A Framework for Wireless Sensor Network Based Mobile Mashup Applications

Shuangquan Wang[1,2], Canfeng Chen[2], Jian Ma[2]
[1]*Beijing University of Posts and Telecommunications, Beijing, 100876, China*
[2]*Nokia Research Center, Beijing, 100176, China*
{ext-shuangquan.wang, canfeng-david.chen, jian.j.ma}@nokia.com

Abstract

We describe a framework for wireless sensor network (WSN) based mobile mashup applications. The framework consists of four functional components: WSN, mobile gateway, context management platform, and mashup server. The WSN deployed in the application domain senses the physical phenomena and the human-environment interaction and expresses them in sensor readings. The mobile phone is used as the gateway to the WSN, which can directly communicate with the sensor nodes and collect the user related sensor data. On the mobile phone, the raw sensor data is preprocessed and the user-centric contexts are extracted. The context management platform stores all contexts according to the predefined context ontology, detects the inconsistent information, deduces the high level knowledge using heuristic rules and provides web service application programming interfaces (APIs) to mashup server. The mashup server integrates the context information obtained from the context management platform with the information from other mashup data sources, and provides services to users through mobile phone. The proposed framework can be used in exhibition center, shopping mall, smart office/home, and many other scenarios.

1. Introduction

Advances in sensor technology, wireless communications, embedded computing, and micro electrical mechanical systems (MEMS) have enabled the development of wireless sensor network (WSN). As the basic component unit, each sensor node integrates sensing, data processing, and communication capabilities together and acts as an autonomous entity. It can measure many physical parameters using different kinds of sensors, including sound, infrared, light, temperature, accelerometer, magnetometer, etc. The sensor nodes in WSN can communicate with each other via radio links, and collaboratively achieve information gathering, processing and dissemination tasks [1].

For most existed WSN applications, the system framework has three main components: WSN, gateway and backend server. WSN senses the physical phenomena and the human-environment interaction and expresses them in sensor readings. The gateway forwards sensor data from WSN to backend server, which is responsible for data processing and service providing.

The above system framework is often application-oriented and has been widely used in environmental surveillance [2], manufacture [3], transportation [4] and military engineering [5]. However, for the WSN based mobile mashup applications where the mobile phone is used as the gateway to the WSN and the service platform and interface to the users, this framework is hardly applicable because: 1) in traditional application-oriented system, the gateway forwards the sensor data to the backend server directly and does not do any processing. In the WSN based mobile mashup applications, each mobile user may produce a large amount of sensor data. Transmitting all the raw data to the backend server is a big problem. Therefore, the mobile phone must do some processing and decrease the transmission amount; 2) in the traditional application-oriented system, the extracted information may only be used by one application. One system can manage its information in a specific format. In the WSN based mobile mashup applications, the information may be shared by many mushup applications. Thus, all the information should be managed in a general format; 3) in the traditional application-oriented system, the services are often provided only based on the information extracted from the sensor data. In the WSN based mobile mashup

978-0-7695-3507-4/08 $25.00 © 2008 IEEE
DOI 10.1109/CSIE.2009.805

applications, the integrated information may come from not only the WSN, but also other mashup data sources.

This paper introduces a framework for WSN based mobile mashup applications. In the application domain, sensor nodes are deployed beforehand. The mobile phone collects the sensor data from sensor nodes, extracts the user-centric contexts and reports the contexts to the backend server for context management. The mashup server integrates the users' contexts with other information, and provides mashup services to the mobile users.

2. The framework for WSN based mobile mashup applications

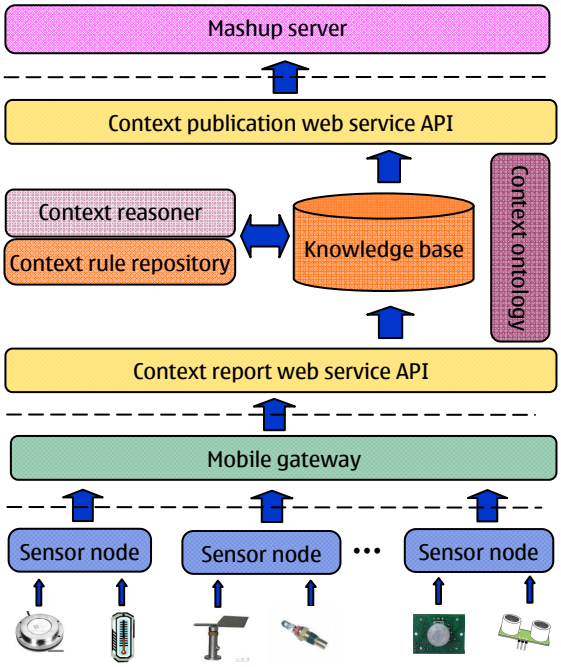

Figure 1. The framework for WSN based mobile mashup applications

Figure 1 shows the framework of WSN based mobile mashup applications. It includes four main functional components: WSN, mobile gateway (i.e., mobile phone), context management platform and mashup server. We explain the function of each component in the following subsections.

