Computer Standards & Interfaces 34 (2012) 171–188

Contents lists available at SciVerse ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

User-configurable semantic home automation

Yung-Wei Kao [a,*], Shyan-Ming Yuan [a,b]

[a] Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu 300, Taiwan
[b] College of Computing & Informatics, Providence University, Taiwan

ARTICLE INFO

Article history:
Received 24 March 2011
Received in revised form 5 August 2011
Accepted 11 August 2011
Available online 8 September 2011

Keywords:
Smart home
Semantic home
Home automation
Web service
OWL

ABSTRACT

The ideas of smart home and home automation have been proposed for many years. However, when discussing homes of the future, related studies have usually focused on deploying various smart appliances (or devices) within a home environment and employing those appliances automatically by pre-defined procedures. The difficulties of supporting user-configurable automation are due to the complexity of various dynamic home environments. Moreover, within their home domains, users usually think semantically; for example, "I want to turn off all the lights on the second floor". This paper proposes a semantic home automation system, USHAS (User-configurable Semantic Home Automation System), which adopts Web Service and WSBPEL for executing automated process; OWL and OWL-S for defining home environments and service ontology; and a self-defined markup language, SHPL (Semantic Home Process Language), for describing semantic processes.

1. Introduction

Previous studies have focused on providing solutions for smart homes [1–4], home automation [5–7], and ubiquitous homes [8–10]. Numerous of these studies [11–13] concentrate not only on controlling devices, but also on combining additional factors before doing so. For example, facial recognition can be performed for opening a door, and hand gesture recognition can be utilized to change the channel of a TV. However, regardless of whether these home appliances are controlled manually by users or indirectly and automatically by computers, most systems provide only pre-defined control of appliances. For example, hand gestures indicating numbers for changing channels on a TV cannot be used as an input for changing the temperature of air conditioner, unless users modify the programs by themselves. Such modification requirement limits the flexibility and scalability of numerous home control and automation systems.

Controlling appliances without pre-defined procedures is difficult, as several requirements must be fulfilled. First, all devices, including home appliances, sensors, and remote controls, should be designed separately with standard interfaces provided. This criterion is reasonable, because users usually purchase devices manufactured by different companies. Second, how automation processes receive information from sensors and control devices should be standardized. Finally, these processes should be designed by users in an easy and understandable manner.

To provide interconnectivity between different parties, various Service-Oriented Architecture (SOA) [14] technologies have been designed. In general, numerous home systems adopted the Open Services Gateway Initiative (OSGi) architecture [15] for smart home implementation. However, the OSGi platform does not provide a complete process management and execution solution; therefore, pre-defined procedures for executing processes are usually designed for the OSGi. By contrast, Web Service [16] focuses on providing interconnectivity between parties. Moreover, the OASIS Web Services Business Process Execution Language (WSBPEL or BPEL4WS) [17] has been defined for process management and execution of Web Services. Therefore, this paper adopted the Web Service standard for defining operations of devices and the WSBPEL standard for defining automation processes.

Although WSBPEL provides an effective process definition standard, a gap remains between what users actually want and what they must define in processes. This is because users usually think semantically; for example, "I want to turn off all the lights on the second floor". In this instance, the WSBPEL process cannot be defined, because the home system does not know about the lights located on the second floor. To solve this problem, this paper adopted the Web Ontology Language (OWL) [18] to create a knowledge base in a semantic home, and the OWL-S [19] technology to describe the Web Services of the devices.

Although the OWL-S standard provides the functionality of process definition, the difference between the OWL-S process and the WSBPEL process is quite small. Only the input and output variables are mapped to the OWL ontology, and users must specify many binding details. To achieve the goal of semantic home automation, this paper adopts OWL-S only for semantic service description and

* Corresponding author.
E-mail addresses: ywkao@cs.nctu.edu.tw (Y.-W. Kao), smyuan@cis.nctu.edu.tw (S.-M. Yuan).

0920-5489/$ – see front matter
doi:10.1016/j.csi.2011.08.002

discovery, but not for semantic process definition and execution. Because no appropriate candidate is available for defining semantic home processes, this paper designed a markup language, SHPL (Semantic Home Process Language), to describe semantic processes. In addition, the SHPL execution runtime is developed, which dynamically generates a WSBPEL process for each semantic process based on the current home environment defined in the knowledge base.

This paper proposes a semantic home automation system, USHAS (User-configurable Semantic Home Automation System), which adopts Web Service and WSDL to execute automation processes; OWL and OWL-S to describe home environments and service ontology; and SHPL to define semantic processes. To prove that the proposed system can satisfy user needs adequately, this paper designed nine demonstration scenarios: the living room, long vacation, home gym, morning rush, dinner time, good student, sweet dreams, bath time, and party night. Furthermore, this paper designed and analyzed a questionnaire to determine which scenarios were more appealing to users. Finally, the usability of USHAS was evaluated.

The research contains seven sections. In Section 2, the backgrounds and related works of the proposed system are introduced. Section 3 discusses the design issues of USHAS. The USHAS ontology and SHPL are described in Sections 4 and 5 respectively. We present the system architecture and detailed system design of USHAS in Section 6. System demonstrations and evaluations of scenarios are presented in Section 7. Finally, we end up with a conclusion and discuss the future works of proposed system in Section 8.

