

A Personalized Stock Investors Alert System

Jameela Al-Jaroodi and Nader Mohamed

The College of Information Technology, UAE University, Al Ain, UAE

{j.aljaroodi, nader.m}@uaeu.ac.ae

Abstract – Stock investors all agree that one of the hardest things they have to do is keep up with the rapid changes in the markets. The Internet carries a huge amount of up-to-the minute information for investors such as current stock prices, exchange rates, sales volumes, latest deals, trading averages, organizations' performance and financial information, oil and other goods prices, and much more. Although such stock-related information is publicly and freely available on the Internet, investors face difficulties tracking the changes and updates fast enough to make sound investment decisions. This paper proposes a system to monitor stock and other stock-related information available from web services or dynamic HTML documents on the Internet. This system can deal with public information available on the Internet on a single or multiple websites and from a single or multiple web services as the information sources. The system allows investors to use these sources to create personalized stock monitoring criteria and then receive updated notifications from the system based on the specified criteria. The public information is treated as a set of Internet variables used to define stock monitoring criteria based on the specific investor's requirements. Based on the defined criteria, SMS and/or email notification messages can be generated and sent to the investor notifying him/her whenever the criteria are met. A number of applications related to stock investments can benefit from this system to provide real-time useful information to the investors.

Index Terms – Information Retrieval, Notification Systems, Web alerts, Web Information, Stock investors.

1. INTRODUCTION

Mr. StockWiz is an avid stock investor who invests large amounts of money in a diversified stock portfolio. He has to follow on information from various sources to identify different possible options to run his investments. In addition to the local stock market which provides Mr. StockWiz with some essential information, he relies heavily on the Internet to get more detailed and up-to-the-minute information. The Internet is one of the main sources of dynamic real-time investment-related information and generally there are three approaches that Mr. StockWiz uses to get the desired information. He uses browsing to scan through multiple pages and read/retrieve some useful data. Mr. StockWiz also relies on search engines to find relevant information. In addition, he uses the notification services available on some information sites such as Yahoo Finance [1]. In those sites, Mr. StockWiz selects some criteria from the

available set and requests to be informed when the changes occur. All the information Mr. StockWiz needs is out there; however, it is very hard to go through all of it and find the required data fast enough to make sound investments decisions. Figure 1 shows the decision process for Mr. StockWiz based on his personal rules.

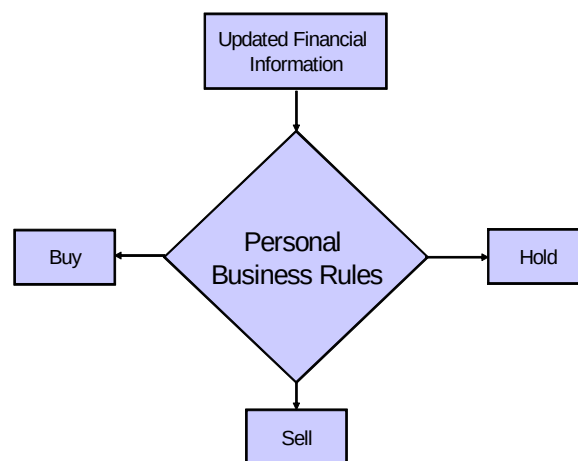


Figure 1. Investors decision Process.

Current notification systems may provide an efficient way to zone in onto relevant data; however, they lack the flexibility and diversity. Each publisher provides a limited set of sources to obtain data from. In addition, publishers currently only provide simple and limited possible criteria for the notification. For someone like Mr. StockWiz, these are not sufficient. For example, he can get notification if a specific stock reaches a specified value, yet it is impossible to set up a notification for a complex condition such as: when the stock price reaches \$Y and the sale volume is lower than Z stocks or the closing average sale price over the past month is \$W. It may also be impossible to get a notification involving changes in two or more different publishers such as asking for notification based on stock price changes on Yahoo Finance and currency exchange rates on another site.

