

OSGi-Based Context-Aware Middleware for Building Intelligent Services in a Smart Home Environment

SHU-CHEN CHENG¹, CHIEN-FENG LAI²

Department of Computer Science and Information Engineering, Southern Taiwan University, Taiwan¹

Department of Engineering Science, National Cheng Kung University, Taiwan^{2,*}

kittyc@mail.stut.edu.tw¹, cinfo@gmail.com^{2*}

Abstract- Context awareness technology is a key technology of digital home that brings a more convenient life. Context awareness technology is a system that catches the information from the sensor network, which is built in environment, the network could get the environment data immediately, and through this data information that network provides, the system could do the exactly the right thing that help people's like getting better. These information data could be location, temperature or humidity. However, these data are used by context awareness system to analysis and process, so as to realize different person in different environment using every kind of devices and through any different existing network system, and still could enjoy the personality applications and services in any time they want. This thesis proposed Intelligent Context-Aware Middleware is including system core modeling, context provider, context analysis model, reasoning system, scenario analysis model and some other APIs to achieve the context awareness system that a digital home needs. Besides, this middleware collaborates with OSGi framework in paralleled. In other words, this Intelligent Context-Aware Middleware is totally associated to each other, which means every new devices could be compatible to either OSGi framework or Intelligent Context-Aware Middleware this thesis proposed immediately.

Index terms: OSGi, Context-Aware, Intelligent Services.

I. INTRODUCTION

Due the scientific and technological progress in recent years, advancements in medicine, extended human life span, lower birth rate, and other factors, an aging society has arrived. For those reasons, people try to use information engineering technology to provide a comprehensive care for both people and environment, so as to enhance the quality of human life, living

convenience and job performance. Facing the advent of a global aging society, many developed countries have proposed issues, such as environment, intelligent housing and health-based monitoring and others.

The focus of intelligent space technology is to establish a healthy and comfortable environment, which can enhance users' quality of life and job performance. With the progress and development in home network and office network, more and more technologies have been applied in digital home and digital office, and context awareness technology becomes the key technology which can enhance users' quality of life and job performance. This technology can sense the data at anytime, such as location, temperature, and humidity, and transmit to the server for users or applications. Thus, users can enjoy personalized application services at anytime and anywhere.

Although the existing home network communications platform Open Service Gateway Initiative (OSGi) standard can integrate heterogeneous networks, provide a standard to control network devices, the OSGi standard does not provide the context awareness technology. [1] [2] Thus, if sensor information in home network can be obtained through gateway, which can issue control command to relevant device, the applicability would be enhanced. To achieve this goal, this study proposed a middleware that has an intelligent context aware platform and combines OSGi home network gateway, and discussed the approach to use sensor information to provide users with a better intelligent space environment when the sensors, computers and consumer electronics need to integrate through network platform, in addition to inter-device communication problems.

The remainder of this paper is organized as follows: Section 2 describes the basic framework and standards of OSGi, and related research on Context awareness technology; Section 3 introduces the system development environment and detailed system operational process. Section 4 introduces the experimental system model and presents the simulation results; Section 5 provides the conclusions.

II. Related work

a. OSGi Framework

OSGi usually contains two meanings: OSGi Alliance, and OSGi framework. OSGi Alliance is an open standardization organization which was established in March 1999, and launched jointly by

Sun Microsystems, IBM, and Ericsson. It was initially called the Connected Alliance. This organization and its standards aimed to help service providers providing a variety of services to smart devices for families through the residential network. At present, the platform has become an open service platform of applications and services for all types of network equipments in all environments, such as indoor, transportations, and mobile phones for delivery and remote management. While the OSGi framework is the practice of standardized service, which was developed by the Connected Alliance, and written with Java. OSGi framework can realize a complete dynamic component model, and make up the deficiencies of Java virtual machine environment, as shown in Figure 1. Its application language or components can be remotely installed, started, stopped, updated and removed without restarting the platform. In addition, there are detailed provisions to manage these Java packages and classes. The original goal of this standard was the Service Gateway, but this specification covers a wider context. It can also be used in hand-held devices, such as mobile phones, and further be maintained through Eclipse IDE of open program code. [3] This standard can also be applied to automotive, industrial automation, building automation, PDA, grid computing, entertainment, and military management.

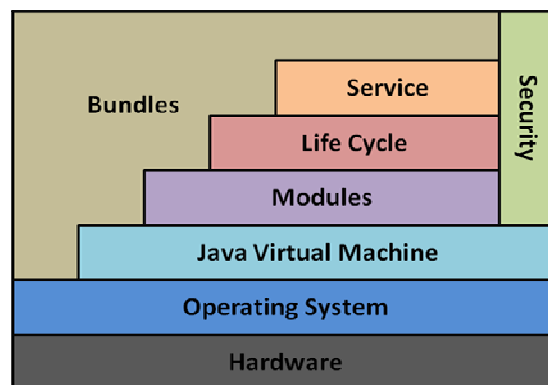


Figure 1. OSGi framework

This study used Knopflerfish 1.x (OSGi R3), which included components defined by OSGi R3, Knopflerfish components, and the Knopflerfish optional components. [4]

b. Context aware

Context aware originates from mobile systems which can detect physical environment; the information is correspondingly used to adapt their behaviors. This system prevails in the

computer environment, and its three main messages are: location, nearby people, and nearby available resources.

The goal of Context aware is to bring convenience to intelligent digital homes. It can transfer the data, such as location, temperature, and humidity to the server at anytime through a sensor network platform built in the environment, for users and applications. After further processing and analysis, the user can enjoy personalized application and services at anytime and anywhere with various devices. The term "context" refers to the detectable information in the environment, and "awareness" refers to the correct feedback after knowing the information. Context aware is the transparent man-machine interface.

The general context aware system uses a function-based framework to intelligently identify specific scenarios. Since the scenario is difficult to determine and it requires many sensors, computers, co-operation between mechanical and electrical products can provide more accurate response. Due to various types of sensors, such as computer and electromechanical products, new problems have arisen. Communications between these devices can strengthen the analysis of information on the environment, but the communications becomes another issue, in particular, interaction among two different types of devices is difficulty. Therefore, this study regarded the context and inter-device communications as a key topic.

