

A Generic Notification System for Internet Information

Nader Mohamed, Jameela Al-Jaroodi, and Imad Jawhar

The College of Information Technology

United Arab Emirates University

Al Ain, P.O. Box 17551, UAE, {nader.m, j.aljaroodi, ijawhar}@uaeu.ac.ae

Abstract

The Internet provides a huge amount of online and dynamic information related to stock information, currency exchange rates, interest rates, expected weather status, oil prices, and many other topics. This information is publicly available on dynamic HTML documents or on web services. This paper discusses a flexible notification system that utilizes the available online information and allows users to define a set of notifications that they are interested in. Based on the defined notification conditions, users will be notified by email and/or SMS messages whenever one or more of the conditions are met. In this system, users use Java-based configurations to define the notification requirements. This system also solves some of the issues facing utilizing public information available on the Internet to build the needed notifications. This includes the problem of capturing highly dynamic Internet information as well as supporting advanced types of notifications.

1. Introduction

There are three approaches to get desired information from the Internet. These approaches are browsing, searching, and notification. The first approach, which started with the concept of the Internet, is browsing. A user can move from one page to another to obtain the required information. The second approach is searching, where a user can search for a topic using some related keywords. The third approach is notification, where a user can define some notification criteria for specific events on some websites. The user is notified whenever his/her criteria are met. The notification can be delivered by SMS, email, or online messages.

Both browsing and searching provide generic ways for obtaining information from the Internet. Users can browse and search for any topic of interest. There are different generic and free search engines available on the Internet that provide basic and advanced search mechanisms for any topic or keyword. On the other hand, current available notification systems are very restricted. The user is

restricted to specific types of information to use and limited types of notifications. Users have no means to define advanced criteria he/she may be interested in. In addition, the data used for notification is usually limited to what is available on the website of the company managing the notification system.

This paper develops a generic and flexible notification system in Java. This system deals with the public information available on the Internet on one or more websites or from one or more web services as the information source of which the user can build his/her notification criteria. The Internet provides a huge amount of dynamic information such as news, stock information, current and expected weather status, currency exchange rates, etc. All this public information is viewed as a set of Internet variables, which are used to define the user needed notifications.

In this paper, Section 2 provides background information. Section 3 discusses how the Internet variables are defined and Section 4 describes the generic notification system for Internet information. Section 5 provides some discussion while Section 6 discusses related work. Section 7 concludes the paper.

2. Background

A notification system is a combination of network services and infrastructure that provides a mechanism of delivering a notification messages to a single or multiple recipients based on the occurrence of an event specified by the recipients. Notification systems usually consist of two types of parties, the publisher/producer. There are different notification systems used for different domains and applications such as telemedicine applications [1], accrual failure detectors [2], e-commerce applications [3], earthquake notification services [4], and finance/banking.

Since the early days of the Internet, notification services were offered on a few websites. For example, on some news website, the user can state that he/she wants to be notified whenever there is news about a specific subject. For example, a user can set a notification condition on yahoo finance that causes him/her to be

notified whenever the Wal-Mart Store stock price drops to a specific value. In addition, a user can subscribe in some journal website to receive an email message whenever his/her previously published paper is cited.

The main limitation of currently available notification services is that they are very restricted. The user is restricted to specific types of information to use and limited types of notifications. Users have no means to define advanced criteria he/she may be interested in. In addition, the data used for notification is usually limited to what is available on the website of the organization managing the notification system.

For example, an investor can set a notification condition on Yahoo Finance such that whenever the Wal-Mart stock price drops to a specific value in US dollars a notification is sent. However, the investor can not define advanced notification 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 Internet, Yahoo Finance does not support that type of notification. Therefore, an investor can not define any advanced or compound criteria.

One possible solution to have unrestricted notification services can be achieved by utilizing the framework that was recently developed [5]. This framework provides an approach to reuse available Internet information to build unrestricted notification services. In this paper, we discuss a system that utilizes that framework in providing flexible and generic notification services.

