phone Store Agent is comparison between phone store product price and other competitors' product price.

1. Introduction

The explosion of Internet and the ensuing applications in electronic commerce (e-commerce) have permanently changed the outlook of traditional business trading behavior. As a result of growing e-commerce, online business are more competitive each other. One of the barriers for e-commerce retailers to overcome is that most consumers are not convinced that the price of a product offered at their sites is the best; and it is always easy for them to surf away to other shopping sites looking for a better offer. This system can overcome this problem by spying competitors' prices over the web and displaying the compared result to the consumer, where it can be configured such that only the prices higher than (or equal to) ours are displayed. This process is performed by the agent called, Price Watcher.

Moreover, there are tremendous amount of business information on prospects, competitors, vendors, suppliers, customers or other companies. These monitoring and collecting of competitors' information would usually employ a team of staff manually. This system also contains another agent, URLs Updater. It can provide automatically collecting and reporting the information of the specified competitors' URLs addresses, and then stored in local database for our site. By using these two agents, this system can improve managers' business decision making and enhances shoppers' confidence.

2. E-commerce

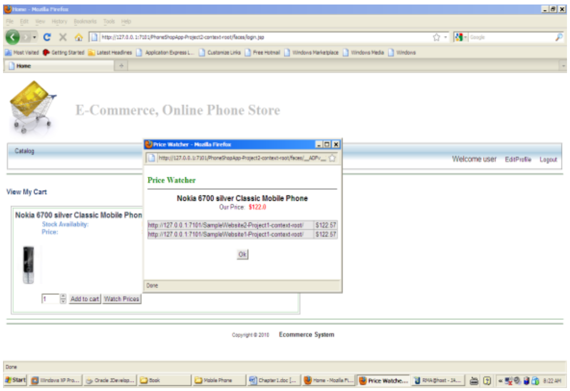
Electronic commerce is sharing business information, maintaining business relationships, and conducting business transactions by means of communication networks. E-commerce includes the relationship between companies (business-to-business), between customers (customer-to-customer), as well as between companies and customers (business-to customer).

Many retail outlets have begun using electronic data interchange systems—the Internet and e-mail—to sell goods and services, a practice called electronic commerce, or e-commerce. In fact, some retailers are opting out of the traditional storefront format entirely to participate in e-commerce, citing lower costs to both businesses and consumers alike.

As online shopping continues to grow, it is likely that an increasing number of buyers will use e-commerce services. Online mobile phone shop purchases must be shown to be as safe and of the same quality as in-store transactions. Until this aspect of online phone shop is traditional buying formats will continue to dominate both the marketplace and marketspace [1].

3. Online Phone Store Agent

Online Phone Store Agent is comparison between phone store and competitors' product price. In this system, admin is authorized person who is permitted to hold this phone store. In administration, the authorized user can entries online phone store and then visiting on phone store website, while browsing our products, can click on the "watch price" button to check how our products are better priced than others. This kind of on-the-spot price comparison provided by Price Watcher is intended to convince online shoppers that the offered price is indeed the best that can be found on the Internet [3], thus discouraging them from going away to the other sites. A snapshot of the application of Price Watcher from the prototype is shown in Figure 1. The agent can be configured such that only the prices higher than (or equal to) ours are displayed [7].



E-Commerce, Online Phone Store

Nokia 6700 silver Classic Mobile Phon
Stock Availabity:
Price:

Price Watcher - Mozilla Firefox
http://127.0.0.1:7101/PhoneShopApp-Project2-context-root/faces/__ADFv...

Price Watcher

Nokia 6700 silver Classic Mobile Phone
Our Price: $122.0

http://127.0.0.1:7101/SampleWebsite2-Project1-context-root/	$122.57
http://127.0.0.1:7101/SampleWebsite1-Project1-context-root/	$122.57

Copyright © 2010 Ecommerce System

Figure 1: Snapshot of the application of Price Watcher

Competitors' URL addresses already exist in local database. Updater agent is downloaded update URLs address of competitors' web pages on daily basis from web and then stored in local database. Online Phone store agent send user selected product name. Price watcher agent receive product name from phone store agent. Price watcher agent is retrieve URLs address of competitors' web pages from local database. Price Watcher agent spies on prices of user selected product from competitors' web pages for E-commerce Online Phone Store. Find prices related with the product name from the competitors' website (web pages) and then reply competitors' product prices to phone store agent. Work flow of the agent communication is shown in figure 2.