Many researches can be used to develop context awareness framework, such as: JADE framework (Java Agent Development framework). [5] One of the popular frameworks is JCAF (Java Context Awareness Framework). [6] It is a context awareness framework based Java, and provides a Java-based, lightweight, efficient and compact interface, which is simply and powerful, so that program developers can develop a more specialized support for context awareness in the relevant fields. Its framework is as shown in Figure 2:

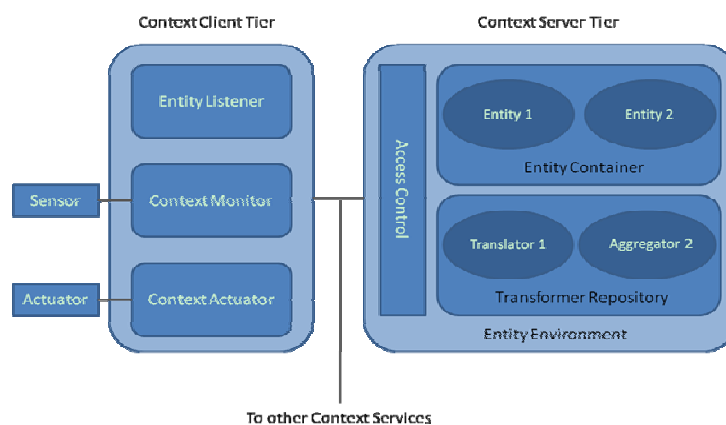


Figure 2. JCAF Architecture

As seen, the context services is a design which uses point-to-point connections, each is responsible for handling context information in this environment. For example, there is an available context service in an operating room, to provide information on the physicians in the operating room, their actions, the patient, and the status of operation. In this network of services, the service can query each other's context information.

Each context service is a long-time execution program, such as J2EE Application Server, because JCAF was inspired and developed by J2EE specifications. [7][8][9][10] The Entity containing context information can be handed over to Entity Container for management. Entity is a small Java program which implements in context service, and in charge of response and changing the context, its life cycle is maintained by the Entity Container. Entity container deals with subscription of context trigger events, and informs the relevant client of changes in entity.

Entity components are in the context service, they cooperate with other components to complete the work, so they need means to access to each other and share resources, such as databases online or RMI call. Entity Environment can achieve this requirement, that is, all the entities have their own handler functions in the execution. In addition, when accessing to general resources, such as the initial parameters, Entity Environment provides functions to access Key-Value property, as well as context transfer. Context transformers are a small customizable Java program, so that developers can write and increase the Transformer Repository, which can be used to inquiries related information at anytime.

To access the Context Service, the Access Control component can be used to ensure clients' permission. This component is basically divided into two parts: Access Control List (ACL) and authentication mechanism. ACL records client's permissions to issue the request. Through standard request, responding model, or subscription of an Entity Listener, context client can access the context information together with the entity. JCAF also supports the subscription with same type, that is, the context client can subscribe to a type of entity, such as patients. There are also two special types of context clients: Context Monitor and the Context Actuator. The former is designed for those that need context information through context sensor to use; the latter is used to interact with other Actuator and change the content of context.

The main purpose of context aware system is to allow the system to respond to users' needs. The corresponding inference system can reach this goal, and there are many related researches. Next section will describe the technologies and literatures referred to by this paper in detail.

III. Construction and practice of intelligent context aware middleware

a. System analysis

The following lists the considerations of this paper when designing the context aware system:

When using the system devices, the device should be able to collect users' behavior patterns. Due to changes in some environmental factors, users may change their needs into another state in the original system settings, although the Rule-based reasoning can provide clear, customized, and pre-determined user settings, it requires rules set by precise settings, so it is more difficult to adapt to the changing needs of dynamic and rapidly changing environment. In other words, once the rules are set but users' behavior patterns are changed, in the Rule-based reasoning framework, it is necessary to re-write the rules. This responding logical reasoning has a high level of dependency, which causes the restrictions on system design.

Intelligent digital home devices should make life more convenient, rather than becoming a trouble. Some services have dependencies on the environment. Usually, more flexible services would require more setting adjustment. Therefore, this study aimed to minimize the need of search, and provide proper assistance, which means that the system would be automatically triggered based on the previous man-machine interaction. [11-15]

Since general users cannot set a proper standard rules, even though the service providers or manufacturers have designed the services for certain environment, there are different devices for each environment, various services and home environments. As a result, learning system must deal with these environmental variables and the preferences of each person, so as to adapt to environmental changes. Furthermore, if the maturity of system and intelligence scenario can change over time, when the service providers or manufacturers add new features for a device, the system should also consider how to update and access to these functions.

To sum up, since OSGi framework can provide excellent service management environment, it is suitable for the development of services required by intelligent digital homes. As OSGi framework lacks the inference system which studies and analyzes user information, as well as of information exchange mechanisms, this study focused on the existing OSGi framework, so it can have context aware components, such as event trigger, reasoning systems, intelligent learning

framework and others, and integrate fragmented environment information into more useful message. [16-25]

b. System structure

This system was based on the OSGi framework. The system structure is shown in Figure 3 below:

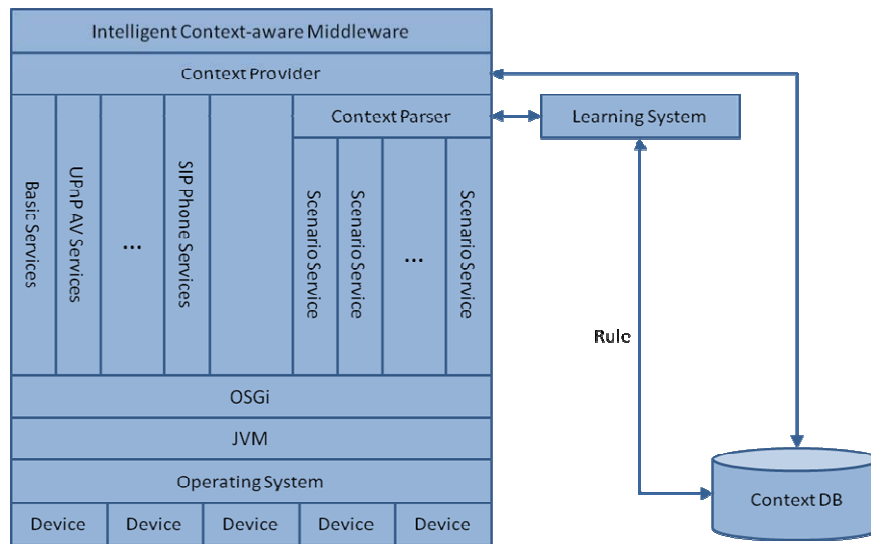


Figure 3. System Architecture

The operational process of system is shown in Figure 4:

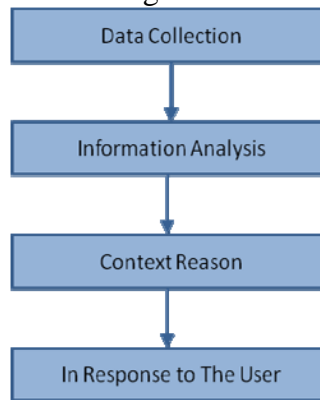


Figure 4. System Process

(1) Information collection: Figure 5 shows the flowchart of sensor data composition, Figure 6 shows the flowchart of identification; Figure 7 shows the flowchart of information collection. Users can interact with devices to perform data collection. The system can deliver related information, which is the quantity of environmental parameters collected and interaction devices (e.g. UPnP, Bluetooth, infrared, ZigBee), to the backend learning system for information analysis,

so as to generate effective information. Furthermore, effective information can be integrated with other information again to form new information. In accordance with the extent of the combination, the level of detail of the users' behavior patterns expressed by the message can be determined, that is, higher level of the combination leads to more accurate description of the ongoing actions of users.

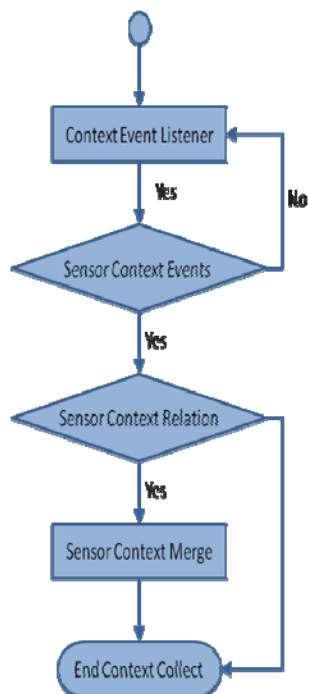


Figure 5. Sensor Data Composition Flowchart

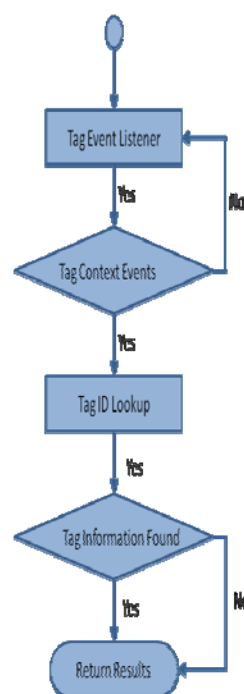


Figure 6. Identification Flowchart

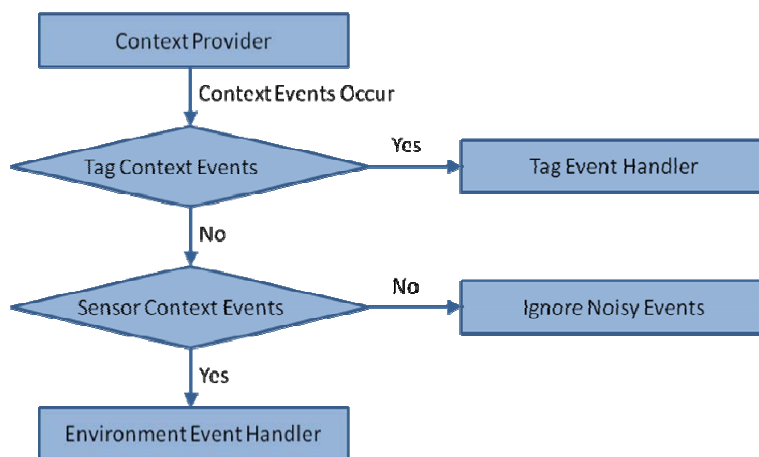


Figure 7. Information Collection Flowchart

(2) Information analysis: its purpose is to determine the user's ongoing behavior patterns. After receiving and analyzing information, system divides interaction information into two categories:

1) a further behavior pattern after interacting with the device; 2) after the operation, the users are not satisfied the environment, or the reaction time of operation is longer, thus need some time to complete. As mentioned above, multiple users in the parallel environment at the same time and device interaction will turn to information analysis, so as to learn the relationship and relevance between the users of this group. Information analysis can also obtain users' series of operations for reasoning system. In short, the composed data is collected in the information collection phase, and is organized and improved into the procedure which can be determined by reasoning system. For example, when a user is ready to go to bed at 0:00, usually he will close the door, turn off lighting, adjust the indoor air-conditioning temperature, and play music. When preparing for particular behavior pattern, environmental information would frequently change until the environmental settings are finished, so when these environmental messages are transmitted to the context aware system, it can analyze that user is ready to go to bed, as shown in Figure 8.

