Designing an Intelligent Agent for Information Appliances

Chiung-Hon Leon Lee[1,*], Wen-Shang Tseng[3], Chin-Mu Chen[1],
Alan Liu[2] , Syuan-Ming Huang[2], and Huang-Bin Huang[2]

[1] Department of Information Engineering and Computer Science, Chung Chou Institute of Technology
Yuanlin, Changhua 510, Taiwan, ROC
{leonlee, kchen}@dragon.ccut.edu.tw

[2] Department of Electrical Engineering and Center for Telecommunication Research, National Chung Cheng University
Min-Hsiung, Chia-Yi 621, Taiwan, ROC
stchang@nchu.edu.tw

[3] Energy and Resources Laboratories, Industrial Technology Research Institute
Chutung, Hsinchu 310, Taiwan, ROC
yappa@itri.org.tw

Received 7 December 2006; Revised 28 February 2007 ; Accepted 16 March 2007

Abstract. An intelligent agent architecture is proposed for integration of information appliances (IAs). Based on the characteristics of different IAs and the proposed agent module, the system designer can design different knowledge bases and communication interfaces for the different IA agents. Therefore, the IA agent will be used to promote the capabilities of the IAs. Also, the agent will follow up the rules in the knowledge base and environment information to make decisions for IA automation. Furthermore, agents can use an agent communication language for communication and cooperation with each other. In addition, by automatic communication and cooperation among the agents, an IA with the IA agents can generate more added-value functions than what a single IA system can provide. An implementation of IA agents in a smart home is used to demonstrate the proposed approach.

Keywords: intelligent agent, information appliances, knowledge-base.

1 Introduction

In recent years, the development of computer systems is focused on faster speed, smaller size, lower price, and easier operation. Computers are widely used by people and become one of the most important appliances at home. Combining the technologies of embedded system, broadband network, Internet, and traditional appliances, the information appliance (IA) [1] has become popular in the computer and electronic marketplace. However, as the number of functions IA increases, it will become even more difficult to use these devices easily and effectively. Users have to refer to complicated manuals and to use their own *reasoning* and *learning* processes for using an IA successfully. Besides, most of the IA applications only concerns with how to bring information network and appliances together. The problem here is how to use the communication and cooperation properties of IA to generate functions that are not performed by a single IA.

In this paper, we propose a software agent architecture for integration of IAs and use an IA agent as the core of intelligent appliances for a smart home [2]. We expect the IA with the IA agent can be promoted to intelligent information appliance (IIA) which is more suitable for constructing a smart home than the traditional IA. An IIA can collect environment information automatically, communicate and exchange information with other IIAs, make decisions by using their knowledge-base, and cooperate with each other to generate more add-value functions to serve users.

The IA with the IA agent can have some capabilities that cannot be performed by a traditional IA. For example, the agent can help the user to detect error actions of the appliance, find the broken appliance out in a smart home, compute the usage of power, detect whether or not the usage of power is overloaded, shut down unused appliances, select proper time to restart appliances, and recover errors of appliances. Moreover, the IA agent can make decisions without user's direction or interact with the user via broadband network for getting more informa-

* Correspondence author

tion to make decisions.

This paper is organized as follows. The related work is introduced in Section 2. In Section 3, we give the requirements for designing the IA agent. We describe the architecture of the proposed IA agent in Section 4. In Section 5, we introduce three different scenarios used for testing applications of the IA agent and a system implementation is also displayed in this section. Finally, Section 6 concludes this paper.

2 Related Work

2.1 Smart Home

Following the appearance of embedded systems, information appliances, and home control network, there are varieties of computer applications introduced in daily life. The human-computer interaction is no longer limited in front of computer only, but all environments. Accordingly, the intelligent environment [3, 4, 5, 6] becomes one of the charming topics in computer science not only for research but also for commercial applications. The goal of intelligent environment is to provide better life quality through a multi-sensing multimodal intelligent system embedded in an environment [7, 8]. An intelligent environment might consist of many different sub-systems such as communication system, location system, IA, home network, and security system, etc. Equipped with different sensors which observe and understand the activities in the environment, an intelligent environment is highly context-aware. After gathering the information from the environment, the system will make appropriate reactions for some events.

A smart home [9, 10] is one kind of intelligent environment and basically must contain three elements: internal network, intelligent control and home automation. The concept of smart homes plays an important role in the planning of future housing-based models of care. Internal network is the basis of a smart home, and it can be power-line, bus-line or radio frequency. Intelligent control means gateways to manage the systems. There are two main purposes of gateways: bridging different technologies within the home and providing access from the home to external services or from external systems to home. Home automation means products within the homes can links to services and systems outside the home automatically.

