The Research on IoT System Construction Method to Support Collaborative Awareness Service

Huaizhou Yang
College of Computer Science
Xi'an Shiyou University
Xi'an 710065, China
yanghuaizhou@sina.com

Liumei Zhang
College of Computer Science
Xi'an Shiyou University
Xi'an 710065, China
rikrun@gmail.com

Abstract—The current IoT (Internet of Things) systems lack the ability of collaborative awareness. Therefore, an IoT system architecture supporting collaborative awareness service is presented. First, in order to hide the heterogeneous factors of IoT, a SOA (Service-Oriented Architecture) based IoT middleware infrastructure is accepted and applied. Then, the components in different levels of middleware platform are designed and described to support dynamic service composition, and provide an event-driven service collaboration mechanism. Finally, the capabilities of the presented architecture to support collaborative awareness service are analyzed and evaluated from three aspects: activity awareness, policy awareness and process awareness. By the presented IoT system infrastructure, the system developers and service process designers can dynamically search, select, and use the real-world services running on physical devices.

Keywords- Internet of Things; Collaborative Awareness; SOA; Middleware

I. INTRODUCTION

The IoT (Internet of Things) systems include many types of services. According to their application types and technology characteristics, the IoT services can be classified into four major types: identification related service, information fusion service, collaborative awareness service and ubiquitous service[1]. This service classification mode represents the inevitable developing trend that the functions of IoT services are enhanced continually.

The identification related services are foundational IoT services. Via the ONS (Object Naming Service) provided by EPCglobal or uID infrastructure of IoT, the identification related services, such as mobile wallet service, can be constructed by RFID (Radio Frequency Identification), barcode, two-dimensional bar code and other identity recognition technologies. The information fusion services often take the form of data acquisition through terminal sensors, information fusion after data transmission, and information usage for a certain application in an IoT system, such as remote meter reading system. In contrast to the above-mentioned two types of services, the collaborative awareness services are mainly used to achieve information cooperation process for a common application goal via distributed information acquisition, data communication, intelligent data computing, and terminal device control. This kind of services require intelligent information interaction between IoT smart objects, and furthermore, require compositive and complementary information process for IoT awareness data from different application systems and different system platforms. The ubiquitous services are a category of idealized and ultimate IoT services, which support information service and exchange at anytime, in anywhere, and between any people and things.

Among the above-mentioned IoT service types, collabora-tive awareness service acts as a connecting link between other types of services, and plays a key role in achieving comprehensive awareness, information sharing, and ultimate ubiquitous service for IoT systems. The Internet, NFC (Near Field Communication) and CPS (Cyber-Physical System) technologies enable us to transform the daily objects into the smart objects, which can understand their own environment and interact with it. Therefore, these intelligent technologies make it possible to construct efficient and real-time awareness services in a collaborative mode[2].

The collaborative awareness service represents the development trend of IoT, but the IoT system construction methods to support the collaborative awareness services are still not studied deeply. Based on the SOA (Service-Oriented Architecture) middle-ware technologies, via system analysis, an IoT infrastructure supporting collaborative awareness service is presented in this study. The designs of main components in SOA middle-ware platform are described in detail. At last, the capabilities of activity awareness, policy awareness and process aware-ness are analyzed and evaluated for the presented IoT system architecture, respectively. The analysis and evaluation results indicate that the services in different system levels can achieve collaborative awareness via the presented IoT architecture.

© 2015. The authors - Published by Atlantis Press

II. SOA-Based IoT System

To realize the collaborative awareness, the various information process units in IoT system are consequentially required to be not only the providers but also the consumers of the awareness information. Therefore, it is a natural choice to design the IoT infrastructure as SOA to support the collaborative awareness service. Based on SOA middleware, it is convenient to hide different technology details, simplify the new service development, and integrate the existing technologies. The SOA-based IoT middleware architecture is shown in Fig. 1. The SOA middleware solution for IoT system mainly exhibits to abstract the device functions and communication abilities as different device services, and then, form a service set and construct a service composition environment.

