Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Design and Implementation of an OSGi-Centric Remote Mobile Surveillance System

Jenq-Shiou Leu
Department of Electronic Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
jsleu@mail.ntust.edu.tw

Wei-Hsiang Lin
Department of Electronic Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
m9702108@mail.ntust.edu.tw

Hung-Jie Tzeng
Department of Electronic Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
b9502082@mail.ntust.edu.tw

***Abstract*— Conventional surveillance systems capture real-time events from cameras mounted at fixed locations. A surveillance user then monitors the occurrences via a browser at a stationary computer. To enhance the mobility at the viewer side and the camera side, we implement a remote mobile surveillance system by integrating some inexpensive techniques - an OSGi service platform, which can easily be implanted with future developed applications, a camera mounted on an embedded system which is carried by a robot - Lego MindStorms NXT, and a J2ME based viewer and controller program on a mobile phone. With assistance from modern wireless networks, surveillance users can handily monitor the remote events in a wider viewing range at anytime and anywhere.**

***Keywords*— Surveillance System, MindStorms NXT, J2ME, OSGi**

I. INTRODUCTION

Thanks to the rapid growth of Internet, surveillance users can monitor events at a remote site by an Internet browser via a surveillance system. Thus the live events can easily be captured at the office, at the school, or at anywhere users can access Internet. In a traditional surveillance system, many wallboard cameras are mounted at fixed locations to capture occurrences. Users then keep a watchful eye on the videos acquired by these cameras via a monitor at a stationary computer. Internet indeed facilitates the remote monitoring but it still lacks some flexibility. For example, viewing at a stationary computer would restrict the user mobility. If the monitoring application can be run on a hand-carried device, such as a cellular phone or a laptop computer for a viewer, acquiring remote events can be achieved at anytime, anywhere via any attachable wireless network, such as the GPRS, Wi-Fi, and UMTS to the emerging 3GPP-LTE or WiMAX. Furthermore, the pre-installed cameras at fixed locations might cause some unseen corners. If the capturing camera can be equipped in a movable robot, a surveillance viewer can watch a wider range as the reviewer wishes. In the past, many efforts are put on improving the inflexibility in a traditional surveillance system. A web-based surveillance system in [1] was proposed for surveillance viewers to perform monitoring and controlling remotely by cellular phones. A network camera equipped on a mobile robot [2, 3] can help capture events in wider and more dynamic angles. However, if the mobile camera and the mobile viewer/controller on cellular phone can be integrated on a common service platform, the service expandability can be achieved. In this article, we aim to implement a remote mobile surveillance system by integrating an open service platform, a movable robot based camera, and a viewing application on a cellular phone. Therefore, we develop a mobile surveillance system in Fig. 1 by some inexpensive techniques: a camera mounted on an embedded system - Dmatek ARM9 DMA-NAV2450 [4] which is carried by a robot - Lego MindStorms NXT [5], a J2ME [6] based viewer and controller program on a mobile phone, and an OSGi gateway.

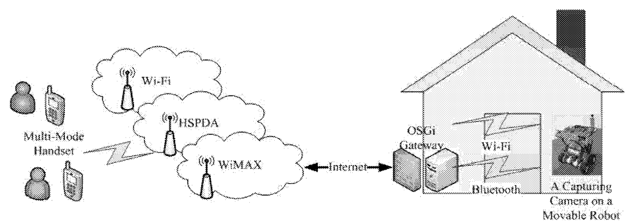

Figure. 1: OSGi-Centric Remote Mobile Surveillance System

The rest of this paper is organized as follows. Section 2 illustrates some related techniques and their applications. Section 3 describes the system architecture of the proposed OSGi-centric remote mobile surveillance system. Section 4 depicts the sequence diagram and class diagrams in designing this system. Section 5, we evaluate the service performance by the actual frame receiving rates on different mobile devices via different wireless networks. A brief concluding remark is finally drawn in Section 6.

