# An OSGi-Based Infrastructure for Context-Aware Multimedia Services

*Zhiwen Yu, Xingshe Zhou, and Zhiyong Yu Northwestern Polytechnical University, P. R. China*
*Daqing Zhang, Institute for Infocomm Research, Singapore*
*Chung-Yau Chin, National University of Singapore, Singapore*

## ABSTRACT

This article proposes an OSGi-based infrastructure for context-aware multimedia services in a smart home environment. A context-aware multimedia middleware (CMM), which supports multimedia content filtering, recommendation, and adaptation according to changing context is presented. It also performs context aggregation, reasoning, and learning. To foster device and service interoperability, CMM is integrated with an OSGi service platform. We envisage the OSGi-based infrastructure to fill the niche of three gateways in a smart home: network connecting, context provisioning, and multimedia personalizing.

## INTRODUCTION

The rapid development of broadband Internet access and wireless communications has enabled residents at home to easily and conveniently access to a broad spectrum of multimedia content and services. Such digital multimedia space is often resided by a networked of electronic appliances, such as TV, hi-fi, PC, PDA, personal video recorder (PVR), and mobile phone. For better audio and visual experience, the provisioning of multimedia content and operations of multimedia devices need to be customized based on changing contexts (i.e. context-aware multimedia services).

Context-aware multimedia services are cognizant of user contexts and able to adapt to them seamlessly. In a smart home environment, a context-aware multimedia service can, for example, record favorite TV programs of family members, show suitable content based on a user's social activities (e.g., holding a birthday party), and present content in an appropriate form according to the capabilities of the display device and network connection.

Context-aware multimedia services have attracted much attention from researchers in recent years, and several context-aware multimedia systems have been developed. However, building context-aware multimedia systems is still complex and time-consuming due to inadequate infrastructure support.

To address this challenge, we propose an Open Service Gateway Initiative (OSGi)-based context-aware multimedia infrastructure for building and rapidly prototyping such services in a smart-home environment. The OSGi [1] is an emergent open architecture that lets us deploy a large array of wide area network services to local networks such as smart homes, work spaces, and automobiles. It defines a standardized and component-oriented computing environment for networked services. OSGi offers the following benefits: platform independence, application independence, multiple service support, service collaboration support, security, multiple home networking support, and simplicity.

Leveraging the OSGi architecture, we develop a context-aware multimedia middleware (CMM). CMM contributes as a platform for multimedia content filtering, recommendation, and adaptation based on changing context. To ensure total exploitation of context information, CMM is enhanced with the capability to aggregate, reason, and learn about context information generated by ubiquitously available context sensors embedded in the smart home. The OSGi-based infrastructure can suitably fill the niche of three gateways in a smart home:

- Network connecting gateway. It connects all kinds of information appliances. It also bridges the external networks with the internal networks and devices.
- Context provisioning gateway. It accomplishes context aggregation, reasoning, and learning.
- Multimedia personalizing gateway. It performs multimedia content filtering, recommendation, and adaptation.

## RELATED WORK

There have been several systems presented to provide generic architectural supports for context-aware multimedia applications. Although Tseng *et al.* [2] propose a multimedia middleware for video transcoding and summarization, they acquire context in an ad hoc manner. Eikerling and Berger [3] propose a service infrastructure to support user mobility for delivery of multimedia content, which is implemented based on OSGi. QoSDREAM [4] is a middleware framework providing context support for multimedia applications. These two systems are similar to our infrastructure; however, they merely handle location context. Our CMM middleware integrates both multimedia middleware and context middleware. The multimedia middleware performs multimedia content filtering, recommendation, and adaptation. The context middleware, on the other hand, is responsible for various processes in context aggregation, reasoning, and learning. The infrastructure therefore outperforms the previous solutions by enabling integrated processing of multimedia content with vast variety of context information in a systematic manner.

