Model Driven Service-Oriented Approach for Smart Building Energy Management

[1]Ding-Yuan Cheng, [2]Nasaraf Shah, [3]Chen-Fang Tsai
[2]Kuo-Ming Chao and [1]Chi-Chun Lo

[1]Institute of Information Management, National Chiao-Tung University
300, Hsin-Chu, Taiwan
[2]DSM Research Group, Department Computing and the Digital Environment
Coventry University Coventry, CV1 5FB, U.K.
[3]Department of Industrial Management, Aletheia University, Taiwan, ROC
cclo@faculty.nctu.edu.tw, kewas@iim.nctu.edu.tw
n.shah@coventry.ac.uk, k.chao@coventry.ac.uk
au1204@mail.au.edu.tw

Abstract. This paper describes a new framework which is featured by integrating two well-known system modeling methods namely model driven approach and service-oriented architecture to facilitate web service development and to effectively satisfy a group of service consumers' subjective requirements and dynamic preferences. These models can be combined at different levels and configured into heterogeneous structures to form various building blocks for development of service-oriented applications. A case study is used to illustrate how to systematically use our proposed framework to construct a smart service-oriented environment system for effective energy management.

1 Introduction

Service-oriented computing has become an important trend in IT development. Especially, the recent convergence of ubiquitous computing and context-aware computing in an attempt to integrate numerous types of sensors, heterogeneous communication protocols, and programming languages to produce an effective and efficient solution to the design of a distributed smart environment has seen as a considerable challenge. A number of methods, frameworks and tools to design a smart environment using service oriented approach have been proposed in the past [1][2][3]. In general, the aim of these systems is to maximize users' comfort level, and minimize the cost of software design, code implementation, application installation, and system maintenance. The communications and interactions among sensors, objects, and human participants are inherently complex, as they involve different protocols and languages. It can be viewed as configuration problem that requires various components to work together in cooperative and coordinated manner in order to produce an optimised environment to meet the system requirements. Service-Oriented Architecture (SOA), which offers a way to take sensors, objects, participants as services, can provide an effective approach to improve their communication and coordination, so the human

Cheng D., Shah N., Tsai C., Chao K. and Lo C.
Model Driven Service-Oriented Approach for Smart Building Energy Management.
DOI: 10.5220/0004465801160130
In *Proceedings of the 4th International Workshop on Enterprise Systems and Technology (I-WEST 2010)*, pages 116-130
ISBN: 978-989-8425-44-7

participants can effectively interact with the system and offer their opinions and preferences through a coherent technology.

A web service can be invoked and executed by a number of consumers simultaneously. Most service discovery and recommending systems recommend a list of services according to the functional or non-functional requirements given by service consumer. Thus, the service consumer can choose the best fit based on a set of criteria. This kind of cases areis based on an assumption that is the cardinality between an instance of service and a service consumer is built upon one to one relationship. However, in some cases such as smart environment, an instance of a service needs to serve a number of consumers. In other words, an instance of a service needs to meet various requirements specified by different service consumers. It does not only satisfy one consumer's needs, but also of most consumers. It, however, can be very difficult when conflicting requirements occur. In order to address the above issue, services should be able to conduct self-adjusting and self-organizing to meet the requirements or preferences given by most consumers along with changing environments.

Hence it is important that the design of service should consider the user preferences. The configuration or composition of services should also take the environmental factors into account. So the system needs to adapt to environment changes by adjusting or configuring its services dynamically in accordance with the data sent by sensors. For example, the staffs in one office use a collection of the same devices or functions provided by the system. Each function is intended as a service such as light service, air condition service, heating service, and ventilation service, etc. Service consumers could have different preferences or requirements on the services such as room temperature. Some users may subjectively feel hot for the current room temperature, but some may feel cool or comfortable. How to set the temperature for air condition/heating service according to these consumers' preferences can be a challenging issue. Existing research works in the context-aware system do not sufficiently address this issue. To increase interoperability among these autonomous services is required, so these services need to have well-defined interfaces and functions. The context-aware system would encompass the capacities such as group consensus reaching mechanism and service self-organizing mechanism to improve system adaptability.

In this study, we design an intelligent context-aware service system based on Service Component Architecture (SCA) and Model Driven Approach (MDA). It combines these two methods to build a ubiquitous computing environment. The goal of this framework or system is not only to effectively integrate sensors and services but it also provides a systematic and adaptive approach to construct services in a dynamic environment.

The remainder of the paper is organized as follows. In Section 2 the awareness management process is outlined along with the basic notions required to model the problem domain. Section 3 discusses a SOA platform. We identify the basic services and briefly discuss the basic components of the platform. Then we present an example to illustrate how smart service in an environment involving human users can be used to capture and convey information. Finally we conclude the paper.

2 Related Works

In this section, we review some existing studies and research work in relation to SOA standards and middleware.

