Context-Aware Middleware for Multimedia Services in Heterogeneous Networks

Liang Zhou, *ENSTA-ParisTech*

Naixue Xiong, *Georgia State University*

Lei Shu, *Osaka University*

Athanasios Vasilakos, *University of Western Macedonia*

Sang-Soo Yeo, *Mokwon University*

This context-aware middleware system facilitates diverse multimedia services in heterogeneous-network environments by combining an adaptive service-provisioning middleware framework with a context-aware multimedia middleware framework.

Efforts to converge heterogeneous access networks (WiMax, Wi-Fi, sensor networks, and so on) have increased in recent years. In fact, such networks are fully integrated into the 4G network design paradigm.[1] Supporting multimedia applications over heterogeneous networks has been one of the main research fields in the networking and multimedia communities. In addition, multimedia devices' content and operations typically must be customized to accommodate changing contexts (that is, *context-aware multimedia services*)—for example, to record favorite TV programs for different family members, show suitable content on the basis of a user's social activities, or present content in a format that the display device and network connection can handle.

Context-aware multimedia services have attracted considerable attention from researchers in recent years, and several such systems have been developed under specific network conditions. However, building context-aware multimedia services in heterogeneous networks is still complex and time-consuming due to heterogeneity in context-aware media content and network conditions. Although Belle Tseng and her colleagues have proposed multimedia middleware for video transcoding and summarization,[2] they acquire context in an ad hoc, rather than heterogeneous, network. Liang Zhou and his colleagues have developed a multimedia rate allocation framework in heterogeneous networks, but

1541-1672/10/$26.00 © 2010 IEEE
Published by the IEEE Computer Society

without taking into account the context-aware multimedia middleware.[3] Zhiwen Yu and his colleagues consider context-aware multimedia services in a wireless network framework and propose corresponding middleware.[4]

The current literature, to the best of our knowledge, considers context-aware middleware and heterogeneous networks only separately. To provide satisfying and unified multimedia services in the context of heterogeneous networks, we consider these two topics jointly. We propose efficient context-aware middleware for multimedia services in heterogeneous networks. Our system includes both an adaptive service-provisioning middleware framework and a context-aware multimedia middleware framework. The adaptive service-provisioning middleware framework makes it possible to provide service to mobile users and professionals anywhere, anytime, and in any context by interoperating with existing heterogeneous networks. The context-aware multimedia middleware framework supports multimedia content filtering, recommendation, adaptation, context aggregation, reasoning, and learning.