2.2 Embedded Agents

How to embed intelligence into devices is the key problem to enable the full benefits of ubiquitous computing to be realized in intelligent environments and to groups of artifacts learning to work together to achieve higher level, user determined goals [11, 12]. An agent [13] is a software entity that has some properties of a human such as autonomy, reasoning, learning, and knowledge level communication, etc. By service ability, the agent technique is widely applied in many fields, such as resource management for cellular mobile networks [14], QoS management for multicast environment [15], multimedia information retrieval [16], media-on-demand on the web [17], and supporting the distribution of Video on Demand (VoD) to portable devices [18], etc.

Recently, the agent technology is widely used in the construction of intelligent environment, because the agent can provide many advantages such as having artifacts in which an agent in the environment can cooperate to generate more complex functions and the environment can provide more user friendly services to users. In [19], the authors combined the use of unobtrusive sensors and effectors with intelligent embedded-agents to realize the vision of ambient intelligence in health care environments. In [20], the authors propose an intelligent agent model for smart home environments. The case-based reasoning and Bayesian Network approaches are implemented in the agent architecture for acquiring the knowledge about user actions and inferring relationships among agents.

A novel life-long learning approach based intelligent agent is proposed in [21]. The agent is embedded in intelligent inhabited environments to improve the quality of users' daily life. Because Type-2 fuzzy systems [22] are able to handle the different sources of uncertainty and imprecision encountered in intelligent environments and give a very good response, in [23] the authors proposed a novel type-2 fuzzy systems based adaptive architecture for intelligent embedded-agent.

It is obvious that there are many advantages for using agent technologies in intelligent environment. However, implementing the properties of an agent in a computer program is non-trivial. The interaction, cooperation, and negotiation issues among agents make the development of a multi-agent system (MAS) become a complex task.

There are many researches of systematic development methodology [24] to discuss how to analyze, design, and implement the MAS. Because of the complexity and different applications of the MAS, using an agent oriented software engineering (AOSE) [25] or agent based software engineering (ABSE) [26] methodology to construct the MAS is still not easy. A large amount of design knowledge and activity will be required. Moreover, the

IA agent has some specific attributes, requirements, and implementation limitations when designing an agent for IA. The designer has to consider different requirements of different IA environment.

The IA agent, proposed in this paper, is a template for the system designer to construct different IA agent for different appliances and use them in different intelligent environments. We use the process of software engineering to analyze requirements, design the system, construct the system prototype, and repeat the process again for constructing a useful system and reusable software components.

3 Requirements Analysis

In many applications, we can build an intelligent environment for further performance of environment management, such as office, office building, factory, storehouse, house, etc. Using an intelligent software agent to manage the environment and to provide a friendly interaction with users is a natural way for building an intelligent environment. The general requirements of an agent for intelligent environment are described as follows.

Software automation: The agent must solve problem actively and autonomously without human guidance.
Friendly interaction: The agent based system should provide a convenient interface for users.
Decision making: When the agent confronts problems, it can make decisions by itself.
Time-limit/emergent task handling: the agent possesses capabilities of handling time-limit or emergent tasks.
Communication and cooperation: Every agent can solve problems with each other through communication and cooperation.
Knowledge sharing: Every agent can share their knowledge with each other.

In the agent and robotics application domains, many researchers have suggested the approach which is combining the deliberative property and the reactive property [27]. Combining the deliberative reasoning ability and reactive real-time response enhances the performance of an integral system. Agents in different applications have specialized skills for its domain. Because the IA agent is designed for IA, it has some specific requirements. However, in our design, an IA agent has to fulfill following basic functional requirements.

1. **Communication requirement:** The agent should provide a mechanism to parse and package FIPA agent communication language [28] for communicating with each other.
2. **Behavior requirement:** The agent should provide a set of low level functions to control the appliance.
3. **Scheduling requirement:** The agent should provide a mechanism for task scheduling.
4. **Reasoning requirement:** The agent should provide a reasoning mechanism which can be used to make decision by referring the message received from the environment and knowledge-base.
5. **Reactive requirement:** The agent should provide a reactive mechanism. If the rule in the reactive module has been triggered, the low level functions will be executed.
6. **Information requirement:** The agent should get environment information by the sensor of related appliance.

4 Agent Architecture

Based on the analysis results of the requirements, we divide the IA agent into five parts: Agent Communication Module (**ACM**)、Control Module, (**CM**)、Reactive Module (**RM**)、Reasoning Module, (**EM**)、 and Device Communication Interface, (**DCI**). The agent architecture is shown in Fig.1. The arrow in the end of line indicates dependency relationship between components. The arrow from component A to component B means that A can use functions or data provided by B. For example, CM can use the functions of DCI, but DCI cannot use any functions of CM.