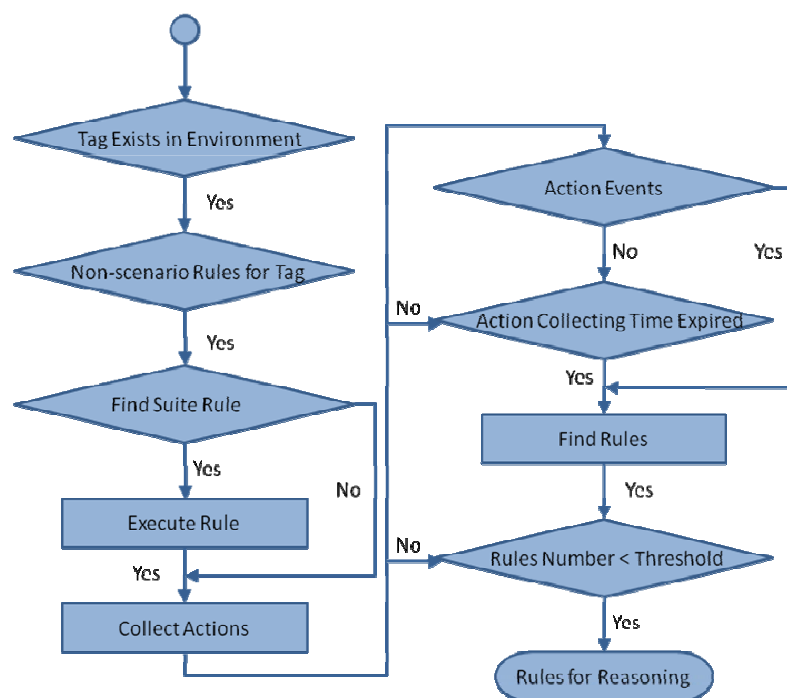


Figure 8. Information Analysis Flowchart

(3) Scenario Reasoning: Figure 9 shows the flow of basic architecture for inference system, which is in the premise of the completing data collection and data analysis. After obtaining environmental parameters, interactive information of devices, behavior patterns, and other data, if inference system is unable to learn from past experience for corresponding record, it would lead these data into a learning system to wait for user feedback, so as to facilitate learning. Otherwise,

the reasoning system would begin. To make the right inference and provide assistance, it is necessary to note the interaction between user and machine, so as to achieve the scenario needed, the operations of which are series with irregular order, it is impossible to reason the conclusions accurately and identify the same scenario with different operational orders. During information analysis, the system has reached a balance of these two objectives, so user's operation order at this stage is completed, as shown in Figure 10. Reasoning system would give judgment standards of possibility to each possible scenario, and the standard is in accordance with the accuracy of its message, as well as the extent of the message combination to give priority. In the flowchart of scenario generation of Figure 11, when a possible scenario has a higher priority, it is decided and executed, while others with lower priority are not adopted as insufficient information.