II. RELATED WORK

The Open Services Gateway Initiative (OSGi) Alliance was formed to conduct open specifications for delivering services to local networks and devices [7]. Recently, many developments relating middleware construction are based on the OSGi standard [8-10]. The OSGi service platform, a general-purpose, secure, managed Java software framework, expands Internet services to homes [11-13], vehicle management [14, 15], automotive [16], health care [17] and so on. OSGi specification defines a service-oriented

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

framework which provides a publishing, finding and binding model for using services so it can support the deployment of extensible and downloadable service applications known as bundles. Bundles are built around a set of collaborative services available from a shared service registry. Therefore, OSGi can play a good broker role to aggregate applications.

As hardware costs are coming down dramatically and capabilities of robot are increasing fast, robotics is becoming more important in everyday life. Lego MindStorms series products have been developed by LEGO Company for years to enable developers to have an inexpensive robotic environment. One of its major contributions is to build simple, portable, and inexpensive experiments for proving the concept of a mobile robot service in the development beginning by not having extensive lab facilities [18, 19].

Thanks to the wireless technology revolution, a wireless user can access Internet services at anytime, anywhere via ubiquitous networks by a multi-mode device. In the mobile world, users visualize the cyber net by their handheld mobile phones with a small screen. WAP (Wireless Application Protocol) and J2ME (Java 2 Micro-Edition) [20] are two typical mobile based development technologies. J2ME sets up 3-tier of J2ME Virtual Machine, Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) to achieve modularity and flexibility across many smart phones [21]. Compared with WAP, J2ME has following advantages: direct access to Internet in a client-server mode without a gateway like WAP and more compatible file formats than WAP. J2ME is more suitable to be used to develop a mobile application with higher computation capability. A J2ME-based wireless intelligent video surveillance system [22] was implemented to make a useful supplement of a traditional monitoring system with its good mobility.

In our work, Java, renowned for its openness and portability across many platforms, is chosen as the main developing technique to realize our remote mobile surveillance system. That includes building Java-based bundles on the OSGi service platform, controlling the NXT brick between bundles and a Lego MindStorms NXT over a Bluetooth connection by the iCommand Java package and the RXTX library [23], delivering the captured videos back to OSGi through a Java-based socket program and using J2ME to develop a handy remote surveillance viewer on a multi-mode mobile phone.

III. System Architecture

The whole system architecture and some main application modules in our OSGi-centric remote mobile surveillance system are shown in Fig. 2.

A. *OSGi Service Platform*

The OSGi platform mainly comprises a JVM (Java Virtual Machine), a set of running applications called bundles, and an OSGi framework. A bundle is a minimal deliverable application in OSGi and managed by the OSGi framework. Meanwhile, a service is the minimal functioning unit in OSGi. Therefore, a bundle is designed as a set of cooperating services, which are discovered after being published in the OSGi service registry.

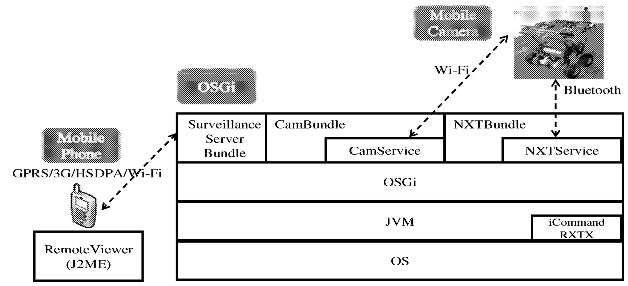

Figure 2: The System Architecture in the OSGi-Centric Remote Mobile Surveillance System

There are three bundles and two services on the OSGi platform:

- SurveillanceServerBundle:

  SurveillanceServerBundle is responsible for authorizing users, handling service requests including delivering captured videos to the remote viewer by invoking functions in CamService and controlling the MindStorms NXT as the remote viewer demands by invoking functions in NXTService.

- CamBundle:

  CamBundle is responsible for registering, publishing CamService, which works for retrieving the remote surveillance videos, in OSGi.

