Control of Information Appliances Using Instant Messaging

Seong Joon Lee and Kwang Seon Ahn

Department of Computer Engineering, Kyungpook National University
1370 Sankyuk-dong, Buk-gu Daegu 702-701, Korea
{imggaibi, gsahn}@knu.ac.kr

Abstract. Many of the systems developed to date for controlling home appliances remotely from outside have used a web server as the interface between the home network and the users. Although the user interface (UI) that uses a web server offers many advantages, this approach requires a fixed IP (Internet Protocol) address, and does not have a push function. Moreover, until the user gets reconnected to the web server, he or she cannot know the result of a submitted order. In particular, internet-based applications using web browsers still require considerable time to execute the initialization process for acquiring state information on home appliances from the home server. To solve these problems, we propose an application for home automation that can efficiently control and monitor home appliances using an instant messaging system (IMS) with real time communication. The proposed system has all not only the advantages of the web server method, but also with additional advantage that the homeowner does not need to continually reconnect to the home server, and that the home server does not need to have a fixed IP address. Our application can be applied using middleware technologies.

1 Introduction

The smart home (SH) is a house or living environment in which devices or services are networked together. If a SH is linked to the Internet, authenticated users can remotely control home appliances. To achieve this networking, home work, home server, and remote control devices are required. The home network is the internal network formed by various appliances and services in the house. To implement this network, appliances must have the computational power to perform predefined functions and the networking ability to share information with other appliances. Appliances with these capabilities are called information appliances or network appliances [9, 10]. Recently developed middleware for building home networks have typically utilized well-known middleware technologies such as uPnP [11], Jini [12], Havi [15], and OSGi [16]. These middleware technologies provide protocols for discovering available services in the network and for distributing required information. The home server acts as a gateway to connect the home network to the Internet and as a server to connect and manage the appliances within the home. Homeowners can remotely control and monitor home appliances through this home server. Remote control devices are the tools used to control and monitor home appliances from distant locations; these devices can be used either when the user is at home or outside. Recently, a key issue in the implementation of SHs has been how to control and monitor home appliances from outside the home. Various methods have been proposed to efficiently solve this problem.

Recent advances in Internet technologies have prompted the development of various Internet-based remote controlling and monitoring of SHs [17-19], most of which adopt a web server as the interface between the homeowner and the home network. Although the use of a web server offers many advantages in terms of the UI, researchers have encountered problems in regard to issues such as fixed IP (fixed internet protocol, real internet protocol) addresses used in web servers and the lack of a push function for notification. To operate a server, the legacy system must have a fixed IP. However, home asymmetric digital subscriber line (ADSL) services offer the possibility of using a dynamic IP service, where the dynamic host configuration protocol (DHCP) server allocates a fixed IP during the lease time. Because a client does not always receive the same IP address in this dynamic IP system, it is impossible to use this system as a server.

Another problem is the push technology. Using this method, users who are subscribers to a server are able to automatically receive up-to-date information from the server. Therefore, mobile users can efficiently obtain the latest information anytime and anywhere, using any available device. Recently, several proposed systems have been worked on to integrate the push technology with various communications technologies. The best-known methods include Short Message System (SMS), Wireless Application Protocol (WAP), or Session Initiation Protocol (SIP), developed within the Internet Engineering Task Force (IETF) Multiparty Multimedia Session Control (MMUSIC) working group. SMS allows mobile phone subscribers to send and receive alphanumeric messages of up to 160 bytes in length in a store-and-forward fashion via an SMS center (SMSC). However, the SMS system allows users to send only one message at a time, and, due to technical restrictions, only short text messages, ring tones, and small graphics can be sent [6, 8]. Above all, SMS is not so much real-time as an immediate data transfer method.

In order for SIP- or WAP-based systems to control and monitor home appliances, each client must first download all the state informations on home appliances, or the file for Common Gateway Interface (CGI) scripts because they use the web serve In addition, the provider must develop various interfaces for the user. Therefore, most systems require complex hardware to reduce the technical complexity facing the user, and users spend considerable time initializing and downloading the related CGI file.