2.1 WSN and data collection

Suppose there are many sensor nodes deployed in the application domain. Each sensor node can act as a data sampling node or a beacon node. The data sampling node senses the physical phenomena and the human-environment interaction using different sensors attached. The beacon node is used to locate the users or other sensor nodes.

One sensor node can transmit its data to the mobile gateway in two ways: broadcasting or query-and-answer. In the broadcasting method, each sensor node packages the data and broadcasts it according to the predefined frequency and transmission power. The mobile gateway only needs to receive the messages. The broadcasting method can be easily implemented at the cost of energy waste of the sensor nodes. As to the query-and-answer method, the mobile gateway sends a query message to the sensor nodes nearby at first, and then these nodes will reply to the mobile gateway. This method can reserve the energy consumption of sensor nodes but, on the other hand, its implementation is a little more complex than the broadcasting method.

Besides the direct communication between one sensor node and mobile gateway, several sensor nodes can form a network and the mobile gateway acts as the mobile sink of the network.

2.2 Mobile gateway

We use the mobile phone as the mobile gateway in the proposed framework. Each mobile phone is embedded a universal sensor data (uSD) entry card, which contains a ZigBee transceiver and can communicate with the sensor node directly [6]. uSD card is a cross-platform device and compatible with standard SD memory card, as shown in Figure 2.

With uSD card, the mobile phone can receive the messages from the sensor nodes nearby. The sensor data can be obtained through parsing the messages according to the predefined message structure. After data preprocessing and fusing, the user-centric contexts will be extracted, which can be divided into two categories: local contexts and global contexts. Local contexts refer to the ones which are only used locally by the user on the mobile phone. Global contexts refer to the ones which need to be reported to the backend server for further processing or aggregation. For example, the user's location information in the exhibition center scenario. With the location information of all visitors, a graphic chart of the visitor distribution can be drawn and shared by all users. In order to facilitate the context management and user-oriented service providing, during context reporting, the time stamp, user ID and other accessorial information should be reported together with the context.

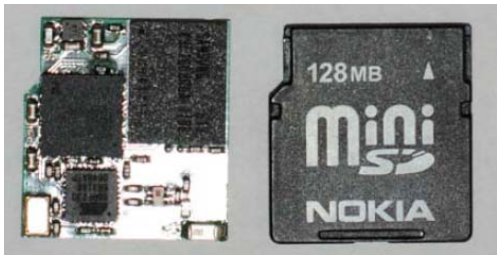

Figure 2. The universal sensor data (uSD) entry card (left) and the standard SD memory card (right)

2.3 Context management platform

Context management platform is located on the backend server, and responsible for managing the contexts reported from the mobile phones and providing application programming interfaces (APIs) for context report and publication. From Figure 1 we can see that the context management platform consists of six modules: context ontology, knowledge base, context rule repository, context reasoner, context report web service API and context publication web service API.

Context ontology defines the concepts and relations for context information. The context ontology consists of a schema which represents the structure and the properties for all ontology's concepts and a vocabulary that enables context information representing and sharing [7]. Knowledge base stores all the ontology knowledge, the meta-data for storage structure description and the ontology instance data. Context rule repository contains all heuristic rules associated with the application domain. The context reasoner implements two kinds of reasoning. One kind exploits the ontology semantics defined by the context ontology to detect the inconsistent information. The other kind exploits the heuristic rules to interpret and aggregate contexts, and deduce high level knowledge. Context report web service API provides interfaces to the mobile gateways for reporting the context information to the backend server. Context publication web service API provides interfaces to the mashup server for extracting the context information from the backend server.