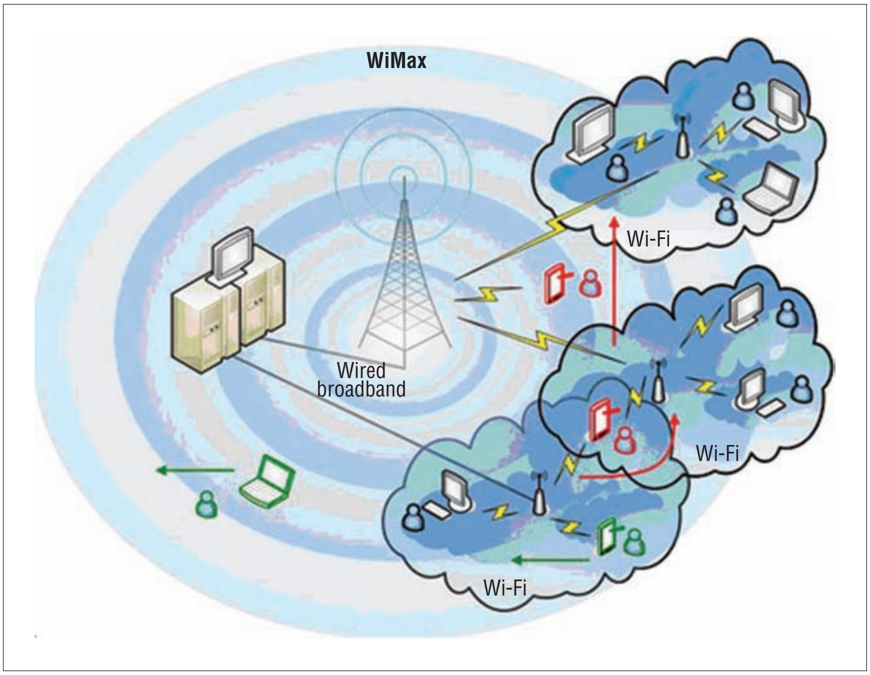

Figure 1. Example of a multimedia streaming architecture in a heterogeneous-wireless-network environment. The IP Multimedia Subsystem platform defines an overlay architecture that provides multimedia services via different wireless networks.

Design Principles and Requirements

Figure 1 shows an example of a multimedia application supported in a heterogeneous-wireless-network environment. In this example, the IP Multimedia Subsystem platform defines an overlay architecture for providing multimedia services via heterogeneous wireless networks (see Figure 1).[1]

Context awareness of heterogeneous applications has recently attracted the attention of many researchers.[5,6] Context awareness is an essential feature of heterogeneous systems because almost all ubiquitous applications use context information in their operations. Generally speaking, there are two technological difficulties and challenges for context-aware multimedia services in heterogeneous networks:

- designing adaptive service-provisioning middleware to handle the heterogeneity of diverse networks and to provide service to mobile users and professionals anywhere and anytime; and
- developing a context-aware multimedia middleware framework on the basis of the proposed adaptive service-provisioning framework to support diverse multimedia services, including multimedia-content filtering, recommendation, adaptation, aggregation, learning, reasoning, and delivery.

We adopt the definition of *context* as "any information that can be used to characterize the situation of an entity."[4] An *entity* is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves. Examples of context information include a user's location or profile, the time, local resources of the mobile device, and available services.

Context awareness characterizes a system that uses context information when it performs its tasks. In this article, we use the task-oriented definition of context awareness: "a context-aware system uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."[4] That is, we place our main emphasis on the context that is relevant to the task.

Context awareness involves performing data acquisition from sensors, context recognition, and other tasks that must be completed before the context can actually be used. Delegating data acquisition and context-processing tasks to applications makes reuse of such tasks almost impossible. One solution to this problem is to decouple the tasks from the applications and move the desired functionality to lower layers. Context-aware middleware, rather than applications, should perform the following context-oriented functions:

- support a variety of sensor devices,
- support the distributed nature of context information (because the data is from different sources),
- provide transparent interpretation of applications and abstraction of context data,
- maintain context storage, and
- control the context data flow.

In addition, multimedia applications set requirements for communication and computation. Multimedia algorithms need a lot of bandwidth and processing power, so the middleware must also have the following capabilities. First, it must be able to effectively use all the available bandwidth. The middleware must be able to use all the available connections and switch to the one that best fulfills the requirements at any given moment. Second, it must control the place of computation. To cope with limited resources, the middleware must decide, on the basis of the situation at hand, where computations should be performed. Third, the middleware must support various multimedia devices (video cameras, microphones, and so on). Finally, the middleware must fulfill certain requirements to make the system adaptable. These include

- triggering adaptation on a system-wide level,
- supporting system-wide adaptation policies, and
- providing a common interface between devices and the middleware.

Developing a generic adaptation mechanism suitable for context-aware multimedia services in heterogeneous networks is a demanding task. The fundamental requirements for such a mechanism include the following:

- The logic for implementing the different phases of the entire adaptation process must be clearly demarcated.
- The adaptation mechanism should support elements of arbitrary complexity, such as profiles and algorithms.
- The mechanism must be independent of any particular type of profile or algorithm. The algorithms used for profile matching should be loaded at runtime rather than a static part of the adaptation mechanism. Thus, any logic pertaining to specific types of profiles or algorithms shouldn't be hard coded in the adaptation system.
- Interoperability and portability must be supported. Ideally, the adaptation system will handle disparate data and algorithms, and enable their seamless integration in a variety of environments. System modularity and decoupling from context are vital to accomplishing this goal.

