

iMMS: Interactive multimedia messaging service

J. Shen
P. Sun
J. Zhang
S. Song

Multimedia messaging service (MMS) promises a dramatic increase in messaging capabilities for mobile phone users that will enrich their experience. For network operators and content and service providers, it will create a major new source of revenue because it adds features such as color pictures, animations, audio samples, and video clips. However, the current MMS standard provides only a few features that support user interaction, limiting the wide use of value-added messaging services. In this paper, we propose interactive MMS (iMMS) technology that can extend the current MMS standard and enhance interaction capability by embedding eXtensible HyperText Markup Language (XHTML) technology into MMS Presentation Language. Moreover, an MMS application server (MAS) is presented in our research work. This framework can help service providers to quickly develop and deploy MMS services. We have implemented and tested a prototype that includes an interactive MMS client on a handheld device and an MAS on a Microsoft Windows® 2000 server.

Introduction

Multimedia messaging service (MMS) [1, 2] is a new global messaging standard that enables a wide range of different media elements—including text, pictures, audio, and video—to be combined and synchronized in messages sent between mobile devices. MMS is designed to be used on 2G, 2.5G [which includes General Packet Radio Service (GPRS)], and 3G networks, with the experience being richer as the network, bearer, and device capabilities permit.

For device users, MMS enhances personal connectivity and productivity through a more immediate exchange of rich content. For instance, while on the road, users can receive a localized city map; or, while at a conference, an up-to-the-minute graph or layout. For network operators, MMS promises additional revenue as a result of increased air time, heavier all-around usage, service differentiation, and customer loyalty. Market studies [3, 4] show that not only are users enthusiastic about MMS, they are also willing to pay as much as five times more for the service

than they currently pay for short messaging service (SMS). By deploying MMS today, operators can secure a strong market position early in the personal wireless multimedia era.

However, most current MMS applications focus only on the transmission of images, ring tones, and text, and there is no specification in the current MMS standard that defines how to support an interactive model for MMS service. This is extremely limiting to widespread use of MMS services, especially when multiple interactions among users, MMS terminals, and back-end services are required.

In this paper, we discuss an interactive presentation markup language and an MMS middleware architecture that support interactive MMS (iMMS) services. A prototype was implemented to support the research work. We begin by presenting the requirements of iMMS features. We then propose our interactive solution, and this is followed by a description of the architecture of an MMS application middleware that supports value-added

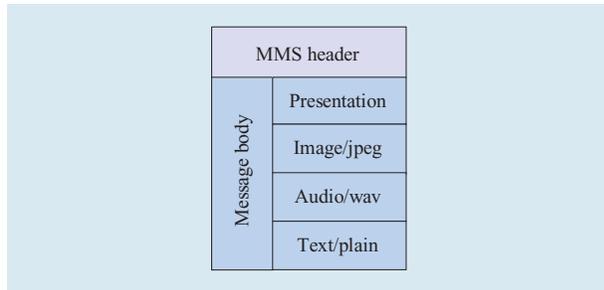


Figure 1
Multimedia message structure.

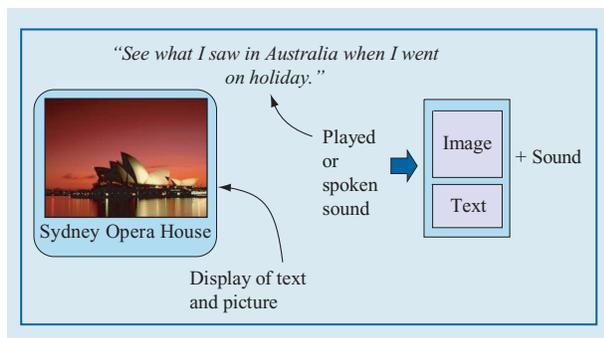


Figure 2
Example message with multimedia content.