Although smart home and smart office concept have recently attracted a great attentions from research community and industry in Europe and United States, a number of Smart Home projects within the past few years [4], [5], [6] have been carried out. Also, AT&T Laboratories at Cambridge [7] built a dense network of location sensors to maintain a location model shared between users and computing entities. Microsoft's EasyLiving [8] focuses on a smart space that is aware of users' presence and automatically adjusts environment settings to suit their needs. More recently, Hewlett Packard's CoolTown [9] provides physical entities with "Web presence" and allows the users to navigate from the physical world to the Web virtual environment by picking up links to Web resources using various sensing technologies. These projects have provided great insight into smart building and environment, but we are particularly interested in three different issues of these technologies, namely service selection, consensus formulation method and service modeling.

2.1 Web Services

Many researchers have focused on approaches or methods on improving service discovery protocols [10][11][12][13][14] to increase accuracy in service discovery and selection. One of the most widely used protocols for publishing service is Universal Description, Discovery and Integration (UDDI) that includes service registry with explicit specifications to enable service advertisement. In our previous studies on the UDDI specification [15], we use the tModel to represent the QoS for web services composition. After that, several researchers have designed the semantic query mechanism into UDDI registry, and mapping RDFS upper concepts to UDDI data model to increase the precision in service selection such as [16] [17] [18].

Services can be classified into atomic or composite [19] services. An atomic service is a basic unit which cannot be decomposed further. A composite service is made of a collection of existing services which can be atomic or composite. Many research works on service composition have used workflow-based approach, Artificial Intelligence (AI) planning, and other modeling methodologies to compose services. In the workflow-based composition methods, [20] proposed composite service definition language (CSDL) to reform the flow. They used a static work flow generation in their proposed platform. Some researchers presented Semantic Web Service composition method based on Model-Driven Architecture (MDA) [20] [22], [23] [25], and UML [21]. These composite services are specified using standard UML model to generate specifications and to produce applications using MDA concepts.

2.2 Consensus Forming Methods

Decision making is one of the most complicated administrative processes in management. In a decision-making process, decision makers may encounter multiple criteria for evaluation. Therefore, Multiple Criteria Decision Making (MCDM) is one of the most well known branches in decision making. MCDM can be divided into two categories: MODM and MADM. A further discussion about MODM and MADM can be found in Yoon and Hwang [26][27]. MADM has been widely used by decision makers in management processes to evaluate and rank possible alternatives.

In a decision making processes, a group of decision makers could be involved and it is called Group Decision Making (GDM), so the all members' options or preferences have to be considered. Most of the GDM problems have strategic dimensions and can be complicated due to their multi-criteria framework involving many subjective and quantitative factors. Optimal utilization of the time and resources is a key element sought by many GDM methods. Various researchers have focused their efforts on increasing the ability of making quality group decisions[28, 29, 30, 31, 32, 33, 34, 35, 36, 37].

An effective web service discovery mechanism should be able to search and assess services based on their QoS and service contents as well as users' requirements. The service assessment or selection often involves multi-criteria decision-making process [38]. So, the GDM is applicable to service selection when the service consumers have inconsistent or conflicting requirements, as it can be considered as a reasoning process for reaching group consensus on their requirements for web service selection.

2.3 Web Service Modeling Methodology

There are several ways to compose services at the design time. Model-driven Architecture (MDA) [39] is a software architecture framework proposed by the Object Management Group (OMG). MDA consists of a set of standards that assist the system in creation, implementation, evolution and deployment. The key technologies of MDA are Unified Modelling Language (UML), Meta-Object Facility (MOF), XML Meta-Data Interchange (XMI) and Common Warehouse Metamodel (CWM).

MDA emphasized the importance of modeling for the software architecture design. MDA suggests a three-layered approach. The Computation Independent Model (CIM) describes a system from the computation-independent point of view to address structural aspects of the system. The Platform Independent Model (PIM) defines a system in terms of a technology-neutral virtual machine or a computational abstraction. The Platform Specific Model (PSM) consists of a platform model that captures the technical platform concepts and a model geared towards the implementation technique. The lifecycle of MDA development is shown as Fig. 1.

In [40], context-awareness is an essential aspect for service utilization, especially when frequent interactions take place between users and environments. In this paper, a solution for developing context-aware web services applications is proposed. The methodology includes a model driven approach to separate the web application functionality development from the context adaptation at the development phases (analysis, design, and implementation). In essence, context adaptation takes place on top of the web application business functionality to facilitate system evolution.