Middleware Infrastructure for Heterogeneous Networks

A unified platform for heterogeneous networks should target all types of access networks, from picocells to wireless local area networks (WLANs), to current 2G and 3G networks. Various middleware solutions are available for these networks. We believe that the increasing diversity of devices (terminals, network elements, and application servers) is moving toward a single dominant middleware platform that will be sufficient for all devices and purposes. As a starting point toward this goal, we assume that various platforms—for example, JAIN (Java APIs for Integrated Networks), OSA (open service access), Parlay, Camel (Customized Applications for Mobile-Network-Enhanced Logic), Parlay, and APIs (application programming interfaces)—will be available over these networks.

Figure 2 shows our proposed middleware platform architecture for heterogeneous networks. To achieve the most open interoperability possible, we adopted the Object Management Group's model-driven architecture.[7] MDA goes far beyond Corba (common object request broker architecture) in terms of interoperability at the level of standard component interfaces by placing formal-system models at the core of the problem. The distinct features of this approach are that the system definition is independent of any implementation model, and formal mappings to many possible implementation technologies (Java, XML, and so on) are provided.

In this approach, services are described using formal models, initially expressed in a platform-independent modeling language such as Unified Modeling Language (UML). Through MDA tools, these services can be instantly mapped onto specific platform technologies, such as Camel, OSA, or Parlay. When heterogeneous networks are used, such services would provide capabilities such as file transfer, voice calls, multiparty video conferences, and terminal positioning.

The adaptability of services is made possible by *adaptive service components*. ASCs are polymorphic self-adaptive components that are specialized for a particular functionality or feature and that can adapt to external triggers. For example, whenever network layer reservations are violated, the relevant ASCs are triggered to