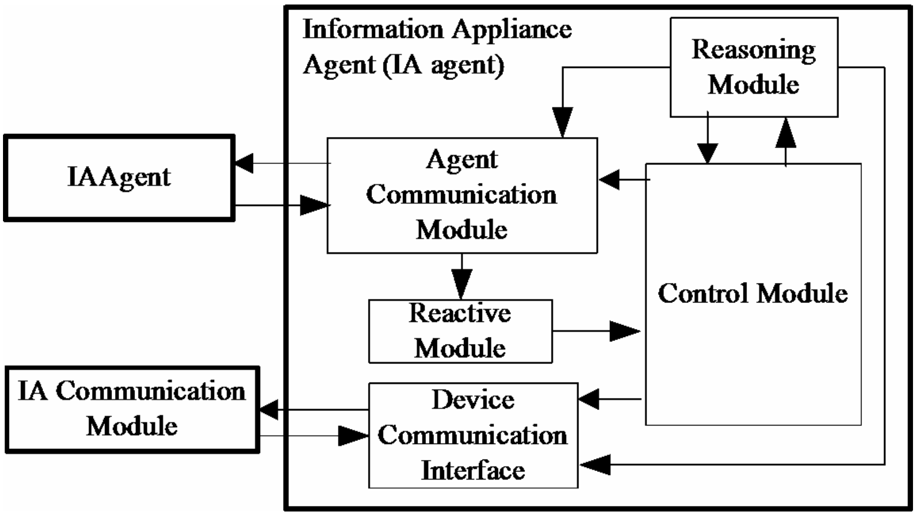

Fig. 1. The IA agent architecture

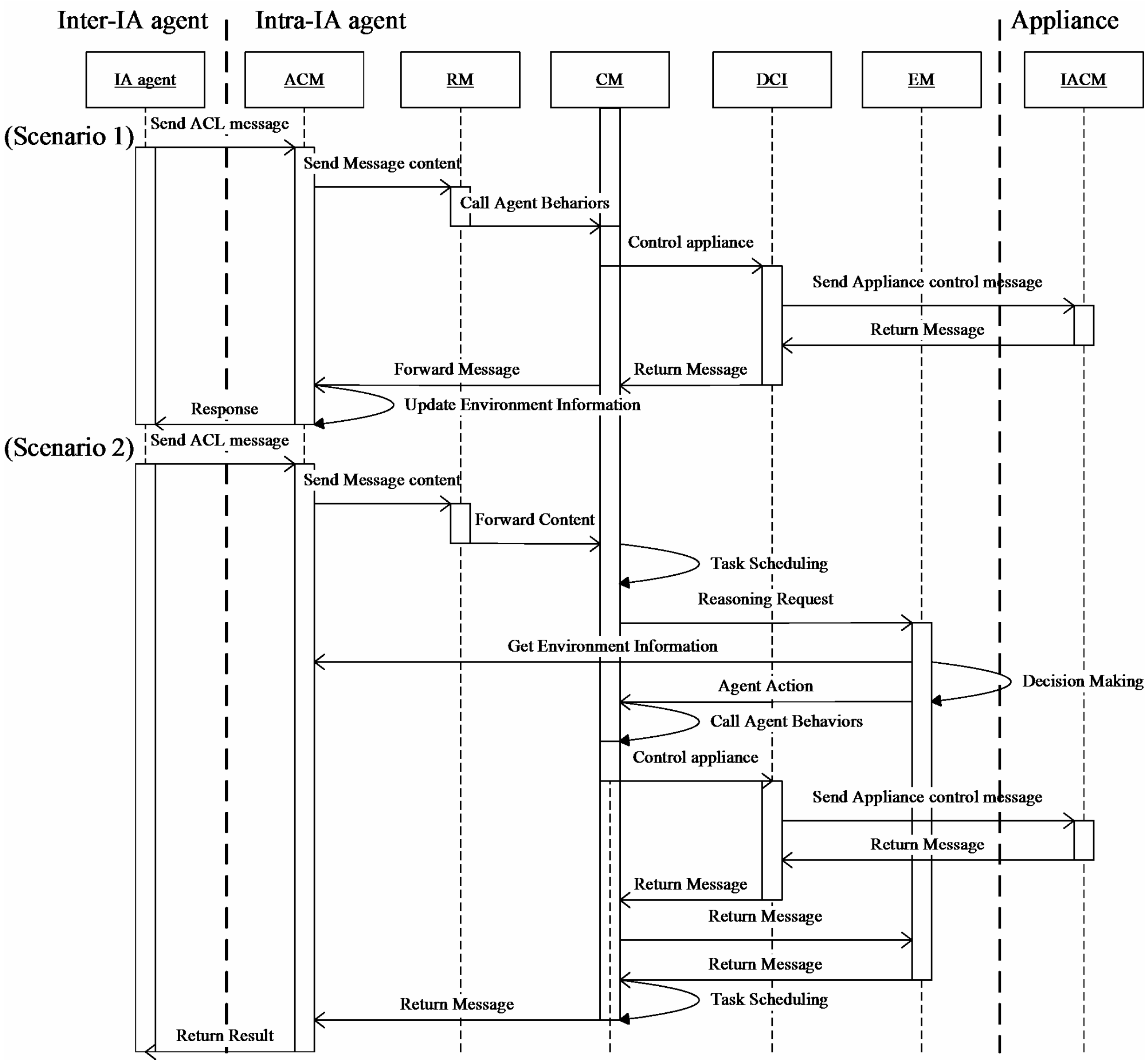

Fig. 2. Two scenarios for reactive behavior and deliberate behavior of IA agent

ACM is the communication interface of IA agents. The main function of ACM is receiving an ACL message from other agents, parsing the content of the message, storing important message, and sending message content to RM. ACM also receives the request from CM, package the request content to an ACL message, and sends the message to other agent.

CM is the control center of the agent. It schedules and controls the message processing process and provides low level functions of the agent. RM provides a set of reactive rules. If received messages fit the description of the reactive rule, RM will execute the related low level function in CM directly. Otherwise, the message will be sent to CM for scheduling. EM uses a reasoning engine to infer facts by referring the message coming from CM and the rules in the knowledge-base. In the reasoning process, the information of the appliance or other agents will be acquired by using DCI and ACM. Finally, DCI provides an interface to communicate with the IA Communication Module (IACM) for controlling the appliance. DCI is a template of communication interface for an IA agent and the appliance. The agent designer can modify this interface for reading and setting parameters of the appliance. The IA agent cans also derive the environment information such as temperature or humidity, etc. from the appliance by the IACM.

The agent has two kinds of behaviors: reactive behaviors and deliberative behaviors. The main difference between these two behaviors is that the reactive actions are triggered by RM and when the reactive tasks do not need be scheduled by CM. However, the tasks of deliberative behavior will be scheduled by CM. Two scenarios are shown in Fig.2 to demonstrate these two different behaviors.

