Architecting Service-Oriented Coordination Architecture for Internet of Things: a Publish/Subscribe-based Approach

BIAO DONG and JUNHUI CHEN

ABSTRACT

This paper presents a service-oriented approach for architecting Internet of Things (IoT) coordination architecture using ontology mechanism. Considering the real time interaction and the dynamic collaboration which are necessary for the application environment of IoT, a common and extensible issue-based resource model is defined to describe the IoT service ontologies and their relationships. A dynamic coordination mechanism based on publish/subscribe (Pub/Sub)-based service router is proposed to support multiple ontology trees fusion, route locating, message forwarding and routing. The analysis indicates that the Pub/Sub-based approach effectively realizes multiple service dynamic collaboration for the service-oriented application in IoT.¹

INTRODUCTION

The IoT application ends to integrate a large number of distributed enterprise information systems, and forms a large-scale complex service system, so large amount of sensed information must be collected and shared across multiple heterogeneous information systems. There are the objects such as users, business processes and physical entities in the application environment of IoT, and these objects have the characters of distributed, dynamic, heterogeneous, autonomous. These characteristics require that there is a dynamic, extensible and open service providing mechanism to reduce the coupling degree of the heterogeneous information systems to the greatest extent, realize dynamic collaboration between

¹Biao Dong, School of Computer & Software, Nanjing Institute of Industry Technology, Nanjing, 210023, China
Jinhui Chen, School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
the systems so as to quickly respond to changes in the physical world in real time. Traditional service-oriented architecture based on request/response is not a good solution to the problem of multiple service dynamic collaboration triggered by a complex event. However, the service provision mechanism of IoT must solve the real time interaction and dynamic collaboration between different business domains. Therefore, how to solve the problem of flexible distributed service collaboration across business domains and even across organizations is a technical challenge for the IoT.

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. The loose coupling between clients has a good effect on service-oriented architecture. By using event-driven mechanism, the numerous and independent IoT services can be coordinated to finish one task. This not only improves IoT application system’s real-time response capabilities, but also reduces the impact on existing application systems to the maximum extent. The Pub/Sub protocol supports service collaboration in dynamic environments where clients and their roles can change frequently, and this paradigm is a suitable paradigm for IoT application.

Guinard described the Web of Things architecture and practices based on the RESTful principles, and discussed several prototypes using these principles, which connect environmental sensor nodes, energy monitoring systems, and RFID-tagged objects to the Web[1]. Soldatos developed an IoT platform enabling the semantic interoperability of IoT services in the cloud, and provided a common standards-based ontology model for representing physical and virtual sensors[2]. Antonić presented an ecosystem for mobile crowd sensing which relies on the cloud-based Pub/Sub middleware to acquire sensor data from mobile devices in a flexible and energy-efficient manner and to perform near real-time processing of big data streams[3]. Bendel introduced a service platform based on the extensible messaging and presence protocol for the development and provision of services for such pervasive infrastructures[4]. Podnar presented the implementation of a Pub/Sub middleware system which is tailored to the requirements of mobile and resource-constrained environments with a goal to reduce the overall energy consumption in mobile crowd sensing environments[5]. Zarko formulated design principles for IoT data management methods and optimisation algorithms by means of pub/sub middleware and linked data which span over mobile networks and cloud infrastructures to produce a coherent IoT ecosystem[6].

The above researches effectively improve the efficiency of IoT application system development, but a key problem for further study is how to provide a flexible and agile service providing architecture for IoT environments in order to solve the real-time interaction and dynamic collaboration across business domains and
organizations. This paper proposes an architecture for designing and developing service-oriented coordination architecture for IoT. Our architecture supports the following features: A common issue-based resource model is proposed for IoT service ontology description. In terms of dynamic coordination mechanism, we use Pub/Sub-based service router which is responsible for multiple ontology trees fusion, route locating, message forwarding and routing.

**RESOURCE MODEL**

An important feature of IoT service is how to distribute, aggregate and share physical perceptual information in objective world between lots of heterogeneous business systems. One of the problems involved in this case is semantic interoperability of event. Through an IoT service system which provides complete decoupling in space, time and control flow, event is a common basis for collaboration and interaction between various business systems. Therefore, we need a standardized event model to handle a large number of different events. Event modeling is closely related to IoT resource modeling which involves a variety of domain concepts, terminologies and generic concepts. So, we use resource model to convert heterogeneous or proprietary information into unified events. Resource model can express resources, context information and domain knowledge. Based on resource model, ontology-based context information and events can be generated.

