An Event-driven Service-oriented Architecture for the Internet of Things Service Execution

http://dx.doi.org/10.3991/ijoe.v11i2.3842

Lina Lan[1], Bai Wang[1], Lei Zhang[1], Ruisheng Shi[1], Fei Li[2]
[1] Beijing University of Posts and Telecommunications, Beijing, China
[2] Vienna University of Technology, Vienna, Austria

***Abstract*—IoT (Internet of Things) bridges the physical world and information space. IoT services are environmentally sensitive and event-driven, so new IoT service architecture should adapt to these features. This paper analyses IoT sensing service characteristics and proposes future services architecture. It is focused on middleware architecture and interface presentation technology. In the middleware layer, traditional SOA architecture is insufficient in real-time response and parallel processes of service execution. This paper proposes a new sensing service system based on EDSOA (Event Driven SOA) architecture to support real-time, event-driven, and active service execution. At the presentation layer, this paper presents new IoT browser features, including using augmented reality technology for input and output and realizing the superposition of the physical world and abstract information. Through a use case and proof-of-concept implementation—road manhole cover monitoring system—we verified the feasibility of the proposed ideas and framework.**

***Index Terms*—Internet of Things services architecture, sensing service, service collaboration, SOA (Service Oriented Architecture), EDSOA (Event Driven SOA), Internet of Things Browser, Augmented Reality**

I. INTRODUCTION

With the rapid development of communication technology, the Internet is gradually evolving from a traditional information publication platform to a distributed computing environment. Cloud computing technology opens up more and more data resources, computing resources and applications as services on Internet.

The sustainable development of computer network technology has created a new world full of sensor, RFID (Radio Frequency IDentification), tags for sensing devices in the physical and social environment [1] [2]. The Internet of Things (IoT) goes beyond the original vision of Internet and the World Wide Web. IoT makes the physical world a network and information world. IoT connects and integrates the physical world with the information space. It is an extension of the Internet to the real world, and it brings Internet to an even larger scale.

By introducing mass sensors, IoT perceives changes in the real world and provides more intelligent services for people. The IoT service has different features from the traditional Internet service. It can provide environmentally sensitive, real-time, and active service. In this paper, we analyze the characteristics of IoT perception service and put forward the new network service system architecture. A new application layer display technology is discussed in this paper. Finally, the feasibility of the framework is demonstrated through a proof-of-concept implementation.

II. IOT SERVICE SYSTEM FEATURES

A large number of small IoT applications have been developed already. Sensor networks, CPS (Cyber Physics Systems) and other emerging intelligent services promote the information revolution of a series of applications such as healthcare, environment monitoring, logistics, etc. These new application services are distributed, real-time, dynamic and interactive. They need to deal with information acquired from the physical world and also need to consider the context of the applications themselves.

A single perception of the physical world is the foundation of intelligent collaborative services. In the complex service scenarios, we need to collect diverse sensory information to achieve intelligent collaborative service. In particular, IoT service system has the following characteristics different from the traditional information service system [9]:

1) Perceptibility of the environment. Perceptibility is the most basic feature of a perceptual awareness service system. A perceived service system interacts with the environment to support dynamic adaptive computing based on scenarios. The sense device at the end of the sensing service system perceives the environment, including the surrounding temperature, humidity, light, air, monitoring the movement of the animal, etc. The sensory data constantly transfer to the service system, and the service system handles it in real-time to support intelligent calculations based on the situation.

For example, according to the user's current location, a mobile phone shows the local weather conditions, real estate prices, and maps information to users. The traditional information service system generally is not contextual. It generally returns response information according to the user's input keyword query information.

2) The service execution is driven by events. The business events are defined by users. For examples, "If a room temperature rises by 10 degrees in 1 minute, there may be a fire. The system should alarm." "If farmland humidity is lower than the threshold, a drought may occur. The system should alarm." etc. The sensing service systems collect and analyze the real-time sense data, get the business events, and then do real-time processing and response to events. As a result, perceived service has the characteristics as the event-driven service execution. And the traditional information service generally uses the request/response model which executes the service to return a response according to requests from the user or client.