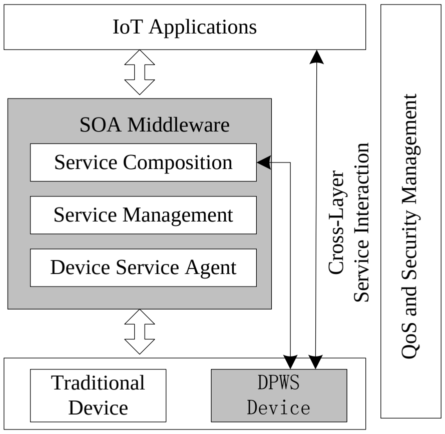

Figure 1. SOA-based IoT middleware architecture .

In the IoT sensing layer, there are many heterogeneous terminal devices whose access modes and functions are very different. The device service agent layer can hide heterogeneous technology details, process the input or output message through standard Web service interface, and meanwhile, transform the device service invocations into particular communication instructions for corresponding physical devices. The service management layer is mainly responsible for dynamic service discovery, service repository, service status monitoring, service configure, QoS management and semantic service, etc., which supports to realize complex service composition. Using the technology of service composition, the service composition layer can orchestrate the fine-grained device services into coarse-grained composite services via a workflow. These composite services represent the main business process and business logic, and can be used to construct complex application systems. DPWS (Device Profiles for Web Services) is a subset of Web service standards, which is used to support seamless and minimum interaction with the Web services running on the resource-limited embedded devices. The IoT applications and the composite services can make service information interaction directly with the DPWS devices without using the device service agent layer.

III. The IoT System Architecture Supporting Collaborative Awareness Service

A. *System Requirements*

On the foundation of SOA middleware technologies, the IoT system supporting collaborative awareness service must satisfy some main system requirements. They are described as below.

(1) Service enabled system functions.

The various types of device functions and system business functions can be flexibly encapsulated into the Web services with different granularities, which can facilitate the flexible system configure and eliminate the different categories of heteroge-neous factors in the system[3]. The services can acquire data, transmit events, accomplish net connection and interact with other services in a mode of high degree of autonomy[4,5].

(2) The event driven service collaboration mechanism.

In one respect, the IoT terminal devices may be mobile devices or other devices using wireless connecting, whose online statuses are often changed. The device services need to be able to send notifications when their statuses changed, which can facilitate other services to use their functions in time and successfully. And meanwhile, the business level services also need to advertise their business process statuses by events. The device services can subscribe these business process events and adjust their execution statuses in time according to the global business process information. In other respect, the system awareness informations usually are multi-source and real-time data flow, and have high degree of time-space sensitivity. The device service events need to be used to drive the execution of high level service functions, which will facilitate the business level services to make task scheduling according to the device events and improve the parallel and distributed process ability for multi-source awareness information. In addition, according to the time and space information of device service events, the business level services can verify the validity of the awareness data, filter the invalid data and improve the reliability of awareness information[6,7].

(3) Dynamic service discovery and service life cycle management.

The quantity and validity of IoT awareness terminals always change dynamically. Therefore, the IoT system needs to be able to discover device services, establish dynamic service repository and manage the lifecycle-related service activities, such as service running, stop, elimination, upgrade and installation[8].

(4) Service composition.

The IoT device services have fine-grained, low reliable and dynamic characteristics. They are mainly responsible for the basic awareness functions. However, the business level services mainly focus on the realization of business logics and functions. In order to achieve the effective collaboration between business level services and device level services, the IoT system need be able to integrate the device services into business services via service composition and provide new application services for the system[9].

B. System Architecture

Researchers divide the SOA-based IoT middleware platform into three layers: business service layer, service management layer and device service layer. The components and their interaction relationships in each layer are shown in Fig. 2.

(1) Device service layer.

The device service agents transform the functions of IoT awareness terminals into Web services, and are responsible for interacting with services in same layer or upper layer via Web service mode. Because the DPWS devices support the Web service standards directly, the agent functions of this type of services can be accomplished by normal DPWS plugins. The service agent of traditional devices can be realized by the third-party plugins, which also can be realized by the DPWS plugins after the transformation of communication mode via gateway or adapter.

In addition, the event notification agent and the event subscription agent are deployed in this layer. The event notification agent sends the evens about the change of device running status (e.g. start, stop and alarm, etc.) to the event repository in the service management layer for event subscription. Meanwhile, in order to achieve the intelligent data acquisition, data fusion, data filtering and status adjustment (e.g. changing to the power save mode), through the event subscription agent, the device services can learn the running statuses of the corresponding services and especially the high level business services, and furthermore, interact with other services via event driving.