To build an infrastructure supporting device interoperability and service management, a standard should be followed in the construction. Several standards are laid to solve interoperability issues among consumer electronics appliances, such as Multimedia Home Platform (MHP), OpenCable, Jini, Universal Plug and Play (UPnP), and Home Audio Video Interoperability (HAVi). MHP and OpenCable attempt to adapt existing Internet and Web standards for digital television (DTV) so that interactive services, especially broadcast streams, can be offered on set-top-boxes (STBs) and PVRs. They can provide interoperability at the application level, regardless of the applications' broadcaster or the manner in which it is broadcast. However, they lack efficient support for data sharing between devices in a smart environment. Besides platform approaches, Jini and UPnP are device identification and service discovery mechanisms that enable devices interoperability and communication. HAVi, which relays on IEEE 1394 buses, purely focuses on connection-specific features but lacks interconnection support.

While each of these standards functions on its own protocol, there is a need to interoperate them in a digital multimedia home. OSGi is proposed to fulfill this task. It focuses on service delivery, home networking, and bridging external networks with internal devices. It can leverage the above mentioned industry standards in its implementations. OSGi defines a lightweight framework for delivering and executing service-oriented applications. In addition, the OSGi framework delineates application programming interface (API) standards for the execution environment of services. These APIs address service cradle-to-grave life cycle management, interservice dependencies, data management, device management, resource management, and security. Using these mechanisms, end users can load services on demand, while the service gateway manages the installation and configuration of these services.

## Representation Model

Multimedia and context representation is an important part of context-aware multimedia systems. Since multimedia metadata and context information are often parsed and processed by automated systems interoperating with third-party services and applications, they need to be represented with standard-oriented, flexible, and interoperable models.

### Multimedia Representation

MPEG-7, formally named "Multimedia Content Description Interface," is the de facto multimedia description standard that has been widely accepted in industry and academiaa, and popularly utilized in many applications. MPEG-7 Multimedia Description Schemes (MDS) specify a high-level framework that allows generic description of all kinds of multimedia, including audio, video, image, and textual data.

We choose MPEG-7 *Creation DS* and *Classification DS* to describe information about the multimedia content, such as the title, keyword, director, actor, genre, and language. This information is very useful to match user preferences and special needs. The *Variation DS* is used to specify variations of media content as well as their relationships. The Variation DS plays an important role in our context-aware media recommendation by allowing selection among different variations of the media content in order to select the most appropriate one for adapting to the specific capabilities of terminal devices and network conditions.

### Context Representation

We propose an ontology-based context model for context representation. In the modeling approach, Ontology Web Language (OWL) [5] is adopted as representation language to enable expressive context description and data interoperability of context. In the domain of knowledge representation, the term *ontology* refers to the formal and explicit description of domain concepts, which are often conceived as a set of entities, relations, instances, functions, and axioms [6]. Using ontologies to model contexts offers several advantages:
- By allowing users and environments to share a common understanding of context structure, OWL ontologies enable applications to interpret contexts based on their semantics.
- Ontologies' hierarchical structure lets developers reuse domain ontologies (e.g., of users, devices, and activities) in describing contexts and build a practical context model without starting from scratch.
- Because contexts described in ontologies have explicit semantic representations, semantic Web tools such as federated query, reasoning, and knowledge bases can be leveraged to support context interpretation. Incorporating these tools into context-aware multimedia services facilitates context management and interpretation.

*We propose an ontology-based context model for context representation. In the domain of knowledge representation, the term ontology refers to the formal and explicit description of domain concepts, which are often conceived as a set of entities, relations, instances, functions, and axioms.*

## CMM Architecture

We designed CMM, which is based on our representation model, to enable rapid prototyping of context-aware multimedia applications (Fig. 1). It consists of the following components:
- *Context aggregator*: aggregates diversity of