So perceived service system needs to support event-driven service execution mechanisms.

3) Service has the characteristics of collaborative work. With the expansion of the sensing service application scope, perceived service may need multiple application systems to work together. For example, the early warning service system for forest fire should be coordinated with security, fire control, and medical departments to work together. To handle the large emergency events, more departments have to work together. Perceived service system commands many components actively. These components include sensors, computation engine, database, responders, actuators, etc. In the future, the programming model will be the agent cooperation pattern, and it is no longer a sequential process.

4) The sensing service has a long lifetime. By configuring operations perceive service system is running in the long period, instead of the traditional service execution by service calls. In the traditional Internet, we enter keywords and Web search engine returns the search results. In the sense service system, the service request is like “if the event occurred matching the keyword, please inform the system”. The service request will remain in effect until the system deletes it initiatively.

5) The sensing service is predictive and proactive. The sensing service system predicts business events and actives service in advance according to the sensory data and context scenarios. For example, the early warning of important event is predicted by the sensing service system. The intelligent active sensing service breaks the traditional passive service mode. It can use all kinds of network resources and actively carry out comprehensive services to meet the individualized service for users.

At present, the application systems based on the traditional information service system architecture lack of connectivity. The information is difficult to be shared effectively. The services lack of collaboration. It is difficult to support the special requirements of a perceived service.

III. IoT SERVICE SYSTEM ARCHITECTURE

A. *Future network service system technology development trends*

The development of future network will deeply change the scale and the behavior of distributed systems. It will bring technological changes to the network transmission protocol, service system architecture, application presentation technologies, etc. The development trend of future network service system technology is shown in Figure 1.

In the traditional Internet, TCP/IP and HTTP are the core protocols in the Internet network services to support communication and information sharing. SOA (Service Oriented Architecture) architecture is dominant in the middleware layer. In the future network environment, perceived service will play a more important role. Between perception and service, new technical framework will need to link them. Traditional SOA should be introduced in a distributed Event Driven mechanism, namely EDSOA (Event Driven SOA) to build large-scale distributed applications in IoT.

Whether TCP/IP network protocols are able to be adapted to the future network and whether we need to create a new more effective network transmission protocol to support the future network services have become hot research issues. Some experts put forward that the future network is based on the content of the network, and content based routing strategies should be adopted [3]. For example, the Internet routing mode may be based on publish-subscribe technology.

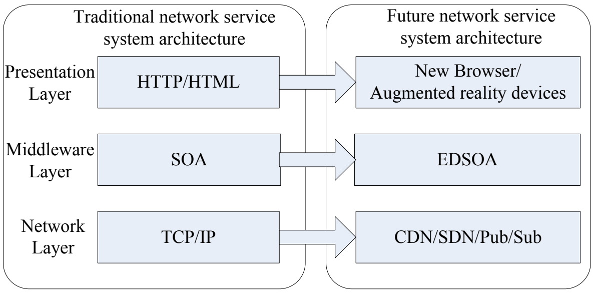

Traditional network service system architecture
Future network service system architecture
Presentation Layer
HTTP/HTML
New Browser/ Augmented reality devices
Middleware Layer
SOA
EDSOA
Network Layer
TCP/IP
CDN/SDN/Pub/Sub

Figure 1. Future network service system technology development trends

In the display layer and human-computer interaction layer, augmented reality devices like Google Glass Prototype [4], the new type of browser technology like IoT Browser [5] challenges the traditional Web browser based on HTTP/HTML.

This article focuses on the middleware layer architecture and presentation layer technology development.

B. *Development from SOA to EDSOA*

SOA is a distributed computing model. It can organize and use all computing ability distributed in different ownership domains to fulfill user requirements [6]. Service is the key concept in SOA. Service is autonomous, loose coupling, and platform independent entity. The entity can be described, published, used and combined.