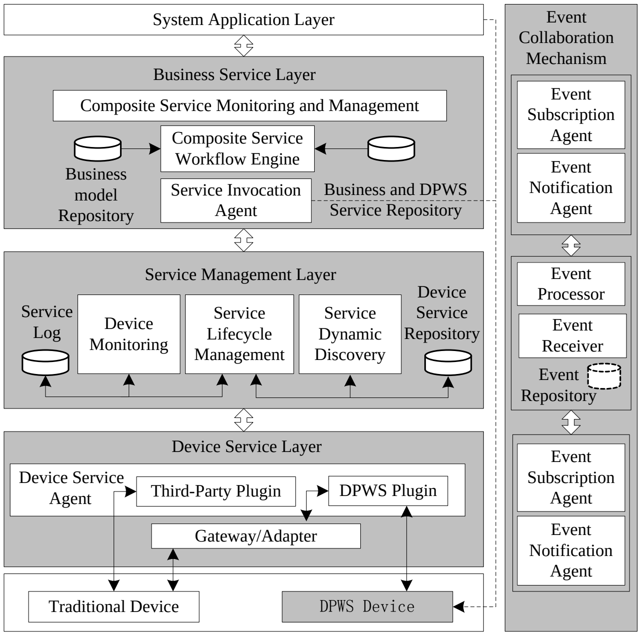

Figure 2. The IoT architecure supporting collaborative service.

(2) Service management Layer.

Through the WS-Discovery technology of DPWS, the service dynamic discovery component fulfills its functions of device service discovery, and then stores the discovered device services into a unified device service repository. Because the WS-Discovery technology is based on UDP (User Datagram Protocol), the service dynamic discovery component needs to be repeatedly deployed in each local physical station[10]. The device monitoring component is responsible for monitoring and recording the running status information of each active device service. The service life-cycle management component is responsible for the static data management of all online and offline devices, and also responsible for the update of the device service repository according to the results of device monitoring and service discovery. After function extend, the service lifecycle management component can accomplish the works of service installation and service update. The service log component facilitates the work of historical data intelligent analysis at a later stage.

The event receiver is responsible for receiving the event notifications from business level services or device level services, and then filters these events. The event processor is in charge of the classification, combination and transformation to the initial simple events, and then records these processed events into the event repository for event subscription[11].

(3) Business service layer.

The business model repository is used to store the pre-defined composite service models which are constructed based on the business logics and rules. This type of model belongs to the business process model, which follows the BPEL4WS, OWL-S or other corresponding standards. These composite service models could have been bound with the static services, but they also can be some abstract business models without any service binding. In order to facilitate the service reuse and the direct invocation to DPWS device services, the business and DPWS service repository is designed to store the information of the existing run-time composite services and the information of those DPWS device services used in cross-layer mode.

The composite service workflow engine is used to execute the service composition process according to the business model, and invoke the functions of component services involved in the service composition orchestration process. Because the online statuses of IoT terminal devices are not stable, the service invocation agent is designed to buffer the service invocation instructions, which enables the composite serve process to be executed continually when the disabled device service can be used again.

The event notification agent and the event subscription agent in this layer are responsible for sending the status change events about business execution process and subscribing the execution status information of corres-ponding device services, respectively. The business process can alter itself to adapt to the device service events. For example, when receiving an alarm event from device services, a business process can change collaboratively to the emergency execution mode. It needs to be noted that the cross-layer service invocation cannot fully use the functions of middleware platform. Only when a given DPWS device can provide stable service, the cross-layer service invocation is suitable. Therefore, this kind of service invocation is illustrated in dash lines in Fig. 2.