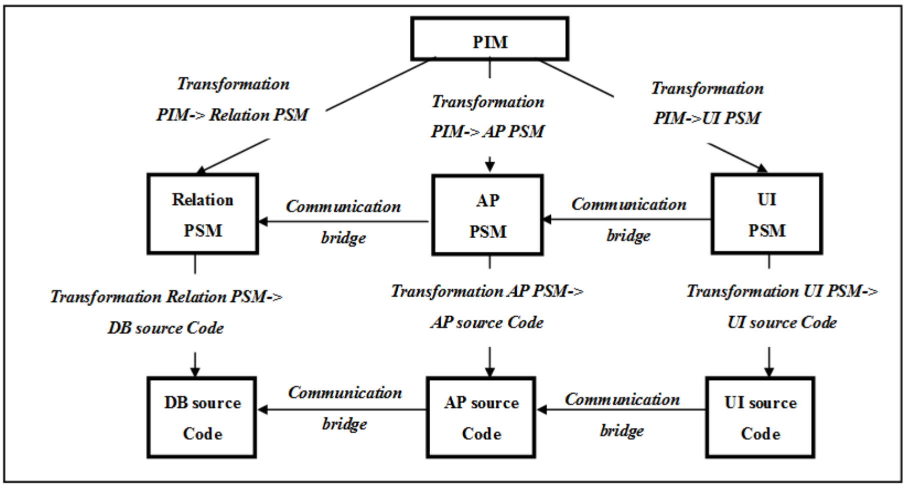

Fig. 1. Lifecycle of MDA Developemnt.

3 Design Framework

Model-Driven Architecture (MDA) is a well-developed concept for software design modeling and implementation. The meta-model plays an important role in MDA. Also, Meta Object Facility (MOF) is the kernel of transformation between different MOF layers. The Unify Modeling Language (UML) is the most widely used in software engineering. The key task of UML modeling is to identify the class, attribute, operation, and their relations. A class might inherit from another class. Moreover, a class can also have many functions which are called operation or method. Therefore, the design concept is not fully applicable to Service-Oriented applications because services are loosely coupled operations or functions and there is no inheritance property and internal facility to store states in services.

Hence, in this work, we proposed an agile SOA modeling methodology which combines MDA and Service-Oriented modeling methods. The main goal is not only to speed up the intelligent system development according to the SOA principles, but also be able to identify group consensus on service's QoS requirement and contents in order to maximize their users' satisfaction. In SOA, a required function can possibly be satisfied by multiple services. Therefore, the process of service selection and discovery needs to consider functional as well as non-functional (i.e QoS) requirements. The characteristic of our proposed methodology allows the services to have ability of self-adjustment in the process of composition in line with emerging requirements or environment states.

3.1 Overview

There are four kinds of models in MDA. During initial phase, the business analyst analyzes the user requirement in CIM. Next, the PIM based on the outcomes derived from the previous phase to define the functionalities, the structure, and the behavior of a system. The PSM focuses more on implementation and execution platform which could be a specific programming language or database.

In our propose framework, we distinguish two types of requirements which are functional requirement and comfort requirement in initial phase. After analysis phase, the users can use any tools or modeling language which they are familiar with to model the ststem. Here, we adopt the service-oriented modeling framework (SMOF) as a modeling framework. In PIM to PSM phases, we use Service Component Architecture (SCA) as transforming methodology. Also in PSM phase, we use the Service Data Object (SDO) to manipulate the connection between application and the database. Our proposed framework is based on MDA that includes SPEF and MOF. The overall architecture is shown in Fig. 2.

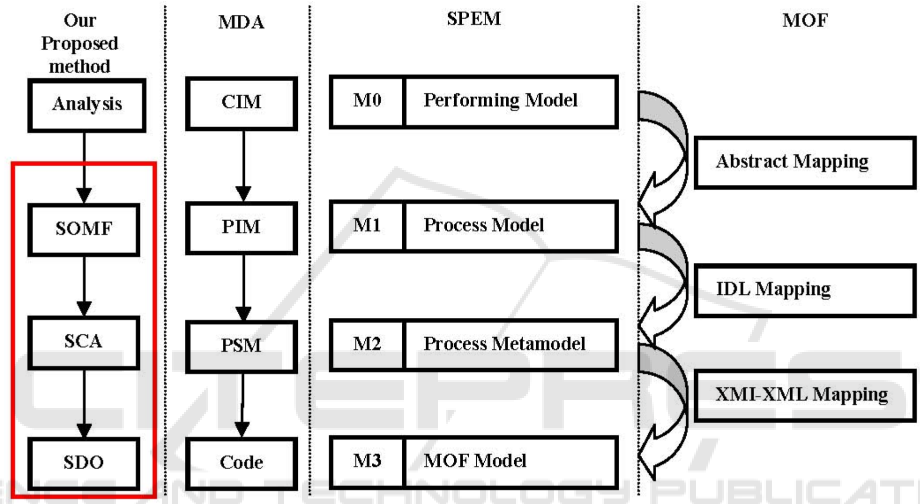

Fig. 2. Life-cycle of our proposed MDA service-oriented approach.

Adopting our proposed methodology to design context-aware systems has the following advantages:

(1) Services-Oriented Modeling: It can reduce the deficiency of object-oriented modeling in service-oriented applications, as UML meta-model cannot provide the necessary support. In service-oriented environment, software and hardware can be represented as services. Services are more transparent and loose coupled, which contain a collection of independent functions or operations as compared to objects. Objects heavily reply on their interdependencies and their internal states to operate.