There are two mainstream modes of Web Service and RESTful (REpresentational State Transfer) style to implement SOA architecture. Web Services is based on SOAP (Simple Object Access Protocol). REST is another technical implementation of SOA [7]. REST defines the network service as the perspective resources. The client uses the URL request resource to the server or proxy server. All interaction in REST is based on HTTP protocol of synchronous request/response message. So RESTful style services don't need SOAP or other application layer protocols, but only need HTTP. The case study in [8] shows that RESTful Web Service is more concise than SOAP-based Web Service, and even provides better performance. So RESTful is named lightweight SOA implementation technology. It gradually becomes the dominant technology architecture of SOA.

SOA deconstructs the system. The entire application is decomposed into a series of independent services. Meanwhile, a variety of standards are formulated to make the service easy to reuse or applied on various platforms. Since SOA pays more attention on static information but not the dynamic information, including random events, the event can't be better responded. Therefore, SOA has deficiencies in the dynamic business matching and inactive service execution.

Event-driven Architecture (EDA) is an event-driven process execution component model. In the EDA system, events can be transmitted between components and services. The generation of an event can trigger one or more services to be invoked.

TABLE I.
DIFFERENT CHARACTERISTICS OF SOA AND EDA

Characteristics	SOA	EDA
Coupling degree	Loosely coupled interactions	Decoupled interactions
Communication mode	One-to-one communications	Many-to-many communications
Trigger mode	Consumer-based trigger	Event-based trigger
Response mode	Synchronous	Asynchronous

The different characteristics of SOA and EDA are shown in Table 1 above.

In Table 1, in the coupling degree of the services, SOA services use request and response interaction which is loose coupling; while EDA services which are driven by event are completely decoupled. EDA is more loosely coupled than SOA. In the mode of communication, there is one-to-one communication between SOA services just as one service sends request to another service. EDA uses many-to-many communication among EDA services, while one service can send event to multiple event receiver services, and one receiver can correspond to more than one event senders. EDA has more flexible communication mode than SOA. Considering the cause and execution of the services, SOA service is triggered by a service requester, while EDA service is triggered by event which responds more immediately. Regarding the service response way, SOA is a synchronous response. The requester needs to wait for a response, and then continue to execute. EDA is an asynchronous response. The incident service can execute continuously without waiting for a response. So EDA has better performance in the service of the coupling, parallelism, and service implementation in real-time.

Any system can be abstracted to be service components and communication channels, just as the computing and communication ability. SOA focuses on components and services, as shown in Figure 2 (a). The service is executed by request/response mode. EDA pays more attention to channel. Event-driven SOA introduces event-driven mechanism in SOA architecture. The service is executed in event-driven way. Services can generate events and also receive events. The service which receives events can trigger the service execution and change status. As shown in Figure 2 (b), the services send and receive events among services through Event Bus. Event Bus realizes the separation of computation and communication ability in the system. The running software likes a network composed of objects (service) and messages (events). EDSOA architecture is better than SOA architecture in concurrency and asynchronous execution. EDSOA is more suitable for IoT services in real time and concurrent, etc.

Now, EDSOA architecture has been preliminary applied in the field of enterprise [9-11], e-government [12] [13] and other fields. Along with the development of IoT application, EDSOA architecture will play an important role in the future IoT services architecture.

C. New Browser Technology of IoT

IoT service has different characteristics from Internet service. It puts forward new requirements on the application layer display technology. It is important to offer rich and intuitive services through the presentation of sensors, sensing object or environment, perceptible data, control operation of equipment, etc.