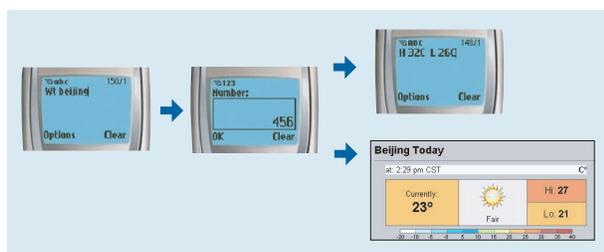


Figure 3
Weather service user interface.

services for service providers. Finally, we summarize our work and briefly discuss future activities.

MMS interaction requirements

As shown in **Figure 1**, a multimedia message consists of a message header and a message body [5–7]. The header contains MMS-specific information in a protocol data unit. This information consists mainly of how to transfer the

multimedia message from an originating terminal to a recipient terminal. The message body is a multipart structure that includes multimedia objects, each in separate parts, and the optional presentation part. The order of the parts has no significance. The presentation part contains instructions explaining how the multimedia content should be rendered to the display and speakers on the terminal. If the presentation part does not exist, the implementation on the terminal determines how the multimedia content is presented.

Synchronized Multimedia Integration Language (SMIL) [8] is used as a presentation profile in MMS. SMIL is a simple language based on eXtensible Markup Language (XML). It consists of a set of modules that define the semantics and syntax for certain areas of functionality. Examples of these modules are the layout module, timing and synchronization module, and animation module. An MMS presentation is shown in **Figure 2**.

MMS is the natural evolution of SMS. Although it delivers much richer content than SMS—color pictures, animations, audio samples, and video clips—it does not solve the user interface problem of SMS when it is deployed in person-to-system (P2S) or system-to-system (S2S) applications, because it has the same poor user interface as SMS. For example, to look up the current weather information for Beijing via SMS, a user submits a service request via SMS to a weather service provider. The service provider then parses the request and delivers the results of the search to the user. However, before making use of the service, the user has to know the special service command format, which is defined by the weather service provider: e.g., the service code (which might be *WT*), service parameters (for example, the city name *Beijing*), and the service provider code number (for example, *456*), as shown in **Figure 3**. Unfortunately, there is no syntax standard to define a service command. It is also difficult for end users to remember so many different service commands to access different services. The same problem exists for MMS applications, because SMIL does not provide a definition to formulate a user’s input. Although MMS could deliver a more attractive weather-searching result, a user has to use a text editor to input such a confusing service command.

We therefore introduce in this paper an interaction feature that can enhance the user’s experience of MMS services. In the above weather scenario, for example, a user simply opens an MMS message delivered as an advertisement by a weather service portal, inputs the location information, and clicks the submit button to send a weather query message to the portal (**Figure 4**). The query result is returned via another MMS message after the portal has processed the query. In this scenario, information that does not concern customers, such as the service and provider code numbers, can be embedded.

Users are provided an interactive interface, and MMS becomes a fascinating service delivery channel for service providers.

Interaction functionality

Generally speaking, two types of interactive functionalities should be added to the MMS standard: client side responsiveness and local resource interaction.

Client side responsiveness

A set of visual controls are described in MMS to provide an interactive user interface between a user and an MMS client, and to define a communication interface between an MMS client and a back-end server. For example, forms—which support multiple kinds of controls, such as text input, buttons, and selection lists—are important because they enable interactive Web applications. If, with iMMS technology, MMS could support the form function, it would be easy to implement many new kinds of interactive messaging applications that go far beyond those built with simple notification and button pushing.

For a control in an MMS message, three attributes should be defined: visual presentation, action, and relationship to other controls. The visual presentation defines how to show a control on the screen. Action defines the function of the control and the reaction when a user interacts with the control. For example, in the weather service user interface depicted in Figure 4, the position, size, type of input area, and submit button are defined. The action of the input area is to obtain a city name via a user's local input. The submit button is responsible for generating an MMS message in XML format that includes the city information and then sends it to the service provider.