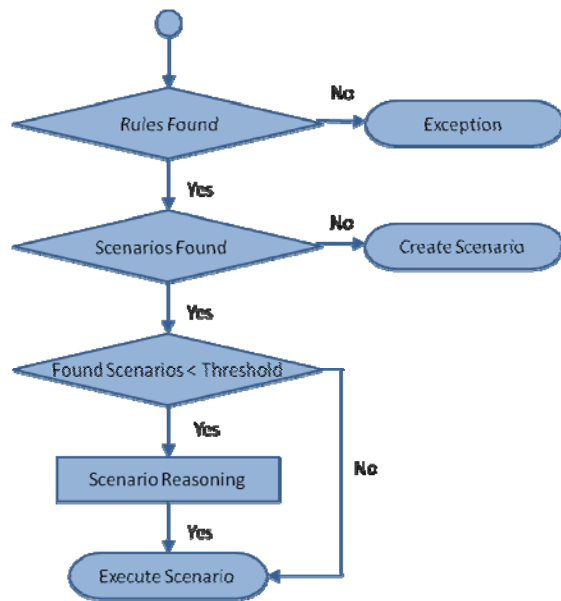


Figure 9. Inference System Flowchart

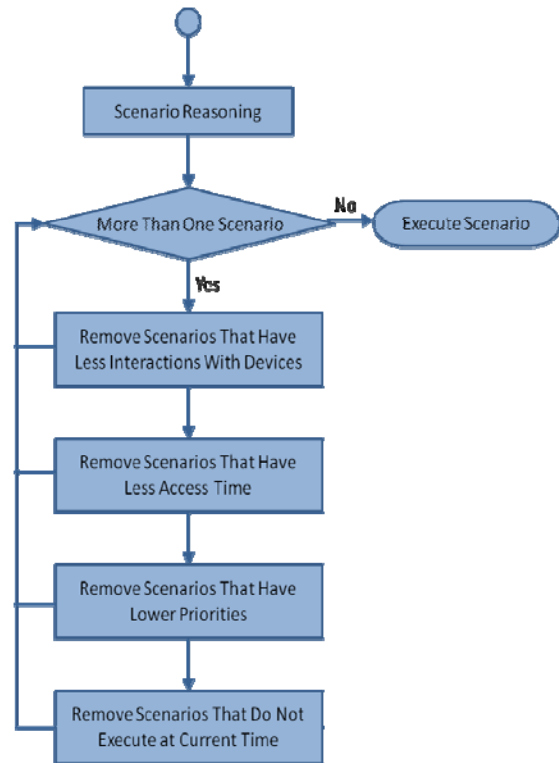


Figure 10. Scenario Analysis Flowchart

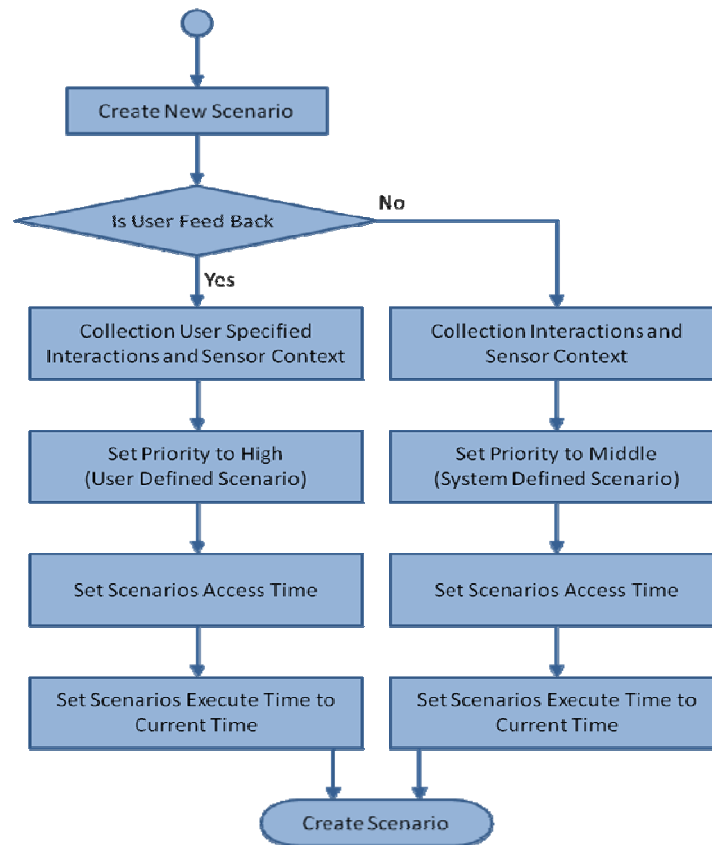


Figure 11. Scenario Generation Flowchart

(4) User feedback: finally, after the implementation of corresponding assistance procedures, system will wait for user response to these assistances whether they have reached the functionality required, and transmit these responses to the learning system, which will update rules for these messages, so as to enhance the subsequent intelligent work. The priority of all possible scenarios predicted by inference system is based on message accuracy, as well as the extent of the message combination. However, if there is user feedback, priority would be enhanced or lowered. Therefore, in this stage, the priority of scenarios executed is adjusted after the prediction, as shown in Figure 12.

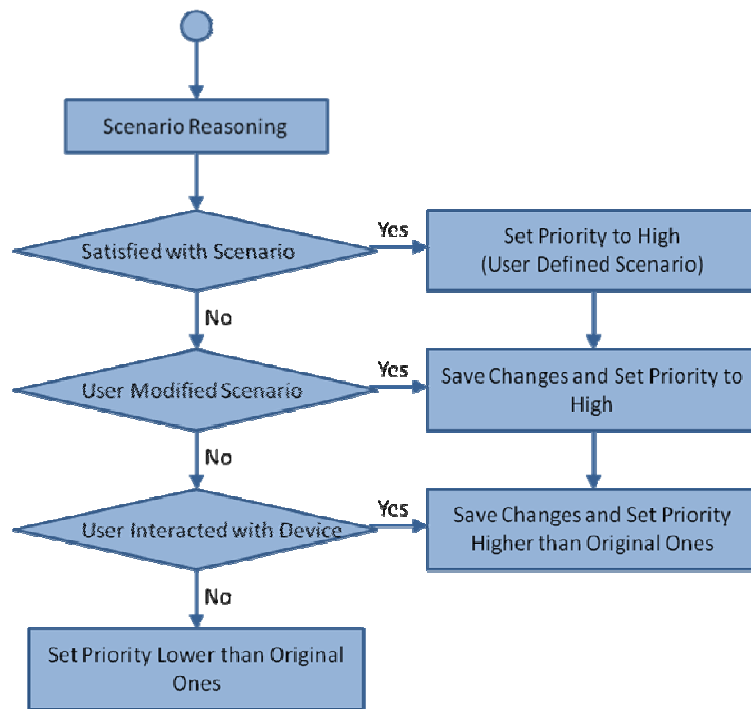


Figure 12. User Feedback Flowchart

The following lists the actual operation of the system. Figure 13 shows that when the signals produced by environmental sensor are received by the system, among them, s1, s2, s3, s4, s5 are the message generated by sensors. According to the flowchart in Figure 5, context provider combines these messages to generate new data E1, E2, E3, E4, as shown in Figure 14; if s1, s2, s3, s4, s5 contain user's label information, then reference to Figure 6 for identification. In accordance with E1, E2, E3, E4 and the flow in Figure 8, context parser determines the act that the user is engaged. When behavior patterns B1, B2, B3 are found, the reasoning system will search for suitable scenario, as shown in Figure 15; otherwise, the system will generate new behavior patterns reference for reasoning system in accordance with the s1, s2, s3, s4, s5, E1, E2, E3, E4, and time relations. When the inference system finds the corresponding scenarios C1, C2, the one to be executed is selected as shown in Figure 10. Similarly, without suitable scenario that meets user demand, Figure 11 shows the user customized scenario provided by system, or create new scenarios by system in accordance with the current scenario s1, s2, s3, s4, s5, E1, E2, E3, E4, B1, B2, B3. In Figure 16, the scenarios selected by system are implemented smoothly, and then wait for user feedback, which flow is described in Figure 12. The user can choose whether to perform scenario changes, if he chooses to amend the relevant settings, whether it is through

the user feedback interface or interact directly with the device, they will be recorded in the system, as shown in Figure 17.