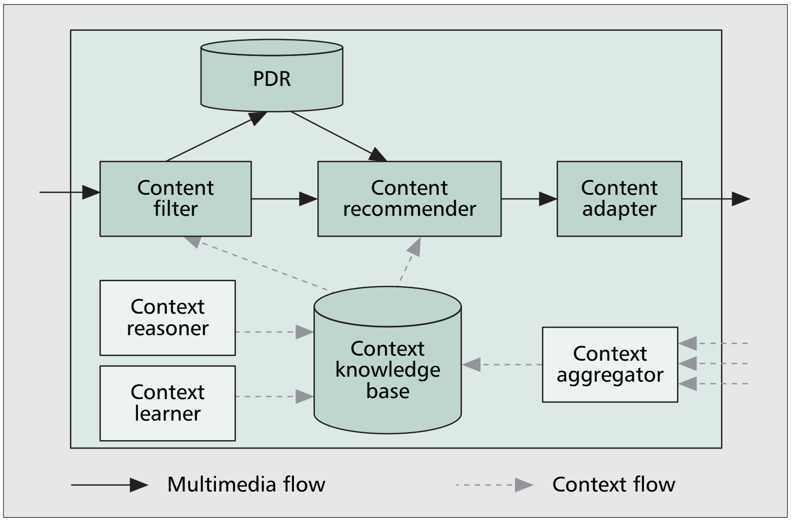



**Figure 1.** *CMM architecture.*

context information from various sources such as hardware sensors and software programs, converts them to OWL representations, and asserts them into the context knowledge base
- *Context reasoner*: infers abstract higher-level contexts (e.g., a user's activity) from basic sensed contexts, resolves context conflicts, and maintains knowledge based consistency
- *Context learner*: deduces user preference of multimedia through implicit machine learning techniques
- *Content filter*: filters the incoming live multimedia content and only records those that the user favors to local storage (e.g., personal digital recorder)
- *Content recommender*: estimates the score for media content and determines its appropriate form for a particular context
- *Content adapter*: performs multimedia content adaptation enabling it to be presented in a particular terminal and transferred under a specific network condition

The components are described in greater detail below.

### Context Aggregator

The context aggregator aggregates a diversity of context information from an array of diverse information sources. Context aggregation helps to merge the required information related to a particular entity (e.g., user) or relevant to a particular context-aware system (e.g., all contexts needed by smart TV service). It then asserts them into the context knowledge base for further reasoning and learning.

We deployed various hardware sensors in our prototype system, including location sensors, lighting sensors, microphones, and video cameras. We also developed some software programs to capture context, such as GUIs for explicitly inputting user preferences and daily schedules, observers for capturing user feedback to specific content, and monitors for detecting terminal capabilities and network characteristics.

### Context Reasoner

The context reasoner infers abstract high-level contexts from basic sensed contexts, resolves context conflicts, and maintains knowledge base consistency. To support various kinds of reasoning tasks, we can specify different inference rules, and preload them into the appropriate logic reasoner. We adopted a rule-based approach based on first-order logic for reasoning about contexts. It provides forward chaining, backward chaining, and a hybrid execution model. The forward-chaining rule engine is based on the standard Rete algorithm. The backward-chaining rule engine uses a logic-programming engine similar to Prolog engines. A hybrid execution mode performs reasoning by combining both forward and backward chaining.

Our current system applies Jena2 generic rule engine [7] to support forward-chaining reasoning over the OWL represented context. To perform context inference, an application developer needs to provide horn-logic rules for a particular application based on its needs. The context reasoner is responsible for interpreting rules, connecting to context KB, and evaluating rules against stored context. We have specified a rule set based on the forward-chaining rule engine to infer high-level contexts (e.g., a user's social activity in the smart home).