In the first scenario, an IA agent sent an ACL message to another IA agent. The ACM of latter agent parsed the ACL message and forwarded the content of the message to RM. There was one of the reactive rules in RM triggered by the content, and RM called Agent Behavior module in CM for controlling the appliance to do something. DCI received the function call from CM and then sent control signal to IACM in the appliance. When the task has been completed by the appliance, an acknowledgement message will be returned.

In the second scenario, the IA agent received another ACL message from the same IA agent. First, the message was parsed by ACM and the content of message was forwarded to RM. RM found that there is no reactive rules triggered by this message and then forwarded the message content to CM. CM arranged the content into schedule and prepared to send a message to EM for decision making. Once the EM received the message from CM, it started to collect related information for reasoning. The agent made decision and controlled the appliance to act by sending messages to DCI. Finally, a result will be sent to the IA agent.

4.1 Agent Communication Module

The ACM is divided into three parts: Communication Interface (CI), Environment Information Interface (EII), and Environment Information (EI). The content of ACL message will be extracted by CI. If the ACL message contains environment information, CI will use EII to update the environment information stored in EI. CI will send the extracted data to RM for future processing. CI can also accept the request from CM for packaging the message to ACL format and send it to related agent. EII provide a function call for EM acquiring environment information in reasoning process. The architecture of ACM is displayed in Fig.3.

