A Remotely Manageable Electrocardiogram Measurement System for Home Healthcare using OSGI Framework

Ing-Yi Chen* Chao-Chi Huang

Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan, 106 R.O.C.

Received 3 Sep 2004; Accepted 15 Sep 2004

Abstract

This paper presents a remotely manageable electrocardiogram measurement system for managing middleware applications using OSGi and LDAP technologies. The platform incorporates a Device Connection Interface, a Data Transmission System, a Service Management Center, and a Health Care Portal System. In this paper we focus on the health care service manage framework which is intended to simplify the provision of applications and reduce the managing effort. A pilot application for telecardiology is presented and discussed. The system, if successfully operated, will offer better service management framework for large scale of homecare service devices.

***Keywords*:** Electrocardiogram, Telecardiology, OSGI

Introduction

Telehomecare is a relatively new field that uses the Internet to deliver access to health care information and services. Particularly, patients with chronic disease require frequent tracking to monitor the progress of the disease, compliance with treatment, and preventive care. Telehomecare extending the range of treatments is essential to improve the independent living for chronic disease patients.

However, chronic disease often occurs to the elder and they often do not easily adapt to new technologies. Complicated machines will hinder the elder from adopting the Telehomecare system. Thus, the development of easy-to-use methods and techniques supporting advanced home health care is the major challenge for the medical and informative community.

In order to support this desired functionality, the following issues need to be addressed: 1) ease of use regarding medical equipment, 2) simplified provision of application and operation, and 3) ease of networking infrastructure management.

A Home Gateway with ease of use can be an important feature for increasing the user acceptance rate. It provides an interface for remote access to the medical service center without complicated setting. However, as the scale of Home Gateway becomes larger, the management issues involved becomes more complicated, and the task of remaining cost effective becomes increasingly challenging. The purpose of the research is to develop a new Telehomecare system from a human-centered perspective with an emphasis on embedded application platform. Furthermore, a solution of centralized telehomecare service management system with remote service management and standardized application model has been developed.

This paper is organized as follows. In section 2, relevant previous works dealing with Telehomecare system are reviewed. In Section 3, the design choices of the system are illustrated and briefly discussed. Later, the detailed architecture and implementation of the Telemedicine System is presented in Section 4, and the result of the study is reported in Section 5. Finally, the conclusions and future works are discussed in Section 6.

Literature Review

There has been a considerable effort to investigate issues of supporting healthy aging adults in maintaining their usual lifestyles in the comfort of their homes [1-4]. Those studies mainly focus on delivering medical information via web technologies. However, those systems are PC-based health care systems to connect the healthcare center. However, the limitation on usability of those systems hampers the elderly patients who are not familiar with computer operations to use them.

The approach of usability provides an important insight regarding use of technology by an elderly patient population. Some Studies [5-8] use wireless technology to directly connect to healthcare center .The MobiHealth BAN system [9] enables remotely monitoring patients with 2.5/3G public wireless

* Corresponding author: Ing-Yi Chen
Tel: +886-2-27712171 ext. 4235; Fax: +886-2-87732945
E-mail: ichen@ntut.edu.tw

infrastructures. Besides, Patient data is collected through a Body Area Network. This system provides a mechanism extending the local services offered by a Body Area Network to remote healthcare center.

Instead of using PC system, it is an issue to provide an easier way for existing monitor device, which using on a PC based system, to link to the service center. One of commonly used options is to provide services to a home-based patient through a residential gateway. We propose a solution as a complete system not limited to PC-system. The main focus of this work is to make devices more supportive and easier to use by creating a common framework for controlling and monitoring remote devices.

Methodology

Design Choice

In this section we describe the main architecture choices that allow the requirements described above to be satisfied.