### Context Learner

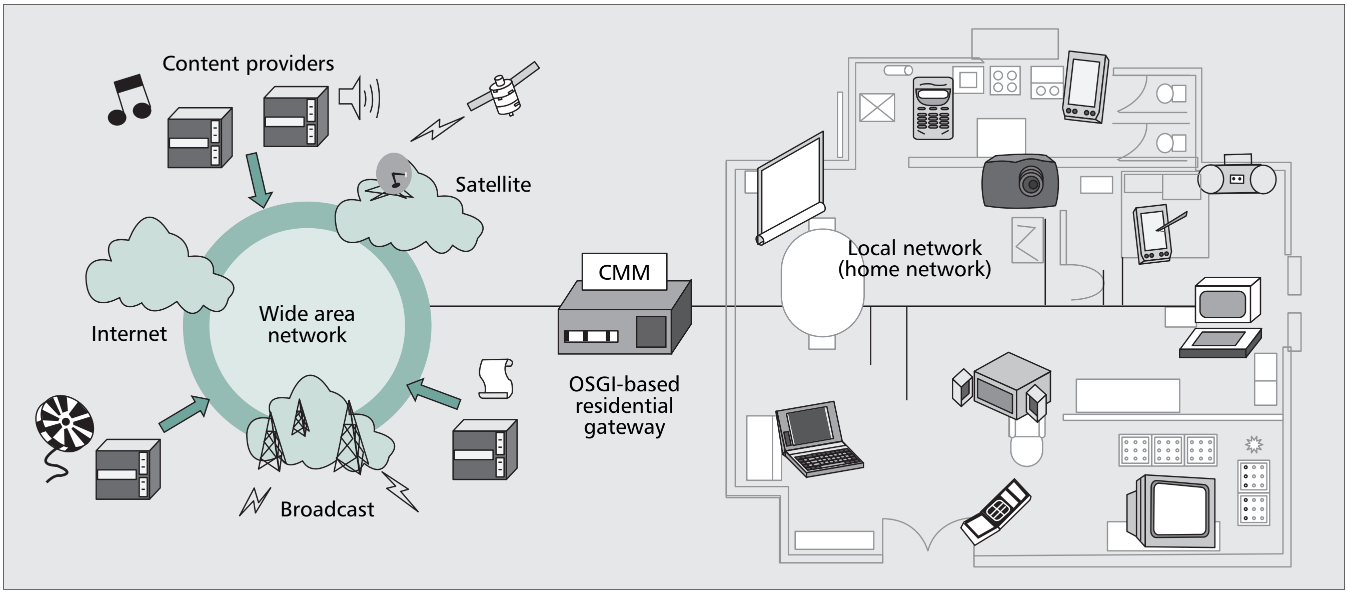
Preference context plays an important role in multimedia personalization services. The context learner deduces and updates user preference by compiling statistical analysis on user viewing history aggregated by the context aggregator from all kinds of media playing devices (e.g. PC, television, PDA). The details of centralized preference learning in a ubicomp environment based on a master-slave architecture and implicit learning algorithm that applies relevance feedback and a naïve Bayes classifier approach were described in [8].

The centralized learning approach has several advantages. First, it learns user preference by utilizing overall feedback information as opposed to other traditional methods that just use partial feedback information in one device. Second, the approach can relieve pervasive devices with limited resources from computation- and storage-consuming learning tasks. These devices are merely responsible for observing user behavior and uploading feedback information to the aggregator.

Intuitively, a multimedia content is often viewed by a group of users, e.g., a family, friends in a party, etc. Therefore, sometimes the common interest of the group users is needed for the purpose of recommendation. The context learner can also deduce the group preference by merging individual user preferences into a common one [9].

### Content Filter

The content filter evaluates between incoming multimedia content and preference context. It compares features in a media item with terms that characterize a user's watching preferences to determine whether the user likes it. Only




■ **Figure 2.** *Overall system architecture.*

media items that have a high degree of similarity to the user's preference would be recorded to the local storage or directly forwarded to the recommender. The multimedia content filtering strategy by using Vector Space Model (VSM) was presented in [10].

The content filter also records content according to a user's situation context. For example, knowing the user is currently participating in a legal course, the filter will record law-related documentaries (e.g., legal cases).

### CONTENT RECOMMENDER

The content recommender provides the right content, in the right form, to the right person based on all categories of context. The recommendation output consists of two aspects for a media item: score and form. The score implies the degree of interest that a user pose about the media item, while the form means the presentation features (e.g., modality, format, and frame size) on a particular device.