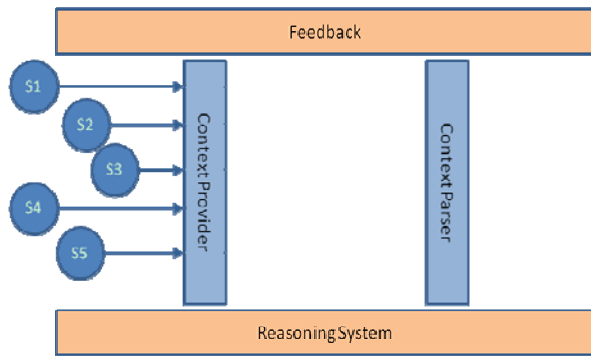


Figure 13. Receive Environmental Signals

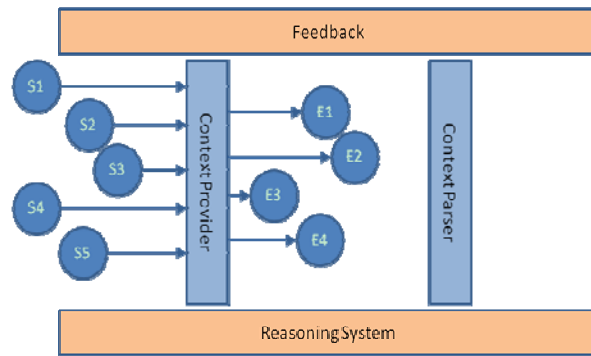


Figure 14. Combine Environmental Signals

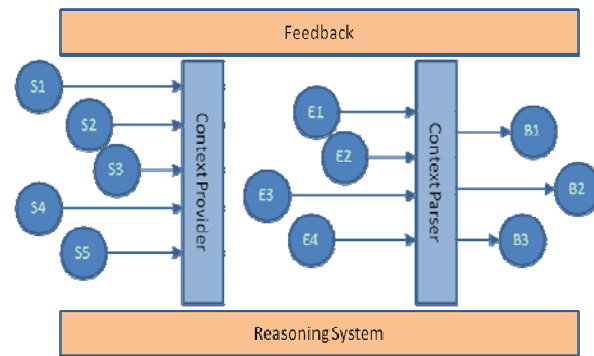


Figure 15. Search for Suitable Scenario

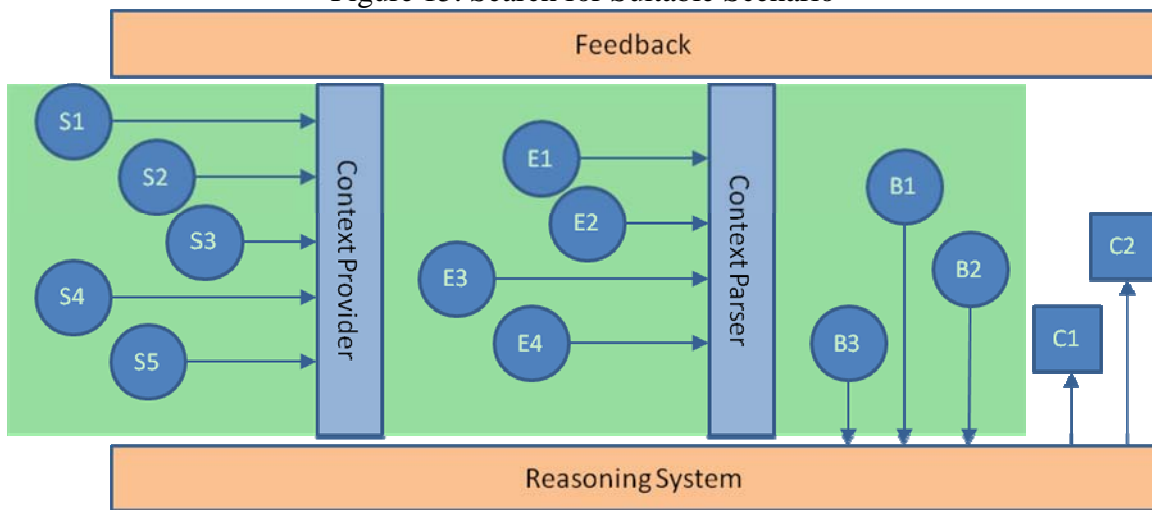


Figure 16. Generate New Behavior Patterns

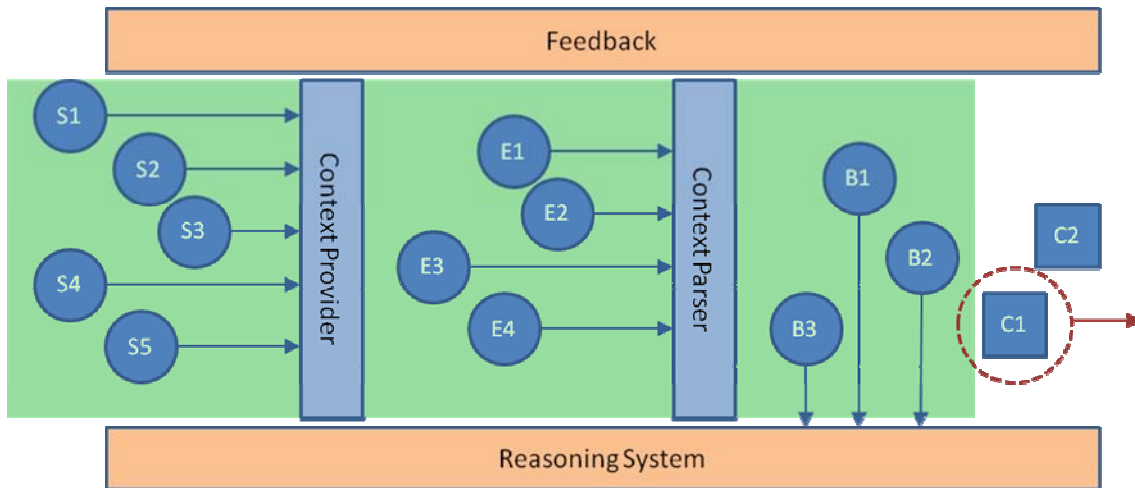


Figure 17. Generate New Scenario

Structure of scenarios and rules, as shown in Figure 18:

ID	Action	Setting Values	Provided Data	Device	Location
Environment Context Structure ID	User Action	Device Set Value	Sensor Provided Data	Device ID	Device's Location

Figure 18. Tag structure

Environmental information, as shown in Figure 19:

ID	Identity	Owner	Priority	Location
Tag Structure ID	Tag Device Built-in Number	Tag Owner	Owner Priority	Owner's Location

Figure 19. Context Structure

Structure of rules, as shown in Figure 20:

ID	Tag	Non-scenario Rule	Environment Context	Access Times	Last Execution Time
Rule ID	Tag Structure ID	This rule belongs to a scenario	Environment Context Structure ID	How many times has this rule been access	The time this rule is executed recently

Figure 20. Rule Structure

Scenario structure, as shown in Figure 21:

ID	Priority	Rules	Access Times	Last Execution Time
Scenario ID	Scenario Priority	Rules Related with This Scenario	How many times has this scenario been accessed	The time this scenario is executed recently

Figure 21. Scenario Structure

All the structural relations are shown in the following diagram:

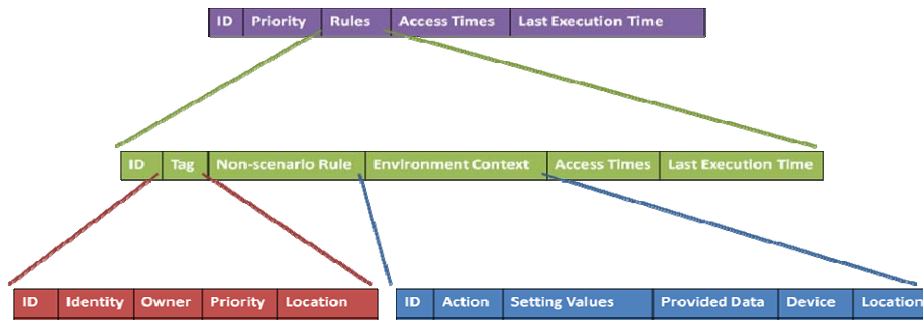


Figure 22. Structure Relation

IV. System Implementation and Testing

To test the implementation of the system, this study built some common devices in home environment on an OSGi framework. In this system, each device had a bundle running in the OSGi framework, before running, the device would register its services provided at the system. In Table 1, real-world sensor changes are simulated through data changes in virtual device, when the value in data source changes, it means that the environmental conditions change.

