System Modeling of a Smart-Home Healthy Lifestyle Assistant

Xinhua Zhu[1], Yaxin Yu[2], Yuming Ou[1], Dan Luo[1], Chengqi Zhang[1], and Jiahang Chen[1]

[1] AAI, QCIS, FEIT, University of Technology, Sydney, Australia
{Xinhua.Zhu,Yuming.Ou,Dan.Luo,Chengqi.Zhang,Jiahang.Chen}@uts.edu.au
[2] College of Information Science and Engineering, Northeastern University, China
Yuyx@mail.neu.edu.cn

Abstract. A system modeling is presented for a Smart-home Healthy Lifestyle Assistant System (SHLAS), covering healthy lifestyle promotion by intelligently collecting and analyzing context information, executing control instruction and suggesting health plans for users. SHLAS is Multi-agent based. Each agent has three levels: the Goal Layer has business rules for representing agent goals; the Strategy Layer provides technical rules and processes for guiding how the agent reacts to events; the Component Layer is made up of components, some components are called by technical rules and processes in the Strategy Layer, some others are used for communicating with third party systems. This agent framework enables the customizability of agents in SHLAS. We also introduce an Ontology-based domain knowledge and context model to capture and represent the agents, and agent behavior which provides agents with reasoning ability. SHLAS helps users with healthy lifestyle promotion by tracking and analyzing their behaviors, and recommending health plans. The paper closes with an empirical evaluation of the approach from the point of view of customizability.

Keywords: Multi-agent, Agent behavior analysis, Customizability, Planning.

1 Introduction

The smart home is always a hot topic in both academe and industry. Science fiction has imagined an idealized vision of a fully integrated smart home, where all the operations of a house can be efficiently controlled by a central application. A good smart home system should be an excellent assistant to home life, able to collect and analyze a range of information from people's daily lives and use it to optimize their living environment. It should be able to flexibly control home appliances, monitor people's health status, advise of any abnormalities, develop personalized health programs for each family member, and push users to execute such plans. Using smart home systems, people would have ease of access to information and services that will improve health and quality of life.

L. Cao et al.: ADMI 2012, LNAI 7607, pp. 65–78, 2013.

A significant feature of a smart home is intelligence. According to Wikipedia, Intelligence has been defined in different ways, including, but not limited to, the ability to exercise abstract thought, learning, understanding, self-awareness, memory, reasoning, planning, communication, emotional knowledge, and problem solving[1]. This is not the case with current smart home products. Most existing smart home systems are not intelligent enough. The main issues include:

1) Oversimplified processing logic. The current smart home systems such as MavHome [8] rely heavily on sensor data acquisition and the passive acceptance of clear control commands, and lack intelligent and complex human-computer interaction and reasoning ability. It is hard for users to convey their intentions in a natural way through speech, text or gesture.

2) Poor flexibility and scalability. Different families may have different requirements and they may change their requirements frequently. Moreover, devices may be added and removed during runtime and they may fail due to connectivity problems. As a result, there is a need for an open complex agile architecture to deal with all these circumstances.

3) Insufficient focus on healthy lifestyle promotion. In the final analysis, people's health mostly depends on their daily behaviors. Early awareness of inappropriate habits could help to improve health and prevent disease. However, to the best of our knowledge, there is limited research studying the smart home from the perspective of healthy lifestyle promotion.

The paper proposes a Smart-home Healthy Lifestyle Assistant System (SHLAS). In respect of the first problem above, oversimplified processing logic, SHLAS draws on the Cyber-physical system (CPS) [2] theory. CPS is a multi-dimensional complex intelligent system integrated with computing, networking and the physical environment. Real-time perception, dynamic control, and information services of large-scale engineering CPS can be implemented by use of 3C (Computation, Communication, Control) technologies. In CPS, a key feature is context-aware computing which focuses on the collection, modeling, and intelligent processing of context information [1].Our SHLAS makes full use of context-aware computing. At the information collection stage, similar to the induction of the human sensory organs to the objective world, SHLAS obtains context information related to people and equipment through human-computer interaction and a Wireless Sensor Network (WSN). At the modeling stage, the collected multi-source heterogeneous information is transformed, represented and stored effectively. At the intelligent processing stage, useful and meaningful know-ledge is discovered from the collected information so that SHLAS can understand users' behaviors and intentions. Finally context-aware services are provided.

