

Architecting Semantics-based Publish/Subscribe Applications for the Internet of Things

Biao Dong^{1, a}, Jinhui Chen^{2, b}

¹School of Computer & Software, Nanjing Institute of Industry Technology, Nanjing, 210023, China

²School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China

^aemail:dongb@niit.edu.cn, ^bemail:cjh@nuist.edu.cn

Keywords: Semantics; Publish/Subscribe; Internet of Things

Abstract. This paper presents a semantics-based approach for architecting the internet of things(IoT) applications using publish/subscribe(Pub/Sub) mechanism. Considering the features of openness and integration which are necessary for reducing the technical threshold for the development of the IoT and ensuring the large-scale distributed deployment, IoT application environment based on Web(AEW) is proposed in IoT applications. Service subscription center(SSC) is proposed to provide a rules-based dynamic situational reasoning service and an execution service. The analysis indicates that the approach effectively realizes knowledge publishing, reasoning and distributed deployment for semantics-based Pub/Sub applications in IoT.

Introduction

The IoT infrastructure needs to be able to support the deployment of large-scale distributed applications of IoT, and to provide appropriate services for various industries and the public. But in the current IoT technical architecture, whether considered from the underlying physical sensing devices or the development of upper layer application platform, the architecture is restricted by the factors such as the heterogeneity of devices, the interoperability between devices and platforms, the heterogeneity of communication protocols and application platforms, and the cost of development. This results in a fault zone between networking technology and end-user demand, and reduces the efficiency of the development of IoT application[1,2]. Currently, the devices in IoT can only respond to users with accurate and complete information. However, a task of IoT is often an abstract task which is described by high-level semantics. Researches in the field of semantic Web have been devoted to the combination of knowledge engineering and artificial intelligence technology in order to represent and deal with the relevant data and knowledge. The semantic technology based on machine interpretation and formal description can be used to solve the problem of heterogeneous object description, information sharing and information integration. In the field of IoT, addition of semantic information can also contribute to machine interpretation and data self-description. A series of techniques, such as ontology, semantic annotation and semantic Web services, which are developed on the semantic Web, can be used as the main solution to realize the semantic of IoT[3,4]. Based on the above analysis, we construct AEW which provides a high-level semantic view to consolidate all kinds of IoT services. According to Web technical standards, each service in IoT can communicate with each other through the high-level semantics based information.

Pub/Sub is an asynchronous communication paradigm that supports many-to-many interactions between a set of clients. A client can be an information publisher, an information subscriber, or both. Client interactions are data-centric: publishers describe their publishable events, subscribers express their interest in events, and the Pub/Sub protocol delivers the published events to their corresponding event subscribers. This loose coupling of clients aids scalability in dynamic environments where clients and their roles can change frequently[5,6]. Pub/Sub is a suitable paradigm for IoT, and this paradigm is used to design a framework that exhibits scalability for target AEW architecture.

In view of the above analysis, an approach, for architecting semantics-based Pub/Sub applications for IoT, is presented. The main characteristics of the approach include two aspects.

Firstly, AEW is divided into three layers. We propose AEW architecture according to the three layer of AEW. Secondly, we construct SSC whose design is closely related to the service Pub/Sub mechanism.

The Architecture

The IoT application environment based on Web is divided into three layers: device access and ability abstraction layer, resource aggregation layer, intelligence service support layer. As the first layer of the application environment, the device access and ability abstraction layer ensures that application can access data and service resources provided by a variety of physical devices, and proposes a seamless scheme of integration of IoT device to Web. This layer also need to combine the characteristics of IoT application to provide some advanced management features, such as abstract resource naming, data caching and scheduling. The resource aggregation layer mainly provides the following services. Firstly, through business process engine, the layer provides resource-oriented business aggregation service and service generation service. Secondly, based on the technologies of business process choreographer and reuse, IoT device resources can be encapsulated into commercial application services to achieve cross-regional, cross-sectoral and cross-domain service collaboration. IoT devices can automatically sense the environment to get all kinds of information in the physical world, which reflects the situation and context of human activities. This intelligence service support layer mainly provides developers with resource-oriented context modeling and reasoning, and provides users with more intelligent and personalized service.