C. The Design of Event-Driven Serive Collaboration Awareness Mechanism

In Fig. 2, the event-driven service collaboration awareness mechanism is comprised of the components in charge of event sending, receiving and processing in the middleware platform and their relationships. Researchers adopt WS-Notification standard to design the corresponding components. This standard includes three specifications: WS-Notification, WS-Brokered Notification and WS-Topic. Specifically speaking, researchers select WS-Brokered Notification to set event notification interface for event receiver and event subscription agent, which are event consumers, and then set event subscription interface for event processor, which is event producer. The event subscription supports two types of event delivery modes, push and pull. Because the event notification agent only does the job of simple event sending, researchers do not set event subscription interface for it. The detailed event-driven service collaboration awareness mechanism is shown in Fig. 3.

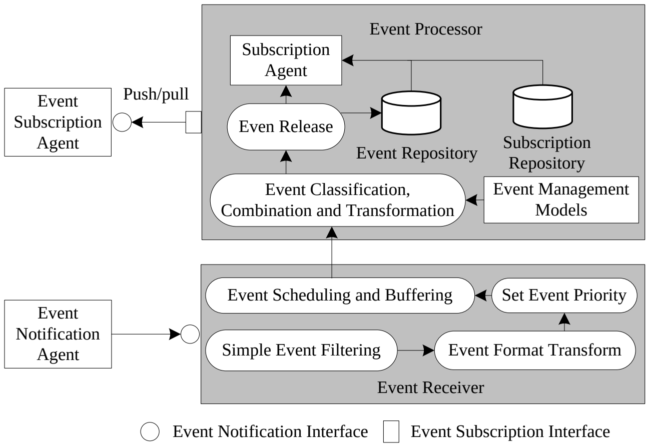

Figure 3. The event-driven service collaboration awareness mechanism.

The IoT awareness events are usually time-sensitive. The device services are possible to send events in high frequency. Therefore, the event receiver needs to filter the initial received events according to the predefined rules, for example to judge whether an event is overdue or redundant. After that, the formats of initial received events are transformed for later processing, and then the appropriate priorities are set to these events. Finally, according to the event priorities, these events are scheduled and buffered.

The event processor firstly fulfills the works of event classification, combination and transformation according to the event management models, which transforms the simple events into the advanced business events for the related services to use them easily. The event management models follow the WS-Topic standard. The WS-Topic defines a hierarchical topic tree. Through the topic tree, the event types can be defined and the events can be filtered when they are subscribed. After the events are released, they are stored into a dynamic event repository for service subscription. The subscription repository is mainly used to store the subscription relationships between services and events. The event subscription agent is used to fulfills the function of event subscription according to the subscription relationships, and actively push the high real-time required events to the related services.

IV. THE ANALYSIS AND EVALUATION OF SYSTEM COLLABORATION AWARENESS CAPABILITIES

Through the SOA middleware platform, the IoT device functions and the high-level core business functions can be uniformly encapsulated into services in different granularity. The awareness types of these IoT services can be classified into three categories: activity awareness, policy awareness and process awareness. The service application modes can be divided into three levels: the function mode, the business rule mode and the workflow mode[12]. Their relationships are shown in Fig. 4.

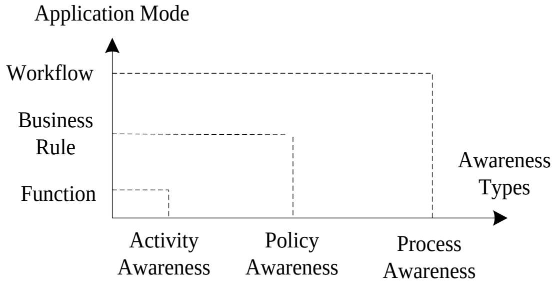

Figure 4. The relationships of IoT service awareness types and applicaton modes.

The activity awareness means that a service unit can detect the external events and activities of other services, and fulfills its specific application functions according to the detection results. However, an activity awareness service only records log data but has not service interaction ability. The policy awareness means that an activity awareness service unit can interpret the detected events and activities according to the predefined application policies. The application mode of policy awareness is usually reflected as a set of business rules for further activities based on the detected events. The policy awareness services provide task-sensitive service status information for the outside, especially the information of service warning and alarm. The process awareness means that a service unit (e.g. a device service) can perceive the current execution situation of a business process and the global system information. Its application model is often related with a workflow. The above-mentioned three awareness types can be used as the design principles of IoT system architecture.

The system architecture proposed in this study can fully support three awareness types.

(1) The activity awareness ability.