The ability to provide sensing and control functions for devices in IoT is abstracted as resource. We model the device class and the resource class in a 1: N relationship, where one device can hold multiple sensing abilities and a sensing ability corresponds to a resource. Resource model describes resource specific information, which includes related resource attribute information such as the resource ID, type, location, access method, etc. and the relationships between those resources. Fig. 1 shows issue modeling for service ontology description. The resource model consists of three layers, such as resource description layer, issue layer and service description layer. At the resource description layer, resource ontology includes not only information on resource function and technical specification, but also the non-functional information such as performance, working environment. In the issue layer, we extract the following dimensions of device, such as the measurement principle, the type of measurement, the measurement capability, the scope of operation and application, the application field, and the space, to construct the theme model. At the service description layer, service ontology describes data entity, scene environment which affects data entity, and some domain knowledge. In this model, a service ontology can be considered as a document that contains one or more issues.
The goal of the source model based on issue mapping is to decouple the up-level applications from the underlying resources. Resource ontology and service ontology are created by different business roles. A resource ontology has nothing to do with a specific business, but a service ontology is oriented to a specific application. The application developers only need to focus on data objects and application logics in application scenarios, and the underlying resource providers only need to describe resources and IoT status. By binding between resources and issues, as well as binding between issues and services, a service ontology is associated with one or more resource ontologies. When the underlying resources associated with the up-level services, the resources can observe or control the specific attribute values of data entity, so the originally meaningless device data have become the scenario data with specific application significance. The binding relationship between services and resources is loosely coupled, and its effectiveness is monitored. When a situation or state of data entity or resource is changed, a binding relationship can be dynamically created or disconnected. The advantages of issue modeling for service ontology are as follows. We can use this model for services that use different description languages. The model maps high dimensional resource to low dimensional issue to improve the computing speed of service ontology similarity. Service is automatically classified according to its issue.

THECOORDINATION ARCHITECTURE

Role

In the application of IoT scenarios, there are multiple non-centralized service publishers in different fields. According to different service domains, services are registered to different registration centers. We introduce the idea of service routing and form a service system framework based on semantic Web. The framework includes four roles: service subscriber, service publisher, service router and service registry. The functions of each role are described as follows. Service publisher
carries out the following works: service registration, service publishing, and service invocation. The main function modules of service publisher include ontology creation, service description and implementation, interaction management during service invocation. Service subscriber initiates service subscription and service invocation. The main function modules of service subscriber include service subscription, analysis and processing of result, and service invocation. Service router provides a semantic-based message delivery mechanism. According to specific service information, service router transfers related subscription information to appropriate roles for subsequent processing. According to the different service fields, the distributed service registries are designed specially. Service registry carries out the following works: manage the services in specific domains, respond to the service request, complete the service registration, and meet user’s service demand.

**Service Router**

A basic supporting platform of the architecture is composed of multiple service routers, which is responsible for multiple ontology trees fusion, route locating, message forwarding and routing. Service router consists of five parts such as controller, message processor, ontology manager, routing table and interface manager. The functions of each part are described as follows. The main function of the controller is to record the distribution and the connection mode of service registry, and to realize routing management, routing location and routing updating. The controller is responsible for service routing business logic, supports service routing, and provides users with an interface to operate the semantic Web. After the message processor receives data with a certain semantic message, parser module extracts semantic information from the data, and then the semantic information is submitted to the controller for further processing. The message processor carries out the following works: data resolution, data encapsulation, conversion between byte stream and the structure which can be understood by service router. Ontology documents in different fields are initially parsed into ontology trees separately by the ontology manager, and then according to ontology tree fusion algorithm, the ontology trees in different fields are fused into an ontology tree in order that the routing location algorithm of the controller can locate and distribute message accurately. The routing table is used to record the following information, such as the status information and the address information of the various roles that are connected to the service router, and the areas of interest. The interface manager is the basic module of service routing, which provides the message processor with an abstract underlying network interface, and a simple and uniform invocation style.

**Dynamic Coordination Mechanism**

In the application environment of IoT, we can adopt an event-driven architecture to collaborate with the widely existing heterogeneous business systems whose
functions are relatively independent. This means that the business systems do not have direct access to business logic in actual application. The systems standardize the resource model and service invocation through the event Pub/Sub mechanism in order to realize the information interaction of multi-source business systems. In this way, there is no an explicit and centralized service orchestration which defines the calling orders of Web service or any other services. Instead, an asynchronous and implicit event chain is used to drive the dynamic collaboration of different business systems. Service collaboration based on event driven is essentially the one-to-many, and many-to-many asynchronous communication paradigm, characterized by multiple services contributing and receiving events, with the event elements often interlinked across different service subscribers and service publishers. As a result, the service collaboration mechanism is helpful to reduce the coupling degree of the business systems across different service domains and even across different organizations, and forms a more flexible and agile service providing system.

CONCLUSION

In this paper, we propose a service-oriented approach for architecting IoT coordination architecture using ontology mechanism. The results show that the Pub/Sub-based approach effectively realizes multiple service dynamic collaboration for the service-oriented application in IoT.

ACKNOWLEDGEMENT

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

REFERENCES