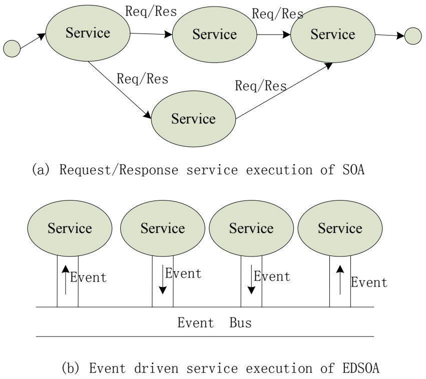


Figure 2. Different Service Executions of SOA and EDSOA

Just like Internet browser is used to display Internet resources, IoT Browser is used to display IoT resource. The resources of IoT are different from the Internet resources at the following characteristics: [5]

1) The character of location-awareness. Internet users consider the information and service on Internet. IoT user not only concerns the information, but also concern about the location of the services provided. For example, the place a high temperature alarm signal of the temperature sensor is located. But Internet users generally don't care about the information is stored in which host in the Internet.

2) The character of physical entity. IoT devices and the perceived objects are specific physical entities. The objects in Internet are almost virtual entities just as software. Sensory data should represent the physical characteristics. For example, humidity in the soil is sensed by the humidity sensor. The information can show whether the flowers lack of water or not. The IoT Browser interface is shown in Figure 3.

4) Semantic sensitivity. With the development of IoT, any two devices can be interconnected. The different IoT applications can access others' equipments. So the communication between the equipments cannot be confined to their own special protocol. The semantic information to explain the device should be adopted to be widely recognized by more applications.

Paper [5] proposed the method using a new device of Brower of IoT - Sentient Visors to browse IoT. The Sentient Visors is a kind of new intelligence software. The user can use it to know the status of the sensors connected to IoT and interact with it. The user can see augmented reality overlay real physical objects and abstract information as shown in Figure 3.

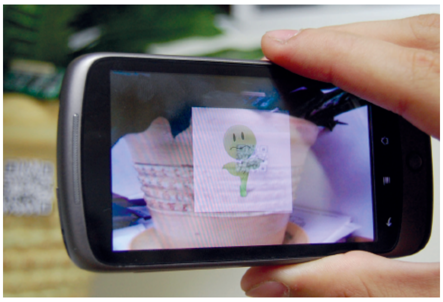


Figure 3. IoT Browser [5]

IV. CASE STUDY

Based on the above, we design and implement a perceived service system - Road Manhole Cover Monitoring System (RMCMS). On the key roads, RMCMS deployed the cameras over road manholes to monitor the status, and the position sensors were installed at the bottom of the manhole covers. Cameras and sensors collect real-time video data and the position data of covers to upload to RMCMS. The system analyzes the sense data to find the business events, such as manhole cover tilt, manhole cover mobile, etc, and then sends alarm in real-time to achieve the perception of the abnormal state of manhole cover.

A. System Architecture

EDSOA is adopted in the architecture of RMCMS as shown in Figure 4. The camera service is defined as taking pictures and videos. The position sensors provide position service. When position sensors detect position changes, such as tilt, mobile, inversion, etc., the sensor position service will generate position changing event. Camera photograph service and video service will receive the event and then take pictures and video services. That means position change event driving photo service and photographic service execution. By working the three services together, the real-time monitoring of the abnormal state of manhole cover is realized. The system alarms when an event happens, and provides photos and videos at that time.

Users can also easy to operate camera equipment through the user interface of RMCMS. The system will create user operation event when user operates, for example changing the camera parameters such as angle, focal length, aperture, and contrast, or taking photograph, video, etc.

B. Sevice and Event Pubilsh and Subscibe Definition

In the system, the service definition is following as:

```
Class Camera{
        int CameraID;
        String type;
        String model;
        String video(int CameraID);
//video service of camera
        String TakePicture(int CameraID);
//photograph service of camera
        ….
}
 Class Sensor{
int SensorID;
String type;
String model;
String longitude;
String latitude;
        String AnalyzeEvent(String data);
//analyze data to detect the event and send event to event bus.
……
}
```

Event Bus software provides the event publish and subscribe interface as following:

```
new EventPublisher().send("TopicA","EventA1");
new EventSubScribe().init("TopicA","SubScribeID",Objectof EventHandler);
```