In terms of the second problem, poor flexibility and scalability, SHLAS introduces an open, complex, agile multi-agent architecture model. Agent is the basic execution unit in SHLAS. Agents are grouped in accordance with their goals and

[1] http://en.wikipedia.org/wiki/Intelligence/
[2] http://en.wikipedia.org/wiki/Cyber-physical_system/

events are used in agent communication. Each agent is modeled carefully to regulate its behavior and interface. An agent model can be divided into three layers: the Component Layer, Strategy Layer and Goal Layer. The Component Layer provides a variety of internal function components called by the Strategy Layer and external interface components for sensor data collection, device control and software control; the Strategy Layer defines basic agent behavior logics through processes and rules; and the Goal Layer defines high level business rules written in natural language to represent agents' goals. Agent behaviors are highly customizable by updating rules and processes, which ensures flexibility on the basis of commonality.

To address third problem of healthy lifestyle promotion, a Health Promotion Procedure Model (HPPM) is introduced which combines health domain knowledge, context, intelligent planning [2] and behavior informatics [3] [4]. HPPM is an iterative four-step management model: planning, reminder, behavior monitoring and performance evaluation. Four important factors are considered in the planning stage: health promotion knowledge; historical behavior analysis; current status; and health goals.

The paper is organized as follows. In Section 2, the concept of a smart environment, typical projects and related tools are discussed. Section 1 introduces the architecture of SHLAS. Section 4 proposes the ontology-based domain knowledge and context modeling. We illustrate two typical application scenarios in Section 5. Several key points in implementation are discussed in Section 6. We conclude the paper in Section 7.

2 Related Work

A smart environment should be able to acquire and apply knowledge about its inhabitants and their surroundings in order to adapt to the inhabitants and meet the goals of comfort and efficiency [5]. It has further been stated that a smart environment can reduce the amount of interaction required by inhabitants, reduce energy consumption and limit other potential waste, and provide a mechanism for ensuring the health and safety of the environment's occupants [6].

EasyLiving [7] at Microsoft Research supports smart environments through the dynamic interconnection of a variety of devices. This middle-ware supports mechanisms such as inter-system communication, location tracking for objects and people, and visual perception. One issue confronting the EasyLiving system is that it only focuses on room control and lacks residential health management.

The MavHome Smart Home project [8] is a multi-disciplinary research project at Washington State University and the University of Texas at Arlington which focuses on the creation of an intelligent home environment. It views the smart home as an intelligent agent that perceives its environment through the use of sensors and can act upon the environment through the use of actuators. MavHome provides many excellent features such as resident behavior analysis and action prediction, but makes no effort to improve flexibility; for instance, how to adjust agent behavior easily.

Rule based tools can provide a smart home system with powerful flexibility. A Business Logic Integration Platform called Drools has been introduced which provides a unified and integrated platform for Rules, Workflow and Event Processing. It has been designed from the ground up so that each aspect is first class, with no compromises [3]. The Drools Planner is used to optimize automated planning by combining search algorithms with the power of the Drools rule engine[4]. All Drools components are used in SHLAS for agent behavior representation and agent communication.

Data Mining and Multi-agent Integration [11] [12] [13] presents cutting-edge research, applications and solutions in data mining. This will improve smart home system on both performance and intelligence.

3 Architecture of SHLAS

Agent is the basic execution unit in SHLAS. As shown in Fig. 1, there are three groups of agents in SHLAS: the Core Agent Group, Interface Agent Group and Service Agent Group. We draw on the agent service abstract model [9] to develop a theoretical model of a smart home agent.

A smart home agent is represented by attributes such as name, type, locator, owner, roles, behavior, protocol, address, event, input variables, pre-conditions, output variables, post-conditions and exception handling.