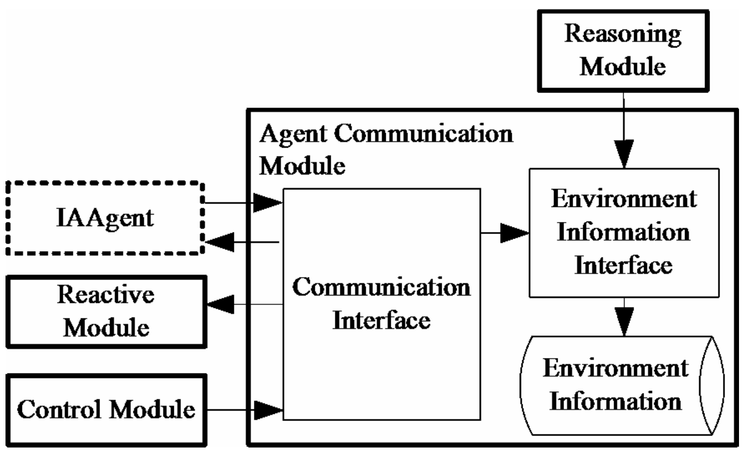

Fig. 3. The agent communication module of IA agent

4.2 Control Module

CM is the task scheduling and event control center of the agent. The main function of this module is generating system events and sending events to an Agent Event Queue (AEQ) for determining the execution sequence of events. These events include sending the RM filtered message to EM for reasoning, acquiring the information of the related appliance, setting parameters of the related appliance, preparing the message for sending to other agents, etc. An Agent Behaviors (AB) module is also designed in CM. AB is a set of implemented low level functions which can be performed by the IA agent to control related appliance or acquire information of appliance. These functions can be triggered by the rules in RM and executed directly without waiting in AEQ for generating the reflex behavior of the IA agent. Fig.4 shows the CM of an IA agent.

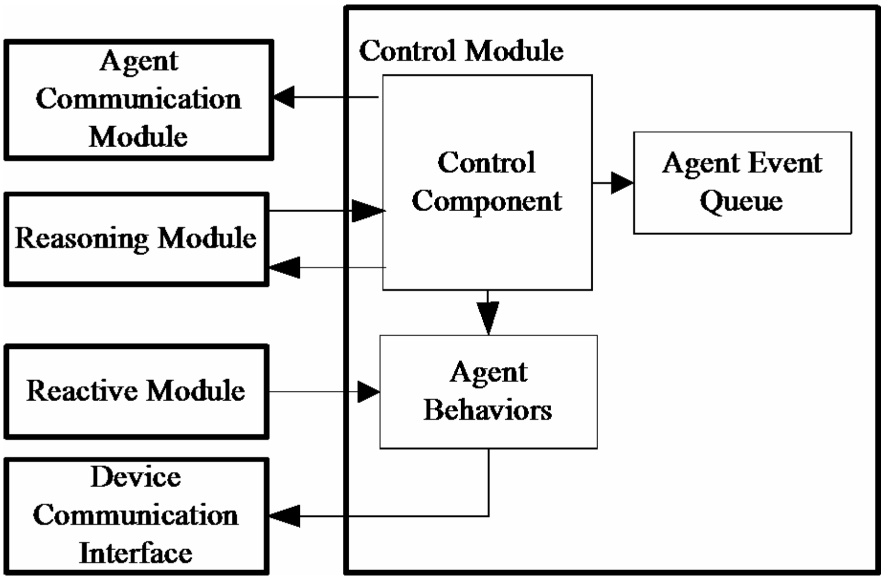

Fig. 4. The control module of the IA agent

4.3 Reactive Module

This module is the main component for the reactive behavior of the IA agent. There are three components in this module. Reactive Component (RC) receives the message from ACM and uses Reactive Rules Interface (RRI) to decide which rule stored in Reactive Rules should be triggered. The IA agent uses RM to response real time and simple requests. The RM is shown in Fig.5. If RC find that there is no reactive rule be triggered by the message fro ACM, the message will be forwardded to Control Module for processing.

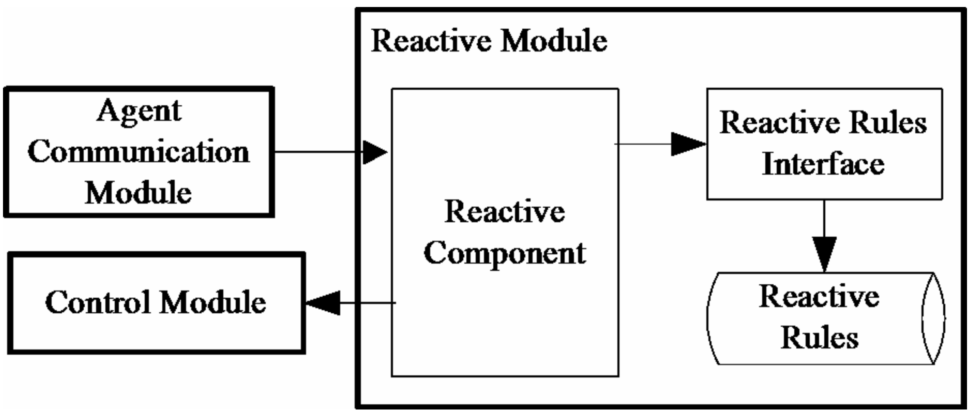

Fig. 5. The reactive module of IA agent

4.4 Reasoning Module

Three are three components in EM: Reasoning Component (RC), Knowledge-Base Interface (KBI), and Knowledge-Base (KB). A forward-chaining reasoning algorithm [29] is implemented in RC. KBI provides a set of application interface (API) for RC to access the rules stored in KB. The agent designer can also implement a user interface for IAA designer to manage rules in KB. The EM is shown in Fig.6.