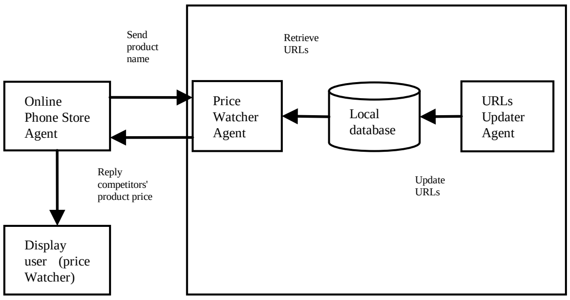


Figure 2: Work flow of the agent communication

Online Phone Store Agent display user (price watcher) selected product price, competitors' URLs address and competitors' product price, as shown in figure 1.

4. Price Watcher Agent

Price Watcher Agent is an autonomous software program that takes the competitors' URLs and spies on price of some selected products. Price Watcher takes the responsibility of retrieving prices of products from competitors' websites. The prices collected from the competitors are stored in a local database. They can be used for price comparison at the front-end of e-commerce online phone store as well as for market research at the back-end.

There are several price comparison services available on the web [2]. The main differences between the conventional price watcher agent and our price watcher agent are follows [7]:

- Designed for individual online shops.
- Neither broker nor public database is used.
- No participation of retailing shops is required.

4.1 Design of the Price Watcher Agent

Price Watcher Agent is an autonomous software program that takes the competitors' URLs and spies on prices of some selected products.

1. The set competitors' URLs obtained from local database.
2. HTML pages are retrieved using these URLs over the World Wide Web by using Web retrieval engine
3. Only HTML pages containing dollar sign like $ are retrieved.
4. The product name (Nokia N97) will be matched by using string matching algorithm. If they have been match in the html page, price extracting will be performed by using KPS (Keyword, Pattern, and Sample) web mining algorithm.

In price watcher, only the HTML texts are downloaded [5]. Finding the level of similarity between our product names and the names provided on the web, as well as extracting the corresponding prices is the two challenges facing us.

4.2 Keyword-Pattern-Sample (KPS) web mining algorithm

The World Wide Web provides a vast source of information. The Web mostly contains semi-structured information. It is, however, not easy to search and extract structural data hidden in a Web page. Compare with traditional databases, Web information is dynamic, semi-structured and interwound with hyperlinks. Querying over the World Wide Web is significantly different from querying the data from traditional databases, e.g. relational databases, which are structured, centralized and static.

KPS algorithm mines the desired information from irregular pages directly using keywords, patterns and/or samples. KPS is one of the Web Information Mining Algorithm and more efficient than any conventional approaches. This algorithm is used to extract information from the semi-structured web pages over the World Wide Web [9].

- Keyword-based mining is used to extract a value associated with a keyword.
- Pattern-based mining performs string matching on the WWW based on the patterns specified by the users.
- Sample-based mining method extracts information based on a sample specified by the user [9].

• Using Keyword Based Mining

Then we apply the following four rules to detect the location of the price in the HTML pages.

1. If the keyword (Product Name) is included in a title (ie. <H1> - <H6>), the target information (Price) is the string after it until the next title.
2. If the keyword is an item of a list (i.e. <LI> or <UI>), the target information is the string after it until the next <LI> or <UI> or the end of the list.
3. If the keyword is in a cell of a table, the price is most likely to be located in the same cell. Or if the keyword is a field in a table, the target information is the field on the right for the first column of 2-column tables. Otherwise, it is the field under it.
4. If the keyword is in textual line, the price is most likely to be located in the same paragraph, or the next paragraph, until the end of the page.

Extract the fragment of HTML code that contain price and then price extraction will be performed by using sample based mining.

• Using Sample Based Mining

- After determine the start and end of the location of the price in the HTML page, split this part into a list of segments.
- Calculate the Pattern Similarity Measure (PSM) and Pattern Similarity Score (PSS) for each segment.

Pattern Similarity

Pattern similarity measures how much two objects match with each other. For objects *p* and *q*, the *pattern similarity measure,* denoted by *PSM.*

(1) PSM - *the maximum size of the sub-objects* p_i *and* q_j *such that* $p_i = q_j$

For example, for objects *p*=(Phone, :, (, 306,), 781, -, 7488) and *q*=(Tel, :, 1, -, 306, -, 781, -, 7453), the *maximum matched sub-objects* are: (:, 306, 781, -, 7488) = (:, 1, 306, -, 781). Thus, *PSM* (*p*, *q*) =5. In order to reduce the effect of *false matches* between large objects, such as *The student is called as Tom* and *A student is reading a book written by Tom.*, we first consider the *ratio* of *PSM* to *the average size* of the objects to reflect the real *pattern similarity*. However, for two objects whose *average size* is 10 and *PSM* is 5, the *ratio* is 50% which is the same as that of objects whose *average size* is 2 and *PSM* is 1. In practice, the former should have a higher *similarity* since the one *matched field* in the latter may be a *false match*. Therefore, we multiply the *ratio* by the *PSM* to reflect it. The final result is called *pattern similarity score* (denoted by *PSS*).