- NXTBundle:

  NXTBundle is responsible for registering, publishing NXTService, which works for controlling NXT, in OSGi.

B. *Mobile Camera*

Lego MindStorms NXT already has a digital camera accessory but the video quality it captures is not good enough for remote viewing. Hence we mount a digital camera on an embedded system - Dmatek ARM9 DMA-NAV2450 which is carried by Lego MindStorms NXT to capture the events at the locations where the viewer can ask NXT to move to. The videos are captured by the camera and sent back to the CamService via a Wi-Fi network interface card. Since the limited viewing screen on the mobile phone, we degrade the original captured video frame dimension from 320x240 to meet the screen sizes of mobile phones, such as 235x279(for SonyEriccson G900), 171x166(for SonyEriccson K618i) and so on. The parameter for the frame size is auto-negotiated when the mobile device connects to SurveillanceServerBundle. Furthermore, a trade-off always happens between the frame transmission cost and viewing quality. To our best experience, downgrading the frame quality to 10% can save considerable transmission cost but still meet the viewing quality on the mobile phone. Frame downsizing and quality downgrading can make the surveillance video transmission more effectively. Meanwhile, we use part of the Java APIs of Lego MindStroms NXT in Tab. 1 to develop the control program - NXTService on OSGi for packing user commands by the iCommand package and communicating with NXT via a Bluetooth connection created by the RXTX library to manipulate Lego MindStroms NXT.

TABLE I. JAVA APIS OF LEGO MINDSTROMS NXT

Class	Function

Motor	forward();
	backward();
	setSpeed();
	stop();
	flt();

Light	activate();
	getLightPercent();
	getLightValue();
	passivate();

Touch	isPressed();
Sound	getdB(); getdBA();
Ultrasonic	setMetric(); getDistance();

The physical construction of the mobile camera is shown in Fig. 3.

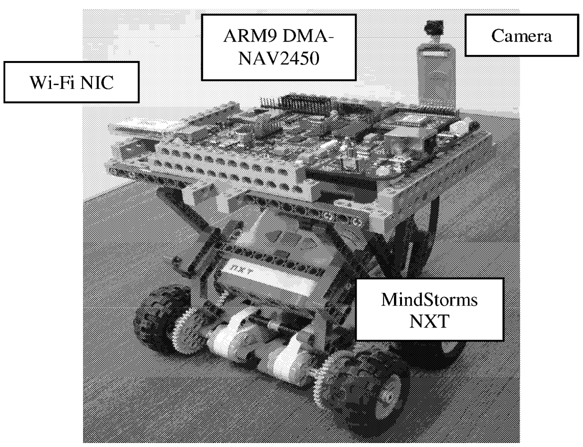

Figure 3: A Mobile Camera Mounted on Dmatek ARM9 DMA-NAV2450 Carried by Lego MindStorms NXT

C. Mobile Phone

J2ME is used to develop the controlling and viewing functions on a mobile phone. A viewer can use the keypad of mobile phone to remotely control the moving direction of the mobile camera via NXTService as well as to retrieve the surveillance videos captured by the remote camera via CamService.

IV. SEQUENCE DIAGRAM AND CLASS DIAGRAMS

The sequence diagram in the OSGi-centric remote mobile surveillance system is illustrated in Fig. 4. The generic process flows take place between four major entities: RemoteViewer, SurveillanceServerBundle, CamBundle, and NXTBundle.

An OSGi service is defined by a service interface, specifying the service's public methods and being implemented as a service object, owned by, and runs within, a bundle. The bundle is in charge of registering the service object with the OSGi service registry so that its functionality is available to other bundles. That means other bundles therefore recover and invoke services that are registered in OSGi.