In this paper we provide Mr. StockWiz with a more flexible solution that will allow him to be notified based on his own criteria. Here we develop a personalized notification system for stock investors. Unlike other notification systems provides, this system is not tied to a single information provider, it is based on all the public information available on the Internet. It uses multiple

websites and/or multiple web services as the information sources of which the investor can build his/her stock monitoring criteria. Using this information and some knowledge of how it is represented on the web will allow the user to create detailed, simple (or complex) conditions to be monitored and used to supply the notifications.

In this paper, Section 2 provides related work and background information. Section 3 discusses how updated stocks information can be obtained from the Internet and Section 4 describes the developed system. Section 5 provides some stock notification applications of the approach and Section 6 concludes the paper.

2. BACKGROUND AND RELATED WORK

An alert system is a combination of network services and infrastructure that provides a mechanism of delivering alert messages to a single or multiple recipients based on the occurrence of events specified by the recipients. Notification systems usually consist of two players, the publisher/producer and the subscriber/consumer [2]. There are different notification systems used for different domains and applications such as telemedicine applications [3], accrual failure detectors [4], e-commerce applications [5], and emergency services [6]. A lot of literature is available on notification systems representing different projects, protocols, and architectures. Examples are CORBA notification service [7], Notification for Open Grid Services [8], web services based notification [9][10][11], and scalable Internet event notification architecture [12].

There are many public alerts services available over the Internet. Users can use these services to define web alerts for information they have some interest in. One of the services is Google Alerts [13] which provides search-based notification. The user defines keywords for a topic of interest and email alerts will be sent whenever any article is published over the Internet about that topic. This service can be used to get up-to-date information about the user's interests. This type of search covers news, blogs, video, or groups.

Other alert services are available to monitor changes in web pages. One example is ChangeDetect [14]. ChangeDetect is a service that monitors web page contents for changes and sends an automatic email alert to the user whenever the web pages are updated. A user can use that service; for example, to monitor new news for a company or to monitor if there are new position openings listed in a webpage of a company.

Another alert service is Yahoo Alerts [15], which provides alert services for different type of interests such as breaking news, fantasy sports, games, health news, hot jobs, weather, traffic, stocks, etc. The stocks alerts provide users with some options to set notifications for stocks information. These options are when stock prices increase or drop with specific amounts or percentages. For example, the user can receive an alert whenever the Wal-Mart stock price increases above \$50.00 or when the

price drops below \$44.00. In addition, the user can choose to receive alerts when the price increases or drops by a certain percentage.

Although, there are some alert services for stocks information, these services are very restricted. The user is restricted to specific types of information and limited types of alerts. Users have no means to define advanced criteria involving multiple values or sources. In addition, the data used for notification is usually limited to what is available on the website of the company managing the alert system. For example, an investor can set an alert condition in Yahoo Finance such that whenever the Wal-Mart stock price drops to a specific value in US dollars an alert is sent. However, the investor can not define advanced alert criteria such as "when the trading volume reaches a specific quantity and when the Wal-Mart stock price reaches to a specific price in Euros;" although, both the trading volume and the USD to Euro exchange rates are available over the Web. Yet Yahoo Finance does not support that type of alert then an investor can not define these advanced criteria

The main difference between our system and other web-based systems is that our system does not have specific publishers of information or events; instead we expand the source of information to every public-domain website. Therefore, all public Internet sites can be considered as publishers for our notification system and all the information available on these sites can be used to build the notification rules. In other words, the users will be able to define notification rules based on information available and updated on any website on the Internet or based on information provided by any web service.

3. STOCKS INFORMATION

The proposed system requires first to access and use stocks information available on the Internet. Therefore, the main component is the monitor, which depends on a set of variables (carrying information from the Internet) defined by the users. Going back to our investor Mr. StockWiz, if he wants to know the price of Wal-Mart stock in Euros, he can define a variable for Wal-Mart current stock price and another variable for US Dollar to Euro exchange rate. Using these two variables, the monitor will find and store the values to be used by the system to calculate the required values. The Internet variable content's sources are from dynamic HTML documents [16], XML documents [17], or from web services [18] available on the Internet. This Section discusses the three main ways for defining Internet variables depending on their sources.