In this paper, we present an efficient real-time solution for home automation systems by using Jini network technology and instant messaging system (IMS) method. This System provides a uniform graphic UI (GUI) for end users. It immediately sends state information from the home network to the homeowner via the Internet. Homeowners with the appropriate IM installed on their mobile phones can remotely control and monitor their home appliances, anytime and anywhere. Furthermore, the system is lightweight, flexible, and extensible. The proposed system is convenient because it can sends a completion or alarm message to a manager via push functionality (NOTIFY message), even if the manager does not reconnect to the home network after it finishes setting the device. The GUI program can be used in other platforms without any changes.

2 Background

2.1 Push technology

There are two traditional approaches to distribute information in an environment. In traditional client-server model, the client can only receive the information after connecting to the server. That is, until the connection between the server and the client has been made, the server cannot send any data to the client. This well-known method is the pull technology. On the contrary, in the push technology, the server can send any data to a client without establishing a connection.

(a) Pull technology (b) Push technology

Figure 1. Pull Technology vs. Push Technology

2.2 Instant Messaging Systems (IMSs)

IMSs have long been one of the most popular applications on the Internet; they allow real-time exchange of messages, independent of locale [21]. Also, IMS merges e-mail, SMS function of mobile phones, multimedia communication, and file transfer. There are two features that make instant messaging unique: rapid-fire asynchronous messaging, and real time presence information. It allows near real-time communication, message interchange with little delay, and lightweight software compared to a web server. IMSs can be implemented on small Internet-enabled devices, such as mobile phones, PDAs, and set-top-boxes.

In this paper, we propose an IMS-based method for continuously controlling home appliances. The main goal was to verify that the proposed system could efficiently operate on different platforms without additional expense. To achieve this, we designed the user interface not only operates in specific environments, but also in others. In existing legacy system, the memory of most clients is too small to use Java. Even though there were many advantages of JAVA APIs, such as JMS APIs and RMI APIs, we did not consider JMS as a communication protocol for client-server interactions because of the reasons mentioned above.

2.3 Jini network technology

Jini network technology, a java-based middleware technology developed by Sun Microsystems, provides a set of APIs and high level network protocols that facilitate the development of distributed systems. Jini provides a simple way to perform the tasks of discovering, registering, and removing devices and services on home networks. First, Jini creates a software infrastructure, called federation of services, to share access to services, and then engages in interactions without any prior knowledge of the other systems or any need for human intervention [13]. Figure 2 shows a simplified schematic of the process involved in using a Jini service.

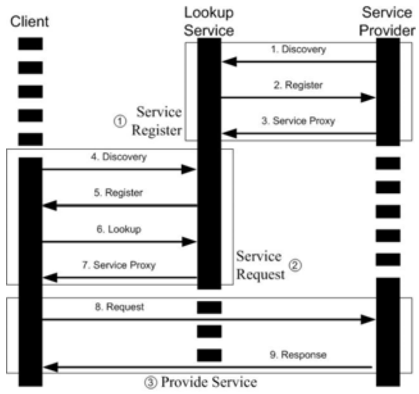


Figure 2. Jini Technology

The system for enabling Jini service consists of three protocols including discovery, join, and lookup that provides dynamic configuration, and must first make a network service available. When a service is plugged into the Jini network, it uses a multicast request to find the local lookup service through the discover protocol and then registers the proxy service object in the lookup table via the join protocol (step 1 in Figure 2). Clients can also use the discovery protocol to find the lookup service. Subsequently, when a client requests a search for a service, the lookup service returns matching proxy service objects to the client (step 2). Finally, the object downloaded from the lookup server communicates directly with the service provider (step 3) [14].