Table1: Virtual Devices List

Virtual Devices	Data number	Device number
Light	10	10
Chair	3	0
Bed	1	0
Window	5	0
Door	2	0
TV	4	2
PC	9	9

Audio	3	3
Location	2	0
Air conditioning	6	2
Humidity	2	0

Figure 23 shows the graphical interface of system for testing, it is user-friendly communication with the device.

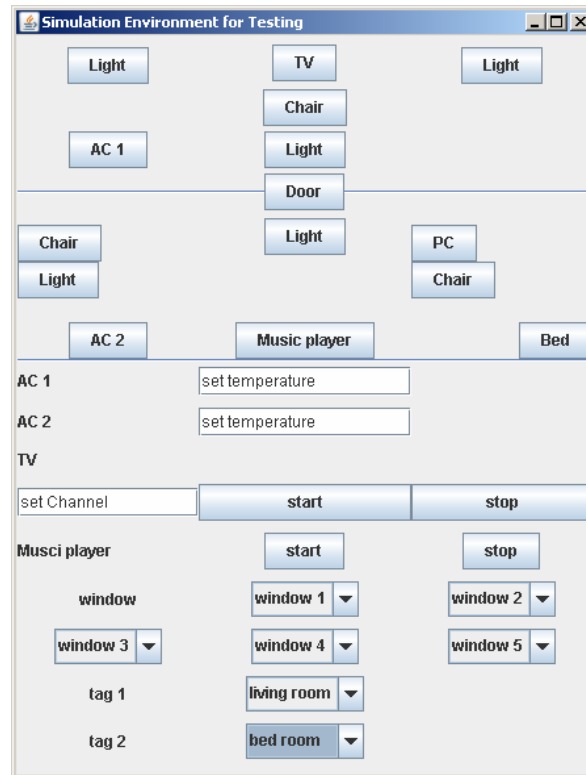


Figure23. Graphical Interface of System for Testing

After completing all the procedures and implementing scenarios, the user feedback graphical interface is shown for users to modify scenario, and even customize the contents of scenario, as shown in Figure 24.

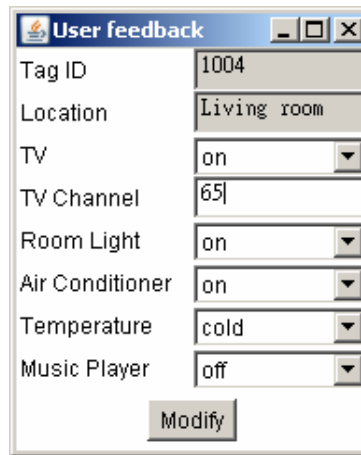


Figure 24. user feedback graphical interface

This study also implemented the system monitoring and control interface, which can monitor the status of implementation of each system component, as shown in Figure 25.