Relationship is an optional attribute for a control. The relations among controls are the internal drivers that enable local interaction. Usually, a message contains information about one or more control objects, no matter how this information is presented. An object is described by some attributes. The internal relationships among these attributes determine the relationship among data, presentation, and action. For example, an attribute is determined by other attributes, or an attribute determines other attributes. There are also external relationships between these objects. The *inclusion relationship* means that an object is an attribute of another object. The *ally relationship* means that two or more objects have linkages between them. The object description and its relations are used to facilitate the user's interactive operation. If data expressing one attribute of an object instance is *select and display*, other attributes of the object instance should show at the same time. If the message is about more than one object, other corresponding instance data should show as well.



Figure 4

Interactive user interface for weather service.

Local resource interaction

Local resource interaction aids a user by means of some immediate implicit and/or explicit action. The most common implicit action is *association*, which constitutes linking to other messages or Web pages. The most common explicit association uses simple buttons or icons to represent actions such as submitting a request or forwarding a message. For instance, a *phone call association*—a special implicit action enabled on mobile telephones—enables a user to click on a phone number to make a call. A *software association* leads to a software download and installation process. A *hyperlink* association causes a wireless application protocol (WAP) browser to navigate a Web page. Through the local resource interaction, MMS applications could support much more complex user actions and manipulation not currently supported by MMS. iMMS could bring the following advantages to service providers and end users.

- *Extend MMS usage scope:* MMS could carry and display richer content than SMS. This could include color pictures, animations, audio samples, and video clips. However, because of the limited input capability of a mobile device, the current MMS standard defines only the MMS presentation to package and communicate text, audio, and picture information. As a result, MMS can provide only some basic user interfaces, such as inserting a picture and audio file into a message or inputting some text information. This usage model also leads to MMS having to focus on P2P or S2P applications, because these applications require only basic interactive operations.

iMMS can overcome this limitation. It could widen MMS use by enhancing MMS interaction functionality so that end users could easily input required information into a message and send it to a back-end system.

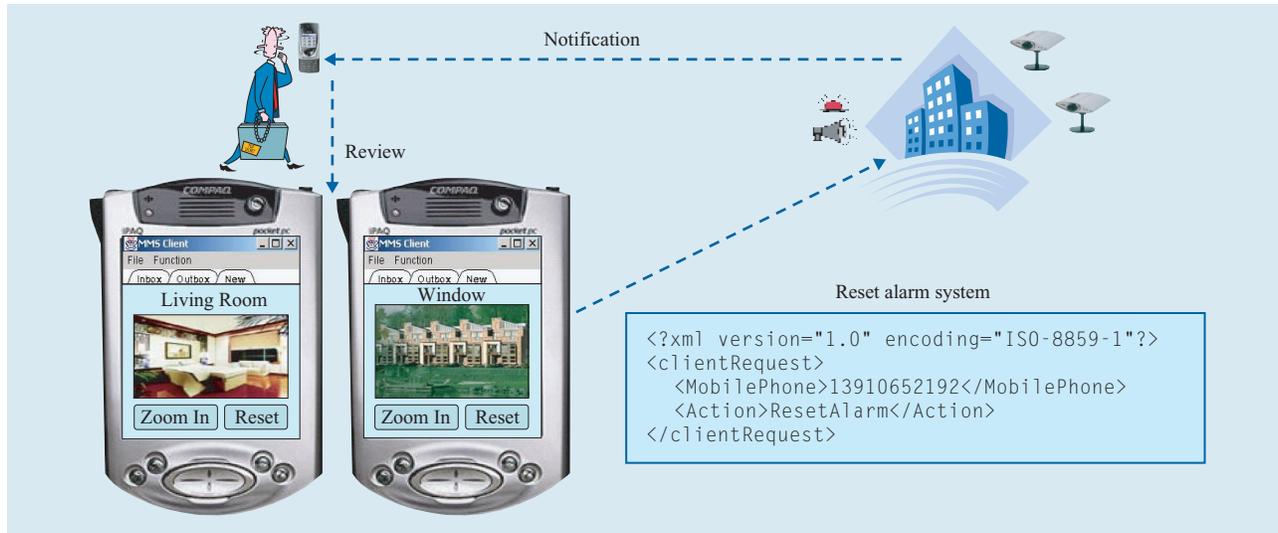


Figure 5

Home surveillance scenario developed as a prototype on an HP iPAQ** pocket PC with Java**.