3 HAMSuIM

The HAMSuIM system is based on the Internet/Intranet architecture, the Jini technology, and the IMS. To implement the HAMSuIM, we assume that the home network uses Virtual Private Network (VPN) and that the IP address of the residential gateway is not fixed. In the proposed system, three main agents are involved in the implementation of application programs: the Mobile Manager Agent (MMA), Home Messenger Agent (HMA) and Information Appliance Manager Agent (IAMA). Each agent must operate well even if the structure of the other two agents changes. To achieve this, we designed and implemented agents that do not depend on the environment of the other two agents. Figure 3 shows a block diagram of the system architecture.

First, the main objective of the IAMA is to create the object necessary for connecting with middleware using messages transmitted from the HMA, and then to monitor the appliances. Therefore, when the homeowner makes a request, the IAMA must rapidly seek out the information required by the homeowner and attempt to complete the specified task. If a problem is encountered, it must transmit an urgent error message to the homeowner. If a task is completed, it must send a result message to the homeowner stating that it is completed. All of this must be performed as quickly as possible. To perform its functions, the IAMA must be able to track all the changes and breakdowns of each appliance. To directly manage the Lookup table, we designed the IAMA to include the Lookup server of Jini. The appliance services are registered at the Lookup server through the modified register function in the IAMA.

Second, the HMA acts as an interface between the MMA and the IAMA. The HMA hooks up with the IAMA to permit management of home appliances and contains the register_owner() function for processing homeowner authentication. Once homeowners are authenticated by the HMA, they can control home appliances, whenever they wish, using the same email address.

Finally, the MMA is a messenger that is executed through a device that allows the homeowner to access the system, thus to control home appliances connected to the home network, via the Internet from either inside the home or from another location. The type of devices that can access the proposed home network is divided into three groups: user, homeowner, and administrator.

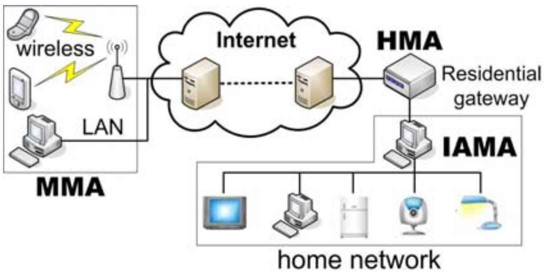


Figure 3. Overall Software Interaction.

The user group comprises all the devices onto which the IMS is installed. Members of this group are candidates for the homeowner group and acceptable to HAM-SuIM but cannot control or monitor any of the home appliances. The homeowner group is defined as the device that is registered with the HMA as a friend. Once a device is added as a friend, the device can control and monitor all home appliances when needed. Finally, if the Medium Access Control (MAC) address using the registration function is enrolled to the administrator list in the HMA, the device is assigned to the administrator group. Administrator functions can only be executed within the home.

3.1 Homeowner Registration

To control home appliances on a home network, the MMA must be enrolled on the list of friends. The homeowner registration process uses the MAC address that is returned by the ARP Protocol. The advantage of such protocol is that each MAC address on a subnet is unique, and the MAC address cannot be delivered from devices that are outside the home. This method based on the ARP protocol has the disadvantage that administrators cannot add a homeowner in other subnets; however, it protects against access by unauthorized users.

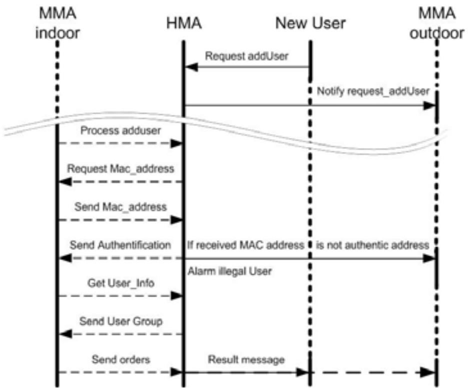


Figure 4. The Algorithm for Adding a Homeowner.