From the view of specific functional components, we propose AEW architecture according to the three layer of AEW. Fig.1 shows AEW architecture used in semantics-based Pub/Sub applications.

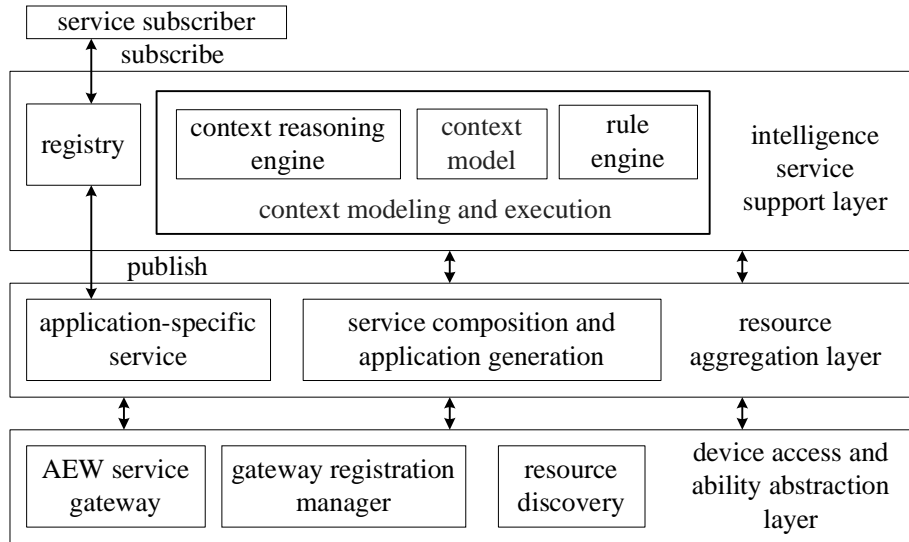


Fig.1. AEW architecture

At the device access and ability abstraction layer, there are the following components: AEW service gateway, gateway registration manager, and resource discovery. Among them, AEW service gateway is the key component to support technologies such as device access and resource abstraction of IoT. The gateway enables IoT application to directly access gateway in Web mode, and so as to communicate with IoT devices. The gateway is deployed in a distributed manner. Each gateway represents an application domain, and is used to access IoT devices in legacy applications. Gateway registration manager is a gateway identifier assignment system based on dynamic domain name service extension, and realizes the registration management of gateway. The manager allocates dynamic domain name and globally unique identifier for gateway. The resource discovery implements the search function of AEW resources. The specific process is as follows: Firstly, through the gateway registration manager, the resource discovery provides lookup to the AEW service gateway. Secondly, according to gateway identifier, the resource discovery realizes the

appropriate gateway resources discovery.

At the resource aggregation layer, there are the following components: application-specific service, service composition and application generation. Among them, the application-specific service includes content-based routing, semantics-based addressing mechanisms, naming and binding services, service dissemination and aggregation. These services would use simpler communication interfaces and abstraction than the raw communication interface. Through business process choreographer engine and corresponding resource aggregation mechanism, the service composition and application generation component is used to compose and choreograph IoT device resources and third-party services.

At the intelligence service support layer, there are the following components: context modeling and execution. The registry publishes semantic-based services according to ontology, and allows service subscriber to subscribe to them. At the heart of this layer is the context modeling and execution component which is divided into three main parts: context reasoning engine, context model and rule engine. For IoT device data which is abstracted as Web resources, the component has a method of context modeling, and provides a rules-based dynamic situational reasoning service and an execution service. Upon receiving a subscription, these processes such as monitoring, analyzing, planning and execution are managed by the component. We detail on the layer in the next section.

Semantics-based Publish/Subscribe

SSC includes a set of interfaces and a knowledge base, its design is closely related to the service discovery mechanism. SSC can use the following directory services, such as universal description discovery and integration(UDDI), lightweight directory access protocol. These directory services are built on structured Web service interfaces and syntax level matching. In order to meet the needs of semantic reasoning, it is necessary to select a semantic extension mechanism. Due to UDDI's universality in application and semantic extensibility, UDDI is selected as the basis for semantic SSC. UDDI is composed of five main data structures, namely business entity, business service, binding template, Tmodel and publisher assertion. The main problem with UDDI is that there is no way to provide a detailed description of the service capabilities, and only to provide keyword matching. We map an OWL-S service description document to UDDI so as to introduce semantic information into UDDI. In UDDI, entity business can be used to represent service provider. Each service provider has several services which are called as business services, these service providers are associated with business keys. Each service is associated with a binding specification. Let's describe the desired service attributes through Tmodel structure, which is a supplement way to represent effectively semantic information. All entities are in compliance with the Tmodel specification.