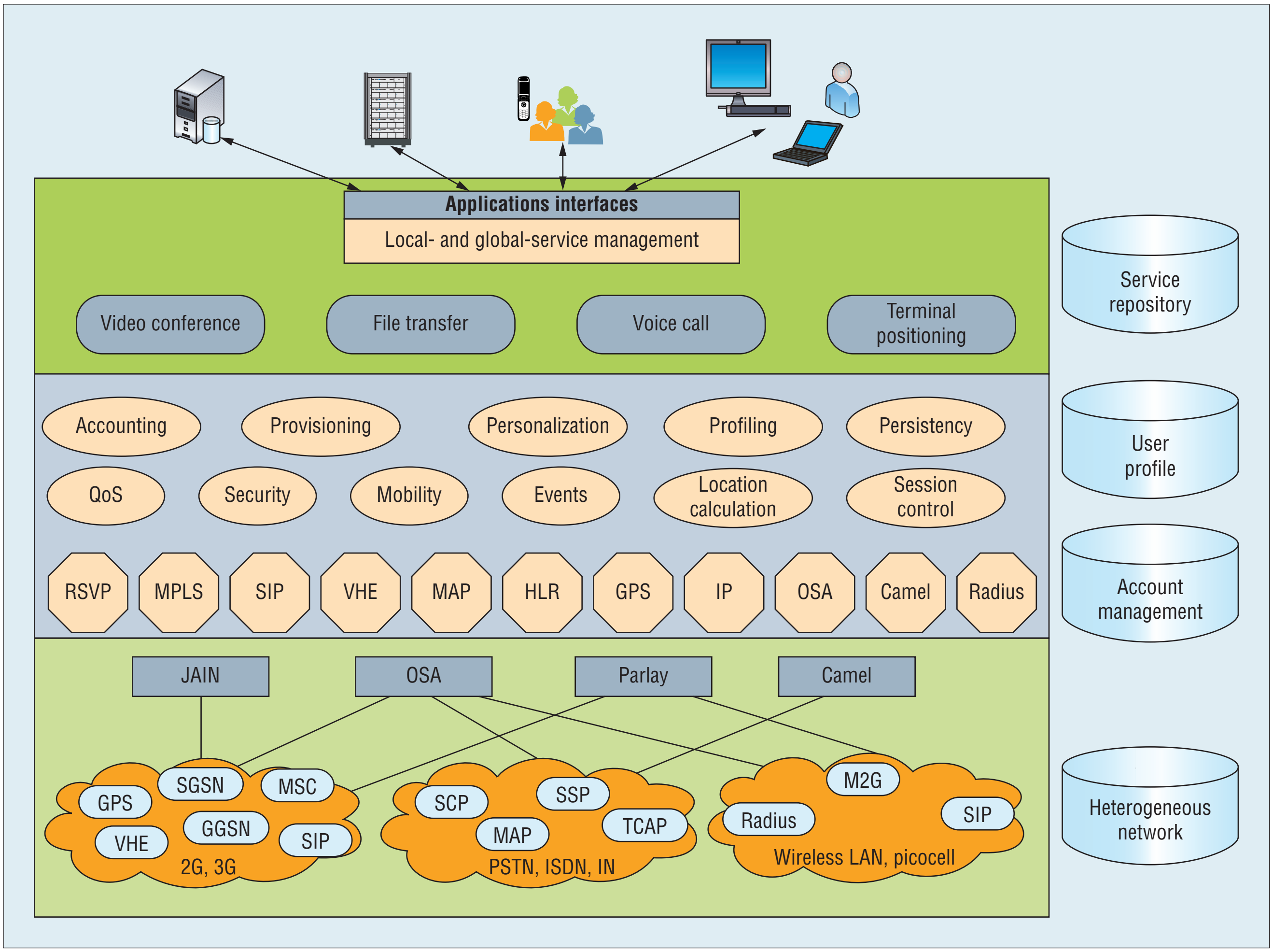

Figure 2. Middleware infrastructure and corresponding function for heterogeneous networks. (Camel: Customized Applications for Mobile-Network-Enhanced Logic; GGSN: gateway GPRS [General Packet Radio Service] support node; HLR: home location register; IN: internetworking; IP: Internet Protocol; JAIN: Java APIs for Integrated Networks; LAN: local area network; MAP: multiple application protocols; MPLS: multiprotocol label switching; MSC: mobile-switching center; M2G: mobile to gateway; OSA: open service access; PAN: personal area network; PSTN: public switched telephone network; QoS: quality of service; RSVP: Resource Reservation Protocol; SCP: structure-conduct-performance; SGSN: serving GPRS support node; SIP: Session Initiation Protocol; SSP: service-switching point; TCAP: transaction capability application part; VHE: virtual home environment.)

adapt themselves to the available network resources on the basis of rules, scenarios, and service-level agreements (SLAs). To achieve such polymorphisms, ASCs follow metamodeling techniques.[4] The metamodeling strategy is ultimately achieved via shared metadata, and understanding metadata involves the automated development, publishing, management, and interpretation of models. Metamodeling technology provides dynamic system behavior on the basis of the runtime interpretation of such models.

ASCs are highly interoperable, easily extended at runtime, and completely dynamic in terms of their overall behavioral specifications (that is, their range of behavior is not bound by hard-coded logic). As Figure 2 shows, ASCs for our adaptive service-provisioning middleware framework include location calculation, session control, mobility, quality of service, security, profiling, personalization, and provisioning.

Platform-independent ASCs are subsequently translated to network path-selecting components (NPSCs) by mapping the ASC models to some implementation of a language or platform (such as Java) using formal rules. Development and integration can be facilitated through common platform services and programming models. For example, J2EE (Java 2 Platform, Enterprise Edition) enables implementation and deployment of component-based distributed applications, and the Java community is developing pure Java programming models in the form of J2EE standard APIs.