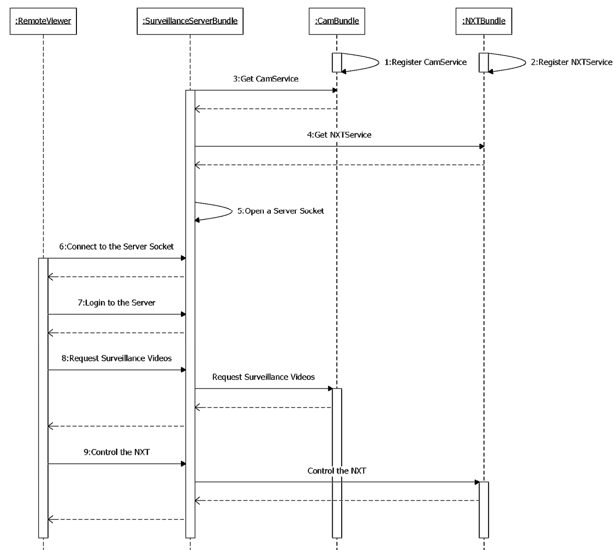

Figure 4: The Sequence Diagram of the OSGi-Centric Remote Mobile Surveillance System

1. Register CamService:

   CamBundle registers and publishes the CamService in the OSGi service registry.

2. Register NXTService:

   NXTBundle registers and publishes the NXTService in the OSGi service registry.

3. Get CamService:

   SurveillanceServerBundle can get the CamService after discovering the registered CamService on the OSGi framework.

4. Get NXTService:

   Similarly, SurveillanceServerBundle can get the NXTService after discovering the registered NXTService on the OSGi framework.

5. Open a Server Socket:

   SurveillanceServerBundle opens server socket and listens to this socket to wait for the client connection requests.

6. Connect to the Server Socket:

   The RemoteViewer activates the viewer/controller program and connects to the SurveillanceServerBundle.

7. Login to the Server:

   Remote users need to input a legitimate pair of username and password and submit them for login. The SurveillanceServerBundle would authenticate the user.

8. Request Surveillance Videos:

   If the surveillance user is authenticated, the user can obtain the videos from the camera on NXT via CamService on OSGi.

9. Control the Lego Robot:

   If the surveillance user is authenticated, the user can control NXT via NXTService on OSGi.

Related class diagrams in the OSGi-centric remote mobile surveillance system are depicted in Fig. 5. RemoteViewerMIDlet is a J2ME based application for users to view the remote surveillance videos and control the remote mobile camera on a cellular phone. RemoteViewerCanvas is designed to display the surveillance videos. The CamService and NXTService define the service interfaces and are registered and published in the OSGi framework by CamBundle and NXTBundle, respectively. SurveillanceServerBundle then invokes functions in CamBundle and NXTBundle to retrieve videos and control NXT.

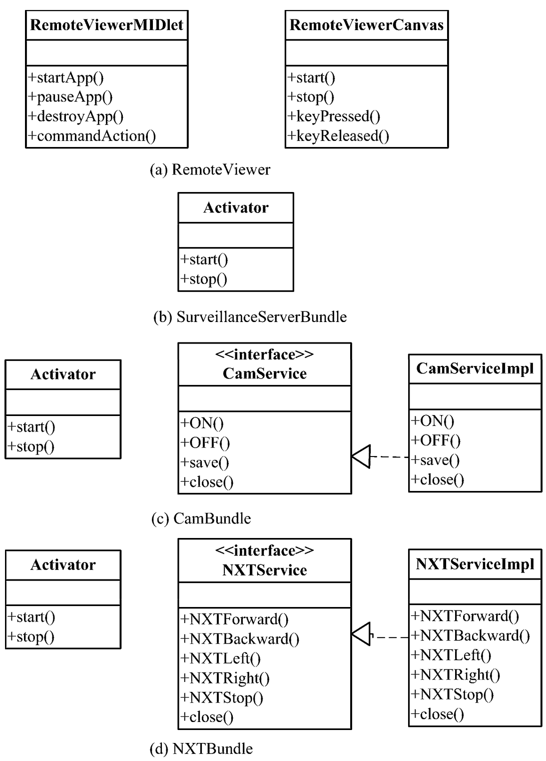

Figure 5: Related Class Diagrams in the OSGi-Centric Remote Mobile Surveillance System