(2) $PSS(p,q)=(PSM(p,q)/((size(p)+size(q))/2))* PSM(p, q)*100$

In the above example, $PSS(p, q)=(5/((8+9)/2))*5*100= 290$; If we assume the object *p* is the *sample,* then the *PSS* is enough to distinguish the object *q* from other parts. However, if there is another object *l*=(Fax, :, 1, -, 306, -, 781, -, 7453) (e.g. in the second example in the *Appendix*), it is hard to determine which (*q* or *l*) should match with *p* since *PSS*(*p*, *q*)= *PSS*(*p*, *l*).

A *segment* is a list of continuous fields in a web page, but does not contain HTML tags. If we consider all HTML tags as *segment delimiters* in web pages, then the body (between <body> and </body>) can be viewed as a list of segments. (Since the *head* (i.e. the text between <Head> and </Head>) of a page usually cannot be viewed by users in a browser interface, we thus do not consider it in the extraction process). Then, we define **Pre-marker** and **After-marker** which are used to identify the *beginning* and *end* of an object. [9]

- After calculating PSS for all of the segments, choose the highest score value as the price value and extract it and store the price and product information in the local database.

5. Competitors' URLs Updater Agent

Competitors' URLs Updater agent is designed as a multi-featured monitoring tool for retrieving relevant information from the competitors' websites. There are many similar commercial products on the markets that help retrieving competitors' information online. Updater agent differs in a way that it is specially built for monitoring sections of the competitors' websites online. Updater agent is specialized to retrieve information that is relevant to our sales team. The information include competitors' product prices, company news, product news, their mentions and their search result rankings from the most popular search engines. With updater agent, sales team will get the very latest and timely information about the competitors' movement on their websites [6].

5.1 Design of the Competitors' URLs Updater Agent

The Competitors' URLs Updater agents are autonomous agent that helps the administrator of the online shop get the latest information about his competitors' web sites.

- Checks if specified URL (URL from competitor) is valid for download.
- Updater agent delete pages related to current competitor from local database.
- Prepare variables for the current competitor.
- It's also automatically collects the competitors' URL addresses and stored in local database on daily basis.
- The input is the competitors' URL address and the output is the URL addresses of the new relevant competitors' web pages.
- The scheduler is downloading on daily basis for update competitors' web pages.

- The final step is downloaded update URLs address of the competitors' website (web pages) and then stored in local database.
- Therefore this updater agent always keeps the local database up-to-date.

6. Agents used in the system

A number of models exists for describing agent systems; however, for discussing phone store issues it is sufficient to use a very simple one, consisting of only two main components: the agent and the agent platform. Here, an agent is comprised of the code and state information needed to carry out some computation. Mobility allows an agent to move, or hope, among agent platforms. The agent platform provides the computational environment in which an agent operates. The platform from which an agent originates is referred to as the home platform, and normally is the most trusted environment for an agent. One or more hosts may comprise an agent platform, and an agent platform may support multiple computational environments, or meeting places, where agents can interact. An application based on JADE (Java Agent Development Environment) is made of a set of components called ***Agents*** each one having a unique name. Agents execute tasks and interact by exchanging messages. Message format is defined by the **ACL** language defined by FIPA, an international organization that issued a set of specifications for agent interoperability [8].

In the network environment, there are so many hosts owned several agents. But these hosts will have their own agent platform (Home platform) or remote platform. Online Phone Store Agent at home platform and Price Watcher and Agent at remote platform. Phone store agent launches this agent on home platform by means of local-host and local port. If a certain user who wants to watch product price from competitors' website (web pages), he/she will create price watcher agent, and then connect phone store's agent platform (remote platform) by means of remote host and port. Phone store and price watcher agents will meet on agent platform, communicate each other.

6.1 Three main agents used in the system

There are three main agents used in the system: Online Phone Store Agent**,** Price Watcher Agent**,** and URLs Updater Agent. Two agent interact with each other in the system are Online Phone store agent and Price Watcher agent. In this system, it deals responsible to communicate between agents of online phone store and Price watcher service. The agent of phone store request the remote message phone name and then agent of Price watcher service response price list of phone according its repository. The price watcher service can provide the list of price related with the client provide product name. URL Updater Agent is autonomous agent that automatically and manually downloaded the competitors' web pages based on pre-defined schedule setting (on a daily basis) [7]. It allows downloading a World Wide website from the Internet to a local database, building recursively all directories, getting html, images, and other files from the server to the computer. URLs Updater agent downloaded competitors' web pages for Price Watcher agent.

6.2 Analysis of the system

Only the competitors' prices that are more expensive than price of the main page are retrieved from local database. Then these competitors' price and main page of price are shown in a tabular form. So, advantages of these system takes as a result of showing more expensive prices of the competitor, the shop can get more customers over their competitors [2]. It will also enhance the shopper’s confidence via price comparison. Moreover, because of the manager can always check the competitors’ product prices, it will help to switch the product prices in the market.