Examples of NPSCs include components to interface OSA, Parlay,

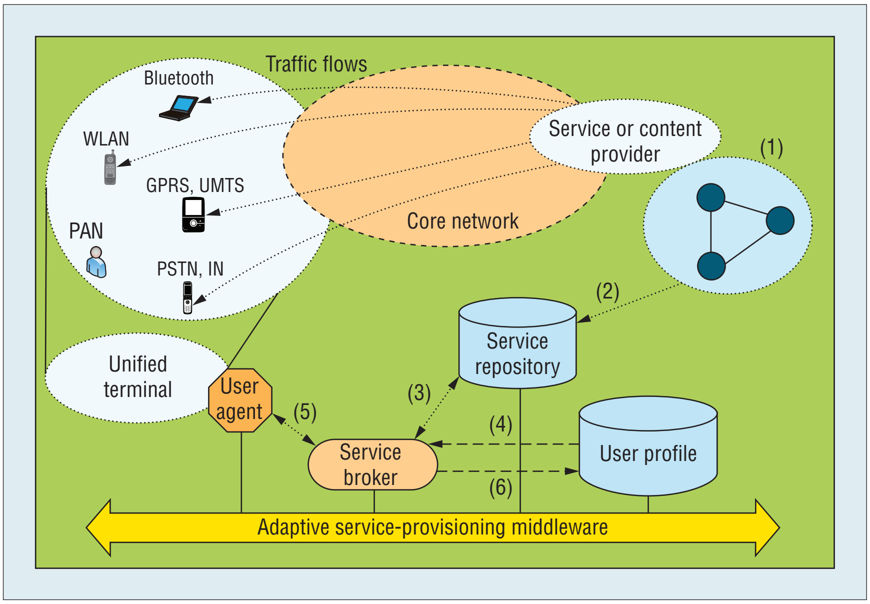

Figure 3. Service-provisioning interactions. Different components interact with one another to realize function-aware middleware in heterogeneous networks. (GPRS: General Packet Radio Service; IN: internetworking; PAN: personal area network; PSTN: public switched telephone network; UMTS: Universal Mobile Telecommunications System; WLAN: wireless local area network.)

VHE (virtual home environment), GMLC (gateway mobile location center), HLR (home location register), GPS, SIP (Session Initiation Protocol), MPLS (multiprotocol label switching), DiffServ (differentiated services), and RSVP (Resource Reservation Protocol). In many cases, the NPSCs just wrap the functionality when the platform offers an open API (as in OSA and Parlay). In other cases (for instance, SIP and GPS), full-fledged NPSCs must be implemented.

Clearly, many composite services can be built from a given set of atomic services and components, but only a subset of these are useful to the individual user. The utility of a given composite service highly depends not only on the user's personal preferences and desired tasks to perform but also on the context in which that user accesses the service. The proposed adaptive scheme provides a method of dynamic incorporation of context information—specifically, information about the current network conditions of the selected service and the composition process. This will contribute significantly to the realization of services that can automatically adapt and reconfigure themselves to handle changes in network conditions—for example, those resulting from a user switching to a different device or roaming among different access networks. Moreover, new services will be able to advertise themselves, and software components can be automatically installed at the user terminal according to user preferences.

The steps of the service-provision interactions (see Figure 3) are as follows:

1. The service is deployed, discovered, and composed.
2. Service descriptions are published at the service repository. These descriptions identify which services (both atomic and composite) are available to complete specific tasks, given specific terminal and network conditions.
3. Given descriptions of user and terminal preferences (which are stored at the user or terminal profile), the tasks to be performed, and the network context in which the user will access the service, the service broker can select a preexisting (atomic or composite) service.
4. If no suitable services exist, the service broker can automatically compose a new service that meets the user's demands.
5. The user agent represents the user and negotiates (via the provisioning ASCs) with the service broker. The user agent sends the service broker the tasks to be performed in terms of agreed ontological elements and indicates the user's preferences for service delivery.
6. The profiling ASCs build up the user profile over time as the user agent reports the user's service usage patterns and changing preferences via the service broker.