- *Provide a rich user interface:* For end users, the major benefit of iMMS is to provide a more friendly user interface. The set of visual controls defined in the proposed design comprise an intuitive interface that can easily be used, as in our earlier weather service scenario example. Furthermore, it is difficult to use an MMS terminal to compose and edit an MMS message. Many MMS clients within existing terminals require 10–20 keystrokes to send a message. The process of sending an MMS message should be simplified and the number of required keystrokes minimized. The recommendation is for a maximum of three keystrokes. User reactions can be predefined in iMMS messages, which can then be generated automatically and sent without requiring manual text input.
- *Format user input and structure data exchange with back-end servers:* For end users, iMMS technology could provide friendlier interfaces to facilitate the use of MMS applications. For an MMS service provider, an important concept in iMMS is that it could format a user's input data to be expressed as XML instance data. Because the structure of the instance data is described by XML, a back-end server could easily process the structured interchange data when it is received by the server.
- *Optimize the communication traffic:* The interaction with local resources could optimize communication traffic, since it could reduce the number of round trips between MMS clients and servers. Generally, the performance of a wireless network is not as good as that of a wired network, so the local interaction feature would enable a

better user experience for mobile phone users, especially when wireless applications are used on mobile phones.

- *Support disconnected operation:* The local interactive feature supports disconnected operation. MMS is a mobile application. When a terminal disconnects with a remote server because the network signal is too weak, the user can continue to use MMS messages received on the terminal and submit a request to the server. The request is sent out automatically when the wireless network is ready. Unlike the Web application, MMS is a messaging system. Since the user is not aware of the message transmission before the whole message has been delivered and received, the communication delay is imperceptible. This reduces the requirement for network performance and makes MMS as convenient and friendly to use as SMS.

Interactive presentation

SMIL describes only how to show a media object on a screen with no interactive characteristics. Therefore, an interactive presentation should be added to the MMS standard. As discussed above, iMMS provides two classes of interaction—client-side responsiveness and local resource interaction. These two classes have similarities with Web application interfaces, since the Web is a typical interactive application. For example, a form tag in a Hypertext Markup Language (HTML) [9] page is used to support a browser to interact with a remote Web server. Currently, XHTML Mobile Profile [10–12] is defined by the Open Mobile Alliance (OMA) as a presentation for WAP applications. Therefore, one way to define iMMS

is to adopt XHTML Mobile Profile as one type of presentation language in MMS.

Three kinds of presentation language are supported in the iMMS description part: MMS SMIL for compatibility with the current MMS standard; XHTML Mobile Profile for interactive capability; and the combination of XHTML Mobile Profile and MMS SMIL as the presentation part of the MMS message. The World Wide Web Consortium (W3C) has an XHTML+SMIL proposal [13] that could be used as a basis to further the work of integrating SMIL functionality into XHTML. It defines a set of abstract XHTML modules that support a subset of the SMIL 2.0 specification. It includes functionality from SMIL 2.0 modules providing support for animation, content control, media objects, timing and synchronization, and transition effects. The proposal also integrates SMIL 2.0 features directly with XHTML, European Computer Manufacturers Association Script Language (ECMAScript) [14], and cascading style sheets (CSS) [15], describing how SMIL could be used to manipulate XHTML and CSS features.