(1) Middleware Platform: To remotely deploy telehomecare services to home terminal which cover numerous operational systems, it is important to have a standard application platform to leverage the coupled environment. Because OSGI [10-13] is based open standard, it can be straightforward to reuse the health service by remotely deploying applications. The OSGi Service Platform provides a platform that service providers can use to deliver to the devices on their local network. Applications are known as bundles in the OSGi framework. Under OSGi architecture, application is encapsulated into a bundle which is a Java Archive file that contains the executable code together with all of its resources and a description specifying the class to start and stop the bundle. With the OSGI technologies, homecare services can be uniformly operated and managed by packing all these items together in a single JAR

(2) Data Transportation Interface: Typically, a gateway serves as a coordinator. It also can act as a buffer between a set of devices and the service center where communication is through a proprietary protocol. However, the devices interacting with a service endpoint may be changed another platform in the future. It will take all the overhead of programming effort but get little of the promised flexibility, scalability in return. Thus, data exchange using standardized formats is needed for cross-platform and cross-vendor to support. It can address cross-platform communication issue that arises with Internet appliances. In the research, we use Web Services [14-15] technologies based on XML to provide an interface in whatever area they emerge in the future. It provides means to offer device capabilities as well-defined connection interface to any other platform with the framework.

In the research, we adopt a standard JSR 172 architecture [16] as a Web Services client enabling developers to create a service once and then deploy it many times to diverse clients on diverse platforms. JSR 172 specifies standardized client-side technology to enable J2ME applications to consume remote services on typical Web Services architectures. The technologies ensure the interoperability of J2ME clients with Web services, and enable developers to reuse Web services components when they design these clients for enterprise services.

Another issue in Data Transportation Interface Design is to deal with the problem of asynchronous communication. Reliable communication is achieved by persistent message queuing facilities. Thus function can be achieved through a MOM (message oriented middleware) system [17-18]. For critical message (i.e. the doctor's advice), a mechanism which can help medical service to build fully reliable transportation is necessary. In the research, the JMS (Java Messaging Service) specification is used for providing a Java API for implementation of asynchronous communication.

(3) Centralized management Center: Directory is adopted as the enabling technology for centralized network operation management. It is well known that a directory is an object-oriented naming model that stores its information in a hierarchical tree structure [19-20]. Various network operation information in a homecare service provider such as home gateway profile, sub-organization profiles, and user information can be easily maintained in a directory. In addition, the information structure of a directory can closely reflect the organization structure of a homecare service provider. This structure makes service management easier since the same rules, policies and restrictions applied to a homecare center can be effortlessly replicated to the directory.

Architecture of the Telemedicine System

The Telehomecare System consists of a device connection interface, data transportation system, service management center and a service portal system. In the use case, we use a 12 Lead Interpretive Electrocardiogram (ECG or EKG) device to collect, store and forward vital signs data .This device provides an USB interface to connect to home gateway box. This Telecardiology service system not only provides a way for prophylaxis but also propose a solution to detect an arrhythmia attack for patients who are receiving rehabilitation care at home. After ECG data collects to medical service cater, the portal system provides a graphic display of the electrical activity of the heart.

Device connection Interface

Compared to UPnP device architecture, an existing device needs developing a custom connection interface independently. Under the OSGI programming model, we design an ECGDeviceBundle for ECG device attached to the home gateway. Three basic components need to be implemented: device driver, Communications API, and access application.

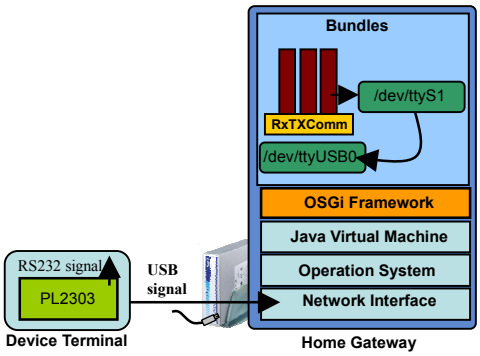

Figure 1. Interaction model between ECG monitor and home gateway

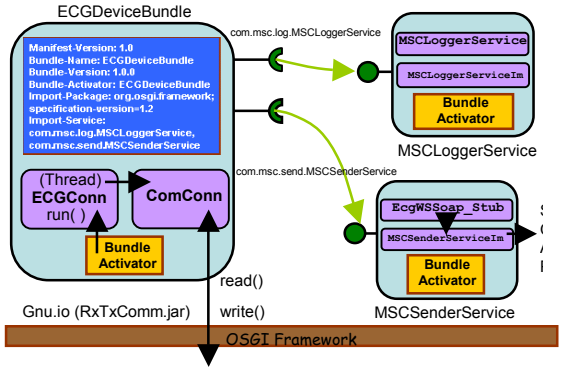

Figure 2. Architecture of the Device connection Interface

The model is illustrated in Figure 1, representing an ECG monitor is based on PL2303 serial-USB adapter which is capable of transforming RS-232 signal to USB signal. In order to control the adapter, we use an open-source communication API called RXTXcomm which is a Java communications API driver for Linux to access the serial port. This significant module provides the infrastructure for obtaining electrocardiogram data.