2.4 Mashup server and mobile mashup applications

The mashup server is responsible for mobile mashup application development. A mashup application is a web application that combines data from more than one source into a single integrated tool [8]. Compared with existed mashup applications [9, 10, 11], the mobile mashup applications introduced in this paper are especially interesting because: 1) the mashup applications are based on the user-centric contexts. The mobile users are not only the context consumers, but also the context contributors. They contribute and share with each other; 2) based on the contexts and user ID, the mashup server can develop context-aware and individual mashup applications for each user; 3) taking the mobile phone as the service platform and interface to the user, the user can access the mashup services in real time and conveniently.

In the proposed framework, there is a backend server for the context management platform and a mashup server for mashup service providing at each application domain (parking lot, exhibition center, shopping mall, smart office/home, etc). The mashup server integrates the context information obtained from the context management platform on the backend server and information (map, video, photo, search, shopping, news, etc) from other mashup data sources, and provides mashup services to users through mobile phone.

3. Example scenario

We take the exhibition center scenario as an example to show the WSN based mobile mashup applications.

Suppose there are many beacon nodes and data sampling nodes deployed in the application domain, with the mobile phone, each user collects messages from the beacon nodes and data sampling nodes nearby. On the mobile phone, user's location contexts and the onsite environmental contexts (e.g. temperature and noise level), together with the user ID and time stamp, are extracted and reported to the backend server through the context report web service API of the context management platform.

On the context management platform, the reported contexts are stored according to the predefined context ontology. Newly arrived context will trigger the context reasoner to 1) detect whether it is consistent with the context ontology definition according to the context knowledge and 2) interpret and aggregate contexts, and deduce the high level knowledge according to the heuristic rules. For example, it can deduce the visitor's interests based on the exhibition booths he visited and the time spent on each exhibition booth.

The mashup server can access above contexts from the context management platform through context publication web service API. Integrating these contexts with the 2D digital map (or 3D virtual scene) on the local server, the mashup server can show real-time

visitor distribution and environmental parameter (temperature and noise level) distribution in the whole application domain to the users through mobile phone. In addition, it also can integrate visitor's real-time location context with the introduction information (photo, text, video, etc) of each exhibition booth on the exhibitor's server, and provide related information to the visitor when he is approaching one exhibition booth.

4. Conclusion

This paper proposed a framework for wireless sensor network (WSN) based mobile mashup applications. In this framework, the mobile phone acts as the mobile gateway to the WSN and collects the sensor data. The extracted user-centric contexts are reported to the context management platform which provides application programming interfaces (APIs) to the mashup server. In the application domain, the mashup server integrates the users' contexts from context management platform and other mashup sources, and provides mashup services to users.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", *IEEE Communications Magazine*, 2002, 40(8), pp. 102-114.

[2] T. Arici and Y. Altunbasak, "Adaptive Sensing for Environment Monitoring using Wireless Sensor Networks", *in Proceeding of the IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, USA, 2004, pp. 2347-2352.

[3] J. J. Evans, "Wireless sensor networks in electrical manufacturing", *in Proceeding of Electrical Insulation Conference and Electrical Manufacturing Expo*, Indiana, USA, 2005, pp. 460-465.

[4] S. S. Yang, Y. G. Kim and H. Choi, "Vehicle Identification Using Wireless Sensor Networks", *in Proceeding of IEEE Southeastcon 2007*, Richomond, USA, 2007, pp. 41-46.

[5] X. Sheng and Y. H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks", *IEEE Transactions on Signal Processing*, 2005, 53(1), pp. 44-53.

[6] X. Zhang, X. Wang, Y. Ren, C. Chen and J. Ma, "A Novel Compatible Hardware Expansion Method Based on General Memory Interface", *in Proceeding of 2009 International Conference on Communications and Mobile Computing*, Kunming, China, 2009.

[7] P. Korpipää, J. Mäntyjärvi, J. Kela, H. Keränen and E.-J. Malm, "Managing context information in mobile devices", *IEEE Pervasive Computing*, 2003, 2(3), pp. 42-51.

[8] Mashup (web application hybrid), *Wikipedia*, http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid).

[9] Y. Li, J. Fang and J. Xiong, "A context-aware services mash-up system", *in Proceeding of Seventh International Conference on Grid and Cooperative Computing*, Shenzhen, China, 2008, pp. 707-712.

[10] A. Kansal, S. Nath, J. Liu and F. Zhao, "SenseWeb: An Infrastructure for Shared Sensing", *IEEE Multimedia*, 2007, 14, pp. 8-13.

[11] A. Brodt, D. Nicklas, S. Sathish and B. Mitschang, "Context-aware mashups for mobile devices", *Lecture Notes in Computer Science*, 2008, 5175, pp. 280-291.