For the purpose of efficient context processing in content recommendation, we classify context into three categories: *preference context* (user's taste or interests for media content), *situation context* (user's spatio-temporal and social situation, e.g., location and time), and *capability context* (physical running infrastructure, e.g., terminal capability and network condition). The content recommender first calculates the *similarity* between the media item and the preference context by adopting Vector Space Model. Then, it evaluates the *probability* of the media item belonging to the situation context. The score is obtained as the weighted sum of the above calculated similarity and probability. The appropriate form is determined by applying rule-based approach to infer presentation details from the capability context.

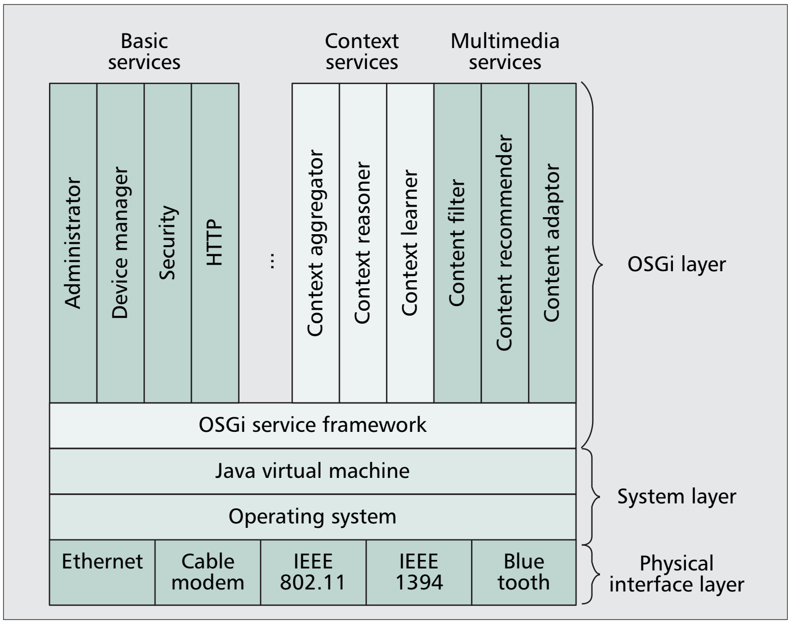
### CONTENT ADAPTER

The content adapter performs multimedia content adaptation by using two techniques: summarization and transcoding. Multimedia summarization means summarizing an audio-video item into a short one that can be viewed within a time constraint. Multimedia transcoding involves transforming the content from one media type to another so that the content can be suitably processed by a particular device or efficiently delivered by a specific network condition. For instance, most handheld computers are not capable of handling video data due to their hardware and software constraints. Therefore, video information can be accessed alternatively through sets of images captured from the video through the transformation/transcoding process.

Multimedia adaptation can be statically done at authoring time prior to delivery or dynamically done on the fly if needed. In our system both strategies are accommodated. If the content comes directly from the filter, online adaptation is done. Otherwise, when the content is recorded to the PDR, offline adaptation is performed to prepare content variations for later consumption. To perform online transcoding of images, we utilize the open source compression/decompression libraries from the Independent JPEG Group (http://www.ijg.org/). The video transcoding is implemented based on the public domain software for H.263. The Power Video Converter (http://www.apussoft.com/) is used for offline adaptation that covers rich functionalities (e.g., converting between different video formats/frame sizes/frame rates, splitting video, transforming video to audio, and extracting images from video).

## OSGI-BASED INFRASTRUCTURE

We integrated CMM with the OSGi service platform to build a reliable and secure system that can deliver and manage context-aware multimedia services in a smart home environment. The overall system architecture is shown in Fig. 2. The OSGi-compliant residential gateway is deployed for wide-area services provisioning and management to the local networked devices, and




■ **Figure 3.** *Software architecture for an OSGi-compliant residential gateway.*

providing a ground for services interoperability among the heterogeneous in-home devices and standards. The CMM architecture leverages the OSGi service platform, allowing multimedia content providers who are distributed in different networks to deliver reliable and secure context-aware multimedia services into the smart home environment. Various multimedia clients in the home can be communicable and monitored by the residential gateway via different home networking technologies, and adapt their operations as the context changes.