In Figure 2 denotes an ECGDeviceBundle programming model:

(1) Create an ECGDeviceBundle implementing Bundle-Activator class as the entry point to the bundle.

(2) In the ECGDeviceBundle.start method, create and open the ECGConn Thread service to automatically track ComConn.java which connects a serial cross link form /dev/ttyS0 to /dev/ttyUSB0.

After ECGDeviceBundle gets ECG data, it stores those records and passes it to service center by invoking two functions which provide logging service and delivering service. However, those functions also are referenced by other bundles. In order to reduce code redundancy and makes it easier to architect complex multilayer applications, we adopt a feature in OSGi framework providing a powerful way for bundles to interact with each other. A bundle can dynamically register services with OSGI container to make those functionalities available to other bundles. In the context of the Framework, a service is an object registered with an interface name. ECGDeviceBundle finds the service reference through a

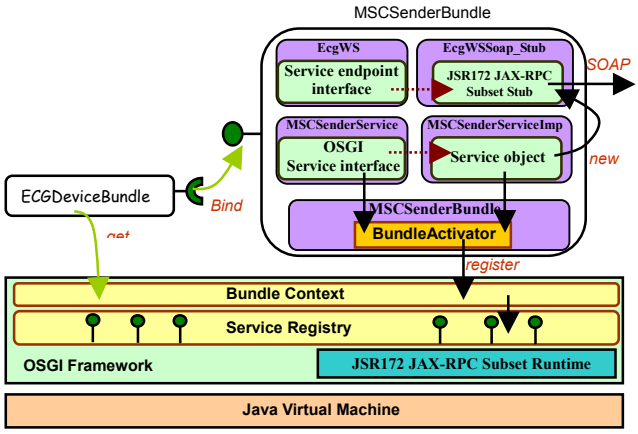

Figure 3. Data transmission with Web Service technologies

lookup API (context.getServiceReference) in the framework to obtain the service implementation object from the MSCLoggerSerice and MSCSenderService.

Data Transmission System

(1) The Data Transmission System is composed of two main components which are MSCSenderBundle and MSCReceiverBundle. The MSCSenderBundle bundles complete data transmission tasks by communicating with each other using a SOAP protocol with HTTP as its transport.

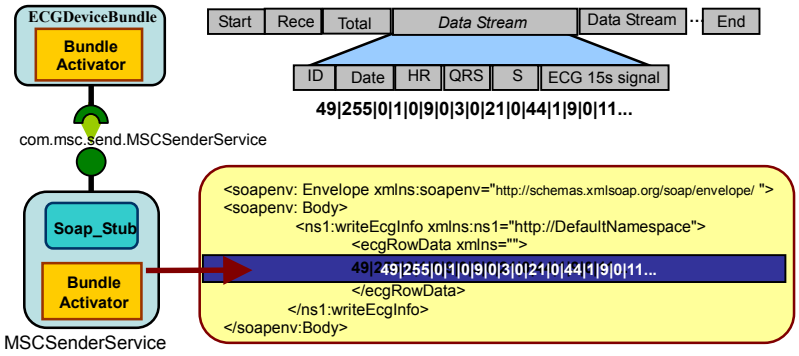

Figure 4. Soap message of ECG information

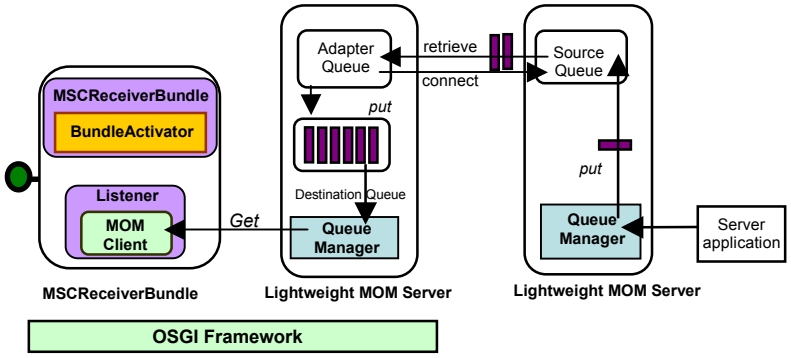

Figure 5. Reliable communication modules between home gateway and service center

The MSCSenderBundle includes five classes (see Figure 3):

(1) EcgWS: The Service endpoint interface (SEI) which defining the service methods of the locator class to call a Web Service.