(2) Autonomous Behavior: A context-aware system based on SOA possesses abilities such as autonomous adjustment, autonomous management, and autonomous deployment to satisfy diverse requirements from multi-user. The group consensus approach collects the preferences from the users and reason over them to provide a basis for system self-adjustment in order to meet the majority of users' requirements.

(3) Annotating sensor data with semantics: Sensor data could be value of temperature, humidity or an expression representing other conditions, but this data could imply a

condition such as light brightness or weather. The sensor devices and their sensed data can be grouped together to become services and annotated with semantics for reasoning.

(4) Information Streams Fusion and resource description: The resource including data, services, computation resource, and device profile will be described explicitly with their location and characteristics. This can benefit locating, allocating and re-deploying resources.

3.2 Top-down Modeling Analysis

Our proposed Model Driven Service-Oriented Approach (MDSOA) provides a top-down modeling analysis method. As mentioned above, we combine MDA concepts in the software design and system implementation with web services. The requirements can be separated into two types: functional requirement, and comfort requirement. Functional requirement defines scale, quantity, and function of all hardware and software. For example, the number of lights, air-condition units, dehumidifier or heater, etc. The comfort requirement is related to users' preferences which are about QoS.

The requirements collected from the previous CIM phase lay the foundation for modeling the required services. An analyst can use any modeling language such as UML to model the requirements. Because some of the service-oriented features cannot be satisfied with UML, it requires another modeling language to specify service flows, service relations, and service capabilities. Hence, we adopt Service-Oriented Modeling Framework (SOMF) [41] in the PIM phase. SOMF is a discipline of modeling business functions and system behaviors based on services.

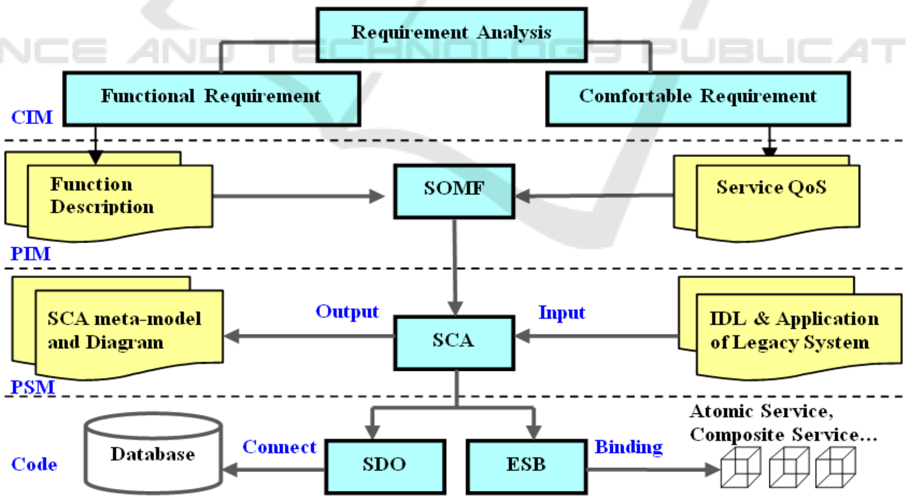

Fig. 3. MDA Mapping Architecture of Top-Down Modeling Analysis.

In PSM phase, the main task is to draw SCA diagram and obtain a system meta-model. After that, the SDO (Service Data Object) and ESB (Enterprise Service Bus) can connect to database and bind services together. SDO aims to provide a consistent means of handling data within applications, regardless of its source and format. It provides a unified way of handling data of databases and services. ESB is used to integrate applications, coordinate resources, and manipulate information. The proposed architecture is depicted in Fig. 3.

3.3 Button-up Assembling Analysis

From the top-down modeling analysis, we can analyze the system development life cycle from abstraction stage to implementation. However, the system should be adaptable and self-manageable according to changes occurring in the environment, so it is able to take the users' feedbacks and environment changes for system adaption. Hence, we proposed a bottom-up analysis to increase the system's capability in adaption to the environment.

We adopt the SCA assembly model which deals with the aggregation of components and their linkages. The assembly model is independent of implementation language. SCA is a set of specifications which is used to build applications and systems by deploying new service and composing existing components. SCA does not only extend and complement prior approaches to implementing services, but also provide a programming model for building applications and systems based on a SOA.

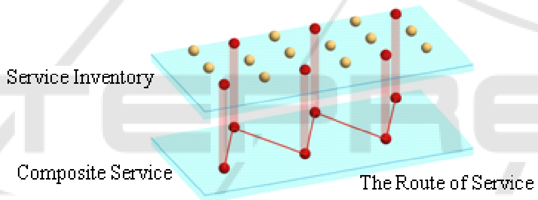

Fig. 4. Bottom-Up Assembling.

The business functions are supported by a series of services which are assembled together to support a particular operational need. The assembly model deals with the aggregation of components and the linking of the components. These services run on the server containers. A service inventory includes a number of server containers.

4 Examples

In this section, we will demonstrate how MDA and SCA approach can be used to develop smart building services for building energy management system. Furthermore, we adopted TOPSIS to resolve conflicts in group decisions.