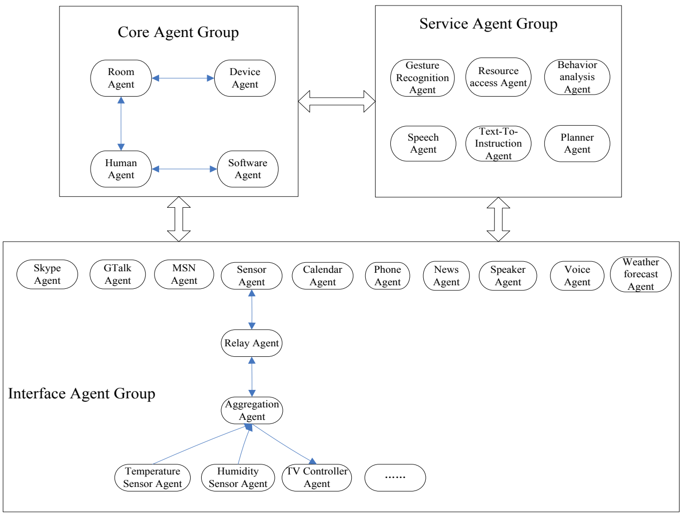

Fig. 1. SHLAS Architecture

[3] http://www.jboss.org/drools/

[4] http://docs.jboss.org/drools/release/5.3.0.Beta1/drools-planner-docs/html_single/#d0e26/

$$< Smart - HomeAgent >:: = f(Name; Type; Behavior; Protocol; Locators; Owner; Roles; Address; Event; InputVariables; Preconditions; OutputVariables; Postconditions; Exception). \quad (1)$$

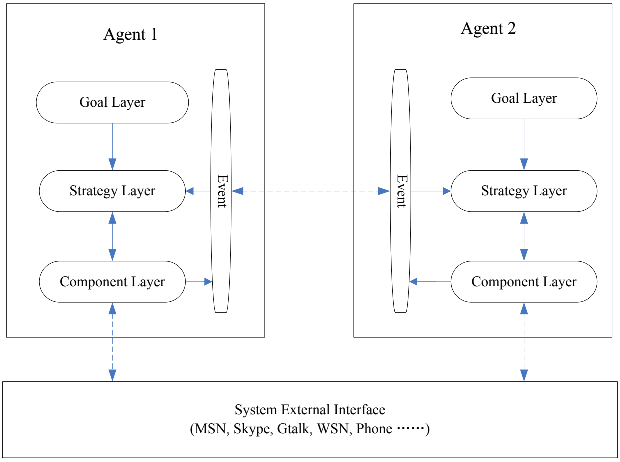

Fig. 2. Smart-Home Agent Model

Behavior is one of the most important elements in the Smart-Home Agent Model. As shown in Fig. 2, behaviors are redistributed into three layers of agent architecture, namely, Goal Layer, Strategy Layer and Component Layer.

Table 1. Agent functional patterns of core Agent Group

Agent name	Behavior
Human Agent	Represent member states, determine control privilege, and analyze behavior patterns.
Room Agent	Represent room states, create control-instruction event to improve room environment or alert user if any abnormality.
Device Agent	Represent device states and control device actions.
Software Agent	Represent software states and control software actions.

– In Goal Layer, high level business rules written in natural language are used to express agent goals. It enables even end users to adjust agent behavior easily by updating the business rules.

- Strategy Layer defines the elements such as technical rules and processes to represent agent behavior. Parts of the technical rules are converted from business rules in the Goal Layer. Others are defined by developers. All the processes and rules are easy to update through a GUI tool. Processes and rules can be triggered by events for agent communication.
- Component Layer provides a variety of basic components whose logic will not be changed frequently. Some components are called by processes and rules in the Strategy Layer. Others interact with system external interfaces, for instance, to collect sensor data, to send control instructions, to interact with MSN or Skype, to call web service, and so on.