(2) EcgWSSoap_Stub: The stub client uses a defined service endpoint interface based on the WSDL document to get a stub for a specific Web service.

(3) MSCSenderService: Define the service interface as a Java interface which exports as a OSGI service.

(4) MSCSenderServiceImp: An Implementation class for the MSCSenderService interface.

(5) MSCSenderBundle: A BundleActivator class implements the required OSGi life cycle methods and registers MSCSenderService service with the container upon startup.

To standard as services in OSGI container, the MSCSenderBundle.java uses registerService method to register as a service in the container. The registerService requires a service interfaces name for reference and a service implement class. By this way, MSCSenderService exposes as aservice to provide transmission channel to send biological signal.

To reduce network transport effort, we use a function to simplify the data format. When the ECG single pass through the function, those data also be dispatched into a web services channel. The channel wraps up ECG signal in a XML tag named “ecgRowData”. Figure 4 shows the soap message of ECG information.

The other issue is how to ensure reliable communication paths between home gateway and service center. We use a lightweight MOM middleware to provide an asynchronous messaging service that guarantees diagnosis messages delivery. By integrating with the relative MOM middleware on the service center, the lightweight MOM can ensure the security of communications and guarantee delivery of usage data despite transient network failures and disconnected mode. When the doctor completes diagnosis, a notification message is generated and sent by application for storing diagnosis result. Then the message will be routed from the Server side MOM server to the lightweight MOM client, and finally to the appropriate medical application.

The kernel of the lightweight MOM is queue manager. Its task is to manage queues and messages for applications. According to the components and functionalities, a queue manager can be configured into various types. On service center, we setup a queue to temporarily store messages for queue manager which pulls messages from the queues on home gateway platform, On home gateway, another type of queue is used by its queue manager to get messages from queues on service center.

Figure 5 depicts the architecture of communication modules which consists of three main parts. The lightweight MOM server resides in the control center, acting as the tools for messaging. It uses an automated pull mechanism initiated by the client application to get messages sent to one of its queues. Two types of additional queues -- the source queue and the adapter queue -- are used in the pull processing.

SMFAdministration(2.0)	smfb-d/SMFAdministration
Device Agent Handler(1.0)	file:/bundlefiles/deviceagenthandler.jar
Device Agent Config(1.0)	file:/bundlefiles/deviceagentconfig.jar
Device Agent(1.0)	file:/bundlefiles/deviceagent.jar
GWAdmin Servlet(1.0)	file:/bundlefiles/gwadmin.jar
HTTPSecurityPluginSample(1.1)	smfb-d/HTTPSecurityPluginSample
MSCViewerBundle(1.0.0)	file:bundlefiles/MSCViewerBundle+1_0_0.jar
ECGDeviceBundle(1.0.0)	file:bundlefiles/ECGDeviceBundle+1_0_0.jar
MSCLoggerBundle(1.0.0)	file:bundlefiles/MSCLoggerBundle+1_0_0.jar
MSCSenderBundle(1.0.0)	file:bundlefiles/MSCSenderBundle+1_0_0.jar
MSCTestLogBundle(1.0.0)	file:bundlefiles/MSCTestLogBundle+1_0_0.jar
MSCReceiverBundle(1.0.0)	file:bundlefiles/MSCReceiverBundle+1_0_0.jar
SMF System Bundle(3.5.3)	System Bundle