A. Information available in Web Services

Web services can provide a structured and simplified way to obtain services or specific information from the Internet. A Web service is defined by the World Wide Web Consortium (W3C) as "a software system designed to support interoperable Machine to Machine interaction

over a network." Web services provide web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. These remote systems support different services including providing information about different aspects or products. For example, different stock markets can host web services to provide current stock prices. Banks can use web services to provide information about loans or information about foreign currency exchange rates. The W3C Web service usually consists of clients and servers that communicate using XML messages that follow the SOAP standard. Web services provide just-in-time integration mechanisms to integrate clients and servers. If desired information is needed by a monitor application, then the user can easily define an Internet variable and link it with the corresponding web service which provides the required information. The main problem with web services is that not all types of information available over the Internet are provided using Web Services.

B. information available in XML documents

Another source of information on the Internet is XML documents. These offer a structured format for data in text-based documents that user programs can scan through and locate the required data easily. The data items in an XML document are associated with special tags that define the data semantics. This allows the monitor to retrieve the required information based on their defined tags. For example, in an XML document the stock price for a company will be included along with a unique tag identifying the company and the type of data (price). Therefore, the user can write a program that will locate the tag and use the data available with it. However, like web services, XML does not offer a large source of information since only a small percentage of the information available on the Internet is written in XML.

C. information available in HTML documents

Most of the Internet information is delivered to users in HTML documents. Unlike XML, HTML documents do not have any semantics for their data. Obtaining specific data from a dynamic HTML document for reuse in other applications can be a complex task. It is very difficult to identify the required parts of the data and dynamically use it in other applications.

We have recently developed a simple and efficient approach for retrieving live HTML-based Internet information [19]. This approach is used to define the notification variables that will have their data updated from the Internet. The main idea is based on finding fixed titles or headers that appear in browsers for HTML documents directly or semi-directly before the needed dynamic data. These fixed headers are used as references to know the position of the required dynamic information. A number of techniques were developed to find the information in an HTML document. These techniques are implemented in a set of methods (See Table 1).

TABLE 1.
Methods used to retrieve data from an HTML documents.

Method	Description
<i>get(header)</i>	To return the next field directly after the defined <i>header</i> . The search starts from the beginning of the page.
<i>get(n,header)</i>	To return the next field directly after the defined <i>header</i> appears <i>n</i> times. The search starts from the beginning of the page.
<i>get(n,header, i)</i>	To return the field after skipping <i>i</i> fields after the defined <i>header</i> appears <i>n</i> times. The search starts from the beginning of the page.
<i>get(header1,header2)</i>	To return the field that is located between <i>header1</i> and <i>header2</i> . The search starts from the beginning of the page.
<i>getWI()</i>	To return the next field from the current read pointer position.
<i>getWI(i)</i>	To return the field after skipping <i>i</i> fields from the current read pointer position.
<i>getWI(header)</i>	To return the field located directly after the specified <i>header</i> from the current read pointer position.
<i>getWI(n, header)</i>	To return the field after the occurrence of the <i>header</i> <i>n</i> times from the current pointer position.
<i>getWI(n, header, i)</i>	To return the field after skipping <i>i</i> fields after the defined <i>header</i> appears <i>n</i> times from the current pointer position.

The first technique is to find information directly after a specific text header. The second technique is to find certain information after the appearance of a specific header for the n^{th} time. The third technique is to find information semi-directly after a specific fixed title or header. The three techniques search for the required information from the beginning of the HTML page.

This approach will be fine if only a single piece of information is needed from the page. However, in most cases, multiple information fields are needed from a single web page. There is no sense in initializing the page read pointer after each *get*. It is better to find the first field in the page and then continue from that position to find the second one appearing on the page, and then the third, etc. This saves processing time. To achieve this, five techniques/methods were developed: *getWI()*, *getWI(j)*, *getWI(header)*, *getWI(n, header)*, *getWI(n, header, i)*, where *WI* stands for Without read pointer Initialization. More information about the implementation and performance of the approach can be found in [19].