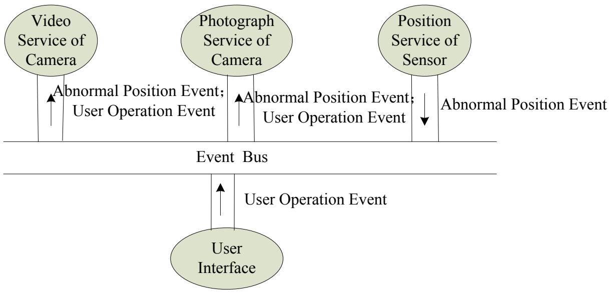
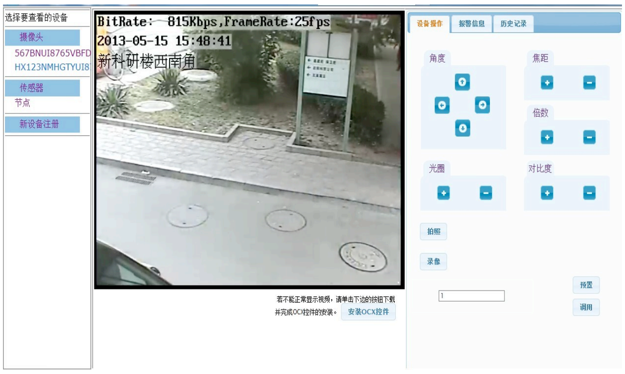


Figure 4. Architecture of Road Manhole Covers Monitoring System



若不能正常显示视频，请单击下边的按钮下载并完成OCX控件的安装。

Figure 5. Interface of Road Manhole Cover Monitoring System

For example, when an abnormal position event occurs, the Sensor service sends the event to Event Bus, and Event Bus notifies this event to all described services such as Camera service and User Interface service. Camera service calls its take picture method and video method. User Interface service also shows the event information on the screen. The two services are all executed by the event driven.

The system uses EDSOA to realize event-driven services collaborative execution. The event is detected in real-time and the services are performed initiatively. These are the main different characteristics compared with the traditional monitoring system.

C. System Interface

The presentation layer of the system presents superimposed images and data to improve visual and convenience. Figure 5 is the interface of RMCMS.

D. Performance Test of the Architecture

To compare the performance between SOA and EDSOA architectures, we test the execution time from the occurrence of abnormal position event to the termination of services execution in the two architectures. The test environment is Windows XP, the CPU is Intel Core i5, and the memory is 4G. The test result is demonstrated in Table II.

The EDSOA architecture takes 5,266 ms while SOA takes 9,156 ms. EDSOA is faster than SOA because the two functions of TakePicture() and video() can be executed concurrently.

V. CONCLUSION

The services in EDSOA can be executed concurrently if they receive the event at the same time. So the performance of EDSOA is better than SOA in which the ser-

TABLE II.
PERFORMANCE TEST OF SOA AND EDSOA

Item	SOA	EDSOA

Service Process	SeviceFlow. AnalyzeEvent();
	SeviceFlow.TakePicture();
	SeviceFlow.video();
		EventPublisher().send();
		Camera.TakePicture();
		Camera.video();

Execution Time(ms)	9156	5266

vices execute in sequence called by another service. The coupling of the services in EDSOA is looser than the services in SOA. For a large amount of services, the complexity of the EDSOA system is less than SOA system.

IoT service system architecture should be developed from SOA architecture to EDSOA evolution to support the real-time concurrent event driven service. The presentation layer will use the new browser and augmented reality technology to achieve a more intuitive image of IoT services.

ACKNOWLEDGMENT

Anonymous reviewers provided useful feedback that helped to improve the quality of this paper.

REFERENCES

[1] Calton Pu: A world of opportunities: CPS, IOT, and beyond. DEBS 2011: 229-230

[2] LIU YUN-hao, From ubiquitous computing and CPS to Internet of things: the next generation Internet horizon. Communications of China Computer Federation, 5(12), Dec. 2009, pp.66-69.