V. SYSTEM EVALUATION

To enhance the mobility and responsiveness, our remote mobile surveillance system broaches the cellular or wireless transmission to surveillance video delivery between two mobile ends. Videos Captured by a Mobile Camera on NXT sent to a mobile phone for viewing as Fig. 6 shows. The mobile camera mounted on NXT would pause 0.1 sec every capturing an image with an origin dimension - 320x240 (around 62KB). The capturer program then downsizes the frame dimension according to the screen size of the mobile phone and downgrades the frame quality to 10%.

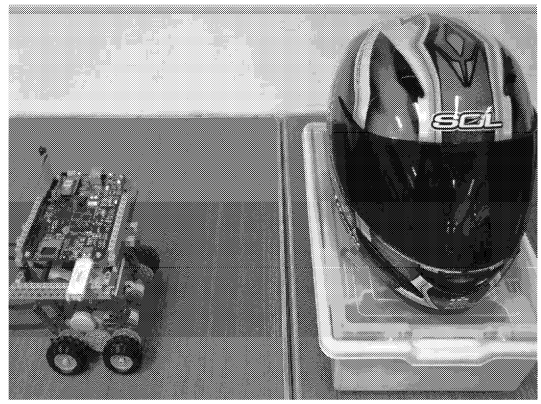

Figure 6 (a): Video Capturing by a Mobile Camera on NXT

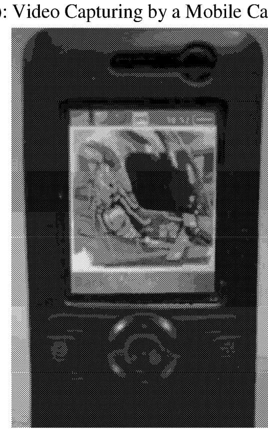

Figure 6 (b): Controlling and Viewing on a Mobile Phone - Sony Ericsson K618i

Afterward, the captured video frame is sent to the OSGi platform via the Wi-Fi network and then relayed to the mobile phone for displaying via the wireless network which the remote viewer attaches. Our evaluation focuses on the actual frame receiving rates on different mobile devices via different wireless networks and the results are shown in Tab.2.

TABLE II. EVALUATION RESULTS

Device	A Phone Simulator on Laptop	A Phone Simulator on Laptop	Sony Ericsson G900	Sony Ericsson K618i
Access Network	Wi-Fi	Wi-Fi	3G	3G
Scaled Frame Dimension	235x279	171x166	235x279	171x166

Device	A Phone Simulator on Laptop	A Phone Simulator on Laptop	Sony Ericsson G900	Sony Ericsson K618i
Scaled Frame Size (KB)	2.2	1.4	2.2	1.4
Average Frame Receiving Rate (f/s)	5.633	7.667	5.617	7.65

VI. CONCLUSION

The OSGi-Centric remote mobile surveillance system proposed in this paper provides an integrated solution of surveillance service. OSGi is chosen as the central service platform because of its future extensibility. The integrated system can increase the mobility at two ends - a mobile user end to monitor and control the capturing and a mobile camera end to raise the event capturing flexibility. We develop this system by some inexpensive techniques - an OSGi service platform, which can easily be implanted with future developed applications, a camera mounted on an embedded system which is carried by a robot - Lego MindStorms NXT, and a J2ME based viewer and controller program on a mobile phone to realize this concept. In this article, we share our design and implementation philosophy and finally evaluate the service performance via different mobile devices to access the remote mobile surveillance service via different wireless networks.

REFERENCES

[1] Y. Imai, Y. Hori, and S. Masuda, "Development and a brief evaluation of a web-based surveillance system for cellular phones and other mobile computing clients," in Human System Interactions, 2008 Conference on, 2008, pp. 526-531.