7. Design of the system

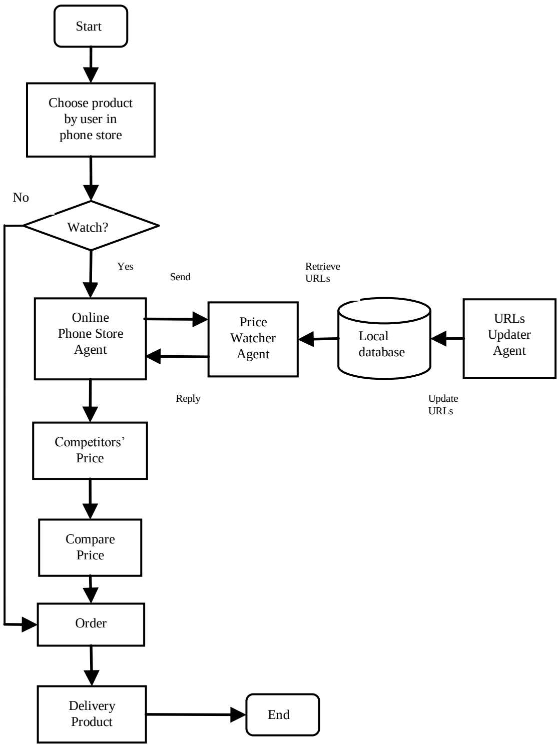


Figure 2: Design of the overall system architecture
The system working process:

- User visiting on phone store website, while browsing our products, can click on the "watch price" button to check how our products are better priced than others.
- Phone store agent send message to the price watcher agent to find prices related with the product name
- Send phone name to price watcher agent.
- Price watcher agent receive product name from phone store agent.
- Find prices related with the product name from the competitors' website (web pages)
- Reply prices to the phone store agent
- Phone store agent receives prices from the price watcher agent.
- And then, Phone store agent is comparison between store product price and competitors' product price.
- User is order and delivery product in Online Phone Store.
- URLs Updater Agent is downloaded URLs address of competitors' website (web pages) and then store in local database for Price Watcher Agent.

8. Conclusion

Agents are a key component in the Internetwide information and electronic commerce systems that are currently being developed across the globe. But it is not an easy task to predict how agents will develop into the future. There is still a long way before software agents transform how businesses conduct business [1]. The greatest change will occur once standards are adopted and evolved to unambiguously and universally define goods and services, consumer and merchant profiles, value-added services, secure payment mechanisms, complex goals, changing environment, etc.

This paper, presented an autonomous software program that collects competitors' product prices, news and monitors their updates on the web. Price Watcher is designed for the use of an e-commerce shopping site at the front-end, and Competitors' URLs Updater is built as a market research tool for the users at the back-end. A proactive use of the price information is to enhance shoppers’ confidence by doing an online price comparison. The application of Price Watcher technology is believed to be relatively new and would create an impact on the way that retail shops market their goods online. The first online shop that applies this technology would benefit most, because it helps to place their business in a market position one step ahead of their competitors. So far this project has implemented the scanning functions. It is envisaged that system will be extended to include data-mining functions on competitors' information and automatic reporting as well, in the near future [7].

9. References

[1] Aleksander Pivk, Matjaž Gams, Intelligent Agents in E-commerce. Jozef Stefan Institute, Department of Intelligent Systems, Jamova 39, 1000 Ljubljana. E-mail: aleksander.pivk@ijs.si, matjaz.gams@ijs.si

[2] Andrew Pollard, "Competitor Intelligence: Strategy, tools and techniques for competitive advantage", Financial Times Management, 1999.

[3] B. Krulwich, "The Bargain Finder Agent: Comparison price shopping on the Internet", Bots and other Internet Beasties, SAMS.NET, http://bf.cstar.ac.com/bf.

[4] B. Doorenbos, D. Etzioni, S. Weld, "A Scalable Comparison - Shopping Agent for WWW", Technical report 96-01-03, Univ. of Washington, Department of Computer Science and Engineering.

[5] J. Allan, R. Papka and V. Larrenko, On-line New Event Detection and Tracking, ACM SIGIR '98, pp.37-45, 1998

[6] M. Pazzani, D. Billsus, Adaptive Web Site Agents, ACM Autonomous Agents '99, pp.394-395, 1999.

[7] S. Fong, A. Sun, K. Wong, Price Watcher Agent for E-Commerce, Intelligent Agent Technology, pp.294-299, 2001.

[8] Sensor Web – TITAN Research, Agent Communication, 12/5/2010.

[9] T. Guan and K. F. Wong, KPS: a Web Information Mining Algorithm. Computer Networks 31(11-16): 1495-1507, 1999.