3. Internet Information Variables

In order for a notification system to be generic and can be used for different types of notification applications, it needs to have access to a wide range of information. Currently, there is no other system that has such huge and up-to-date information like the Internet. It is very useful to reuse the Internet information for building customized notifications. Making the whole Internet information available for a notification system, the user has endless ways to define his/her notification needs.

The notification system for internet information depends on a set of variables defined by the users. These variables represent dynamic information available on the Internet. He/she can for example define a variable for Wal-Mart current stock information and another variable for US Dollar to Euro exchange rates. The Internet variable content's sources are from dynamic HTML documents, XML documents, or from web services available on the Internet. This information may be for stock market information, currency exchange rates, temperatures, oil prices, gold prices, interest rates, etc. The proposed

system depends on the Internet for obtaining the required and updated information to build the notifications. This information is usually available on the Internet in structured or unstructured formats. This Section discusses two ways for defining Internet variables.

3.1. Web Services Based Variables

Web services can provide a structured and simplified way to obtain services or specific information from the Internet. 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 can provide different services including providing information about different aspects or products. For example, different stock markets can provide web services to provide current stock prices. Banks can provide web services to provide information about loans or information about foreign currency exchange rates. If desired information is needed by a notification application, then 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 by Web Services. Most of useful information on the Internet is still available in HTML format.

3.2. HTML Based Variables

Although XML documents have self-defined structures that make them easy to deal with, most of the Internet information is delivered to users in HTML. Unlike XML documents, HTML documents do not have any semantics for their data. HTML documents usually contain tags, scripts, links, and user defined 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 [6]. This approach can be used to define the notification variables that will have their data updated from the Internet. The main idea of this approach is based on finding fixed titles or headers that appear in browsers for HTML documents directly or semi-directly before the needed dynamic information. These fixed titles or headers are used as references to know the position of the required dynamic information. The proposed approach is developed as a Java class, *urINFO*. Multiple objects can be created from this class for different Internet HTML documents that contain some of the required information. A number of techniques were developed to find this

information in any HTML document. These techniques are implemented in a set of methods listed in Table 1.

All these techniques can be used to define Internet variables by the users before defining the notification criteria. As soon as the fields are identified the user can specify the arguments for the *get* or *getWI* methods, which will allow the system to retrieve the required information. Users can use any HTML documents on the Internet to define Internet variables that they need. More information about the implementation and performance of the mentioned approach can be found in [6].

Table 1. Methods to retrieve HTML-based information.

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 <i>skipping 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 header <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.

4. The Generic Notification System

The system is developed as a client that monitors one or more Dynamic HTML documents from one or several web servers on the Internet as shown in Figure 1. In addition, the system can deal with one or more web services to get some other information. 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 some calculated variables form from the defined Internet variables. These calculated variables are usually functions of 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 that contains some of the defined Internet variables or the defined calculated variables will be sent to the user. The user needs to define first all Internet variables and all calculated variable he/she needs for preparing his/her single or multiple 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.

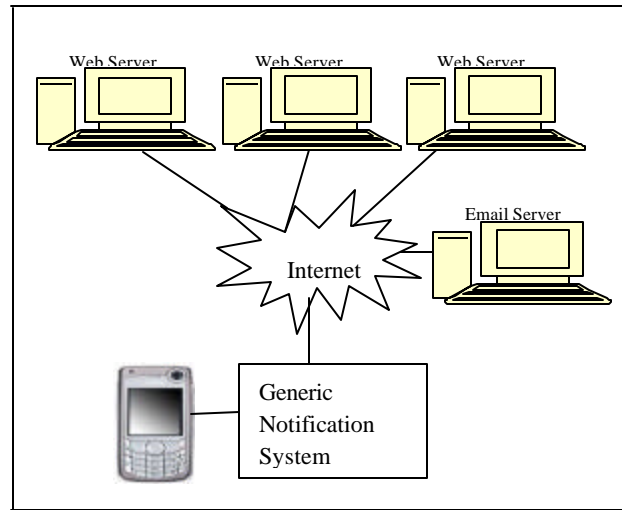


Figure 1. The generic notification system works as a client monitoring the Internet.