For example, if a user submits a request to register as a homeowner to the HMA, the HMA transmits this registration request to the other administrators. Such requests can only be handled by the administrator situated within the house. If the administrator operates the homeowner manager function, the HMA asks the administrator for the MAC address using the ARP protocol. If the returned MAC address is found on the administrator list, the HMA sends the administrator a list of e-mail addresses of users wishing to become homeowners. If the administrator agrees to any of these user requests, the HMA then sends a list of the e-mail addresses of new homeowners to all existing homeowners. Figure 4 shows the homeowner registration process.

After a user is enrolled as a homeowner, that homeowner receives various messages from the HMA. The types of message are classified below: information message (DESKEY, NEWDEVICE, UPDATE, and NOTIFICATION), execution message (DIRECTIVE), and emergency message (ALARM). When using IMS, The messages to be interchanged between clients are not encrypted. Therefore, we designed all messages to encode and to send. To achieve this, the MMA receives a key to use for cryptograph from the HMA by using *DESKEY message*. This key that is to be used as message encryption is periodically delivered from the HMA. And the MMA receive of the metadata list of modified home appliance using *UPDATE message*. If a new home appliance is detected, *NEWDEVICE message* is delivered to the MMA. This message consists of program layout. When the Homeowner tries to modify state variables in the IA which can be found in the metadata list (UPDATE message), the MMA receives modified state information from the HMA using *NOTIFICATION message* before the set-up window pops up.

The unique message that states the MMA notifying the HMA is the *DIRECTIVE message*. It can be classified into two orders: get_attribute to require the HMA to send the state variable of IA, set_attribute to control the home appliance. The MA receives the DIRECTIVE message from the MMA, verifies it, and then forwards the message to the IAMA. Then, the IAMA creates a new object, begins the operation, and returns the completed result to the HMA. According to this result, the HMA sends either an ALARM or NOTIFICATION message to the MMA.

Finally, *ALARM message* is not encoded by 3DES to ensure processing without delay. When an error or fault occurs in the home appliance, the IAMA transmits an urgent message to the last bidder and administrators through the HMA. In addition, if one MMA (MMA A) has already instructed the IAMA to run a modification of the home appliance either immediately or at some future time, any subsequent attempt by another MMA (MMA B) to modify the same appliance will cause the IAMA to send an ALARM message to MMA A that includes the email address of MMA B, and an ALARM message to MMA B stating that the home appliance was not modified as instructed. Figure 5 show relation and flow between these messages.

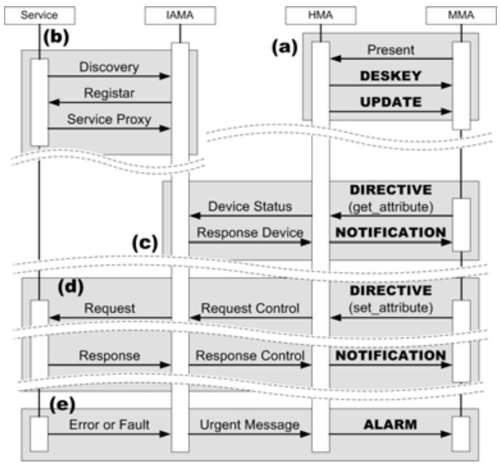


Figure 5. Relation and Flow between Messages

4 Implementation

The aim of this research was to develop a system for controlling and managing home appliances using a home automation system onto which an IMS was installed. To implement this system, we used Jini middleware to establish an efficient home network and MSN Messenger to connect to the home server. We defined the various types of messages that are used in communication between the MMAs and the HMA.

Prototype implementations of the three agents have been developed on Java2 v1.4. We referred to the MSN analysis document [25] for creating a MSN clone, and the J-Sim Library to implement the ARP protocol. J-Sim Library is an object-oriented library written in Java for network simulators [24]. Jini API is designed to enable the simultaneous control of multiple information appliances.

In the proposed system, the MMA has a cache directory in which state information passed to the MMA by the HMA is stored. The state information stored in this directory is kept as consistent as possible with the true current state of the system. In addition, the MMA deletes data from the cache if that data has not been updated within a pre-specified time. The developed HAMSuIM has the ability to fully control and manage all appliances within the house when using Jini-enabled devices. The detailed architecture of the HAMSuIM is shown in Figure 6.