2. Backgrounds and related works

2.1. Smart home and home automation

Smart homes and home automation are popular topics, referring to devices and appliances in the home environment that can be controlled automatically in an intelligent manner. Thus far, numerous studies have proposed system designs and even smart home products [1–4]. In smart home systems, devices and appliances are usually controlled to facilitate the duties of daily life. Home automation systems generally contain an internal network, as well as intelligent rules and devices in the home network for convenient or special purposes. Devices and appliances can be controlled automatically by, or provide environmental information to, these home automation systems. In addition, changing the state of a device may also change the state of another device, or trigger actions in other devices within the smart home environment [6].

2.2. SOA and OSGi-based smart home

Before defining a high level of logic for automation processes, the most crucial challenge is to facilitate the interconnectivity between different devices. Due to highly varied standards for home devices, such as X10 [20], INSTEON [21], UPnP [22], and Jini [23], communication between different devices is difficult to establish using dissimilar interfaces under various standards. For such heterogeneous network integration, the Service-Oriented Architecture (SOA) [14] design principle provides interoperability between various loosely coupled services. Open Services Gateway initiative (OSGi) [15] is one of the technologies that implement the SOA paradigm, and numerous researchers have implemented smart home systems based on OSGi. Li et al. [24] designed a home network system, providing a Web interface for users to control appliances directly on the OSGi platform. Ishikawa et al. [25] proposed SENCHA, a smart appliance integration middleware framework based on OSGi. They indicated several limitations of OSGi in implementing smart home automation, such as the lack of multiple views of abstraction levels. In other words, the capability of multilevel abstraction of OSGi is not enough. Wu et al. [1] combined the OSGi platform with mobile agents [26] in their design, which involves multiple mobile agents responsible for different tasks, distributed among multiple OSGi platforms. Liao et al. [8] adopted the Message-Oriented Middleware (MOM) [27] paradigm for event handling in their context-aware smart home system. Rui et al. [9] presented a physical structure model and a multi-agent [28] based software architecture based on OSGi; this architecture encapsulated the device sensing as well as control operations into the AmI-Adaptor, and encapsulated the computation logic into the AmI-Box.

The primary problem of OSGi is that only bundles installed on the same OSGi container can inter-communicate. Therefore, technologies such as mobile agents in [1] must be designed to establish communication between different containers. Another problem of OSGi is that no standard process definition is provided. Hence, in the OSGi based systems, high level device control decisions are usually made by multi-agents and pre-defined by programmers. As a result, automation processes are fixed and not allowed to be configured by users. Although the system designed in [9] provides the capability of user configuration to AmI systems, the configuration level is at the AmI-Adaptor and AmI-Box levels, not at the process level; users must select and configure which AmI-Adaptor or AmI-Box to communicate.

2.3. Web Service, WSBPEL, and Web Service Based Home Automation

Web Service [16] is another technology that implements the concept of SOA, consisting of three main standards: Web Services Description Language (WSDL) [29], Simple Object Access Protocol (SOAP) [30], and Universal Description, Discovery and Integration (UDDI) [31]. Similar to OSGi, Web Service provides interoperability between different services; therefore, it is also a candidate for smart home platforms. Uribarren et al. [32] proposed a middleware system based on Web Service for controlling devices with different protocols. Unlike OSGi, a process execution standard, Web Services Business Process Execution Language (WSBPEL) [17], is designed to support process definition for executing Web Services. Because devices are usually provided by different manufacturers, the concept of device functionalities can be mapped to services provided by enterprises; and the concept of process execution using different devices can be mapped to cross-business process execution, including different enterprises. Anke et al. [33] revealed the drawback of OSGi: OSGi bundles are not directly accessible from clients outside of the OSGi container. To solve this problem, Anke et al. designed a system that exposes OSGi bundles using Web Service interfaces, and then executes these bundles using processes defined in WSBPEL. However, using both OSGi and Web Service involves a duplicate design, because both of them follow the SOA paradigm. Systems supporting WSBPEL process definition and execution can adopt device drivers directly implemented by Web Service, instead of by OSGi bundles, to reduce system overhead.

2.4. Semantic Web, Context-Aware Home, OWL-S, and Semantic Home Automation

Although WSBPEL is already a higher layer of both Web Service and OSGi, it is still difficult for users to write a WSBPEL document directly by themselves; many details of static binding, such as portTypes or URLs of services, must still be provided. To enable communication based on semantic ontology between different programs across the boundaries of different organizations, Semantic Web [34] technology is designed on the basis of Resource Description Framework (RDF) [35] and Web Ontology Language (OWL) [18]. In general, Semantic Web technology is usually used for context-aware home automation systems. Wang et al. [36] designed the CONON ontology for context reasoning in pervasive computing environments, including home environments. Moreover, several reasoning engines, such as OWL-QL [37], have been developed for understanding semantic meanings of OWL, and can be highly useful in semantic home