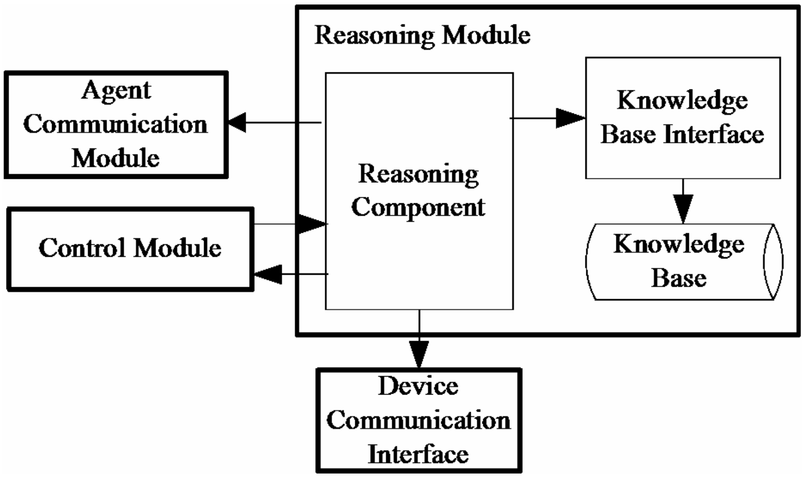

Fig. 6. The reasoning module of IA agent

5 Implementation

We defined three kinds of user when implement the IA agent based system: agent designer, system designer, and general user. The general user uses the implemented Home agent and IA agents to control appliances at home. The system designer uses different kinds of pre-designed IA agent as a template and designs reactive rules and deliberative rules for implementing different interaction scenarios among agents. The agent designer use the proposed IA architecture as a template to design different kinds of IA agent such as air-condition agent, washing machine agent, or refrigerator agent, etc.

In order to test the proposed agent architecture, we implement two kinds of agents: Home agent and IA agent. Home agent provides a graphical user interface (GUI) to interact with the user and manages related IA agents. The user can control an IA agent by using the GUI of Home agent. Home agent and IA agent have the same architecture. The main difference between them is that the KB of Home agent stores the rules of how to manage agents at home. Therefore, the KB of IA agent stores the rules of how to control appliances. A GUI if Home agent is shown in Fig.7.

Fig. 7. A snapshot of home agent GUI

The agent is implemented in Windows CE.net framework 1.1 and running in an embedded system testing board with IXP740 CPU, 64M RAM, and 64M Flash ROM. The GUI of the Home agent is displayed on a LCD television as a control center for the user to monitor the action of IA. Each IA in our environment is communication via the power line. Once an IA adds to the environment, the registry of the Home agent will be updated. The IA agent for different kinds of appliances should be implemented for controlling related appliance. By using the GUI, the user can add, remove, pause, and restart IA agents for IAs. The relationships among the embedded board, IA, and IAagents are shown in Fig.8.

A brief description of IA agents is also shown in the GUI of Home agent. The agents can detect and report their conditions or errors to the user by showing messages in the GUI of Home agent. A GUI showing a air-condition message is shown in Fig.9. Therefore, the Home agent can use broadband network to connect with the appliance store for calling appliance engineer to repair broken appliance.

In order to test the proposed IA agent, we design and implement three scenarios for IA agents. First, we assume that an user is watching TV in living room and the user set parameters of the air-condition in living room. The Home agent received the status reports from the TV agent and the air-condition agent. If the home agent detects the TV and the air-condition in living room are turned off and the air-condition in bedroom is turned on, the air-condition agent in the bedroom will be started. The newly started agent will ask whether or not there is any air-condition has been opened recently, and ask the setting parameters of the air-condition agent in living room for adjusting its setting. By using this scenario, the newly opened air-condition can be adapted itself setting for the user.