Table 2. Agent functional patterns of Interface Agent Group

Agent name	Behavior
Sensor Agent	Transfer data between wireless sensor network (WSN) and PC
Voice Agent	Get voice from microphone for speech recognition and voiceprint recognition.
Camera Agent	Get video from camera for gesture recognition.
Skype Agent	Communicate with people via Skype.
GTalk Agent	Communicate with people via GTalk.
MSN Agent	Communicate with people via MSN.
Phone Agent	Communicate with people via Phone.
Calendar Agent	Get and set Calendar Item.
News Agent	Get news from news RSS websites.
Weather Forecast Agent	Get weather forecast information from weather forecast RSS websites.
Speaker Agent	Play speech.
Relay Agent	Relay transferred data.
Aggregation Agent	Connect and transfer data between Terminal Sensor Agent and Relay Agent.
Terminal Sensor Agent	Collect sensor data and control devices.

Family members, rooms, domestic devices and software applications are the major management objects in SHLAS. As shown in Table 1, the Core Agent Group consists of Human Agent, Room Agent, Domestic Device Agent and Domestic Software Agent.

The Interface Agent Group deals with data exchange between SHLAS and third party systems. As shown in Table 2, the Interface Agent Group includes

Table 3. Agent functional patterns of Service Agent Group

Agent name	Behavior
Gesture Recognition Agent	Map gesture to instruction.
Resource Access Agent	Get context resource.
Behavior Analysis Agent	Analyze people behavior.
Speech Agent	Map speech to text.
Text-To-Instruction Agent	Map text to instruction.
Planner Agent	Make health plan.

Sensor Agent, Speech Agent, Camera Agent, Skype Agent, GTalk Agent, MSN Agent, Phone Agent, Calendar Agent, and more.

As shown in Table 3, the last group is the Service Agent Group, which comprises Gesture Recognition Agent, Resource Access Agent, Behavior Analysis Agent, Speech Agent, Text-To-Instruction Agent and Planner Agent. These agents provide passive services for other agents to call.

4 Ontology Based Domain Knowledge and Context Modeling

Domain knowledge and context are significant resources in SHLAS. As shown in Fig. 3, we overview the Ontology Based Domain Knowledge and Context Model.

Domain knowledge is composed of terms and their relationships. For instance, in the domestic domain, there are open spaces (such as the balcony and garden) and rooms (such as the bedroom, toilet, living room, etc). A living room may contain devices such as TV, air conditioner, washing machine, etc. A TV has instructions such as turn on/off, channel tune and volume tune. An authorized family member may execute particular instructions. Fig. 4 shows parts of the domain knowledge; users can construct a domain knowledge model by dragging and dropping a word from the left column to the right column, naming it, and linking it to other terms.

Some special terms can be further defined as object templates represented in an xml scheme (XSD file). Fig. 5 shows a TV scheme template example. The template describes the object name, instructions and corresponding instruction execution method, and so on.

Context represents the instance of domain knowledge. For example, Family A has a master bedroom and a secondary bedroom; in master bedroom there are a SHARP TV and a TOSHIBA Air Conditioner, etc. Context information can be easily managed. As shown in Fig. 6, when an object type is selected, such as TV, information in the corresponding XSD scheme will be extracted and displayed on the GUI, which is very convenient for users because they do not need to spend much time defining all the objects themselves.