4. THE STOCK INFORMATION MONITOR

The system is developed as a client that monitors one or more dynamic stocks information from one or several web servers on the Internet as shown in Figure 2. The user can define the needed Internet variables as discussed in the previous section. These variables are updated by the system periodically by downloading the HTML documents defined by the user and extracting the required fields or calling the corresponding web services. The user can add functions to derive additional calculated variables from the defined Internet variables. The system periodically checks the set of notification criteria defined by the user. Whenever one of the criteria is met, a predefined notification message will be sent to the user.

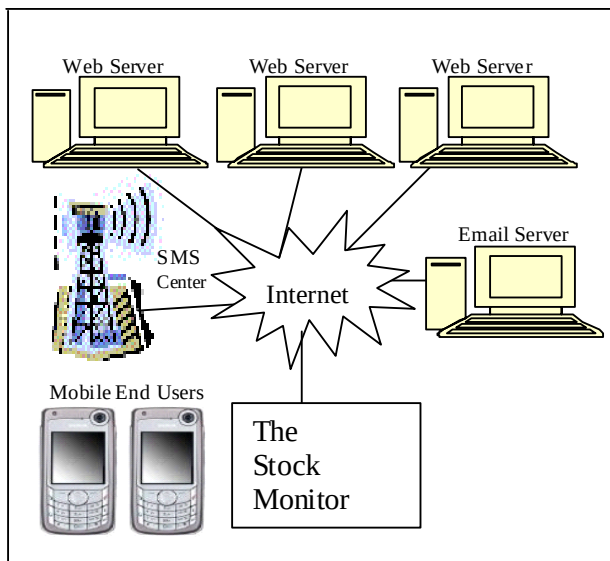


Figure 2. The Monitor works as a client monitoring the Internet and sending notification messages for end users.

The user defines one or more notification criteria. With each notification criteria, a notification message should be defined. The message consists of some static text defined by the user and some Internet variables that are extracted from the Internet. In addition, each message can have some variables that are calculated from the defined Internet variables. The variables can be embedded in the user defined text. Furthermore, the user needs to define the emails or phone numbers where the notification message should be sent.

A prototype system was implemented in Java to demonstrate the concept. The main component of that system is a monitor program, which executes the notification policies defined in a configuration file. Users can define their notification requirements using a Java-based configuration file. Variety of notification policies can be implemented without changing the monitor program itself. However, only Java code configurations need to be developed. An example of a user defined configuration is shown in Figure 5. This configuration file is a simple case needed by Mr. StockWiz who is currently using the Euro in his trainings and would like to sell *Emaar* stock listed in UAE in Dirham if the price goes above 2 Euros.

Two types of information are needed, the stock price in Dirham offered by the market website shown in Figure 3 and the Dirham-Euro exchange rate offered on a different website shown in Figure 4. Both values are dynamic and Mr. StockWiz will need to continuously take note of the updated price and convert the currency to see if the 2 Euro price was reached. However, using our system Mr. StockWiz will only need to specify the two variables: the current stock price in Dirham, *EmaarAED*, and the current exchange rate from Dirham to Euro, *AEDEUR*. Then he uses these two Internet variables to generate a calculated variable, *EmaarEUR* =

*EmaarAED***AEDEUR*, for the current stock price in Euro. In addition, he needs to specify a condition at which a notification message will be sent, which is if *EmaarEUR* becomes more than 2 Euros.

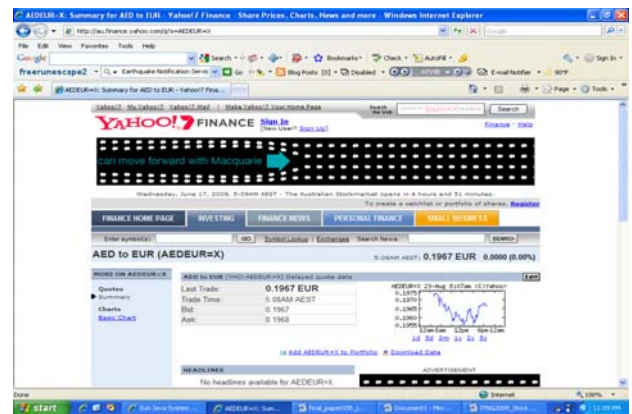


Figure 3. Dubai Financial Market Website.