The OSGi service platform refers to the software stack embedded in the OSGi-compliant residential gateway. As shown in Fig. 3, the service gateway software architecture consists of three layers: the OSGi layer, the system layer, and the physical interface layer.

The OSGi layer is composed of two key components: the service framework and service bundles. The service framework provides a service hosting environment as well as a set of common APIs to develop application bundles. It also includes several basic service bundles such as configuration management, user management, device management, permission administration (security), and HTTP service. The context service and multimedia service components are built on top of the OSGi framework. Each component can be constructed as an independent bundle. In the OSGi environment, bundles are the entities for deploying Java-based applications. A bundle is a Java Archive (JAR) file that comprises Java classes and other resources. The system layer refers to the Java Virtual Machine and operating system (OS) in the residential gateway. Specifically, the OS performs the functionality of IP forwarding, firewall, and Network Address Translation (NAT). The physical interface layer deals with the low-level communications, comprising various WAN and LAN connectivity interfaces to and from the residential gateway.

## Developing and Deploying Context-Aware Multimedia Service

This section illustrates how the CMM and OSGi-based infrastructure described above can be used to author a novel prototype context-aware multimedia service. First, we present a general-purpose process for context-aware multimedia service development using CMM. Then we describe a real context-aware multimedia service, called *ContAwareMovie*, and deploy it on the OSGi-based infrastructure.

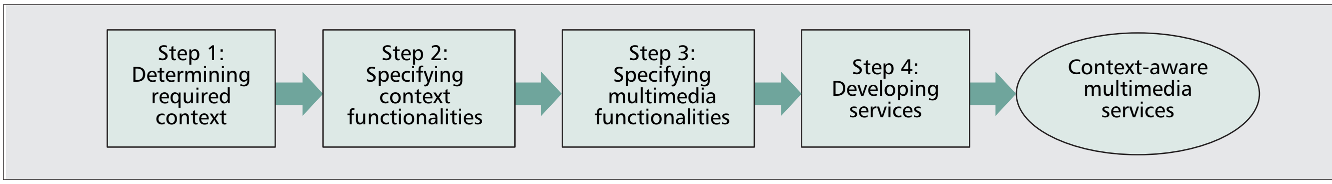
### Service Development Process Using CMM

With CMM, the process of developing context-aware multimedia services is as simple as four general steps (Fig. 4):

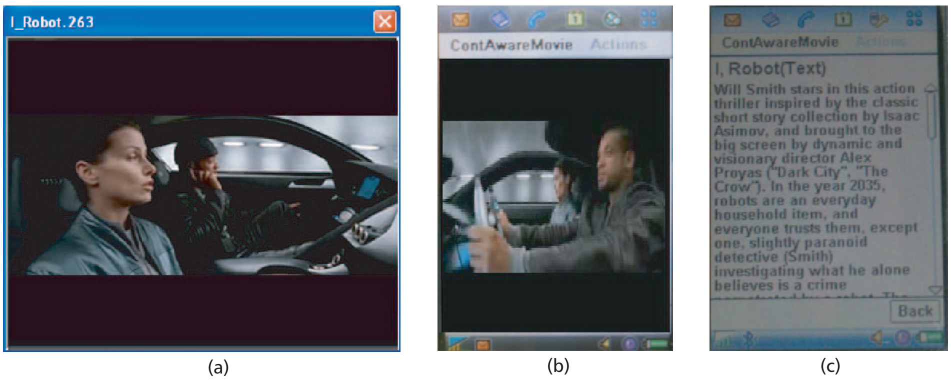
- **Step 1**: The development process begins by determining the collection of context that is required by the context-aware multimedia service to fulfill its functionality.
- **Step 2**: The context-related functionalities in CMM are specified. If the required context is a kind of low-level context (e.g., location and time), the context aggregator is directly assigned to collect it. If the required context on the other hand is the abstract high-level context (e.g., social activity), the service-specific inference rules are specified and preloaded into the context reasoner.
- **Step 3**: The multimedia-related functionalities in CMM are specified according to the context that needs to be processed. For instance, if the capability context is considered, the content adapter is definitely required. Within content recommendation, if only the preference context is taken into consideration, then the weighting factors of preference context and situation context should be assigned 1 and 0, respectively.
- **Step 4**: The service logic and interaction UI are developed. Context-aware behavior is implemented, with access to context through submitting queries to the corresponding listener APIs.