A prototype system was implemented in Java to demonstrate the concept of the generic notification for Internet information. The main component of that system is a monitor program. The main function of the monitor is to execute the notification policies that are in a user-defined 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 are needed to be developed. An example of a user defined configuration is shown in Figure 2. This configuration file is a simple case needed by a European investor (using the Euro as trading currency) who would like to sell *AirArabia* stock listed in UAE in Dirham if the price increases above 0.4 Euro.

```

// Notification File: AirArabia price in Euro.
class AirArabiaNotificationInEuro 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 AirArabiaAED; // AirArabia current price in Dirham
    private double AEDEUR;      // Dirham to Euro currency exchange rate
    private double AirArabiaEUR; // AirArabia current price in Euro

    public void VARIABLES()throws Exception
    {
        // Get AirArabia price from DFM
        urlINFO dfm = new urlINFO("http://www.dfm.ae/marketwatch/defaultt.aspx");
        dfm.download();
        String _AirArabiaAED = dfm.get(1,"AirArabia",10);
        AirArabiaAED = Double.parseDouble( _AirArabiaAED );
        // Get Dirham to Euro exchange rate
        urlIFNO exchangeRate = new urlIFNO("http://finance.yahoo.com/q?s=AEDUSD=X");
        exchangeRate.download();
        String _AEDEUR = exchangeRate.get("Last Trade:", "Trade Time:");
        AEDEUR = Double.parseDouble( _AEDEUR );
        // Calculate AirArabia current price in Euro
        AirArabia = AirArabiaAED * AEDEUR; }

    // notification #1 condition and message
    Public void NOTIFICATION1()throws Exception
    {
        if( AirArabiaEUR > 0.4 )
        {
            // Prepare a notification message
            String M1 = message("AirArabia price is %1% Euro",AirArabiaEUR);
            sendEmail("nader.m@uaeu.ac.ae","AirArabia Price",M1);
            sendSMS("+971507533620",M1);
        } } }

```

Figure 2. Notification example for AirArabia when the price researches above 0.4 Euro.

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 rates offered on a different website shown in Figure 4.

Both values are dynamic and finding the correct price normally requires the investor to open both sites and continuously perform currency conversions on the listed prices until the desired value is reached. However, using our system the investor will need to specify two variables for the current stock price in Dirham, *AirArabiaAED*, and the current exchange rate between Dirham and Euro, *AEDEUR*. He/she can use these two Internet variables to generate a calculated variable, *AirArabiaEUR*, for the current stock price in Euro. In addition, he/she needs to specify a condition at which a notification message will be sent to him/her as soon as the price reaches the desired value in Euros.

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 *NOTIFCATION* prefix as shown in Figure 2.

The screenshot shows the Dubai Financial Market website. At the top, there is a banner for "DUBAI FINANCIAL MARKET". Below the banner, there is a news ticker: "discussing the resolutions of its AGM held on 21/05/2008 *** DHF: Suspend trading on ALHADID". The main content area is divided into two sections. The first section is a table of market indices, and the second section is a table of company stock prices.

Index	Value	Volume	Index	Value	Volume	Market Info
General Index	5,696.63 ↑↑ 1.0 (1.0%)		Transportation	536.10 ↑↑ 0.2 (1.0%)	76,103.712	Status
Bank	3,498.23 ↑↑ 0.7 (1.0%)	45,994.611	Materials	428.21 ↑↑ 0.9 (1.0%)	2,082	Trades
Insurance	5,095.88 ↑↑ 0.4 (1.0%)	6,110.090	C. Staples	687.00 ↑↑ 0.8 (1.0%)	9	Trade Value
Investment	6,481.73 ↑↑ 0.8 (1.0%)	36,100.762	Telecommunication	578.96 ↑↑ 0.1 (1.0%)	1,127.361	Trade Volume
Real estate	11,745.43 ↑↑ 0.1 (1.0%)	77,016.394	Utilities	4,669.47 ↑↑ 0.1 (1.0%)	6,616.145	