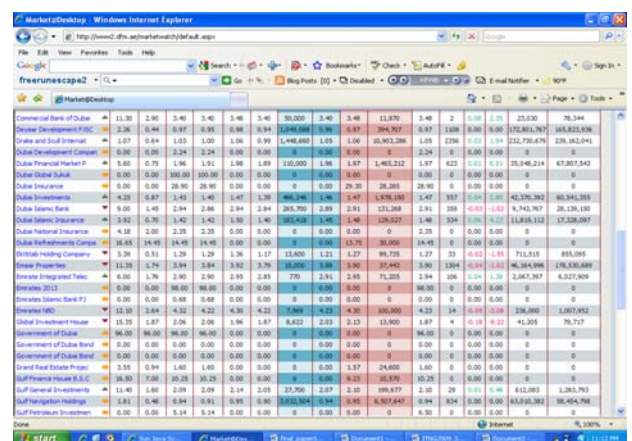


Figure 4. Dirham to Euro Rate from Yahoo Finance.

All Internet and calculated variables should be declared as attributes in the user defined configuration file. The user also needs to define all Internet and calculated variables in a method called *VARIABLES* and the notification policies in methods that start with the *NOTIFICATION* prefix as shown in Figure 5.

In the notification methods, Mr. StockWiz defines the notification conditions, notification messages, and notification destinations as shown in Figure 5. The user also needs to define values of two static attributes: the email server address, *EmailSERVER*, where emails can be sent through and *INTERVAL* which defines the rate at which Internet downloads should be done by the monitor. When the user defines the notification configuration file it should extend the *GenericNOTIFICATION* class, which provides support for creating and sending messages.

```

// Notification File: Emaar price in Euro. This notification is to:
// - get Emaar price in Dirham from Dubai Financial Market website
// - get Dirham to Euro Exchange rate from Yahoo Finance website
// - calculate current Emaar price in EURO
// -send notification Email and SMS message if: The current Emaar price is 2 or more Euros
class EmaarNotificationInEuro extends GenericNOTIFICATION
{
    // define the initial parameters
    public static String EmailSERVER="webmail.uaeu.ac.ae";
    public static int INTERVAL = 2;
    // Declare all Internet and calculated variables
    private double EmaarAED;    // Emaar current price in Dirham
    private double AEDEUR;      // Dirham to Euro currency exchange
    private double EmaarEUR;    // Emaar current price in Euro
    public void VARIABLES()
    {
        // Get Emaar price from DFM
        urlINFO dfm = new urlINFO("http://www.dfm.ae/marketwatch/default.aspx");
        dfm.download();
        String _EmaarAED = dfm.get(1,"EMAAR Properties",10);
        EmaarAED = Double.parseDouble( _EmaarAED );
        // Get Dirham to Euro exchange rate
        urlINFO exchangeRate = new urlINFO("http://au.finance.yahoo.com/q?s=AEDEUR=X");
        exchangeRate.download();
        String _AEDEUR = exchangeRate.get("Last Trade:");
        AEDEUR = Double.parseDouble( _AEDEUR );
        // Calculate Emaar current price in Euro
        EmaarEUR = EmaarAED * AEDEUR;
    }
    // notification #1 condition and message
    public void NOTIFICATION1()
    {
        if( EmaarEUR > 2.0 )
        {
            // Prepare a notification message
            String M1 = message("Emaar price is %1% Euro",EmaarEUR);
            sendEmail("nader.m@uaeu.ac.ae", "Emaar Price",M1);
            sendSMS("+971507533620",M1);
        }
    }
}

```

Figure 5. Notification example for Emaar when the price researches 2 or more Euros.

The monitor will read the user defined file to execute the process of getting the defined Internet variables and calculating other variables. It executes all defined notification methods through multiple threads where each thread will start after the defined *INTERVAL* to execute the defined *VARIABLES* and notification methods. If the *INTERVAL* is two, then a thread will start every two seconds. The *INTERVAL* can be set based on the rate the required information changes. The monitor will not execute a notification method if the corresponding condition was met during a current day and notification messages were sent. If the required information is in web services, the user uses an interface that accesses web services instead of the HTML interface, *urlINFO*.