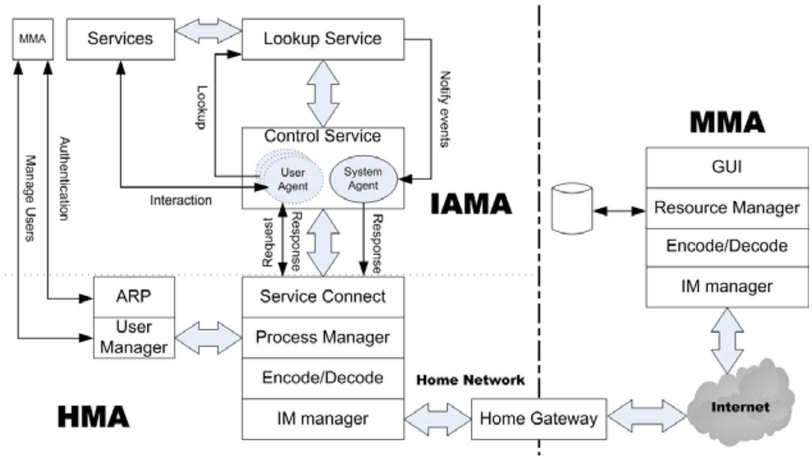
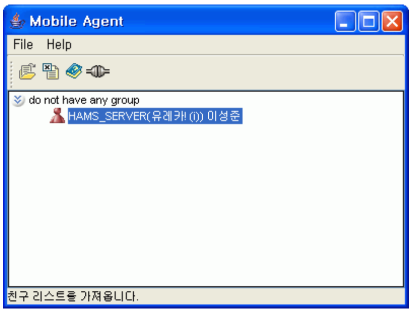


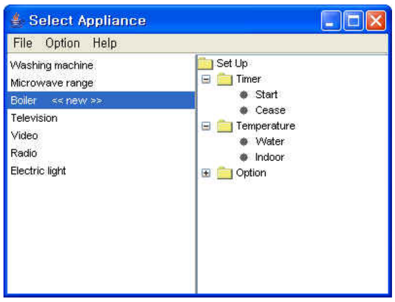
Figure 6. Overall Architecture of the HAMSuIM

A homeowner accesses the HAMSuIM after completing the logon process. Figure 7-a) shows the main GUI window for a MMA. If the item (HOME_SERVER) is selectable, double clicking this item will cause monitor windows to be displayed on the user screen (Figure 7-b). The homeowner is then able to control and monitor his or her home appliances. In the monitor window, the left pane is called the appliance list and the right pane is called the attributes window. If a MMA clicks a home appliance in the appliance list, all state variables of that home appliance will appear in the attributes window.

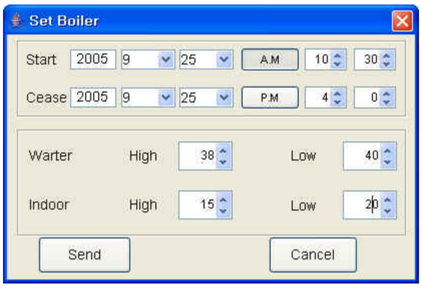
When a new appliance is enrolled into the appliance list by receiving NEWDEVICE message, the label <<new>> appears beside the name of that home appliance for 24 hours. When setting a home appliance, the homeowner can double clicks on the attributes window, and the resource manager of the MMA will search the updated metadata list stored in cache directory. If the appliance is included in the list, the MMA send DIRECTIVE message for receiving modified state variable. And, the setup window will be popped up (Figure 7-c). The setup window is configured by XML code which describes new devices that will be included to a NEWDEVICE message. In the setup window, the MMA homeowner can insert values for the attributes that he or she wishes to control. When the send button is clicked, a DIRECTIVE message is prepared and sent to the HMA. A type of control word is defined by immediate_run if the value of the timer included in the message is less than or equal to the current system time of the HMA, or by reserve_run if it is greater than the current system time of the HMA. The process manager of the HMA analyzes the received message and, if it does not identify a fault, it uses a NOTIFICATION message to broadcast updated attributes to the other homeowners. The homeowner who performed the update will also receive this message, causing his or her cache directory to be updated and his or her window to be refreshed (Figure 7-d). In brief, the DIRECTIVE message does not update the cache directory; only the NOTIFICATION message can update the cache directory.