The event ASCs report information regarding current network conditions to the service broker. This information serves as a key constraint in the selection of potential service compositions for performing a required task. Furthermore, once a service session is ongoing, the proposed adaptive middleware notifies the service broker if changes in network conditions result in a service composition that no longer satisfies the required QoS levels. The broker then searches for alternative service compositions that can satisfy the required task in a different manner or, if that isn't possible, in a way that closely approximates what the user requires.

Context-Aware Multimedia Service

Because multimedia metadata and context information are often parsed and processed by automated systems interoperating with third-party services and applications, they must be represented with standard-oriented, flexible, and interoperable models. We propose an ontology-based context model for context representation, and we adopt the Web Ontology Language (OWL) to enable expressive context descriptions and data interoperability of context.[8] 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. Here, we discuss the design of our context-aware multimedia service framework (see Figure 4) and its components.

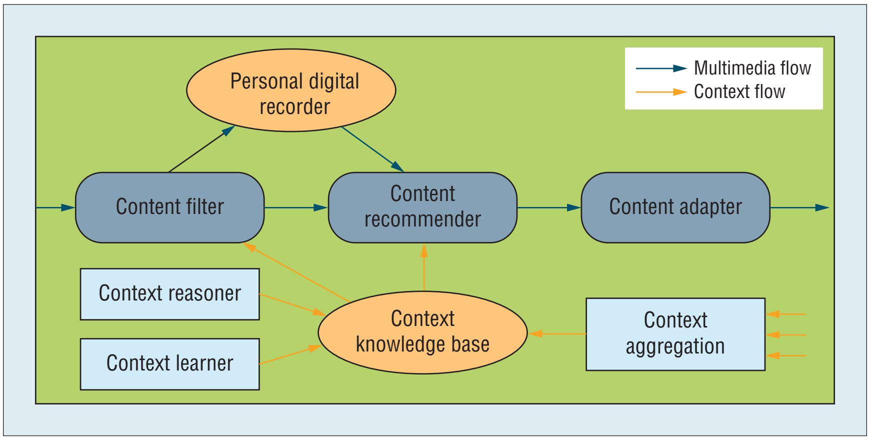

Figure 4. Context-aware multimedia service framework for heterogeneous networks.

Context Aggregation

The context aggregation component synthesizes context information from an array of diverse information sources. This component helps merge required information that is related to an entity (the user, for instance) or relevant to a context-aware system (for example, all contexts needed by a smart TV service). It then inserts this information 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 to explicitly input user preferences and daily schedules, observers to capture user feedback regarding specific content, and monitors to detect 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 adopt a rule-based approach based on first-order logic for reasoning about contexts. The context reasoner provides forward chaining, backward chaining, and a hybrid execution model. The forward-chaining rule engine is based on the standard-rate algorithm. The backward-chaining rule engine uses a logic-programming engine similar to Prolog engines. Hybrid execution mode performs reasoning by combining both forward and backward chaining.

Our current system applies the Jena2 generic rule engine to support forward-chaining reasoning over the OWL-represented context.[9] To perform context inference, an application developer must provide horn-logic rules for a particular application on the basis of its requirements. The context reasoner is responsible for interpreting rules, connecting to a context KB, and evaluating rules against stored context. We've specified a rule set based on the forward-chaining rule engine to infer high-level contexts (for instance, a user's social activity in a smart home).