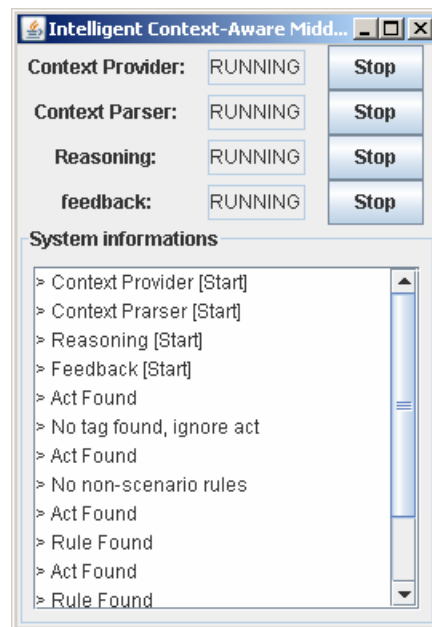


Figure 25. System Monitoring and Control Interface

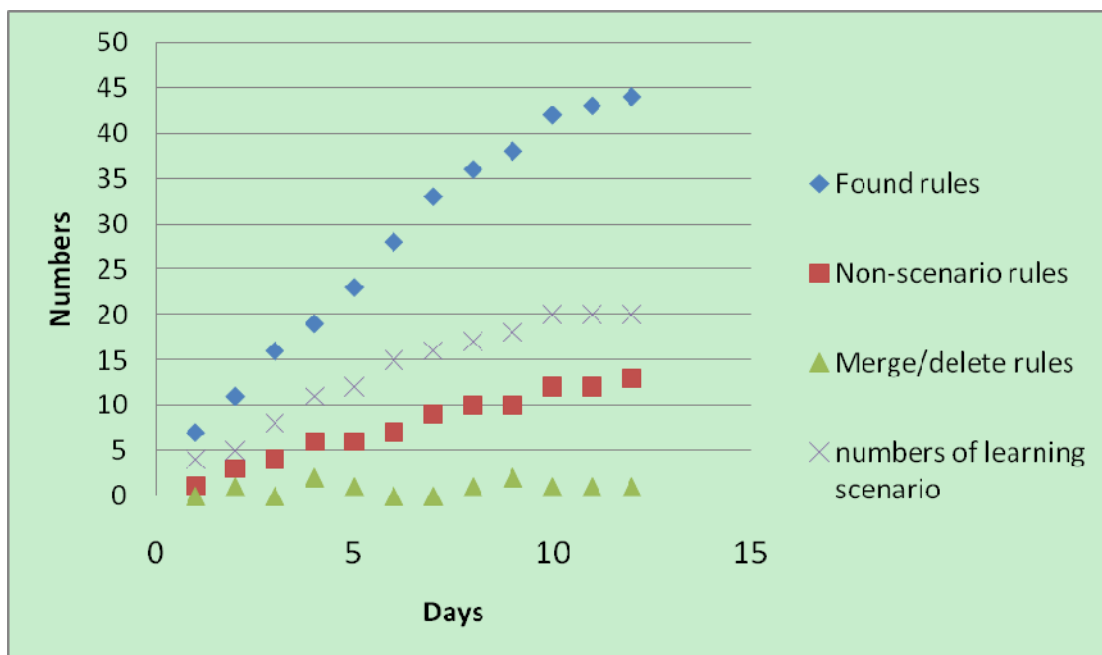


Figure26 experiment results

According to Figure26, system can receive a number of scenarios and rules on the first day for the second day; also, in about 7-10 days, the system can collect most of the user scenarios and rules. This conclusion is judged based on the number of system learning scenario, as well as the new rule curves. Series "Found rules" refers to the number of rules found by system; series "Non-scenario rules" refers to the number of non- scenario rules found by system; series "Merge/delete rules" refers to the number of rules deleted by systems when combining; series "Numbers of learning scenario" refers to the number of scenario for system learning. According to Figure 28, through the curvature of curves, the user behavior patterns collected by system in a week are observed, which can provide the convenience from context awareness for users in the second week.

Through information collection and the reasoning module of information analysis, scenario analysis, and related interfaces, this study proposed an intelligent context aware system that contains the core of the system modular. This system cooperates with the OSGi framework on a parallel architecture, thus, other service providers or manufacturers need not to understand it fully, or transfer their services to this architecture. While their services would be received by system through the OSGi framework, from the service developer's point of view, this intelligent context aware system and OSGi framework are closely connected.

V. Conclusions and future development

Through information collection and the reasoning module of information analysis, scenario analysis and related interfaces, this study proposed an intelligent context aware system that contains the core of the modular system. Since it cooperates with the OSGi framework on a parallel architecture, other service providers or vendors do not need to fully understand the system, or transfer their service to this framework. Their services can be received by system through the OSGi framework, from the service developer's point of view, this intelligent context aware system and OSGi framework are closely integrated.

This intelligent context aware system could establish the hidden scenario data according to the behavioral patterns and the temporary rules, if the new devices or services are developed, users only need to install the device on the OSGi framework, then system could obtain contents of service through service package, so all procedures are ready. If the new device only causes little impact on users' living habits, the system would modify the scenario; otherwise, the system would generate new scenario data.

REFERENCES

- [1] OSGi Service Platform Release 4
- [2] T. Gu, H. K. Pung, and D. Q. Zhang, "Toward an OSGi-based infrastructure for context-aware applications," *Pervasive Computing, IEEE*, vol. 3, no. 4, pp. 66-74, 2004.
- [3] Eclipse is available from <http://www.eclipse.org/>
- [4] Knopflerfish is available from <http://www.knopflerfish.org/>
- [5] JADE Framework - Java Agent DEvelopment Framework, available from <http://jade.tilab.com/>
- [6] J. E. Bardram, "The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications," in *Pervasive Computing*, 2005, pp. 98-115.
- [7] Sun JDK is available from <http://java.sun.com/javase/>
- [8] IBM JDK is available from <http://www.ibm.com/developerworks/java/jdk/>
- [9] BEA JRockit JDK is available from <http://dev2dev.bea.com/jrockit/>
- [10] GNU JDK is available from <http://gcc.gnu.org/java/>

- [11] Newell, Allen, J.C. Shaw, and H. A. Simon. 1957. "Empirical Explorations of the Logic Theory Machine", Proceedings of the Western Joint Computer Conference, pp. 218-239.
- [12] B. Oliver, M. Jerome, and R. Patrick, "Automatic detection of interaction groups," in Proceedings of the 7th international conference on Multimodal interfaces Toronto, Italy: ACM Press, 2005.
- [13] B. Oliver, R. Patrick, and L. C. James, "Automatic Development of an Abstract Context Model for an Intelligent Environment," in Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops: IEEE Computer Society, 2005.
- [14] R. Oriana, F. Cristiano Di, R. Stefano, and R. Kimmo, "Unearthing Design Patterns to Support Context-Awareness," in Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops, 2006, pp. 383-387.
- [15] C. A. Petri, "Kommunikation mit Automaten, PhD thesis," in Institut fur Instrumentelle Mathematik Bonn, 1962.
- [16] K.-C. Lee, J.-H. Kim, J.-H. Lee, and K.-M. Lee, "Implementation of Ontology used Context-Awareness Framework for Ubiquitous Environment," 2007, pp. 78-282.
- [17] C. Jonghwa, S. Dongkyoo, and S. Dongil, "Research and implementation of the context-aware middleware for controlling home appliances," Consumer Electronics, IEEE Transactions on, vol. 51, no. 1, pp. 301-306, 2005.
- [18] K. Nishigaki, K. Yasumoto, N. Shibata, M. Ito, and T. Higashino, "Framework and rule-based language for facilitating context-aware computing using information appliances," 2005, pp. 345-351.
- [19] Brdiczka, P. Reignier, J. L. Crowley, D. Vaufreydaz, and J. Maisonnasse, "Deterministic and probabilistic implementation of context," in Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on Volume , Issue , 13-17 March 2006 On pages: 5 pp.-
- [20] J. L. Crowley, O. Brdiczka, and P. Reignier, "Learning Situation Models for Understanding Activity," in 5th International Conference on Development and Learning(ICDL06) Bloomington, IL., USA, 2006.
- [21] T. Gu, H. K. Pung, and D. Q. Zhang, "Toward an OSGi-based infrastructure for context-aware applications," Pervasive Computing, IEEE, vol. 3, no. 4, pp. 66-74, 2004.
- [22] R. Khosla, C. Lai, B. Chowdhury, and Q. Li, "Patterns of Behaviour Mediated by Cognitive Scripts and Emotional Attitudes - Context-Aware Engineering of Data Mining Systems," 2006, pp. 3-10.
- [23] M. C. Mozer, "The Neural Network House: An Environment that Adapts to its Inhabitants.," in Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments. AAAI Press, 1998, pp. 110-114.

[24] M. Sven and R. Andry, "A survey of research on context-aware homes," in Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21, Adelaide, Australia, 2003, pp. 159-168.

[25] S. Thomas, K. Kay, S. Frank, and Y. Ming, "Middleware Support for Context-Awareness in 4G Environments," in Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, 2006, pp. 203-211.