[2] T. Yu-Chee, W. You-Cbiun, C. Kai-Yang, and H. Yao-Yu, "iMouse: An Integrated Mobile Surveillance and Wireless Sensor System," Computer, vol. 40, pp. 60-66, 2007.

[3] P. Jung-Hyun and S. Kwee Bo, "A design of mobile robot based on Network Camera and sound source localization for intelligent surveillance system," in Control, Automation and Systems, 2008. ICCAS 2008. International Conference on, 2008, pp. 674-678.

[4] " Dmatek ARM9 DMA-NAV2450," http://www.dmatek.com.tw/en/tn/index.asp.

[5] "Lego MindStorms NXT," http://mindstorms.lego.com/.

[6] "Java 2 Platform, Mobile Edition," http://java.sun.com/javame/index.jsp.

[7] D. Marples and P. Kriens, "The Open Services Gateway Initiative: an introductory overview," Communications Magazine, IEEE, vol. 39, pp. 110-114, 2001.

[8] L. Choonhwa, D. Nordstedt, and S. Helal, "Enabling smart spaces with OSGi," Pervasive Computing, IEEE, vol. 2, pp. 89-94, 2003.

[9] T. Gu, H. K. Pung, and D. Q. Zhang, "Toward an OSGi-based infrastructure for context-aware applications," Pervasive Computing, IEEE, vol. 3, pp. 66-74, 2004.

[10] D. Valtchev and I. Frankov, "Service gateway architecture for a smart home," Communications Magazine, IEEE, vol. 40, pp. 126-132, 2002.

[11] W. Chao-Lin, L. Chun-Feng, and F. Li-Chen, "Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 37, pp. 193-205, 2007.

[12] J. E. Loez de Vergara, A. Villagra Victor, C. Fadon, J. M. Gonzalez, J. A. Lozano, and M. Alvarez-Campana, "An autonomic approach to offer services in OSGi-based home gateways," Computer Communications, vol. 31, pp. 3049-3058, 2008.

[13] L. Cheng-Liang, W. Pang-Chieh, and H. Ting-Wei, "A wrapper and broker model for collaboration between a set-top box and home service gateway," Consumer Electronics, IEEE Transactions on, vol. 54, pp. 1123-1129, 2008.

[14] K. Hyun-Chul and S. Dong-Ryeol, "OSGi Based Subway Management System," in Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on, 2008, pp. 564-566.

[15] J. Yi-Seok, N. Choon-Sung, J. Hee-Jin, and S. Dong-Ryeol, "Train Auto Control System based on OSGi," in Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on, 2008, pp. 276-279.

[16] L. Yuantao, F. Wang, H. Feng, and Z. Li, "OSGi-based service gateway architecture for intelligent automobiles," in Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, 2005, pp. 861-865.

[17] C. Ing-Yi and T. Chen-Hsin, "Pervasive Digital Monitoring and Transmission of Pre-Care Patient Biostatics with an OSGi, MOM and SOA Based Remote Health Care System," in Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on, 2008, pp. 704-709.

[18] B. S. Heck, N. S. Clements, and A. A. Ferri, "A LEGO experiment for embedded control system design," Control Systems Magazine, IEEE, vol. 24, pp. 61-64, 2004.

[19] S. H. Kim and J. W. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms," Education, IEEE Transactions on, vol. 52, pp. 99-108, 2009.

[20] K. Read and F. Maurer, "Developing mobile wireless applications," Internet Computing, IEEE, vol. 7, pp. 81-86, 2003.

[21] D. S. Kochnev and A. A. Terekhov, "Surviving Java for mobiles," Pervasive Computing, IEEE, vol. 2, pp. 90-95, 2003.

[22] L. Xu, Z. Wang, H. Wang, A. Shi, and C. Li, "A J2ME-Based Wireless Intelligent Video Surveillance System Using Moving Object Recognition Technology," in Image and Signal Processing, 2008. CISP '08. Congress on, 2008, pp. 281-285.

[23] "RXTX," in http://www.rxtx.org/.