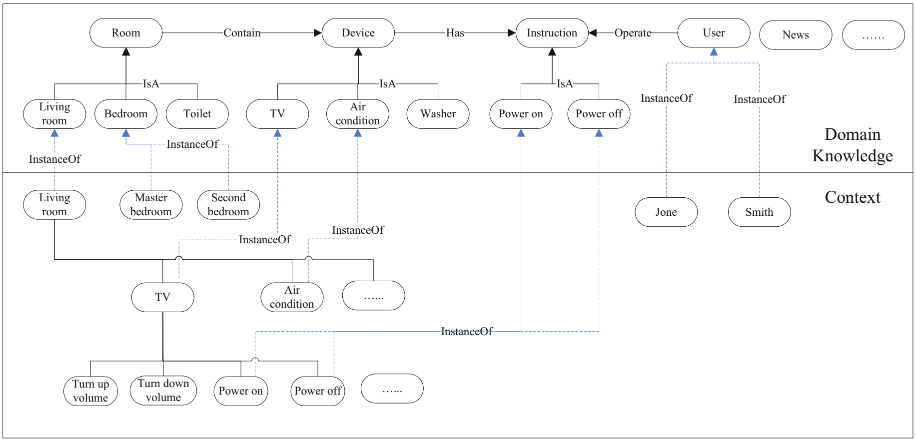

Fig. 3. Domain knowledge and context model

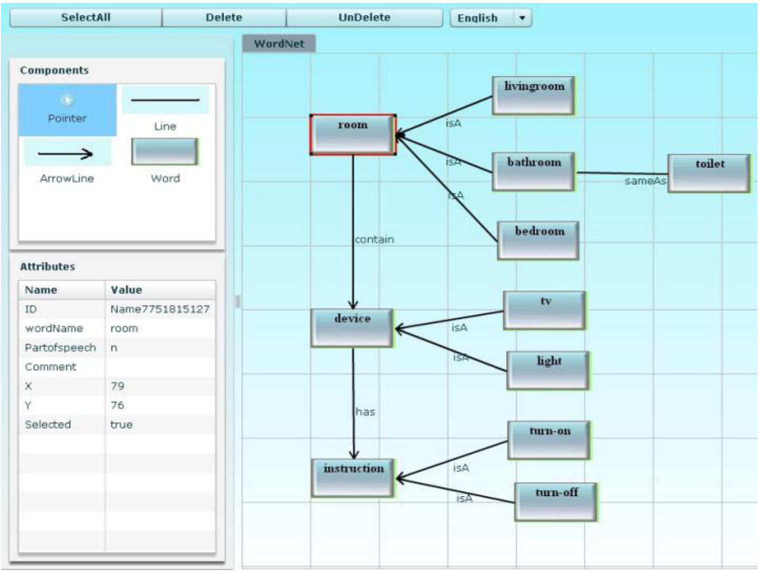

Fig. 4. Domain Knowledge Management

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="SHLAS.Schema"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="SHLAS.Schema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
>
<xsd:element name="TeleVision">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Instructions">
                <xsd:complexType>
                    <xsd:all>
                        <xsd:element name="TurnOn">
                            <xsd:complexType>
                                <xsd:sequence>
                                    <xsd:element name="JavaBeanName" type="xsd:string" fixed="SHLAS.Instruction.InfaredSend"></xsd:element>
                                        <xsd:element name="Parameters">
                                            <xsd:complexType>
                                                <xsd:sequence>
                                                    <xsd:element name="Port">
                                                        <xsd:complexType>
                                                            <xsd:attribute name="value">
                                                                <xsd:simpleType>
                                                                    <xsd:restriction base="xsd:integer">
                                                                    </xsd:restriction>
                                                                </xsd:simpleType>
                                                            </xsd:attribute>
                                                        </xsd:complexType>
                                                    </xsd:element>
                                                </xsd:sequence>
                                            </xsd:complexType>
                                        </xsd:element>
                                        <xsd:element name="Roles">
                                            <xsd:complexType>
                                                <xsd:attribute name="Type" default="Adult">
                                                </xsd:attribute>
                                            </xsd:complexType>
                                        </xsd:element>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
```

Fig. 5. XSD template of TV

Domain knowledge and context are used in reasoning. Details are described in Section 6.

5 Scenario Analysis

In this section, two typical scenarios and the corresponding agent behaviors are introduced to illustrate how SHLAS works. Generally speaking, people's health includes physical health, mental health, social relations health, and so on. Whether or not their daily lifestyle is rational will directly affect a person's health. Through behavioral analysis, habits can be acknowledged and personalized services can be recommended to help the member achieve a healthy lifestyle. Scenario 1 is an example of this.