Company	51 Week High	52 Week Low	Ann. Closing	Opening	High Price	Low Price	Vol. Volume	Bid	Offer	Offer Volume	Current Closing Price
A. Scandinavian Insurance	4.25	1.81	2.96	2.96	2.95	2.95	30,000	2.82	3.11	60,000	2.95
Ararated Concrete Industri	7.47	3.67	6.95	6.95	6.95	6.95	0	6.90	6.90	0	7.30
Air Arabia	3.22	3.68	1.97	1.97	2.00	1.96	1,703,938	1.96	1.99	845,151	1.98
Al Fintous Holding	4.02	3.15	2.64	2.64	2.67	2.60	26,525	2.60	2.66	30,000	2.67

Figure 3. Dubai Financial Market website.

Internet variables are defined using the approach mentioned in Section 3.2. The user needs to define a URL

and then specific information can be extracted using the different *get* methods. In the notification methods, the user can define notification conditions, notification messages, and notification destinations as shown in Figure 2. 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 *GenericNOTIFICATION* class. This class provides some support for creating and sending notification messages.

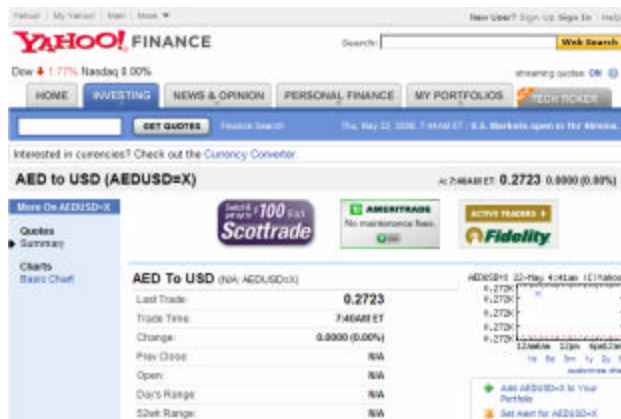


Figure 4. Dirham to Euro Rate from Yahoo Finance.

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. The execution is done through multiple threads where each thread will start after the defined *INTERVAL* to execute the defined *VARIABLE* 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.

The monitor uses the reflection API [7] 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 find out 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* defined by

the user. In addition, it uses the reflection APIs to execute the defined *VARIABLES* and all notification methods.

The monitor sends messages through either emails or SMS messages. The monitor utilizes JavaMail APIs [8] for sending emails and SMSLib [9] for sending SMS messages. SMSLib is a java library for sending and receiving messages via a mobile phone or GSM modem. A mobile phone or a GSM modem needs to be connected to the machine where the monitor is executing in order to enable the SMS message transmissions.

5. Discussion

In this paper, we discussed two approaches to retrieve Internet information: web services and the *urlINFO* class. However, users are not restricted with these two approaches. The users can use any other retrieval libraries to define required Internet variables. Users just need to use any available library in the *VARIABLES* methods to retrieve the desired Internet information. In addition, within a single notification application the user can use heterogeneous techniques to define the required Internet variables. For example, the user can define a notification by use the *urlINFO* class to define one Internet variable while he/she can use a web service to define another Internet variable.

Using Java to define the user notification policies provides endless ways to build advanced notifications and solves some of the open issues listed in [5]. For example, temporal notifications can be supported. An example of this type of notifications is that a notification message should be sent if a temperature drops 10 degrees within one hour or a notification message should be sent if a stock price increase 5% in the same day. Both these mentioned notifications require keeping track of changes over an extended period of time. This can be achieved using a database to store all changes as well as to search for the required temporal information. With each new value for a defined Internet variable, the user can define some additional code to store the new value and the current time in a database. Furthermore, the user needs to add some code to search the database to check if his/her temporal notification conditions are fulfilled.