4.1 Architecture

In the proposed smart office system, each device has a unique ID and it provides specific service. For example, the motion sensor detects object movement within its designated range. The temperature sensor measures temperature and humidity. These sensors can be considered as an atomic service. All the service descriptions in smart office can be stored in a registry. The overall architecture is shown as Fig 5.

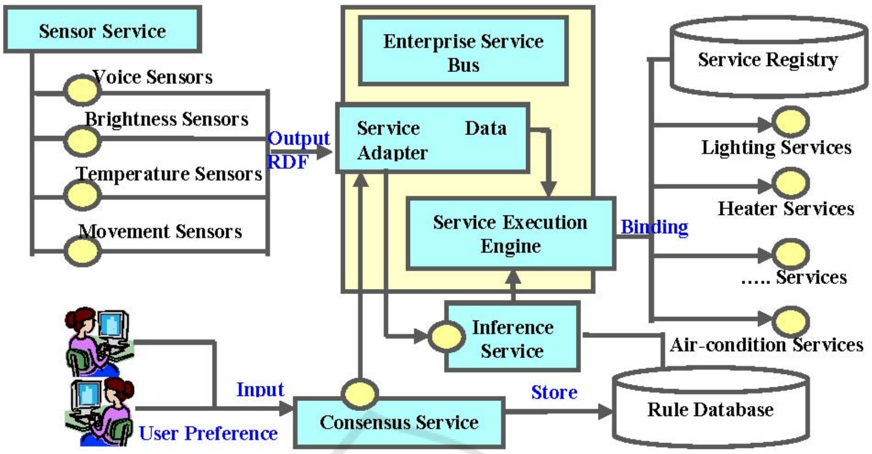

Fig. 5. System architecture.

(1) Services Repository. The aim of semantic web service is to locate services automatically based on the functionalities provided by the web services. These services are registered in a service repository which is UDDI. Therefore, we use the JUDDI to build UDDI environment which provides Business Entities, Service Entities, Binding Templates, and tModels to represent the business details and its services. Services registered with in JUDDI can be searched by name, location, business, bindings or tModels.

(2) Consensus Service. We design a consensus service which is based on the TOPSIS method to reason over a group of users' preferences to identify their potential agreements. These preferences can be very subjective and inconsistent and they could be represented in different ways. The preferences can be associated with uncertainty, fuzziness, and incompleteness. The consensus service is able to identify common requests from the majority of users and made recommendations to the device service to adjust the devices such as the light or air-condition services.

(3) Service Execution Engine. Service Execution Engine (SEE) provides a run time environment for service binding and execution. We use an open source system, Tuscany, which is an Enterprise Service Bus (ESB) for our SEE. SEE is not only a web server container but also can forward the message to other web services accordingly which are deployed on the other server. For achieving goal of service execution, we use the Synapse, which is an open source project as a routing server on Apache web server. Synapse provides a simple, lightweight and fully open source SOA infrastructure to assemble and manage composite applications as well as route messages. Synapse supports HTTP, SOAP, SMTP, JMS, FTP and file system transport for message exchange using XSLS, XPath and XQuery to bind the web services. Fig 6 shows the service execution engine architecture.

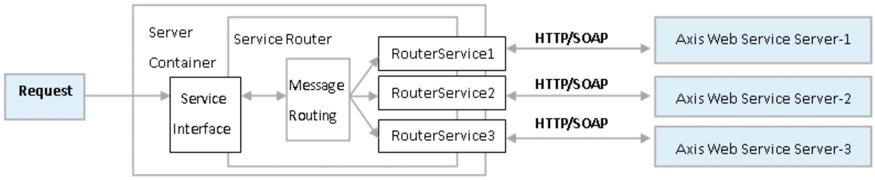

Fig. 6. Service Execution Engine architecture.

(4) Rule Database. The rule database is used to store the expert rules and user preferences. Moreover, we used a tool kit, FuzzyJ, as the fuzzy inference engine in our prototype system. The inference engine contains essential pre-defined knowledge for interpreting and classifying the information e.g., Very Cold, Normal, Warm, Hot, Very Hot. It consists of primitive and composite fuzzy terms, modifier and quantification of fuzzy terms, and fuzzy rules. Primitive terms are a set of atomic terms that represent a collection of raw data collecting by service broker.

(5) Sensor Services. The sensor device is associated with a sensor service which collects and senses the environmental parameters. Sensor service transmits the sensor data to service execution engine. The description of service is represented in RDF. The sensor data in stored in XML format can be parsed consistently by other services.

4.2 Module Design

The prototype system is a smart office system and it is designed by following SCA principles. We designed several composite services which are called SensorComposite, InferenceComposite, UserOpionComposite, ConsensusComposite and DeviceComposite. The SensorComposite is composed of several components which include FireSensorComponent, SmogSensorComponent, VoiceSensorComponent, MovementSensorComponent, TemperatureSensorComponentand BrightnessSensorComponent. The SensorComposite is responsible for all kinds of sensor services. All sensor data is transmit to InferenceComposite to evaluate whether the change of the device parameters such as brightness of light is necessary or not. Another important mechanism is related to handling the group preferences. The UserOpinionComposite is used to collect users' preferences. The feedbacks and preferences from users are sent to ConsesusComposite to calculate and obtain their consensus if there is any. The result of group consensus is then sent to the DeviceComposite.