Figure 6. Bundle Management Interface

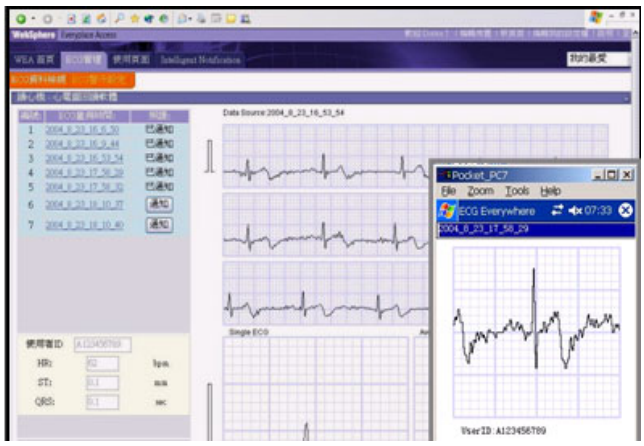

Figure 7. Homecare service portals for diagnosis

The source queue resides on the server and holds messages destined for the client queue manager. When a server-side application attempts to put a message on a client-side queue, the message gets placed in the destined queue. The home-server queue resides on a client. When triggered, it attempts to initiate communication with the server and checks with the destined queue for messages waiting to be delivered. Pending messages are pulled to the client by the target queue and put into the correct destination queue. Therefore the designed model which provides asynchronous communication can address the requirement of reliability.

Service Management Center

Bundle Management Center is an Internet services manager that enables you to centrally manage Bundle application. We propose architecture based on LDAP technology for remote management on the Telehomecare. The hierarchical structure of the directory enables the delegation of administrative control. If a hierarchical relationship is defined in the directory, various deployment scenarios can be realized for centralized management. For updating local configurations and software Device agent need to be designed, the device agent handles device enrollment, device configuration jobs, software distribution jobs, and hardware and software inventory jobs.

Whenever an application needs to monitor the gateway access in a specific department, the device management center looks up device information by name in the directory through a programming interface. GUID can be a pointer to the relevant information stored in an external device profile table. Whenever network administrators login to the web-based management system, related device profiles are presented according to the GUIDs.

Results

We have developed a Telehomecare system based on OSGI, Web Services, MOM, and LDAP standards. We have also successfully implemented application systems which can extend the range of treatments and manage monitor clients. Furthermore, we have achieved the following:

(1) Using OSGI and LDAP technology to build a flexible platform on which homecare services provider can host management services for multiple healthcare equipments.

(2) Utilizing Web Service for realizing a transportation interface for interoperating heterogeneous networks. Thus, the healthcare services bundles have the possibility to send message to target whichever service center platform it runs on.

(3) Implementing MOM services as OSGi bundles. The function is defined as a reliable message delivery channel between home gateway and service center.

Furthermore, an approach has been introduced to manage service around home environments with LDAP technology. Such result can not only be applied to remote management but also reduce manpower cost. As depicted in figure 6, the administrator can quickly enable Telehomecare services by leasing a shared directory.

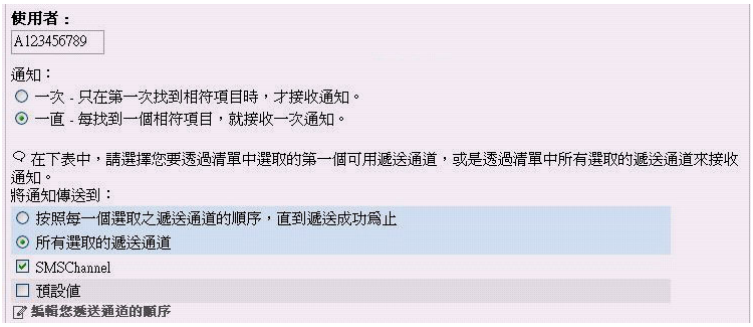

Figure 8. Administrator's interface of Intelligent Notification service

In this project, the diagnosis portal is the other main subsystem of the whole system including the modules for "consulting service", "diagnosis service", that provides the efficient ways for doctors, patients and medical specialists to communicate with each other through the Internet and whenever the home-care patient takes the patient commands. The bundles load all the information stored in the ECG monitor and send message to the PDA screen to display the status of the patient. As shown in Figure 7, ECG data collected form patient will reflect on the user-interface.

Furthermore, Device Management Center includes an Intelligent Error Alert engine which is designed to judge if the ECG data is abnormal. Messages can be delivered via three mechanisms: WAP, SMS, and mail. The Administrator's interface of the Intelligent Notification service is depicted in Figure 8.

Conclusion & Future works

In this research, we have presented our work on a Telehomecare solution for remote management for large scale distributed service. In addition, we have developed a bundle management center for delivery of new services and applications via the Internet. Comparing to the traditional Telehomecare system, the proposed system is not only more effective but also providing real-time remote diagnosis and maintenance service.