Context Learner

Preference context plays an important role in multimedia personalization services. The context learner applies a centralized learning approach, in which it deduces and updates user preference by compiling statistical analysis on the user's viewing history via context aggregation from all kinds of media-playing devices (PCs, televisions, personal digital assistants, and so on). (Yu and his colleagues also describe a centralized preference-learning approach in a pervasive environment. This approach comprises a master-slave architecture and an implicit learning algorithm that applies relevance-feedback and naive-Bayes-classifier techniques.[7])

This centralized learning approach has several advantages. First, the context learner discovers user preferences by using overall feedback information as opposed to other traditional methods, which use only 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 context aggregation component.

Multimedia content is often viewed by a group of users—for example, family and friends at a party. Therefore, the common interests of the group should also be considered. The context learner can deduce such group interests by merging individual user preferences into a common set of group preferences.[10]

Content Filter

This component distinguishes between incoming multimedia content and preference context. It compares features in a media item with terms that characterize a user's preferences to deduce whether the user would like this content. Only media items that have a high degree of similarity to user preferences are recorded in local storage or directly forwarded to the recommender. (Yu and his colleagues have presented a filtering strategy for multimedia content that uses a vector space model.[4])

The content filter also records content according to the context of the user's situation. For example, knowing that the user is currently participating in a legal academic course, the filter would record law-related documentaries (such as those describing particular legal cases).

Content Recommender

The content recommender provides the right content in the right format to the right person on the basis of 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 has in a media item; the form refers to the presentation features (modality, format, frame size, and so on) of a particular device.

For efficient context processing in content recommendations, we classify context into three categories: *preference context* (the user's tastes and interests regarding media content), *situation context* (the user's spatial-temporal and social situation—for example, location), and *capability context* (the physical running infrastructure, including terminal capabilities and network conditions). The content recommender first calculates the similarity between the media item and the preference context by adopting a vector space model. Then, it evaluates the probability of the media item belonging to the situation context. The score is the weighted sum of the calculated similarity and probability. The appropriate form is determined by applying a 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* involves summarizing an audio-video item into a shorter one that can be viewed under some time constraint. *Multimedia transcoding* involves transforming content from one media type to another so that a particular device can suitably process it or a specific network condition can efficiently deliver it. For example, because of their hardware and software constraints, most handheld computers can't handle video data. Therefore, video information can alternatively be accessed through sets of images captured from video through the transformation and transcoding process.

Multimedia adaptation can be statically handled at authoring time before delivery, or it can be dynamically handled on the fly. Our system accommodates both strategies. If the content comes directly from the filter, online adaptation is performed. To prepare content variations for later consumption, on the other hand, offline adaptation is performed when the content is recorded to the personal digital recorder (PDR). For online image transcoding, we use the open-source compression and decompression libraries from the Independent JPEG Group. We implement video transcoding using public-domain software for the H.263 video compression standard. For offline adaptation, we use a power video converter that provides rich functionality.

In light of the ongoing heterogeneity of access networks, and the anticipated provision of seamless context-aware multimedia services, there is a need for integrating service delivery platforms and corresponding middleware platforms. Our context-aware middleware system facilitates diverse multimedia services in heterogeneous-network environments by combining an adaptive service-provisioning middleware framework with a context-aware multimedia middleware framework. In the future, we plan to study some practical issues for implementing our proposed system. For example, we'd like to reduce the dependence of multimedia content on specific multimedia services by having our middleware system automatically adapt this content.

Acknowledgments

Lei Shu's work related to this article was supported in part by the Lion project of Science Foundation Ireland under grant SFI/08/CE/I1380 (Lion-2) and in part by the Grant-in-Aid for Scientific Research (S)(21220002) of the Ministry of Education, Culture, Sports, Science, and Technology in Japan.

References

1. S. Karlich et al., "A Self-Adaptive Service Provisioning Framework for 3G+/4G Mobile Applications," *IEEE Wireless Comm. Magazine*, vol. 11, no. 5, 2004, pp. 48–56.