We have developed a prototype on an HP iPAQ pocket personal computer with Java to demonstrate this kind of interactive technology. In a home surveillance scenario, illustrated in **Figure 5**, many different detectors, such as infrared sensors, gas detectors, and cameras, may be used to monitor the situation at home. When not at home, the householder wants to know what is really happening when those detectors encounter something unusual. To satisfy that need, when the home monitoring system is triggered, home cameras record pictures, package these pictures in an MMS message, and send it to the householder. The householder can then examine the home situation by viewing the pictures taken from different rooms. If there is no problem (for example, the alarm system was accidentally triggered by wind or by a pet), the user can reset the alarm system by clicking the *reset* button, which causes an MMS message to be sent that includes an XML reset request (see Figure 5). The home monitoring system receives and processes the reset request.

In this scenario, we can see that iMMS combines the advantages of a general message system and a browser system. Not only do mobile users receive information in nearly real time through messages being “pushed” to them, but they can also easily exchange information with a back-end server via a user-friendly interface such as a browser application.

An iMMS client is designed to implement this scenario. As shown in **Figure 6**, the communication module provides communication stacks, including MMS, SMS, and a WAP protocol stack over GPRS. The basic function of the viewer is as an interface for users to view messages. The viewer includes a *parser and renderer* component to parse, verify, and interpret the message body, which is composed

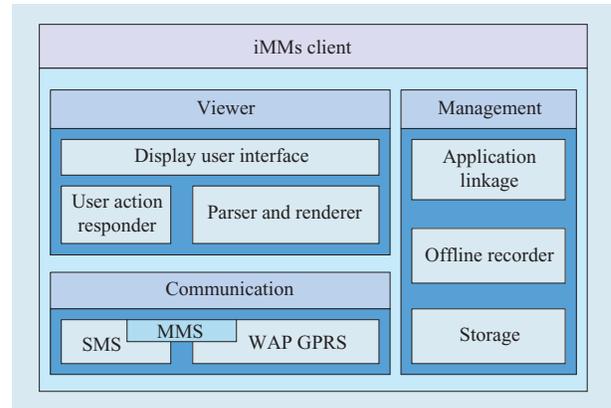


Figure 6

iMMS client stack.

with markup language. The qualified message is delivered to the display module and displayed for users. The predefined actions in the message are passed to the user action responder module to create an event “listener,” which responds immediately to a user’s action.

The management component is composed of application linkage maintenance, an offline recorder, and message storage. Some associate interaction may request opening another application to view or play a message attachment. The application linkage maintains the relationship between the media type and the application. To support user operations performed while the device is offline, the offline recorder records the user’s action (for instance, *submit data to server* or *forward message* to other users). When the device goes online, the recorded action is performed. The storage retains some messages for later reuse.

MMS application middleware

The MMS application server (MAS) provides middleware for quick MMS service development and deployment. As shown in **Figure 7**, the MAS is composed of two parts. The first part is an MMS service gateway, which is the kernel of the MMS application server. The gateway is made up of several components. The interface component provides a communication interface with an operator’s MMS center. Currently, the 3rd Generation Partnership Project (3GPP) standard MM7 [16, 17], which is the communication interface between a value-added service provider application and the MMS relay/server, is implemented by the component. The parser module is responsible for parsing the basic message information—such as service type, service name, and user identification number—and passing the information to the MMS service

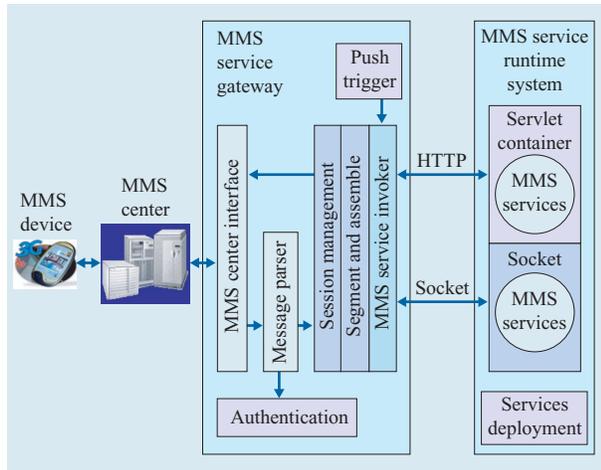


Figure 7

MMS application server architecture. (HTTP = HyperText Transfer Protocol.)

invoker module to call relevant MMS services according to the parameters of the message.