The DeviceComposite would also receive the sensor data and preferences from sensors and users. All the data and signals are stored in XML-format file. We model these composite services by using SCA diagram which is shown in Fig. 7.

The system gathers environment related data to generate appropriate information for use. For example, a user may leave his seat, and move to conference room. The motion detection sensor would sense the situation, and the light will be turn off. The patterns of interactions among these appliances and their controls can be driven by analyzing sensor data. Thus, the detailed workflow design or relationship between user and appliance are not necessary to be prescribed at design time. Hence, the proposed framework includes a SCA approach which consists of service-oriented modeling and assembling mechanism. Sensor and intelligent mechanisms can be considered as services. It can not only activate component automatically, but also improve the service capability on semantic interpretation and to reuse the existing service to build new composite services to meet new requests.

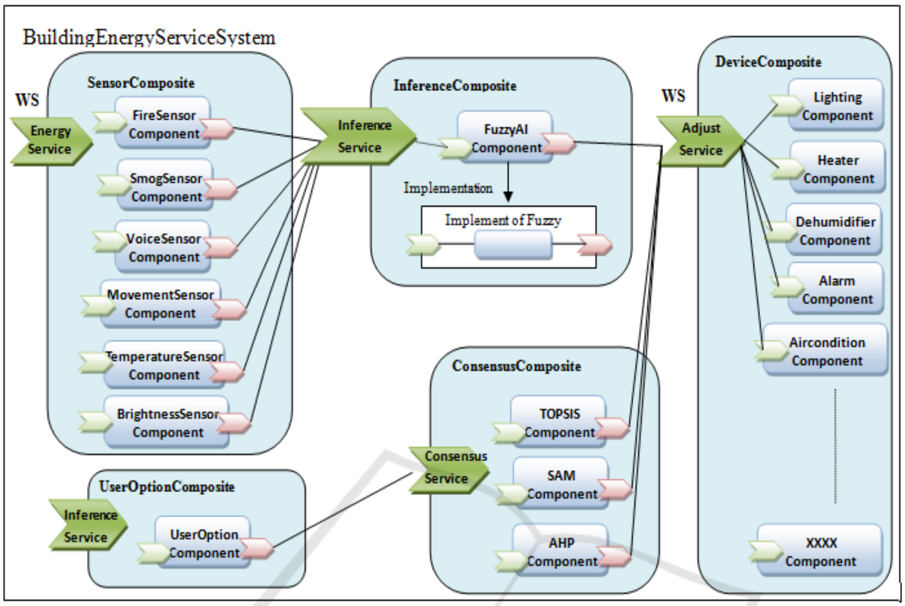

Fig. 7. Service Composite based on SCA.

4.3 Implementation Details

We use a sequence diagram to help readers understand how services are activated and adjusted to meet users' requirements. The system has sensors to detect brightness, voice and temperature and the corresponding devices to control them. The sensor service transmits the sensor data to ESB. The ESB binds the inference service with rules to reason over data and pass the outcomes to the device services. The device services control and adjust the devices by giving appropriate commands and parameters (such as dimmable light and air-condition).

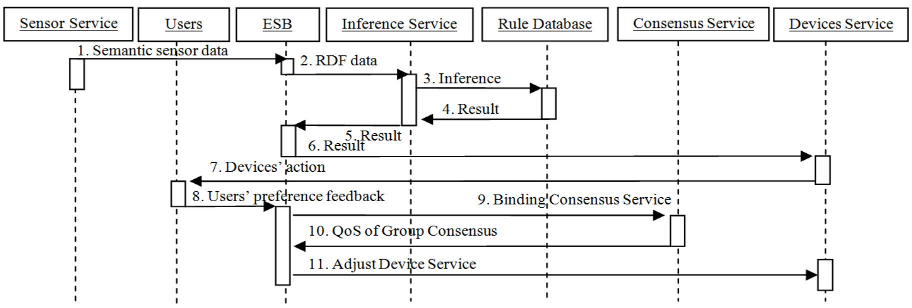

Fig. 8. Sequence Diagram of service support the self-adjust between ESB and users.

If the users feel too hot or gloomy, they can make requests to adjust the light or air-condition. These requests are collected and submitted to the consensus service. Since not every user wants to change the lighting or temperature and they may have conflicting requests or preferences, the consensus service will reason over these data to produce the recommendations which satisfy most users' preferences or requirements. If there is any change required in the environment, the device services will set new parameters and send commands to the device such as light and air-condition according to the recommendations.