#### Deploying ContAwareMovie

Following the above mentioned developing steps, we have built a smart movie prototype, namely *ContAwareMovie*. The prototype environment consists of an OSGi-compliant residential gateway, a media server, a PC, a mobile phone (Sony Ericsson P900), and various sensors. The residential gateway is built on an Intel Celeron 600 MHz processor and 256 Mbytes of memory. It runs the embedded Linux (kernel 2.4.17) operating system and the ProSyst mBedded server (http://www.prosyst.com/) as the OSGi service platform. The ContAwareMovie is packaged as an OSGi bundle and published as an OSGi service. We use Prosyst's framework package (www.prosyst.com/osgi.html) for editing and managing bundles, managing users and security, and so on. The media server continuously broadcasts media content to the gateway. Then the ContAwareMovie delivers multimedia content to




■ **Figure 4.** *Context-aware multimedia service development process using CMM.*




■ **Figure 5.** *ContAwareMovie screenshots: a) playing high-quality video on a PC terminal; b) playing low-quality video on a mobile phone; c) displaying text on a mobile phone.*

users based on various contexts. The client-side application was running on the PC and the mobile phone.

In a particular scenario, we mainly considered the watching time (situation context) and terminal/network condition (capability context). Radio frequency identification (RFID) technology was adopted to identify the user. The terminal capabilities were obtained through explicit input when the device was registered to the system. For example, based on the watching time, the movie with the highest score is *I, Robot*. Under different terminal/network conditions, the displaying forms may vary accordingly. First, we compare the content presented in different devices with sufficient network bandwidth for H.263 video streaming. The video playing on the PC terminal features 576 × 240 frame size and 30 frames/s frame rate (Fig. 5a), while a small size (OCIF, 176*144) and low frame rate (10 frames/s) video on the mobile phone (Fig. 5b). Second, we compare the content presented in the mobile phone in different conditions of transmission bandwidth. In this case, with high network bandwidth over Ethernet (e.g., 112.6 kb/s), the content displaying is video (Fig. 5b), while in the condition of very low bandwidth over Bluetooth (e.g., 3.7 kb/s), the displaying modality is text that summarizes the plot of *I, Robot* (Fig. 5c).

## Conclusion

In this article we propose an OSGi-based infrastructure for context-aware multimedia services in a smart home environment. We first present a context-aware multimedia middleware (CMM) and then integrate it with an OSGi service platform. Developing CMM on top of the service-oriented OSGi open standard can provide a robust and interoperable infrastructure for building, provisioning, and managing context-aware multimedia services in smart homes and beyond. We also present a general-purpose process for context-aware multimedia service development using CMM, and built a real service on the OSGi-based infrastructure.

We plan to implement a few improvements to our system. First, GUI-based toolkits will be built for service visual design, implementation, and testing. Then mechanisms for interspace context discovery [11] will be added to the infrastructure.

## Acknowledgment

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by the National Science Foundation of China (no. 60573161) and the Doctorate Foundation of Northwestern Polytechnical University of China (no. CX200417).