A service unit can accurately detect the target events through event subscription. The subscription relationships can be configured flexibly. The service log can be recorded in detail in the service management layer.

(2) The policy awareness ability.

According to the specific business rules, the initial events can be interpreted by the event filtering, event priority setting and event management models. After that, the initial events can be converted to the target events for directly driving the service activities. The device service agent can also be used as the policy set point to deal with the received events intelligently. The above-mentioned mechanism relieves the policy awareness stress to the service units, especially to the device services. This mechanism also achieves the separation between the collaboration policy and the collaboration behavior, which facilitates the flexible adjustment to the collaboration policies.

(3) The process awareness ability.

In the presented IoT architecture, the composite services can advertise their business process execution statuses in time through events. Via these composite service events, other services, especially device services, researchers can learn the global system information in time and accomplish necessary collaboration works.

The implementation of the above-mentioned awareness abilities makes the presented system architecture successfully meet the IoT system construction requirements to support the collaborative awareness service.

V. Conclusion and Future Work

In this study, on the basis of SOA-based middleware platform, the functions of IoT terminal devices can be converted into services, which can shield the heterogeneous system technology details. In order to adapt to the dynamic characteristics of IoT terminals, the functions of dynamic service discovery and service lifecycle management are designed for the middleware platform. The complex business service functions are realized by the service composition. Meanwhile, in order to improve the collaborative awareness abilities of IoT services, the event-driven service collaboration mechanism is added into the basic SOA architecture. In the future works, in order to satisfy the requirement of real-time service collaboration awareness for large IoT system, the problem of how to enable the IoT middleware platform to provide high throughput for event processing will be studied deeply.

Acknowledgment

This research was sponsored by Scientific Research Program Funded by Shaanxi Provincial Education Department under the grant No. 14JK1585 and Xi'an Science and Technology Project of China under the grant No. CXY1437 (7).

References

[1] X.J. Xin, J.L. Wang and M.D. Li, “The Business and Key Technology of the Internet of Things,” ZTE Communications. 2010,16(2): 27-30.

[2] M. Chen, J.F. Wan and F. Li, “Machine-to-Machine Communications: Architectures, Standards and Applications,” KSII Transactions on Internet and Information Systems. 2012,6(2): 480-497.

[3] V. Venkatesh, P. Raj, V. Vaithiya and R. Amirtharajan, “Smarter Environments: A Compact Service-Oriented Framework for Context-Awareness,” Journal of Artificial Intelligence. 2014, 7(4): 145-160.

[4] J. Neto, T. Silva, R. Assuncao, R. Mini and A. Loureiro, “Sensing in the Collaborative Internet of Things,” Sensors. 2015, 15(3): 6607-6632.

[5] A. Kalburgi, C. Smita, O. Deshmukh and V. Chaudhary, “i-IoT (Intelligent Internet of Things),” International Journal of Computer Science and Mobile Computing. 2015, 4(4): 636-642.

[6] J. Kong, J. Jung and J. Park, “Event-Driven Service Coordination for Business Process Integration in Ubiquitous Enterprises,” Computers & Industrial Engineering. 2009, 57(1): 14-26.

[7] Y. Zhang, L. Duan and J.L. Chen, “Event-Driven SOA for IoT Services,” 2014 IEEE International Conference on Services Computing (SCC). Anchorage, AK: IEEE Press, 2014: 629-636.

[8] H. Jo, J. Kwon and I. Ko, “Distributed Service Discovery in Mobile IoT Environments Using Hierarchical Bloom Filters,” Lecture Notes in Computer Science. 2015, 9114(6): 498-514.

[9] Z. Yang and D.H. Li, “IoT Information Service Composition Driven by User Requirement,” 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE). Chengdu, China: IEEE Press, 2014: 1509-1513.

[10] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess and D. Savio, “Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services,” IEEE Transactions on Services Computing. 2010,3(3): 223-235.

[11] S.Z. Liu and J. Cao, “Event Collaborative Awareness Model Based on SOA,” Computer Engineering. 2009,35(3):48-50.

[12] G. Kortuem, F. Kawsar, D. Fitton and V. Sundramoorthy, “Smart Objects as Building Blocks for the Internet of Things,” IEEE Internet Computing. 2010,14(1): 44-51.