2. B.L. Tseng, C.-Y. Lin, and J.R. Smith, “Video Summarization and Personalization for Pervasive Mobile Devices,” *Proc. SPIE Electronic Imaging: Storage & Retrieval for Media Databases*, vol. 4676, SPIE, 2002, pp. 359–370.
3. L. Zhou et al., “Cross-Layer Rate Allocation for Multimedia Applications in Pervasive Computing Environment,” *Proc. Global Telecommunications Conf.* (GLOBECOM 08), IEEE Press, 2008, pp. 2303–2307.
4. Z. Yu et al., “An OSGi-Based Infrastructure for Context-Aware Multimedia Services,” *IEEE Comm. Magazine*, vol. 44, no. 10, 2006, pp. 136–142.
5. Y. Isoda, S. Kurakake, and K. Imai, “Context-Aware Computing System for Heterogeneous Applications,” *Proc. 1st Int’l Workshop Personalized Context Modeling and Management for UbiComp Applications* (ubiPCMM 05), IEEE Press, 2005, pp. 17–25.
6. F. Lopes et al., “Context-Based Heterogeneous Middleware Integration,” *Proc. Workshop Middleware for Ubiquitous and Pervasive Systems*, ACM Press, 2009, pp. 13–18.
7. Z. Yu et al., “Content Provisioning for Ubiquitous Learning,” *IEEE Pervasive Computing*, vol. 7, no. 4, 2008, pp. 62–70.
8. D.L. McGuinness and F. van Harmelen, eds., *OWL Web Ontology Language Overview*, World Wide Web Consortium (W3C) recommendation, Feb. 2004; www.w3.org/TR/owl-features.
9. T. Lukasiewicz and U. Straccia, “Managing Uncertainty and Vagueness in Description Logics for the Semantic Web,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 4, 2008, pp. 291–308.
10. G. Tao, H. Pung, and D. Zhang, “Information Retrieval in Schema-Based P2P Systems Using One-Dimensional Semantic Space,” *Computer Networks: The Int’l J. Computer and Telecommunications Networking*, vol. 51, no. 16, 2007, pp. 4543–4560.

THE AUTHORS

Liang Zhou is a postdoctoral researcher at ENSTA-ParisTech. His research focuses on multimedia communications and networks. He has a PhD in electrical engineering, jointly from École Normale Supérieure, Cachan, in France and Shanghai Jiao Tong University, Shanghai, in China. He is a member of IEEE and the IEEE Communications Society. Contact him at liang.zhou@ieee.org.

Naixue Xiong is a research scientist in the Department of Computer Science at Georgia State University. His research interests include communication protocols, network architecture and design, and optimization theory. He has a PhD in computer science (software engineering based on networks) from Wuhan University and a PhD in computer science (dependable networks) from the Japan Advanced Institute of Science and Technology. He is a member of IEEE. Contact him at nxiong@cs.gsu.edu.

Lei Shu is a specially assigned Research Fellow in the Department of Multimedia Engineering of the Graduate School of Information Science and Technology at Osaka University in Japan. He performed the work described in this article while working as a research scientist in the Digital Enterprise Research Institute at the National University of Ireland, Galway. His research interests include wireless sensor networks, context-aware middleware, and sensor-network middleware and security. He has a PhD in digital enterprises from the National University of Ireland, Galway. He is a member of IEEE and the ACM. Contact him at lei.shu@ieee.org.

Athanasios Vasilakos is a professor in the Department of Computer and Telecommunications Engineering at the University of Western Macedonia in Greece and a visiting professor in the graduate program of the Department of Electrical and Computer Engineering at the National Technical University of Athens. His research interests include wireless networks, cooperative communications technology, context-aware middleware, cross-layer design, and communications security. He is a member of IEEE. Contact him at vasilako@ath.forthnet.gr.

Sang-Soo Yeo is a professor in the Division of Computer Engineering at Mokwon University in Korea. His research interests include communication protocols, and network architecture and design. He has a PhD in computer science from Chung-Ang University in Seoul. He is a member of IEEE. Contact him at ssyeo@msn.com.