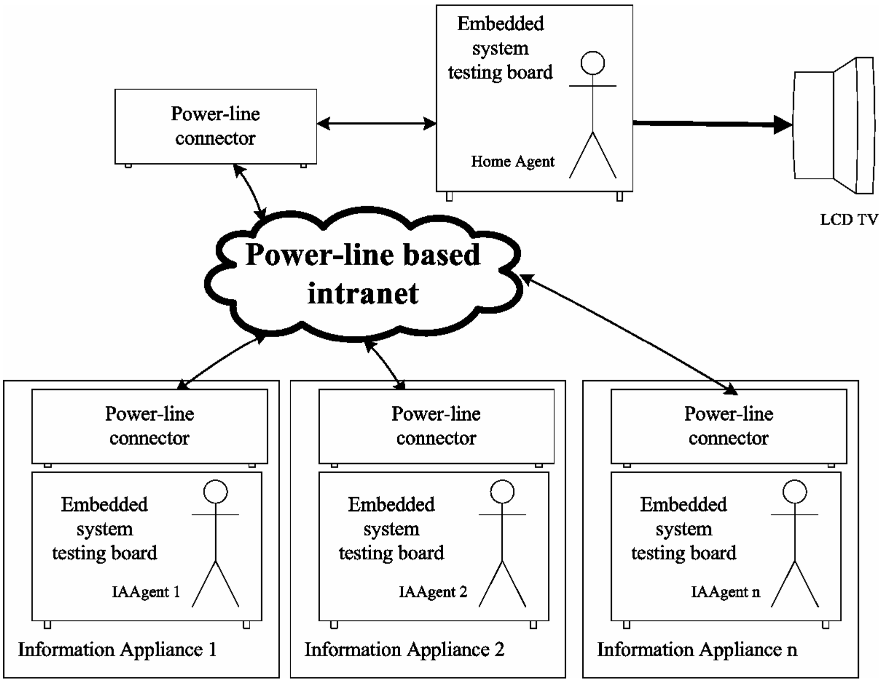

Fig. 8. The deployment diagram of IAAgents

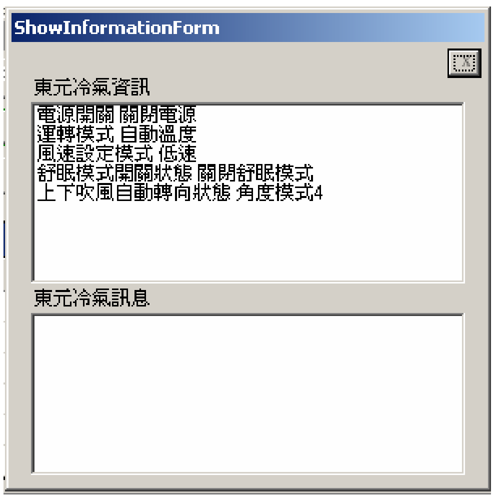

Fig. 9. The GUI for appliance messages

In the second scenario, we assume that there are one air-condition and one dehumidifier in a room. When temperature is higher than 26 ℃, air-condition will be turned on. In contrast, if the temperature is lower than 22 ℃, air-condition will be turned off. Furthermore, when humidity is higher than 55 %, the dehumidifier will be turned on. In contrast, if the humidity is lower than 50%, the dehumidifier will be turned off. If the air-condition is running, the dehumidifier will never be turned on. That's because air-condition can dehumidify when it is running.

In the third scenario, we assume that there are five appliances at home: TV and air-condition in the living room, air-condition in the bedroom, refrigerator in the kitchen room, and washing machine in the balcony. Because air-condition, washing machine, and refrigerator needs more power when the compressor or motor is running. Hence, do not let these three kinds of appliances running at the same time is better. When the compressor of air-condition is off, compressor of the refrigerator and motor of the washing machine can work. If the compressor of air-condition is turned on and the agent find that washing process of the washing machine is almost done, air-condition can delay for waiting the washing machine. By using the communication and negotiation capabilities of the IIA, the appliances can make a better schedule for doing their works and do not violate power usage constrain.

6 Conclusions

We proposed an IA agent template for setting up an intelligent environment. Our IA agent combines the deliberative reasoning module and the reactive module to accomplish the core part of the agent. The reasoning module can provide a complete decision making method, and the reactive module that can increase the near real time response. Through these two modules, the IA agent can solve emergent and complex tasks in intelligent environment. In a smart home, the IA agents which based on the proposed agent template can cooperate with each other to achieve a mutual goal and shares information to complete more complex tasks. In the analysis method, we use software engineering methods to analyze our architecture. Therefore, we provide a method and a suitable architecture for the users who want to have an existing environment turn into an intelligent environment.

Acknowledgement

This research is a part of the Reactive Module for Intelligent Agent of Smart Home project which is supported by the Industrial Technology Research Institute, Taiwan.

References

[1] R. Want and G. Borriello, "Survey on Information Appliances," *IEEE Computer Graphics and Application*, Vol.20, No.3, pp.24-31, 2000.

[2] T. Okugawa, H. Masutani, and I. Yoda, "A Home Network Service Environment for Wide-Area Communications," in *Proceedings of Asia-Pacific Conference on Communications*, pp.14-18, 2005.

[3] Z.G. Deng and H.S. Hu, "A framework for mobile-service based co-ordination of embedded web agents in intelligent buildings," i*n Proceedings of IEEE ASME 2001*, Vol.1, pp.284-289, 2001.