The three components described above build up the basic architecture of the MAS. Another three components are designed to enhance the functionality of the server. Because most message clients do not provide session management, this function must be offered by the middleware to support interactive applications, so a session module is included in the architectural design. Furthermore, a segment and assemble component deals with the problem of message size. An MMS package size is currently limited by an MMS center. The maximum size of a package is generally about 100 KB, but sometimes, to support local interactions, MMS applications have to download multiple files whose size is beyond the MMS limitation. Therefore, these lengthy messages must be divided into several smaller segments before sending and must be recombined upon receipt. A push trigger component is provided to implement the most important feature for a message system—the ability to push a message or notification to end users according to predefined trigger rulers.

The second part of the MAS is the service runtime system, which supports two programming models. One is a servlet programming model, which is supported by a servlet container implemented with a Web application server in the system. Another programming model is the socket model, which enables any program to communicate with the MMS service invoker component via the socket. One important feature of the socket model is that it makes it easy to integrate with enterprise legacy systems and applications. A deployment module is needed to

manage all of the MMS services, and an MMS template database is used to store the MMS templates used by the MMS services.

Conclusion and future work

MMS is the natural evolution of SMS. However, the success of SMS in business does not mean that MMS will be a success. One key way to promote MMS rapidly is to provide a large number of value-added MMS services for customers. In this paper, we propose expanding the current MMS standard to enhance its interaction and presentation capability by embedding XHTML technology in MMS presentation language. As part of the iMMS project, both the client and MMS application servers, based on the Windows 2000 server, have been developed and presented. They can enable service providers to develop, deploy, and show the benefits of various MMS services using iMMS. The prototypes not only show enriched user experience by enhancing MMS interactive capability, but also facilitate service delivery for service providers.

The functionality of the MMS middleware can be enhanced by the deployment of portal technology in the future, for example, by integrating the MMS middleware into a portal server to support interactive multimedia messaging services and adding disconnection management on a portal. Another important effort is to develop a visualized MMS authoring tool to help users create interactive MMS applications. A messaging-based programming model will also be explored to leverage the current Web-based programming model in the project.

**Trademarks or registered trademarks of Microsoft Corporation, Hewlett-Packard Development Company, L.P., or Sun Microsystems, Incorporated, in the United States, other countries, or both.

References

1. L. Novak and M. Svensson, "MMS—Building on the Success of SMS," *Ericsson Rev.*, No. 3, 102–109 (2001).
2. WAP MMS Architecture Overview, WAP Forum, WAP-205-MMSArchOverview; see <http://www.wmlclub.com/docs/especwap2.0/WAP-205-MMSArchOverview-20010425-a.pdf>.
3. Multimedia Messaging Services: the Developing Picture, ARC Group, December 2002; see http://www.gii.co.jp/english/ar11839_multimedia_messaging.html.
4. Mobile Messaging in Western Europe, 2003-2007, IDC, May 2003, R104-12870; see <http://www.mindbranch.com/listing/product/R104-12870.html>.
5. WAP MMS Encapsulation Protocol, WAP Forum, WAP-209-MMSEncapsulation; see <http://www.openmobilealliance.org> and/or the WAP Forum, <http://www.wapforum.org/>.
6. WAP MMS Client Transactions, WAP Forum, WAP-206-MMSCTR; see <http://www.wapforum.org/what/technical.htm>.
7. MMS Conformance Document 1.2, Open Mobile Alliance, OMA-MMS-CONF-v1_2-20030623-D.
8. Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation, August 2001; see <http://www.w3.org/AudioVideo/>.

