Towards Automated Service Composition using Policy Ontology in Building Automation System

Son N. Han, Gyu Myoung Lee, Noel Crespi
Department of Wireless Networks and Multimedia Services
Institut Mines-Telecom, Telecom SudParis
91011 Evry Cedex, France
{*son.han, gm.lee, noel.crespi*}@*it-sudparis.eu*

***Abstract*—Automated service composition is critical for successfully implementing Building Automation Systems which facilitate Web services over physical devices. In this paper, we propose a new service composition model using a policy ontology represented by Semantic Web languages. Our proposed model can be a powerful tool to automatically compose services with associating meaningful policies.**

***Keywords*-service composition; semantic web; policy; ontology; building automation system;**

I. Introduction

In building automation industry, there are several communication protocols and a variety of products from various vendors available on the market. It is a challenge for coordinating heterogeneous devices or appliances, in which, topologies, communication protocols, security policies, and such are dynamic and different from one to another. An approach from the Internet of Things (IoT) paradigm is to make physical devices accessible as Web services that are identified by Uniform Resource Identifiers [1]. Many innovative architectures for automation systems based on the concept of Web services and Service-Oriented Computing have been proposed [2]. In order to successfully deploy Web services for automatic control systems like Building Automation System (BAS), we need an automatic mechanism to combine separate services into a composite one to meet the diversity of user's requirements. This can be achieved by a process called automated Web service composition. Semantic Web [3] visions the possibility of semantically understanding between services. So that practical and powerful applications can be written by using semantic annotations and suitable inference engines to automatically compose Web services.

So far, there have been several efforts to use semantics for BAS by designing ontologies [4], [5]. However, these approaches aim at different frameworks of BAS rather than Web service-based. In [4], a comprehensive ontology on the domain of BAS was proposed. However, BAS ontology with knowledge and terms for describing BASs makes a little impact on the process of service composition that is usually governed by composition rules.

In this paper, we propose a model for automated service composition using a policy ontology that is represented by Semantic Web languages. Policies are means to dynamically regulate the behavior of system components. By changing policies, a system can be continuously adjusted to accommodate variations in externally imposed constraints and environment conditions. The adoption of a policy-based approach for controlling a system requires an appropriate policy representation. In our approach, we aim at representing policies semantically by Semantic Web languages, putting them all together in a Building Automation Policy Ontology (BAPO). In the following sections, after outlining the BAPO-based service composition model, we present a design of BAPO and an algorithm for Workflow Generation.

II. BAPO-based Service Composition Model

Figure 1 shows the main components of BAPO-based service composition model. User gives input requirements semantically into User Requirements unit and the system automatically generates an output composite service and then store it back in the Services Repository. BAPO is written in Web Ontology Language (OWL) [6] to describe policy concepts and relationships between them, which are represented in classes and properties respectively. BAPO covers all necessary types of policies and is updated constantly. Workflow Designer consists of two parts: Workflow Generator is for automatically generating service workflow with reference to Workflow Templates database and Workflow Editor is to assist users for designing workflow or

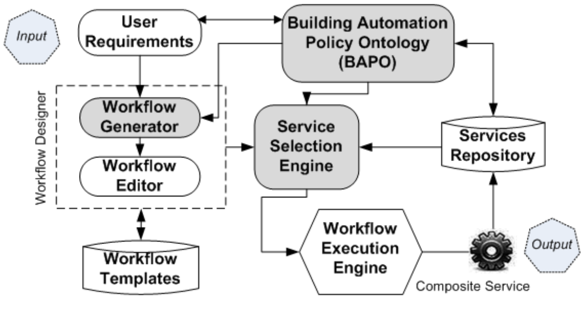

Figure 1. BAPO-based Service Composition Model

correcting workflow generated from Workflow Generator. The workflow created in Workflow Designer is stored in Workflow Templates database for future use. Service Selection Engine is the implementation of selection algorithms to automatically select an appropriate service among available ones from Services Repository. In Services Repository, services are described by Semantic Annotations for WSDL (SAWSDL) [7]. SAWSDL provides semantic descriptions of policies of Web services in service profiles. The profiles are therefore used for advertising registry, discovery, and matchmaking for the Service Selection Engine to find the service that meets required policies.