[4] V. Callaghan, M. Colley, G. Clarke, and H. Hagras, "Embedding intelligence: Research issues for ubiquitous computing," i*n Proceedings of the 1st Equator IRC Workshop on Ubiquitous Computing Environments*, pp.110–130, 2001.

[5] N.S. Liang, L.C. Fu, and C.L. Lin, "An integrated, flexible, and Internet-based control architecture for home automation system in the Internet era," *In Proc. of IEEE ICRA02*, Vol.2, pp.1101-1106, 2002.

[6] H. Hagras, et al, "Creating an ambient-intelligence environment using embedded agents," *IEEE Intelligent Systems*, Vol.19, No.6, pp.12-20, 2004.

[7] C. L. Wu, C. F. Liao, and L. C. Fu, "Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology," *IEEE Trans. on Systems, Man and Cybernetics, Part C*, Vol.37, No.2, pp.193-205, 2007.

[8] D. W. Kim and D. J. Kim, "An Intelligent Smart Home Control Using Body Gestures," in *Proceedings of the ICHIT 2006*, Vol.2, pp.439-446, 2006.

[9] T. Ma, et al, "Context-aware implementation based on CBR for smart home," i*n Proceedings of IEEE WiMob2005*, Vol.4, pp.112 – 115, 2005.

[10] Y. J. Lin, H. A. Latchman, M. Lee, and S. Katar, "A power line communication network infrastructure for the smart home," *IEEE Wireless Communications*, Vol.9, No.6, pp.104-111, 2002.

[11] M. N. Huhns, "Networking embedded agents," *IEEE Internet Computing*, Vol.3, No.1, pp.91-93, 1999.

[12] H. Hagras and M. Colley, "The role of embedded agents in creating ambient intelligent environments," In *Proceedings of IEE Seminar on Intelligent Building Environments*, pp.127-145, 2005.

[13] M. Wooldridge, "Agent-based computing," *Interoperable Comm. Networks*, Vol.1, No.1, 1998, pp.71-97, 1998.

[14] Y. Iraqi, R. Boutaba, "A Multi-agent System for Resource Management in Wireless Mobile Multimedia Networks," *LNCS*, Vol.1960, pp.0218.

[15] B. Ahn, H. Cha, K. Cho, "A QoS management platform for multicast environments," *In Proc. of IWS 99*, pp.112 -118, 1999.

[16] J.A. Lay, P. Muneesawang, L. Guan, "Multimedia information retrieval," *Electrical and Computer Engineering*, Vol.1, pp.619-624, 2001.

[17] E. C. L. Wai and Q. Li, "Media-on-Demand for Agent-Based Collaborative Tutoring Systems on the Web," *LNCS*, Vol.2532, pp.976-984, 2002.

[18] P. Bellavista and A. Corradi, "How to support Internet-based distribution of video on demand to portable devices,**"** *In Proc of ISCC 2002*, pp.126-132, 2002.

[19] F. Rivera-Illingworth, V. Callaghan, and H. Hagras, "A connectionist embedded agent approach for abnormal behavior detection in intelligent health care environments," in *Proceedings of IEEE SMC2004*, Vol.4, pp.3565-3570, 2004.

[20] N. Kushwaha, M. Kim, D. Y. Kim, and W. D. Cho, "An intelligent agent for ubiquitous computing environments: smart home UT-AGENT," in *Proceedings of IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, pp.157-159, 2004.

[21] F. Doctor, H. Hagras, and V. Callaghan, "A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments," *IEEE Trans. on Systems, Man and Cybernetics, Part A,* Vol.35, No.1, pp.55-65, 2005.

[22] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 Fuzzy Logic Systems," *IEEE Trans. on Fuzzy Systems*, Vol.7, No.6, pp.643-658, 1999.

[23] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez, "An Incremental Adaptive Life Long Learning Approach for Type-2 Fuzzy Embedded Agent in Ambient Intelligent Environments," *IEEE Trans. on Fuzzy Systems*, Vol.15, No.1, pp.41-55, 2007.

[24] S. Lee and T. Kim, "Generic Engineering Approach for Agent-Based System Development," in *Proceedings of MATA 2002, LNCS,* Vol.2521, pp.58-71, 2002.

[25] A. Tveit, "A Survey of Agent-Oriented Software Engineering," *Report, Norwegian University of Science and Tech.,* May 2000.

[26] C. Petrie, "Agent-Based Software Engineering," *In Proceedings of AOSE 2000, LNCS,* Vol.1957, pp.59-75, 2001.

[27] M. Wooldridge, *Reasoning about rational agents*, MIT Press, London, 2000.

[28] http://www.fipa.org/repository/aclspecs.html

[29] S. Russel and P. Norvig, *Artificial Intelligence: A Mordern Approach 2ed.*, Prentice Hall, London, 2003.