## References

[1] D. Marples and P. Kriens, "The Open Services Gateway Initiative: An Introductory Overview," *IEEE Commun. Mag.*, Dec. 2001, pp. 110–14.
[2] B. L. Tseng, C. Y. Lin, and J. R. Smith, "Video Summarization and Personalization for Pervasive Mobile Devices," *SPIE Electronic Imaging 2002 — Storage and Retrieval for Media Databases*, San Jose, CA, Jan. 2002, pp. 359–70.
[3] H. Eikerling and F. Berger, "Design of OSGi Compatible Middleware Components for Mobile Multimedia Applications," *Proc. Joint Int'l. Wksps. Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems: Protocols and Systems for Interactive Distrib. Multimedia*, *LNCS*, vol. 2515, Jan. 2002, pp. 80–91.
[4] H. Naguib, G. Coulouris, and S. Mitchell, "Middleware Support for Context-Aware Multimedia Applications," *3rd IFIP Int'l. Conf. Distrib. Apps. and Interoperable Sys.*, Krakow, Poland, Sept. 17–19, 2001, pp. 9–22.
[5] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," W3C Rec., 2004

[6] T. Gruber, "A Translation Approach to Portable Ontology Specification," *Knowledge Acquisition*, vol. 5, no. 2, 1993, pp. 199–220.
[7] J. J. Carroll *et al.*, "Jena: Implementing the Semantic Web Recommendations," *Proc. 13th Int'l. Conf. World Wide Web*, New York, 2004, pp. 74–83.
[8] Z. Yu *et al.*, "User Preference Learning for Multimedia Personalization in Pervasive Computing Environment," *Proc. 9th Int'l. Conf. Knowledge-Based Intelligent Info. & Eng. Sys.*, *LNCS*, vol. 3682, 2005, pp. 236–42.
[9] Z. Yu *et al.*, "TV Program Recommendation for Multiple Viewers Based on User Profile Merging," *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, Mar. 2006, pp. 63–82.
[10] Z. Yu and X. Zhou. "TV3P: An Adaptive Assistant for Personalized TV," *IEEE Trans. Consumer Electronics*, vol. 50, no. 1, Feb. 2004, pp. 393–99.
[11] D. Zhang, C. Y. Chin, and M. Gurusamy. "Supporting Context-Aware Mobile Service Adaptation with Scalable Context Discovery Platform," *Proc. IEEE VTC 2005-Spring*, Sweden, May 30–June 1, 2005, pp. 2859–63.

## Biographies

ZHIWEN YU [M] (zhiwenyu@nwpu.edu.cn) received his Ph.D. in computer science from Northwestern Polytechnical University, P. R. China, in 2005. His research interests include context-aware computing, intelligent information technology, and personalization. From September 2004 to May 2005 he was a visiting researcher at the Institute for Infocomm Research, Singapore. He joined the Nagoya University, Japan as a post-doctoral researcher in February 2006.

DAQING ZHANG (daqing@i2r.a-star.edu.sg) is manager of the Context-Aware Systems Department at the Institute for Infocomm Research, where he leads research in connected-home and context-aware systems. His research interests include pervasive computing, service-oriented computing, and context-aware systems. He received his Ph.D. from the University of Rome La Sapienza and the University of L'Aquila, Italy.

XINGSHE ZHOU [M] (zhouxs@nwpu.edu.cn) is a professor and dean of the School of Computer Science, Northwestern Polytechnical University, P. R. China. His research interests include distributed computing, embedded computing, and sensor networks. He received his M.S. degree in computer science from Northwestern Polytechnical University.

CHUNG-YAU CHIN (chungyau.chin@nus.edu.sg) is a graduate student at the National University of Singapore. His research interests include pervasive computing, context-aware systems, and service discovery. He received his B.E. in computer engineering from the National University of Singapore.

ZHIYONG YU (yzyhb@126.com) is a Ph.D. candidate at the School of Computer Science, Northwestern Polytechnical University, P. R. China. His research interests include embedded computing and pervasive computing. He received his B.E. from China Agricultural University in 2004.