The monitor uses the reflection API [20] to handle the user define configuration class. With the reflection API, it is possible to determine the class of an object and get information about a class's modifiers, fields, methods, constructors, and superclasses. It is also possible to know what constants and method declarations belong to an interface, create an instance of a class whose name is not known until runtime and get and set the value of an object's field, even if the field name is unknown to your program until runtime. In addition, one can invoke a

method on an object, even if the method is not known until runtime. The monitor utilizes these features to read the values of *EmailSERVER* and *INTERVAL* and execute the defined *VARIABLES* and all notification methods.

The monitor sends messages through either emails or SMS messages. The monitor utilizes JavaMail APIs [21] for sending emails and SMSLib [22] for sending SMS messages. SMSLib is a java library for sending and receiving messages via a mobile phone or GSM modem.

5. STOCK NOTIFICATION APPLICATIONS

Using the proposed system, real-time personalized stock alert applications can be easily developed to suit any investor's needs. Investors will have to go through the process of identifying the required websites and the specific data fields to be used for their notification criteria. The system will then monitor the websites and generate the notification messages whenever the defined criteria are met. This provides flexible and extended features that are not generally available in the current notification systems. There are several possible ways to use this system for example:

I. Notification System for Stock Information Changes in Stock Markets: Many some stock markets provide live information about stock prices; however, they do not provide notification mechanisms for users to get specific price change information based on some defined criteria. The proposed system can be used to define notification variables and price change criteria. The system will notify the users by emails or other messaging mechanism whenever those criteria are met. Examples of the criteria may be as simple as notifications of prices reaching a specific value or could be more complex to cover multiple variables and markets. For example, Mr. StockWiz needs to be notified when a particular stock reaches a certain price level and the sell volume increases by a given percentage or he needs to be notified when another stock's price increases or decreases by a certain percentage while the sell orders are above or below a specified level. In both examples, without the proposed system, Mr. StockWiz will have to constantly monitor the market listings and compare the values until he gets the right conditions. This may happen very rapidly such that Mr. StockWiz will not be able to catch the changes fast enough to react or in other cases this may take several hours or days to happen which will basically waste Mr X's time monitoring the changes. Using the proposed system, Mr X. instead will spend some time defining the variables and conditions for the notifications and then move on to do other stuff. The monitor will keep track of the values of the defined variables and will send the notifications whenever the conditions are met.

II. Monitoring System for Stocks listed in Multiple Securities Markets: Some investors like to monitor prices of particular stocks that are listed in multiple securities markets to buy or sell them based on price differences in those markets. An application can be designed to monitor the same stocks listed in multiple securities markets. In GCC countries, for example, there are some stocks that are listed in multiple markets, which open and close around the same time. The price of a specific stock listed in multiple markets may decrease or increase in one market before the others. This price difference may only happen for short time periods (e.g. a few minutes) before the stock prices in the other markets adjust. The price decrease/increase event in one market is a good indication for near future price decreases/increases in the other markets. Some investors would like to have a tool that notifies them about the price differences in order to make the right investment decisions to buy or sell stocks. The proposed system can be used to get a notification message whenever there is a certain price difference between the markets listing the same stock. For example, let us say that Mr. StockWiz is trading a stock listed in both Abu Dhabi and Doha stock markets. To monitor the difference, her needs to define two Internet

variables for the stock price in the two markets. Mr. StockWiz can then define his notification criteria based on these defined variables such as requesting a notification if one of the two prices becomes lower or higher than the other by a specific percentage (5% for example). The system will then constantly monitor and compare the two listed prices and will notify Mr. StockWiz as soon as the 5% difference occurs.