Scenario 1: James worked hard all week. At the weekend, he slept until noon. When he woke, he said, "Open the curtain". The system knew that he was in the bedroom, so the cur-tain in the bedroom was opened. The system recognized that James got up later than usual, so he was reminded: "You got up a little later than usual; please take care of your sleep quality". James always bathed and listened to music after getting up, so the shower was powered on automatically and he was asked: "Would you like to listen to music?" If the answer is "Yes", the system collects and plays a music list based on his historical preference.

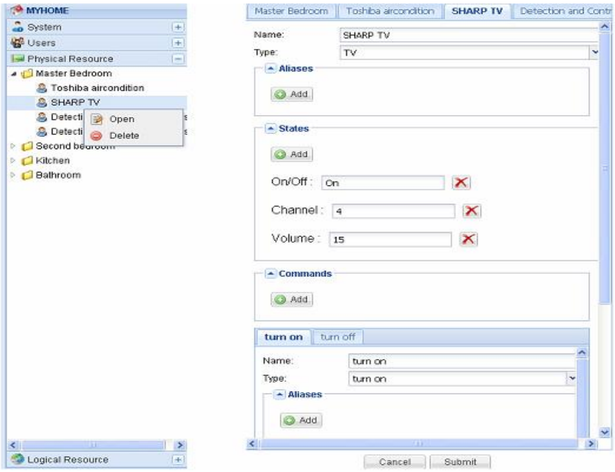

Fig. 6. Context information management

Fig. 7 shows the agent behavior diagram in Scenario 1. The figure omits the message passing procedure from Sensor Agent to Terminal Sensor Agent via Relay Agent and Aggregation Agent.

Scenario 1 is an application of behavior pattern analysis and personalized recommendation. To realize the scenes, SHLAS needs access to a variety of contexts, such as the basic states of the family members, behavioral preferences and environment information. To obtain behavioral preferences, SHLAS uses the FP-growth algorithm [10] to analyze the accumulated historical behavior data regularly in advance, and then stores the identified behavior patterns. Based on these behavior patterns, SHLAS determines the impact factors of health and recommends the most appropriate activities to family members.

It is rarely easy to achieve a long-term healthy state without any expertise or professional guidance. Diet and fitness rules are stored in SHLAS. Professional health plans can be established based on these rules if there are health requirements. Scenario 2 is an example.

Scenario 2: James stood on a weighing scale. The system found that he weighed 2kg more than last month and was 1kg more than standard weight, so James was told: "You are 1kg overweight. Would you like a health plan?" If the answer is "yes", the system collects information about his current status, analyzes his historical behaviors and goals, and generates a three-month plan. The plan is generated into Google Calendar. James can make modifications through any web browser. In the following three months, James will be reminded by speaker or email to execute the planning items wherever he is at home or in the office. James's behavior will be monitored and his ongoing performance will be evaluated for plan refinement.

Fig. 8 shows the sequence diagram of generating a health plan.

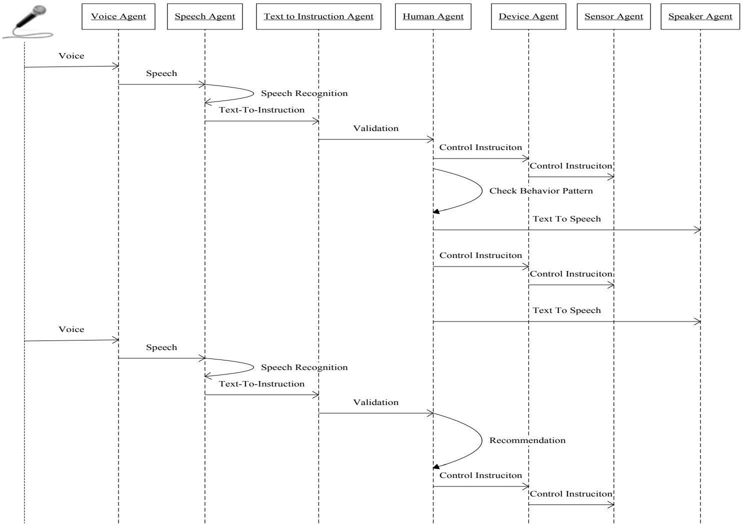

Fig. 7. Agent behavior in Scenario 1

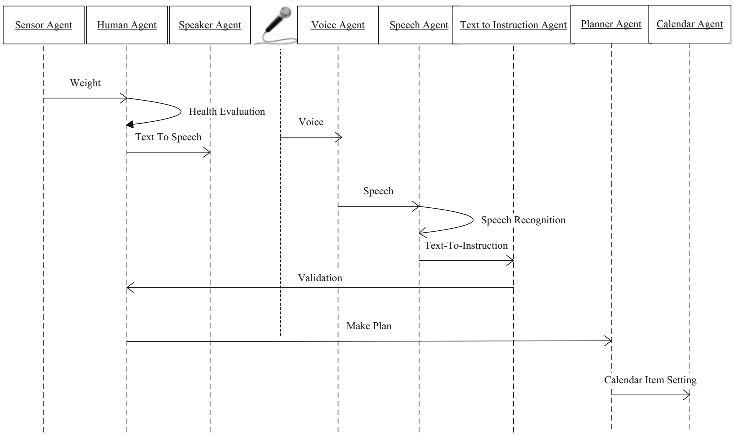

Fig. 8. Health plan generation in Scenario 2

SHLAS provides users with a complete Health Promotion Procedure Model (HPPM) which includes planning, reminding, monitoring, evaluating and plan refining. Scenario 2 is an application scenario of HPPM. Planning is based on the rules related to each family member's behavior patterns, current health status, home environment, expectations for health, and so on. The following constraints in the rules are especially important:

– Hard constraint which must not be broken. For example, only one exercise program for each member can be in progress at any given time.

– Weak constraints which should not, as far as possible, be violated. For example, daily running time cannot be more than one hour.
– Rewards which should be fulfilled as far as possible. For example, walking is the most recommended exercise program for the elderly.

SHLAS develops scoring rules for the above-mentioned constraints, and the rules are used to score a number of health plans generated automatically. One or more sets of plans with higher scores will be used by family members.

From the two scenarios, it is obvious that SHLAS comprehensively helps users with the promotion of a healthy lifestyle.

6 Implementation

Agents in SHLAS are implemented in Java Thread. Drools Expert provides the rule engine, while JBPM provides the process engine.

In the Component Layer of Agent, the components are written in JavaBean. A component interface specification is defined so that updating a component will not cause damage to other components.

In the Strategy Layer, agent behavior can be refactored by tuning the processes and rules of Drools. Fig. 9 shows two rules defined in the Text-To-Instruction Agent. The first rule is to segment text into phrases, while the second is to create an instruction event from phrases. Suppose the agent receives a text event: "Please turn on the TV in the master bedroom", the first rule will segment the text into three phrases, "turn on", "TV", "master bedroom", and will retrieve their types from the domain knowledge repository: the type of "turn on" is "instruction; the type of "TV" is "device"; and the type of "master bedroom" is "room". The second rule will then build up a completed instruction event. The event will trigger a rule in Human Agent to verify the privilege and finally will trigger a rule in Device Agent to execute the control action. All classes (e.g., PhraseEvent, InstructionEvent) and methods (e.g., segment) are defined in the Component Layer of the Text-To-Instruction Agent.

The rules can be refined by a developer or advanced user in runtime and enabled immediately.

In the Goal Layer, end users can modify business rules which are easier to under-stand than technical rules. The business rules in the Goal Layer will be translated into technical rules automatically. Fig. 10 shows a business rule defined in Room Agent; the first line is business rule, the second and the third lines contain the corresponding technical rule. During runtime, end users can only access the business rule. Once end users update "number" in the business rule to another value, the agent behavior will change correspondingly.

Compared to other smart home approaches, SHLAS is special in treating refinement as an optimization problem. Elements in the three layers are updatable and can be accessed by three levels of customers. A developer can refine the low level behavior of the agent by updating the source code of components. An advanced user can refine the medium level behavior of an agent by managing technical