9. HTML 4.01 Specification, W3C Recommendation, December 24, 1999; see <http://www.w3.org/TR/html401/>.
10. XHTML Basic, W3C Recommendation, December 19, 2000; see <http://www.w3.org/TR/xhtml-basic/>.
11. XHTML 1.1 Module-based XHTML, W3C Recommendation, May 31, 2001; see <http://www.w3.org/TR/xhtml11/>.
12. XHTML Mobile Profile 1.1, Open Mobile Alliance, OMA-WAP-XHTMLMP-V1_1-20020904-D, January 14, 2003; see <http://www.w3.org/TR/SVGMobile/>.
13. XHTML+SMIL Profile, W3C Note, January 31, 2002; see <http://www.w3.org/TR/XHTMLplusSMIL/>.
14. ECMAScript Language Specification, Standard ECMA-262, 3rd Edition, December 1999; see <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>.
15. Wireless CSS Specification Version 1.1, Open Mobile Alliance, OMA-WAP-WCSS-V1_1-20030506-D.
16. Multimedia Messaging Service: Service Aspects; Stage 1, 3GPP 3G TS 22.140, Release 1999; see <http://www.3gpp.org/>.
17. Multimedia Messaging Service: Functional Description; Stage 2, 3GPP 3G TS 23.140, Release 1999; see <http://www.3gpp.org/specs/specs.htm>.

Received October 17, 2003; accepted for publication January 23, 2004; Internet publication August 30, 2004

Jun Shen *IBM Research Division, IBM China Research Laboratory, 2/F, Haohai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (shenjun@cn.ibm.com).* Dr. Shen has been working as a Research Staff Member in the IBM China Research Laboratory since 1999, currently working in the Exploratory Solutions Group. He received a B.S. degree in electrical engineering from Tsinghua University in 1992, an M.S. degree in computer science from Beijing University of Aeronautics and Astronautics in 1995, and a Ph.D. degree in computer science from Tsinghua University in 1999. Dr. Shen's research work focuses on wireless and mobile technology.

Pei Sun *IBM Research Division, IBM China Research Laboratory, 2/F, Haohai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (sunpei@cn.ibm.com).* Ms. Sun is a Research Staff Member in the Exploratory Solutions Group. She joined the IBM China Research Laboratory in 2000 after receiving an M.S. degree in computer architecture from Xi'an Jiaotong University. Ms. Sun's research interests include the areas of pervasive computing, mobile computing, and computer networking.

Jianming Zhang *IBM Research Division, IBM China Research Laboratory, 2/F, Haohai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (zhangjm@cn.ibm.com).* Mr. Zhang is a Research Staff Member in the Exploratory Solutions Group. He received both B.S. and M.S. degrees in computer science from Tsinghua University in 1994 and 1997, respectively. Prior to joining the IBM China Research Laboratory in 1998, he worked as a software engineer at Putian and at a start-up company. Mr. Zhang currently works on industry solutions.

Song Song *IBM Research Division, IBM China Research Laboratory, 2/F, Haohai Building, No. 7, 5th Street, Shangdi, Haidian District, Beijing 100085, People's Republic of China (soong@cn.ibm.com).* Mr. Song received a B.S. degree in electrical engineering and an M.S. degree in biomedical engineering from Tsinghua University in 1986 and 1988, respectively. From 1988 to 1995, he worked on the research and development of the diagnostic ultrasound imaging and picture archiving and communications system (PACS) as an associate professor and professor at the Chinese Academy of Medical Sciences and Peking Union Medical College. Mr. Song joined the IBM China Research Laboratory in 1995 as a Senior Research Member and manager of the Pervasive Computing team. He is currently working on wireless and life science solutions in the Exploratory Solutions Group.