5 Discussion

A number of research projects have been established in an attempt to solve the issues associated with context-aware systems. Although sensing and perceptual technologies have been increasingly recognized as key methods to develop smart environment, the main problem is that a group of users might have different requirements in the same space. Achieving context-awareness needs to take into account the variations present in the environment and users' opinions. The system does not only analyze the system functional requirements, but also need to consider the users' preferences.

Combining SCA and MDA could provide an adaptive platform to develop an autonomous adaptation system. The main characteristic of SCA is that it supports declarative foundation which enables it to access and compose services of diverse appliances. Thus a context-aware system can use semantic annotations to locate and bind services dynamically.

6 Conclusions

In this paper, we have demonstrated MDA approach along with SCA concept to develop Service-Oriented applications. The appliances in the framework modeled as services are annotated with semantics to alleviate the difficulty in development. The composition mechanisms, message routing and data driven functions support the intelligent control between appliances and sensors.

Our main contribution is a system modeling methodology which is based on MDA and SCA to facilitate the development of smart building energy management system which often involves complex activities and interactions among sensors, devices and human users. Although sensors can control the devices, it needs to work cooperatively with other devices and human users. In a context-aware system with multiple users involved, it is important to develop a synchronous information stream and fusion. Our proposed Model Driven Service-Oriented approach provides a solution for developers to extend service features to accommodate existing devices or applications without rewriting them.

This study presents an overview of MDA with emphasis on the application of smart office based on SCA framework which provides a mechanism to support necessary steps required for service composition. The developers can take advantage of these existing services and supporting functions to produce composite service for intelligent control. A prototype of the proposed framework based on a number of existing hardware such as Arduino and software such as Synapse, and Tuscany has been developed to test the feasibility of the proposed approach. Further development of the system by introducing more intelligent rules and repositories is needed.

Acknowledgements. The research is supported by the National Science Council of Taiwan under the grant No. NSC 96-2416-H-009-008-MY3. This research is also partially supported by EU FP7 program under grant agreement no 224609.

References

1. Kapitsaki, G., Kateros, D. A., et al.: Service Composition: State of the art and future challenges. In: Mobile and Wireless Communications Summit, Budapest, 2007
2. Chakraborty, D., Yesha, Y., Joshi, A.: A distributed service composition protocol for pervasive environments. In: Wireless Communications and Networking Conference, Atlanta, 2004
3. Josef, N., Alam, S., Mohammad, C.: Integrating Mobile Devices into Semantic Services Environments. In: The 4th International Conference on Wireless and Mobile Communications, Athens, 2008
4. Georgia Institute of Technology–Georgia Tech Aware Home- available at http://www.cc.gatech.edu/fce/ahri, 2002
5. Massachusetts Institute of Technology–House, available at http://architecture.mit.edu/house_n, 1999
6. University of Rochester–Smart Medical Home, available at http://www.futurehealth.rochester.edu/Smart_Home, 2004
7. A. Harter et al., The Anatomy of a Context-Aware Application, Proc. 5th Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (Mobi-Com 99), ACM Press, p.p. 59–68, 1999
8. B. Brumitt et al., "Easy Living: Technologies for Intelligent Environments," Proc. 2nd Int'l Symp. Handheld and Ubiquitous Computing (HUC 2000), LNCS 1927, Springer-Verlag Publication Inc., 2000, p.p. 12–29.
9. T. Kindberg et al., "People, Places, Things: Web Presence for the Real World," Proc. 3rd IEEE Workshop Mobile Computing Systems and Applications (WMCSA 2000), IEEE CS Press, p. 19, 2000
10. UPnP Device Architecture 1.0, UPnP Forum, Dec. 2003
11. JINI Technology Core Platform Specification, v. 2.0, Sun Microsystems, June 2003
12. Service Location Protocol, v. 2, IETF RFC 2608, June 1999
13. Salutation Architecture Specification, Version 2.0c, Salutation Consortium, June 1, 1999
14. Tom Bellwood, IBM. Luc Clement, Microsoft, Claw von Riegen, SAP, UDDI Version 3.0.2, UDDI Spec Technical Committee Specification, Dec. 2004
15. Chi-Chun Lo, Ding-Yuan Cheng, Ping-Chi Lin, Kuo-Ming Chao, A Study on Representation of QoS in UDDI for Web Services Composition, Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, 2008, p.p. 423-428, ISBN: 978-0-7695-3109-0
16. Anton Naumenko, Sergiy Nikitin, Vagan Terziyan, Jari Veijalainen, Using UDDI for publishing metadata of the semantic web, Springer Boston Publication Inc., Vol 188, p.p. 141-159, 2005