```
rule "Segment text into phrases"
    when
        $text:TextEvent()
    then
        String[] phrases = segmentText($text.getContent());
        for(String phrase:phrases){
            PhraseEvent phraseEvent = new PhraseEvent();
            phraseEvent.setContent(phrase);
            phraseEvent.setType(getType(phrase));
            phraseEvent.setSource($text.getSource());
            insert(phraseEvent);
        }
end

rule "Create instruction event from phrases"
    when
        $room:PhraseEvent(type == PhraseType.ROOM)
        $device:PhraseEvent(type == PhraseType.DEVICE && source == $room.getSource
        $command:PhraseEvent(type == PhraseType.COMMAND && source == $room.getSourc
    then
        InstructionEvent instructionEvent = new InstructionEvent();
        instructionEvent.setLocation($room.getContent);
        instructionEvent.setDevice($device.getContent);
        instructionEvent.setCommand($command.getContent);
        instructionEvent.setSource($room.getSource());
        insert(instructionEvent);
end
```

Fig. 9. Rules in Text-To-Instruction Agent

processes and rules. A general end user can refine the high level behavior of the agent by managing business rules.

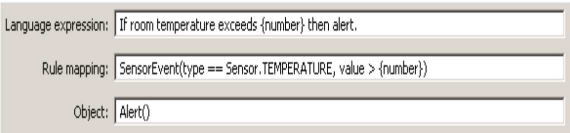

Fig. 10. Business rule and corresponding technical rule

7 Conclusion

This paper proposes a SHLAS system which can assist family members in the promotion of a healthy lifestyle by collecting and analyzing context information, executing control instruction and making health plans.

A multi-modal human-computer interaction style, healthy domain knowledge, and model-based and rule-based reasoning ability make SHLAS intelligent. Multi-agent architecture and a component-strategy-goal agent framework make SHLAS behavior much easier to customize than other smart home approaches. SHLAS helps users to promote a healthy lifestyle by making health plans and reminding the user to enact the plan. Health planning is based on health domain knowledge, collected context information and users' goals. Compared to other smart home approaches, the agent behavior in SHLAS is easier to refine for on-demand requirements due to the flexible multi-agent architecture used in this approach. In fact, this approach suggests a general architecture for managing SHLAS which can easily be extended to other intelligent spaces, such as hospital, office, school, and restaurant, as long as the agent behavior and domain knowledge are updated correspondingly. Our future work will focus on human computer interaction (e.g., emotion recognition) and agent behavior impact analysis (e.g., how the behaviors of Human Agent, Room Agent and Device Agent impact on one another).

References

1. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: IEEE Workshop on Mobile Computing Systems and Applications, WMCSA 1994, Santa Cruz, CA, pp. 89–101 (1994)
2. Wilkins, D., Desjardins, M.: A call for knowledge-based planning. AI Magazine 22(1), 99–115 (2001)
3. Cao, L., Yu, P.S. (eds.): Behavior Computing: Modeling, Analysis, Mining and Decision. Springer, New York (2012)
4. Cao, L.: Behavior informatics and analytics: Let behavior talk. In: IEEE International Conference on Data Mining Workshops, ICDMW 2008, pp. 87–96 (2008)
5. Cook, D., Das, e.S.: Smart Environments: Technologies, Protocols and Applications. Wiley (2004)
6. Das, S.K., Cook, D.: Health monitoring in an agent-based smart home. In: Proceedings of the International Conference on Smart Homes and Health Telematics, ICOST, Singapore (September 2004)
7. Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S.: EasyLiving: Technologies for Intelligent Environments. In: Thomas, P., Gellersen, H.-W. (eds.) HUC 2000. LNCS, vol. 1927, pp. 12–29. Springer, Heidelberg (2000)
8. Das, S.K., Cook, D., et al.: The role of prediction algorithms on the MavHome Smart Home Architectures. IEEE Wireless Communications (Special Issue on Smart Homes) 9(6), 77–84 (2002)
9. Cao, L., Zhang, C., Zhou, M.C.: Engineering open complex agent systems: A case study. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews 38(4), 483–496 (2008)
10. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation. Data Mining and Knowledge Discovery 8, 53–87 (2004)
11. Cao, L.: Data Mining and Multi-agent Integration (edited). Springer (2009)
12. Cao, L., Weiss, G., Yu, P.S.: A Brief Introduction to Agent Mining. Journal of Autonomous Agents and Multi-Agent Systems 25, 419–424 (2012)
13. Cao, L., Gorodetsky, V., Mitkas, P.A.: Agent Mining: The Synergy of Agents and Data Mining. IEEE Intelligent Systems 24(3), 64–72 (2009)