Another example of advanced notifications is allowing users to use locally available information from a local database as part of the defined notifications. Thus notification conditions will be combinations of Internet variables and local information obtained from local applications or databases. This feature can be very useful to build advanced notifications. For example, a user is interested to receive a notification message about a stock if and only if he/she has a specific amount of cash. If the available amount of cash is kept in another application or a

database, the user can integrate that variable with other Internet variables to define the notification conditions.

Some Internet information can be very dynamic. An example is the current price of a stock, where the values change very frequently (e.g. every second). The system utilizes multithreads to support capturing highly dynamic Internet information. The user can change the static variable *INTERVAL* to define the speed of the download. For highly dynamic information, *INTERVAL* can be set low while for slower changing information, *INTERVAL* can be set high.

6. Related Work

A lot of literature is available on notification systems representing different projects, protocols, and architectures. These include web services based event notification systems [10], CORBA notification service [11], notification services for Grid Infrastructure [12], and the publish/subscribe communication paradigm [13].

The main difference between our system and others 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 the notification rules based on information available and updated on any website on the Internet or based on information provided by any web service.

In our other work [14], we developed a development environment to help users generate web monitors. With this development environment, the users can use a user interface to define the desired notifications. This environment can generate a basic web monitor based on user requirements. Although with this environment the user can develop some notifications without knowing or dealing with Java, the environment is still restricted for some types of basic notifications. In contrast, the user can use the proposed system in this paper to have unrestricted ways for building advanced notification applications. This is mainly due to using Java to define user notification requirements.

7. Conclusion

This paper introduced a generic notification system for Internet Information. This notification system is based on information available from web services or dynamic HTML documents available on the Internet. Users could be given the freedom to define and set their personal criteria of notification without being limited to one notification

provider not to a single type of simple notifications. The user defines his/her notification requirements in a Java class. In this class, the user can also implement advanced notification functions such as temporal notifications as well as notifications that are integrated with other local applications and databases. The proposed system also supports capturing highly dynamic Internet information by using multiple threads to download the required information.

Acknowledgments

This work was supported in part by UAEU Research grant # 01-03-9-11/08.

References

- [1] 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-COMSTAR'06), pp. 214-218, July 2006.
- [2] 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.
- [3] 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.
- [4] Earthquake Notification Service, <http://earthquake.usgs.gov/eqcenter/ens/>
- [5] N. Mohamed, J. Al-Jaroodi, I. Jawhar, "A Framework for Generic Internet-Based Information Notification," in Proc. of The Third International Conference on Internet and Web Applications and Services (ICIW 2008), IEEE, Athens, Greece, 8-13 June 2008.
- [6] 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.
- [7] Sun Java Reflection website, <http://java.sun.com/javase/6/docs/technotes/guides/reflection/index.html>
- [8] Sun JavaMail API documentation website, <http://java.sun.com/products/javamail/javadocs/index.html>
- [9] SMSLib website, <http://smslib.org/>
- [10] 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.
- [11] P. Gore, R. Cytron, D. Schmidt, C. O'Ryan, "Designing and Optimizing a Scalable CORBA Notification Service," Proc. of the 2001 ACM SIGPLAN workshop on Optimization of middleware and distributed systems pp. 196-204, 2001.
- [12] S. Tuecke, I. Foster, et al., "Open Grid Services Infrastructure," Available: http://www-unix.globus.org/toolkit/draft-ogf-ogsigridservice-33_2003-06-27.pdf
- [13] P.T. Eugster, P. Felber, et al., "The Many Faces of Publish/Subscribe," ACM Computing Surveys, vol. 35, 2003.
- [14] N. Mohamed, A. Romaitih, E. Samahi, M. D. Kendi, and E. Jabrie, "Generating Web Monitors," in Proc. of The International Conference on Internet Computing (ICOMP 2008), Las Vegas, USA, July 2008.