[3] Deborah Estrin, K. Mani Chandy, R. Michael Young, Larry Smarr, Andrew M. Odlyzko, David D. Clark, Viviane Reding, Toru Ishida, Sharad Sharma, Vinton G. Cerf, Urs Hlzle, Luiz André Barroso, Geoff Mulligan, Adrian Hooke, Chip Elliott: Internet Predictions. IEEE Internet Computing 14(1): 12-42, 2010.

[4] Google, Inc. Project Glass; http://g.co/projectglass

[5] J. Antonio García-Macías, Jorge Alvarez-Lozano, Paul Estrada, Edgardo Avilés-López: Browsing IoT with Sentient Visors. IEEE Computer 44(5): 46-52, 2011.

[6] HAN Yan-bo, XU Xiao-fei, HE Ke-qing, Service Computing in Internet Environment, Communications of China Computer Federation, Vol. 6, Issue 9, Sep. 2010, pp.10-11.

[7] Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures. Owen Branch, UC, 2000.

[8] Fatna Belqasmi, Jagdeep Singh, Suhib Younis Bani Melhem, Roch H. Glitho: SOAP-Based vs. RESTful Web Services: A Case Study for Multimedia Conferencing. IEEE Internet Computing 16(4): 54-63, 2012. http://dx.doi.org/10.1109/MIC.2012.62

[9] Karina Hauser, Helgi S. Sigurdsson, Katherine M. Chudoba. EDSOA: An Event-Driven Service-Oriented Architecture Model For Enterprise Applications. International Journal of Management & Information Systems, 14(3), 2010.

[10] Shi Ruisheng, Zhang Yang, Chen Junliang, Xie Hui, Tan Yulin, Guo Wenqi, Chai Zhaohang, Publish/Subscribe Network Service Infrastructure Design for EDSOA Service Platform, in Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems (CIMS), 18(8), August 2012.

[11] Li, Songtai and Qiao, Xiuquan and Li, Xiaofeng, Study on the Architecture of Platform for IoT Service Based on EDSOA, 2012 IEEE 12th International Conference on Computer and Information Technology (CIT), Pages 953--959, 2012.

[12] Sietse Overbeek, Bram Klievink, Marijn Janssen: A Flexible, Event-Driven, Service-Oriented Architecture for Orchestrating Service Delivery. IEEE Intelligent Systems 24(5): 31-41, 2009. http://dx.doi.org/10.1109/MIS.2009.90

[13] Klievink, Bram and Janssen, Marijn and Lankhorst, Marc and van Leeuwen, Diederik, An Event-driven Service-Oriented Architecture for Coordinating Flexible Public Service Networks, Electronic Government, 7th International Conference, Turin, Italy, 2008.

AUTHORS

Lina Lan is with School of Network Education and Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, BUPT (Beijing University of Posts and Telecommunications), Beijing, 100088, China (e-mail: lindalan2002@sina.com).

Bai Wang is with Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, BUPT, Beijing, 100876, China (e-mail: wangbai@bupt.edu.cn).

Lei Zhang is with Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer, BUPT, Beijing, 100876, China (e-mail: zlei@bupt.edu.cn).

Ruisheng Shi is with Key Laboratory of Trustworthy Distributed Computing and Service Ministry of Education, School of Humanities, BUPT, Beijing, 100876, China (e-mail: shiruisheng@bupt.edu.cn).

Fei Li is with Information Systems Institute, Vienna University of Technology, Vienna, Austria (e-mail: li@infosys.tuwien.ac.at).

This work was supported in part by National Grand Fundamental Research 973 Program of China under Grant No.2013CB329606; Chinese Universities Scientific Fund under Grant No.BUPT2014RC0701; Transformation Project of Scientific and Technological Achievements in Henan Province (2014) under Grant No.142201210009; Key Project of Science and Technology in Henan Province (2014) under Grant No.144300510001. The anonymous reviewers provided useful feedback that helped improve the quality of this paper. Submitted 09 April 2014. Published as resubmitted by the authors 10 March 2015.