III. Analyzing Stocks Potential: In many cases investments decisions are not made solely on the changes of stock prices. Investors usually need to monitor several other variables that may affect the stock markets in general or a single stock in particular. For example, when working with stocks for an industrial company that rely on oil for its operations and is affected by raw material prices, it may be necessary to monitor oil and goods prices in global markets to establish some idea about the possible losses or profits the company may have, which in turn may affect its stock price. Also as an example, a technology company like Intel is highly affected by its announcements of new technologies and also by competitors' announcements of similar successes or failures. As a result the stock price may change significantly due to such information. Therefore, an investor needs to know of information like current oil and goods prices, currency exchange values and companies financial performance numbers as soon as they are posted or updated, which gives him/her an advanced warning of what to expect in stock prices. The proposed system can help the investor define specific websites where such information may be found and use them as notification criteria. For example, if Mr. StockWiz is trading in stocks for a petrochemicals company, he will be very interested in knowing about price changes of oil and all derivatives of oil. Therefore, it will be very useful for him to set up some monitoring variables from websites that list these prices and use them to receive notifications for sudden changes in prices or for steady changes over some periods of time. This way, Mr. StockWiz will not have to spend so much time going over these numbers every day. He would simply wait for the notifications to arrive.

The examples given here demonstrate some possibilities of utilizing the proposed system. The possibilities are endless and investors can customize the system in so many different ways to match their personal needs. As it is clearly shown, it is not just the stock prices that are needed to make investment decisions. Therefore, an investor with the right tools that provide to-the-minute updates and notifications based on his/her own interpretation of the markets and the factors affecting them will provide a competitive edge. Investors will be able to grab short living opportunities in time and benefit from the real-time information made available by applications developed based on the proposed system.

They will also save a lot of time they would have spent sifting through loads of information on the Internet on a continuous basis.

6. CONCLUSION

This paper introduced a personalized alert system for stock investors. This notification system is based on information available from web services or dynamic XML and HTML documents available on the Internet. The system allows a user to define a set of Internet variables, calculated variables, notification conditions, notification messages, emails, and mobile phone numbers. Users are given the freedom to define and set their personal criteria without being limited to one notification provider or a single type of simple notifications. Several stock investment notification applications can be developed using this framework to provide personalized sophisticated notification systems for investors. The methods designed in the framework allow the retrieval of data values from various locations in the HTML documents or from XML or web services. This allows for providing personalized notifications for investors based on their own needs and requirements. This will result in better and more timely investment decisions and much less time spent looking for and monitoring information on several websites over long periods of time.

Most HTML documents have a relatively fixed structure and are mostly updated within the original format. However, a problem may occur in some cases when the component formats of the HTML document are changed. By this we mean adding extra fields with the same labels we used to extract our variables. Our plan for future work is to implement an automatic validation mechanism for the system to make sure that the formats of the defined HTML documents were not changed before attempting to extract the information. One possible solution for this problem is to automatically capture and store the format patterns of the HTML documents. These patterns can be used by the system to capture any future changes in the downloaded documents. In case there are some changes, the system notifies the system administrator to configure new parameters for the Internet variables.

ACKNOWLEDGEMENT

This work was supported in part by UAEU Research grants # 01-03-9-11/08 and 01-04-9-11/09.

