RuCAS: Rule-Based Framework for Managing Context-Aware Services with Distributed Web Services

Hiroki Takatsuka, Graduate School of System Informatics, Kobe University, Kobe, Japan
Sachio Saiki, Graduate School of System Informatics, Kobe University, Kobe, Japan
Shinsuke Matsumoto, Graduate School of System Informatics, Kobe University, Kobe, Japan
Masahide Namamura, Graduate School of System Informatics, Kobe University, Kobe, Japan

ABSTRACT

Machine-to-Machine (M2M) systems and cloud services provide various kinds of data via distributed Web services. A context-aware service recognizes real-world contexts from such data and behaves autonomously. However, it has been challenging to manage contexts and services defined on the heterogeneous and distributed Web services. In this paper, the authors propose a framework, called RuCAS, which systematically creates and manages context-aware service using various Web services. RuCAS describes every context-aware service by an ECA (Event-Condition-Action) rule. For this, an event is a context triggering the service, a condition is a set of contexts to be satisfied for execution, and the action is a set of Web services to be executed by the service. Thus, every context-aware service is managed in a uniform manner. Since RuCAS is published as a Web service, created contexts and services are reusable. As a case study, RuCAS is applied to a real home network system.

Keywords: Context-Awareness, Event-Condition-Action Rule, Home Network System, Sensor Services, Web Services

INTRODUCTION

The recent spread of cloud computing and Machine-to-Machine (M2M) technologies allows us to acquire various kinds of data from heterogeneous and distributed systems (Velte, Velte and Elsenpeter, 2010; Wu, Talwar, Johnson Himayat and Johnson, 2011). The cloud computing provides computational resource and data as networked services, whereas the M2M enables devices to communicate with each other without human intervention. Typical data include temperature, power consumption, weather, system state, operation of a device. Data from the cloud or M2M systems can be obtained usually through Web services or Web-
API. Variety of data achieves a context-aware service (Cohen et al. 2004), which recognizes a real-world context and behaves autonomously for the context. The context-aware services implement smarter services, which are sensible for the environment and human activities.

Traditionally, the context-aware services had been studied in the field of ubiquitous computing (Randell and Muller, 2000; Gellersen, Schmidt and Beigl, 2002). Many studies were reported on context acquisition, context reasoning and utilization, using ubiquitous sensors deployed on local smart space. Now, the context-aware services must evolve so that the services can deal with global and distributed contexts obtained from Web services of heterogeneous systems (e.g. information services, sensor services, networked appliances, etc.). However, there are few studies adopting distributed Web services for creating context-aware services. In our previous work, we proposed a sensor service framework (Nakamura, Matsuo, Matsumoto, Sakamoto and Igaki, 2011), which invokes Web services based on contexts with physical sensors. However, the focus was limited on the sensors only.

Using Web services for inputs and outputs can significantly improve the functionality and flexibility of the context-aware services. However, a major challenge lies in managing complex relations among distributed data sources, defined contexts, and actions caused by the contexts. Unless managed systematically, the service provision would be quite difficult. Therefore, it is essential to have a unified framework for managing advanced context-aware services based on the heterogeneous and distributed Web services.

In this paper, we present a framework called RuCAS (Rule-based management framework for Context-Aware Services), which systematically creates and manages context-aware service using various Web services. The framework consists of five layers: Web service layer, adapter layer, context layer, action layer and ECA rule layer. The existing Web services for data acquisition are managed in the Web service layer. The data acquisition from heterogeneous Web services is adapted to the standard API in the adapter layer. In the context layer, every context is defined based on the data obtained via the adapter. Every Web service that is triggered by a context is managed in the action layer.

Using these elements, RuCAS defines every context-aware service as an event-condition-action (ECA) rule. For this, the event defines a context that triggers a service. The condition refers to a guard condition to execute the service. The action defines Web services executed by the service. Thus, every context-aware service is simply defined and created as a uniformed rule.

To see the feasibility of the proposed method, we conduct a case study that applies the RuCAS framework to creating the context-aware services in a practical home network system. In the case study, it is shown that a smart air-conditioning service with environmental sensors can be easily created by a sequence of RuCAS API.

**PRELIMINARIES**

**Context-Aware Service**

A context refers to a situational information (e.g. human activity, environment, etc.) derived from information of sensors and systems. A context-aware service is a service that automatically detects change of a context and performs appropriate actions corresponding to the context change. For instance, a context “Hot” can be derived from information that “the value of a temperature sensor in a room is higher than 28 degrees”. A context-aware service “Automatic Air-conditioning” starts air-conditioning when the context “Hot” holds.

Traditionally, the context-aware services had been studied extensively in the ubiquitous computing area. The conventional studies include a method that uses sensor information to reason contexts in a smart space, and a method that uses smart phone sensors to reason human behavior (Yamamoto, Kouyama, Yasumoto and Ito, 2011; Chon and Cha, 2011).
10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

www.igi-global.com/article/rucas/126616?camid=4v1


www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

Reducing the Complexity of Modeling Large Software Systems
www.igi-global.com/chapter/reducing-complexity-modeling-large-software/8242?camid=4v1a

Using Executable Slicing to Improve Rogue Software Detection Algorithms
Jan Durand, Juan Flores, Travis Atkison, Nicholas Kraft and Randy Smith (2011). *International Journal of Secure Software Engineering* (pp. 53-64).
www.igi-global.com/article/using-executable-slicing-improve-rogue/55269?camid=4v1a

User Interface Generation from the Data Schema
www.igi-global.com/chapter/user-interface-generation-data-schema/30020?camid=4v1a
Integrating Access Control into UML for Secure Software Modeling and Analysis
[www.igi-global.com/article/integrating-access-control-into-uml/39006?camid=4v1a](www.igi-global.com/article/integrating-access-control-into-uml/39006?camid=4v1a)