17. Jim Luo, Bruce Montrose and Myong Kang, An Approach for Semantic Query Processing with UDDI, SPRINGER-VERLAG, Lecture notes in computer science, 2005, NUMB 3762, p.p. 89-98, ISSN 0302-9743.
18. Asuman Dogac, Ibrahim Cingil, Gokce Laleci, and Yildiray Kabak, Improving the Functionality of UDDI Registries through Web Service Semantics, Lecture Notes in Computer Science (2002), Volume 2444, Springer Berlin, p.p. 9-18, ISBN 978-3-540-44110-6.
19. M. Papazoglou and D. Georgakopoulos. Guest editor introduction: Service oriented computing. ACM SIGSOFT Software Engineering Notes, 46(10):24–28, 2003
20. F. Casati, S. Ilnicki, and L. Jin. Adaptive and dynamic service composition in EFlow. In Proceedings of 12th International Conference on Advanced Information In Proceedings of 13th International Conference on Advanced Information Systems Engineering (CAiSE), Interlaken, Switzerland, Springer Verlag Publications Inc., June 2001
21. H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker. Modeling and composing service-based and reference process-based multi-enterprise processes. In Proceeding of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden, Springer Verlag Publications Inc., June 2000
22. Merabti, M., Fergus, P., Abuelma'atti, O., Yu, H., Judice, C.: Managing Distributed Networked Appliances in Home Networks. Proceeding of the IEEE 96(1), p.p. 166–185 , 2008.
23. Fatima-Zahra Belouadha, Hajar Omrana and Ounsa Roudiès, A model-driven approach for composing SAWSDL semantic Web services, International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010
24. Bernhard Bauer, Jörg P. Müller, MDA Applied: From Sequence Diagrams to Web Service Choreography, ICWE 2004, LNCS 3140, pp. 132–136, 2004
25. Roy Gronmo, Michael C. Jaeger, Model-Driven Semantic Web Service Composition, Proceedings of the 12th Asia-Pacific Software Engineering Conference table of contents, Pages: 79 - 86, 2005, ISBN ~ ISSN:1530-1362 , 0-7695-2465-6
26. Zhengdong Zhu, Yanping Chen, Ronggui Lan and Zengzhi Li, Study of MDA Based Semantic Web Service Composition, Information Technology Journal, Volume 8, Issue 6, 2009
27. B. Bauer and J. P. Muller. Mda applied: From sequence diagrams to web service choreography. In ICWE'04 (LNCS 3140), p.p. 132-136, 2004
28. Simona Motogna, I. Lazăr1, B. Pârv1, and I. Czibula, An Agile MDA Approach for Service-Oriented Components, Electronic Notes in Theoretical Computer Science, Volume 253, Issue 1, p.p. 95-110, Oct. 2009
29. Hwang, C. L. and Yoon, K. Multiple Attribute Decision Making: Methods and Applications. New York: Spriner-verlag, 1981
30. Yoon, K. and Hwang, C. L., Multiple Attribute Decision Making: An Introduction. Thousand Oaks: Sage Publications Inc., 1985.
31. G. Bordogna, M. Fedrizzi, G. Pasi, A linguistic modeling of consensus in group decision-making based on OWA operators, IEEE Transactions on Systems, Man and Cybernetics: Part A. Systems and Humans 27, p.p. 126– 132, 1997
32. N. Bryson, Group decision-making and the analytic hierarchy process: exploring the consensus-relevant information content, Computers & Operations Research (23) (1996) 27– 35.
33. C. Carlsson, D. Ehrenburg, P. Eklund, M. Fedrizzi, P. Gustafsson, P. Lindholm, G. Merkuryeva, T. Riissanen, A.G.S. Ventre, Consensus in distributed soft environments, European Journal of Operational Research 61 (1992) 165–185.
34. A. Efstathiadsea, S. Tassoub, A. Antoniouc, Strategic planning, transfer and implementation of advanced manufacturing technologies (AMT), Technovation 22 (4). p.p. 210– 212, 2002.
35. F. Herrera, E. Herrera-Viedma, J.L. Verdegray, A model of consensus in group decision-making under linguistic assessments, Fuzzy Sets and Systems 78 p.p. 73– 87, 1996

36. J. Kacprzyk, M. Fedrizzi, Multiperson Decision-Making Models Using Fuzzy Sets and Possibility Theory, Kluwer Academic Publishers Inc., Dordrecht, 1993.
37. J. Kacprzyk, M. Fedrizzi, H. Nurmi, Consensus Under Fuzziness, Kluwer, Norwell, MA, 1997
38. E.E. Karsak, Distance-based fuzzy MCDM approach for evaluating flexible manufacturing system alternatives, International Journal of Production Research 40, 2002, p.p. 3167–3181.
39. Object Management Group. MDA Guide V1.0.1. OMG, 2003
40. Marco Brambilla, Stefano Ceri, Ferderico Michele Facca Politecnico di Milano and Irene Celino, Dario Cerizza and Emanuele Della Valle Ceferiel, Model-Driven Design and Development of Semantic Web Service Applications, ACM Transactions on Internet Technology (TOIT) archive Volume 8, Issue 1 No.3 , Nov. 2007, ISSN:1533-5399
41. Bell, Michael, Introduction to Service-Oriented Modeling. Service-Oriented Modeling: Service Analysis, Design, and Architecture, Wiley & Sons Publishers Inc., 2008, ISBN 978-0-470-14111-3.