REFERENCES

- [1] Yahoo Finance, <http://finance.yahoo.com/>.
- [2] P.T. Eugster, P. Felber, et al., "The Many Faces of Publish/Subscribe," ACM Computing Surveys, Vol. 35, No. 2, pp. 114-131, 2003.
- [3] A. Sadat, G. Sorwar, M. U. Chowdhury, "Session Initiation Protocol (SIP) based Event Notification System Architecture for Telemedicine Applications," 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAAR'06), pp. 214-218, July 2006.
- [4] N. Hayashibara and M. Takizawa, "Design of a Notification System for the Accrual Failure Detector", 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06), pp. 87-94, April 2006.
- [5] J. Jeng and Y. Drissi, "PENS: A Predictive Event Notification System for e-Commerce Environment," The Twenty-Fourth Annual International Computer Software and Applications Conference, October 2000.
- [6] Earthquake Notification Service, <http://earthquake.usgs.gov/eqcenter/ens/>
- [7] P. Gore, R. Cytron, D. Schmidt, C. O'Ryan, "Design and Optimizing a Scalable CORBA Notification Service," Proc. of The ACM SIGPLAN Workshop on Language, Compilers and Tools for Embedded Systems, pp. 196-204, 2001.
- [8] S. Tuecke, I. Foster, et al., "Open Grid Services Infrastructure," Available: http://www-unix.globus.org/toolkit/draft-ggf-ogsigridservice-33_2003-06-27.pdf
- [9] Y. Huang and D. Gannon, "A Comparative Study of Web Services-based Event Notification Specifications," Proc. of ICPP Workshop on Web Services-based Grid Applications (WSGA), 2006.
- [10] D. Box, L. F. Cabrera, et al., "Web Services Eventing," Available: <http://ftpna2.bea.com/pub/downloads/WS-Eventing.pdf>
- [11] D. Chappell and L. Liu, "Web Services Brokered Notification (v1.2)," Available: <http://docs.oasis-open.org/wsn/2004/06/wsn-WSBrokeredNotification-1.2-draft-01.pdf>
- [12] A. Carzaniga, D. S. Rosenblum, et al. "Achieving Scalability and Expressiveness in an Internet-Scale event Notification Services," Proceeding of Nineteenth ACM Symposium on Principles of Distributed Computing (PODC 2000), 2000.
- [13] Google Alerts, <http://www.google.com/alerts>.
- [14] ChangeDetect, <http://www.changedetect.com/>.
- [15] Yahoo Alerts, <http://alerts.yahoo.com/>.
- [16] RFC 1866, Hypertext Markup Language – 2.0, November 1995.
- [17] Extensible Markup Language (XML) 1.0 (Fourth Edition), <http://www.w3.org/TR/REC-xml/>, September 2006.
- [18] F. Curbera et al. "Unraveling the Web Services: an Introduction to SOAP, WSDL, and UDDI," IEEE Internet Computing, Mar/Apr issue 2002.
- [19] N. Mohamed and J. Al-Jaroodi, "A Simple and Efficient Approach for Retrieving Live HTML-based Internet Information," in System and Information Sciences Notes, Vol. 1, No. 3, pp. 221-224, July 2007.
- [20] Sun Java Reflection website, <http://java.sun.com/javase/6/docs/technotes/guides/reflection/index.html>
- [21] Sun JavaMail API documentation website, <http://java.sun.com/products/javamail/javadocs/index.html>
- [22] SMSLib website, <http://smslib.org/>



Jameela Al-Jaroodi received her Doctor of Philosophy degree in Computer Science from the University of Nebraska-Lincoln, USA, 2004. Since August 2006, she has been with the College of Information Technology, at the United Arab Emirates University, UAE as an Assistant Professor. Prior to joining UAEU, Dr. Al-Jaroodi was a research assistant professor at Stevens Institute of technology in

New Jersey, USA. Currently, her research interests involve middleware, distributed collaborative systems, information systems, and mobile and pervasive computing. Her research generated over 50 refereed articles in international Journals and conferences such as Concurrency and Computation: Practice and Experience, IEEE Transactions on Distributed Systems, and IEEE International Conference on Cluster Computing. Dr. Al-Jaroodi received the Research Excellence Grant from Sun Microsystems, Inc. In addition, several areas of her research were also supported by the United States National Science Foundation (NSF), Nebraska Foundation, and the National Center for Information Technology in Education (NCITE).



Nader Mohamed is a faculty member at The College of Information Technology, United Arab Emirates University, Al-Ain, UAE since August 2006. Before that, he was an assistant professor of Computer Engineering at Stevens Institute of Technology in New Jersey, USA. He worked at the University of Nebraska-Lincoln as a research assistant from May 2001 to May 2004 in National Science Foundation (NSF) supported research

and development projects. His current professional interest focuses on communication software and middleware, Internet computing, systems integration, cluster and Grid computing, and sensor networks. His research was published in prestigious international journals such as IEEE Transactions on Parallel and Distributed Systems (TPDS), Journal of Parallel and Distributed Computing (JPDC), International Journal of High Performance Computing Applications, and IEEE Distributed Systems Online. He has 8 years of industry experience in the information technology field as IT Consultant and IT projects leader. He obtained his Ph.D. in Computer Science from The University of Nebraska-Lincoln, Nebraska, USA in 2004. In addition, He obtained The American Management Association's Certificate in Business Management for IT and Technical Professionals, New York, USA.