Acknowledgments

The authors would like to express our gratitude for the support of the National Science Council, Taiwan, R.O.C., under Grant NSC-93-2213-E-027-045-. We also thank several anonymous referees for their constructive reviews and comments.

Reference

[1] Istepanian, R.S.H., "Telemedicine in the United Kingdom: Current Status and Future Prospoects", *IEEE Transactions on Information Technology in Biomedicine*, 3(2):158-159, 1999.

[2] E.J. Nordal, D. Moseng, B. Kvammen, and M.L. Lochen, "A Comparative Study of Teleconsultations versus Face-to-Face Consultations," *Journal of Telemedicine and Telecare*, 7(5): 257-265, 2001.

[3] J.J. McKee, N.E. Evans, and F.J. Owens, "Digital Transmission of 12-Lead Electrocardiograms and Duplex Speech in the Telephone Bandwidth*," Journal of Telemedicine and Telecare*, 2(1): 42-49, 1996.

[4] Perednia, D.A. and Allen, A., "Telemedicine Technology and Clinical Applications," *Journal of American Medical Association*, 273(6): 483-488, 1995.

[5] E. Jovanov, et al, "Stress Monitoring Using a Distributed Wireless Intelligent Sensor System", *IEEE Engineering in Medicine and Biology Magazine*, 22(3):49-55, 2003.

[6] Hung, K. and Zhang, Y-T., "Implementation of a WAP-Based Telemedicine System for Patient Monitoring", *in IEEE transactions on Information Technology in Biomedicine*, 7(2):101–107, 2003

[7] Woodward, B., Istepanian, R. S. H**.**, and Richard, C., "Design of a telemedical system using a mobile phone", *IEEE Trans. Information Technology in Biomedicine*, 5(1):13-15, 2001

[8] Pattichis CS, Kyriacou E, Voskarides S, Pattichis MS, Istepanian R, Schizas CN, "Wireless Telemedicine Systems: An Overview", *IEEE Antennas & Propagation Magazine*, 44(2):143-153, 2002.

[9] Dimitri Konstantas, Val Jones, Richard Bults and Rainer Herzog, "MobiHealth _ innovative 2.5 / 3G mobile services and applications for healthcare", 11th IST Mobile and Wireless Telecommunications, June 2002, Thessaloniki , Greece.

[10] Open Services Gateway Initiative, "OSGI Service Platform," Specification Release 3.0, March 2003

[11] H. Cervantes and R. S. Hall. "Beanome: A Component Model for the OSGi Framework", Workshop on Software Infrastructures for Component-Based Applications on Consumer Devices, September 2000.

[12] H. Cervantes and R.S. Hall. "Automating Service Dependency Management in a Service-Oriented Component Model*",* ICSE CBSE6 Workshop, May 2003, Portland, United-Stated.

[13] R. S. Hall and H. Cervantes "Gravity: Supporting Dynamically Available Services in Client-Side Applications*",* ESEC/FSE, September 2003, Helsinki, Finland.

[14] Curbera, F., Duftler, M., Khalaf, R., Nagy, W.; Mukhi, N., and Weerawarana, S.,"Unraveling the Web Services: an Introduction to SOAP, WSDL, and UDDI", *IEEE*, 6(2): 86-93, 2002.

[15] Vinoski,S., " Web Services Interaction Models, Part 1: Current Practice", *IEEE Internet Computing*, 6(3),89-91,2002.

[16] Java Community Process, "JSR-00172 J2ME Web Services Specification," Specification Release 2.1, March 2003

[17] Jun-Jang Jeng,Youssef Drissi,IBM T.J. Watson Research Center," A Predictive Event Notification System for e-Commerce Environment", The Twenty-Fourth Annual International Computer Software and Applications Conference ,October 2000, Taipei, Taiwan.

[18] P.Pietzuch and J.Bacon,Hermes "A Distributed Event-Based Middleware Architecture" on DEBS'02 with IEEE ICDCS, 2002, Vienna.

[19] Fabián E. Bustamante, Patrick Widener, and Karsten Schwan, "Scalable directory services using proactivity", Proceedings of the 2002 ACM/IEEE conference on Supercomputing, November 2002.

[20] Park, J., Sandhu, R., and Ahn, G., "Role-based access control on the web." *ACM Transactions on Information and System Security*, 4(1): 37-71, February 2001.