SSC provides a necessary infrastructure for a centralized organization of service, such as subscription, publishing and matching. On the basis of the traditional UDDI registration center, we propose a semantic based UDDI registration center. Its architecture can be divided into three layers: semantic management layer, service management layer, and interface management layer. As the lowest layer of SSC, the interface management layer is responsible for receiving external requests, updates and maintains service, and provides three interfaces such as registration, inquiry and discovery for service. As an intermediate layer of SSC, the service management layer is composed of the following components: service registration, service maintenance, mapping transform, and a semantic UDDI service library. When a request to the service registration is received, the service registration component stores the service document in the service description library. Generally, a service provider needs to submit a service description document based on OWL-S format to the service registry. If a service provider provides other types of documents, the service registration component need to first convert the document into OWL-S format, and then register. The service maintenance component includes three types of operations: lookup service, update service, and remove service. The service lookup operation is responsible for service queries based on specific parameters, and provides guarantee for the updating service operation and the removing service

operation. The service update operation is responsible for updating service description documents stored in database, which should be implemented in an automatic way. The service removal operation is responsible for deleting the expired or unavailable service description documents. The mapping transform component includes two types of operations: WSDL/OWL-S transform and OWL-S/UDDI semantic mapping. Among them, the WSDL/OWL-S transform converts a WSDL service description document as an OWL-S document. The OWL-S/UDDI semantic mapping converts an OWL-S service description document as an UDDI document, and save the UDDI document into UDDI database. As the upper layer of SSC, the semantic management layer completes the following two tasks: Firstly, the layer manages the related IoT ontology, and completes ontology construction, ontology saving, ontology editing and ontology query. Secondly, the layer adds semantic information to some service descriptions on syntax level, and performs semantic annotation.

The semantics-based IoT service subscription process is divided into the following three steps. Firstly, service registration information is handled by the service registration component. The component determines what format the service description document is. If a document is submitted in OWL-S format, the document is handled directly by the OWL-S/UDDI semantic mapping operation. If a document is submitted in WSDL format, the document is converted to OWL-S format by the WSDL/OWL-S transform operation, and then is processed by the intermediate layer of SSC. Secondly, the semantic management layer adds semantic information to the OWL-S service description document, namely, performs semantic annotation according to the corresponding ontologies in ontology knowledge base. Finally, the OWL-S service description document with semantic information is converted into UDDI format and stored in the semantic UDDI library by the OWL-S/UDDI semantic mapping operation.

Conclusion

In this paper, we propose a semantics-based approach for architecting the IoT applications using Pub/Sub mechanism. We derive conclusions of AEW architecture and SSC process from analysis, and design AEW architecture from three aspects: device access and ability abstraction, resource aggregation, and intelligence service support. The results show that the approach effectively realizes knowledge publishing, reasoning and distributed deployment for semantics-based applications.

Acknowledgement

This work was sponsored by Qing Lan project(Jiangsu province, china).

References

- [1] J.Gubbi, R.Buyya, S.Marusic, et al.: Future Generation Computer Systems, Vol.29(7) (2013), p.1645-1660.
- [2] C.Perera, A.Zaslavsky, P.Christen, et al.: Communications Surveys & Tutorials, Vol.16(1) (2014), p.414-454.
- [3] L.Atzori, A.Iera, G.Morabito: Communications Magazine, Vol.52(1) (2014), p.97-105.
- [4] M.Kovatsch, M.Lanter, Z.Shelby: *Californium: Scalable cloud services for the internet of things with coap* (IEEE, USA 2014).
- [5] J.Soldatos, N.Kefalakis, M.Hauswirth, et al.: *OpenIoT: Open Source Internet-of-Things in the Cloud* (Springer, Germany 2015).
- [6] S.Bendel, T.Springer, D.Schuster, et al.: *A service infrastructure for the Internet of Things based on XMPP* (IEEE, USA 2013).