III. BAPO Design and Workflow Generation Algorithm

A. A BAPO Design

BAPO consists of information about classes and properties in the domain of building automation policies. Classes describe concepts of policy and properties of each class describing various features and attributes of the concept. For example, in Figure 2 class *UserPreferencePolicy* has five subclasses with *is-a* relationship. That means all properties from *UserPreferencePolicy* are included in subclasses. We designed BAPO version 1.0 by Protege ontology editor software [8]. Figure 2 partly shows the class hierarchy of BAPO with some of its classes without their properties. All classes are derived from the root OWL class $owl : Thing$.

B. A Workflow Generation Algorithm

We made an algorithm for automatic generation of a workflow following user's requirements. User can use a Web-based application to give semantic requirements that are instances of BAPO classes. By representing policies in BAPO, the following algorithm can easily generate the desired workflow.

Definition: *Semantic Requirement*
A semantic requirement is an instance of a class in BAPO

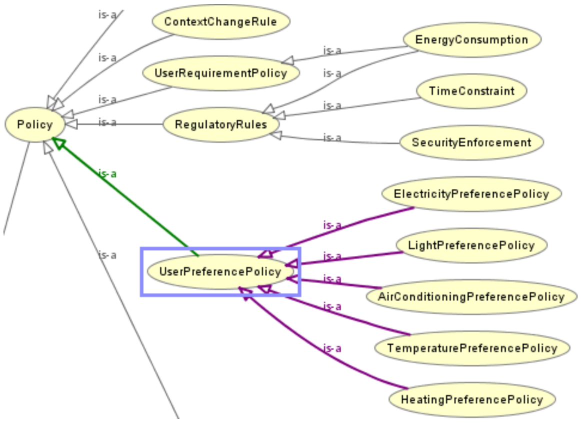

Figure 2. BAPO 1.0 Class Hierarchy

Algorithm: *Workflow Generation*

```
Input: User $U_k$ semantic requirements $\{sr_1, sr_2, ..., sr_n\}$
Output: Service workflow $\{s_1, s_2, ..., s_n\}$
Given user BAPO and Services Repository

BEGIN
Denote $\{C_1, C_2, ..., C_n\}$ are concept classes of semantic requirements
For each $C_i$ find a list of services $\{s_{ij}\}$ that satisfy $C_i$ and user $U_k$ AuthenticationPolicy
Repeat
- Pick one in each list $s_{ij}$ to form a set
- Match them with ServiceLocationPolicy
- Continue if there is ServiceLocationPolicy conflict
Until exist one perfect match.
END.
```

For example, one employee working in an office at the building want to remotely turn on heater and light in his office 10 minutes before his arrival. He can have access to the Web-based application and semantically ask for such composite service.

IV. Conclusion and Future Research

We have proposed a preliminary model for automated service composition in BAS. We have already built the first version of BAPO and implemented an algorithm for workflow generation. We still have a substantial workload to fulfil including continuously updating the BAPO, designing related algorithms for Service Selection Engine, developing a prototype system as well as the Web-based application with a graphical user interface.

References

[1] A. Kamilaris, V. Trifa, and A. Pitsillides, "Homeweb: An application framework for web-based smart homes," in *Telecommunications (ICT), 2011 18th International Conference on*, may 2011, pp. 134 –139.

[2] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," *Industrial Informatics, IEEE Transactions on*, vol. 1, no. 1, pp. 62 – 70, feb. 2005.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, pp. 29–37, May 2001.

[4] C. Reinisch, W. Granzer, F. Praus, and W. Kastner, "Integration of heterogeneous building automation systems using ontologies," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, nov. 2008, pp. 2736 –2741.

[5] S. Runde, H. Dibowski, A. Fay, and K. Kabitzsch, "A semantic requirement ontology for the engineering of building automation systems by means of owl," in *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, sept. 2009, pp. 1 –8.

[6] "Web ontology language." [Online]. Available: http://www.w3.org/2004/OWL/

[7] "Semantic annotations for wsdl and xml schema." [Online]. Available: http://www.w3.org/2002/ws/sawsdl/spec/

[8] "Protege." [Online]. Available: http://protege.stanford.edu