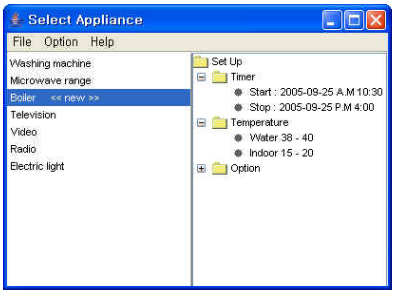
(a) MSN clone



(b) Select an appliance to modify its status



(c) Change the attributes of the selected appliance



(d) Display with new status

Figure 7. Home Automation Management System Interface

5 Conclusions and Future Work

In this paper, we have proposed a system that supports remote control and monitoring of home appliances on a home network through an IMS with real time communication. The advantage of the proposed system is to transfer changed state of IA spontaneously. That is, it can receive messages described IA state information without reconnecting to the Home Server. This eliminates the disadvantage of the administrator needing observe information appliance attentively. And, it can increase access security, because it will make use of the system with dynamic IP as Home server. Finally, since this system is a light-weight process, it can be ported to mobile phones easily. In addition, the proposed system can implement more convenient systems by using the add-on functions of IMSs, such as voice chatting, SMS, and multimedia. In summary, the proposed system can be ported to any networked legacy system, and can control and monitor home appliances anytime, and anywhere. The only disadvantage of the proposed system is that it processes incoming packets from the home server without the user's request.

In the future, we plan to implement an interface with a sentence-based and speech recognition interface.

References

1. Daqing Zhang, Zhengning Dai, Xiaohang Wang, Xiao Ni, Song Zheng, "A new service delivery and provisioning architecture for home appliances," *Consumer Electronics*, 2003. ICCE. 2003 IEEE International Conference on 17-19 June 2003 Page(s): 378 - 379.
2. M. Rahman, C. Akinlar, I. Kamel, "On secured end-to-end appliance control using SIP," *Networked Appliances*, 2002. Liverpool. Proceedings. 2002 IEEE 5th International Workshop on 30-31 Oct. 2002 Page(s): 24 -28.
3. M. Nikolova, F. Meijs, P. Voorwinden, "Remote mobile control of home appliances," *Consumer Electronics*, IEEE Transactions on, Feb. 2003. Vol. 49, Issue 1, pp. 123 – 127
4. IRENE C. Y. MA, J. IRVINE, "Characteristics of WAP traffic," Wireless Networks, Jan 2004, Vol. 10, Issue 1, pp 71 - 81
5. E.M.C. Wong, E.M.C, "Phone-based remote controller for home and office automation," *Consumer Electronics*, IEEE Transactions on, Feb. 1994. Vol. 40, No. 1, pp. 28 -34
6. Ghaderi, M., Keshav, S., "Multimedia Messaging Service: System Description and Performance Analysis, " *Wireless Internet*, 2005. Proceedings. First International Conference on, 10-15 July 2005, pp. 198 - 205
7. Ismail Coskun and H. Ardam, "A Remote Controller for Home and Office Appliances by Telephone," *Consumer Electronics*, IEEE Transactions on, Nov. 1998. Vol. 44, No. 4, pp. 1291-1297
8. Ivanov, R.S, "Controller for mobile control and monitoring via short message services," *Telecommunications in Modern Satellite*, *Cable and Broadcasting Service*, 2003. TELSIKS 2003. 6th International Conference on, Vol. 1, 1-3 Oct. 2003, pp.108 - 111
9. Li Jiang, Da-You Liu, Bo Yang, "Smart home research," *Machine Learning and Cybernetics* 2004. Proceedings of 2004 International Conference on, Vol. 2, 26-29 Aug. 2004 Page(s): 659 - 663
10. Neng-Shiang Liang, Li-Chen Fu, Chao-Lin Wu, "An integrated, flexible, and Internet-based control architecture for home automation system in the Internet era," *Robotics and Automation*, 2002. Proceedings. ICRA '02. IEEE International Conference on, Volume 2, 11-15 May 2002 Page(s): 1101- – 1106.
11. B.A. Miller, B.A., T. Nixon, C. T., Tai, C., M.D. Wood, M.D., "Home networking with Universal Plug and Play," *Communications Magazine*, IEEE, Vol. 39, Issue 12, Dec. 2001 Page(s): 104 - 109.

12. K. Arnold, A. Wollrath, B. O'Sullivan, R. Scheifler, and J. Waldo, "The Jini Specification," Reading, MA: Addison-Weslley, Reading, MA, USA, 1999.
13. D. Reilly, D., A Taleb-Bendiab, A., "An Jini-based infrastructure for networked appliance management and adaptation," *Networked Appliances*, 2002. Liverpool. Proceedings. 2002 IEEE 5th International Workshop on 30-31 Oct. 2002 Page(s): 161 - 167.
14. Q.H. Mahmoud, Q.H., "Using Jini for high-performance network computing," *Parallel Computing in Electrical Engineering*, 2000. PARELEC 2000. Proceedings, International Conference on 27-30 Aug. 2000 Page(s): 244 - 247.
15. R. Lea, R., S. Gibbs, S., A. Dara-Abrams, A., E. Eytchison, E., "Networking home entertainment devices with HAVi," *Computer*, Vol. 33, Issue 9, Sep 2000 Page(s): 35-43.
16. P. Dobrev, D. Famolari, C. Kurzke, B.A. Miller, "Device and service discovery in home networks with OSGi, " *Communications Magazine*, IEEE, Vol. 40, Issue 8, Aug. 2002 Page(s): 86-92.
17. K. Tan, T. Lee and C. Yee Soh, "Internet-Based Monitoring of Distributed Control Systems--An Undergraduate Experiment," *IEEE Transactions on Education*, Vol. 45, No. 2, May 2002.
18. Chi Chung Ko, Ben M. Chen, Shaoyan Hu, Vikram Ramakrishnan, Chang Dong Cheng, Yuan Zhuang, and Jianping Chen, "A Web-Based Virtual Laboratory on a Frequency Modulation Experiment," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Application and Reviews*, Vol. 31, No. 3, pp. 295-303, August 2001: 295-303.
19. N. Swamy, O. Kuljaca and F. Lewis, "Internet-Based Educational Control Systems Lab Using Net-meeting," *IEEE Transaction on Education*, Vol. 45, No. 2, pp. 145-151, May 2002: 145-151.
20. Peng Gong, Feng-jiao Qiu, Meng Liu, "A new algorithm based on DES and ECC for CSCW," Computer *Supported Cooperative Work in Design*, 2004. Proceedings. The 8th International Conference on,Volume 1, 26-28 May 2004 Page(s): 481-486. Vol.1
21. Christian Dewes, Arne Wichmann, Anja Feldmann, "Applications: An analysis of Internet chat systems," October 2003. Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement.
22. John Stone, Sarah Merrion, "Features: Instant Messaging or Instant Headache?," April 2004. *Queue*, Vol. 2, Issue 2.
23. Sun Microsystems Inc., "Jini Architecture Specification," Downloaded from, http://ww.sun.com/software/jini/specs
24. J-Sim Org., "J-Sim API Specification," Downloaded from http://www.j-sim.org/
25. MSN Messenger Protocol Documentation. Downloaded from http://www.hypothetic.org/docs/msn
26. Seong Joon LEE, Gwang Seon Ahn, "Remote Control for Information Appliances Using Instant Messenger," June 2005. Proceedings of the 2005 International conference on Embedded System and Applications.