2005 한국경영과학회/대한산업공학회 춘계공동학술대회
2005년 5월13일~14일, 충북대학교

유비쿼터스 헬스케어서비스를 위한 목표기반 지식에이전트 접근법

김지홍[1], 하병현[1], 강석호[1], 이우기[2], 김철영[3], 허원창[3]
[1] 서울대학교 산업공학과
valentine@ara.snu.ac.kr, pepper@netopia.snu.ac.kr,shkang@snu.ac.kr
[2] 성결대학교 컴퓨터학부
wook@sungkyul.edu
[3] 사이버메드, 강남구 논현동
kaster@cybermed.co.kr, hwc@cybermed.co.kr

Goal-Based Knowledge Agent Approach for Ubiquitous Healthcare Services

Ji-Hong Kim[1], Byung-Hyun Ha[1],Wookey Lee[2], Suk-Ho Kang[1]
[1] Department of Industrial Engineering, Seoul National University,Seoul 151-742, Republic of Korea
[2] Department of Computer Science, Sungkyul University,Anyang, Korea
[3] CyberMed, Inc., 5F Won Bldg., Nonhyeon-dong, Gangnam-gu,Seoul 135-814, Korea

Abstract. Ubiquitous Computing has introduced one of the most innovative international research contributions on the design and evaluation of new generations of handheld and mobile information appliances. New devices are being invented and existing devices are being improved with smaller size and increased mobility. These changes of computing paradigm are enabling the enterprises' legacy services to be automated and value-added all the more. We suggest a service framework and algorithms of provisioning healthcare services in a ubiquitous computing environment. We construct domain ontology that is related to diabetes and minimum common sense ontology for service composition. The results of this research enable integration and interconnection of devices, applications, and functions of enterprises within the healthcare services.

1 Introduction

Ubiquitous Computing has introduced one of the most innovative international research contributions on the design and evaluation of new generations of handheld and mobile information appliances. Since late 90' it has provided a global perspective on new developments, uniting technical accounts with studies of the social, cultural and organizational impacts of new personal technologies. With ubiquitous computing the quality of human life can be improved by interoperation among various devices and services. [10] Invisible devices, applications, and services of enterprise are interconnected and accessed transparently through communication networks, thus they together can provide users with beneficial services. These changes of computing paradigm are enabling the enterprises' legacy services to be automated and value-added all the more.

To realize the new computing paradigm, various efforts are being conducted. New devices are being invented and existing devices are being improved with smaller size and increased mobility. New network and protocol technologies that form the basis of interoperability are also being

developed. By the help of these efforts, simple and direct needs of users, e.g. turning on and off lights or playing music automatically according to a user's preference, can be easily accomplished. However when the needs are indirect and the tasks that fulfill the needs are complex, we need the ingenious mechanism that coordinates a variety of objects including the users themselves. [5]

When services are modeled for the functionalities of devices, the services need to be combined dynamically according to the situation using knowledge base in other to achieve high-level requirements of users. The reasons for dynamic composition of services can be stated with three perspectives of users, devices, and domains.

First, at user perspective, the needs of users in reality seem too diverse to define all possible schemes of services in advance. If we define every service scheme statically, it will be have limited scalability. This is because almost all services need to be redefined, even when the details of user's needs change slightly. At the device perspective, the availability of devices and applications making up ubiquitous environment changes according to time and place. Therefore the services that they can provide are impossible to predict and it is not applicable to assume fixed conditions. Lastly, in order to apply one successfulsystem in a certain domain to anther domain, the system requires to be designed in a generic form. If a system is implemented using generic rules and domain-specific knowledge to meet the needs of users, the system can be easily applicable in another domain only after modifying the domain-specific knowledge. As a result, it is important to compose services dynamically based on predefined knowledge to build successful ubiquitous computing environment and it can become more important if enterprises are to efficiently provide ubiquitous services on a commercial scale.

In this research we suggest a service framework and algorithms of provisioning healthcare services in a ubiquitous computing environment. The results enable integration and interconnection of devices, applications, and enterprises'legacy services related to healthcare services.

The framework is composed of atomic services, user's goals, goal-based agents, dynamic service composition algorithm, and knowledge bases. Currently most researches of composing services dynamically depend on the conventional planning methods of Artificial Intelligence domain. Basically these approaches consider the inputs, outputs, preconditions, and effects of services and then search the plans of achieving a user's goal. However they have the limitation of computational feasibility. Incase there are services at various levels or unavailable, they do not function properly. To overcome these weaknesses we repeatedly refine the user's goal into sub-goals with commonsense knowledge until there are appropriate services for sub-goals and after, employ the services. This approach stands on the Mikrokosmos ontology [8] which enables querying and answering based on shared knowledge base and on the HowNet [3] for causality inference based on the set of lexical knowledge bases. All these show that the network of concepts is used for extracting more refined knowledge.

2 Ubiquitous Healthcare Services

As aging population grows and economic standards improve, there are more stress and chronic diseases than ever before and therefore the generic public is becoming more interested in heath management. Because of the decline in economic productivity and increase in public welfare cost of an aging society, healthcare policies are moving from treatment towards the prevention for reducing rapid increases in healthcare expenses. Thereafter, there grows the need for prevention of a disease, early diagnosis, tailor-made healthcare service, and solving inequality of service level due to the concentration

of medical manpower in the cities. However, in case of home care services, the proliferation is hindered by inconvenience of user interfaces, required attention of medical experts, and so on.

When healthcare services are provided in a ubiquitous computing environment, the tailor-made services are easily employed in everyday life and enterprises can realize commercial in-house services by minimizing the medical experts' intervention accompanyinghigh costs. To put it concretely, possible healthcare services that enterprises can provide according to the health state of an individual are as follows: health improvement for normal people, early diagnosis of disease, improvement of life quality for elderly people, healthcare for chronic invalids and prognoses, management of medical conditions like pregnant women and athletes, and so on.

3 Motivating Examples and Implications

In this section we first present the situation of perceiving the need of a user and providing appropriate services for the need in a general ubiquitous computing environment. Then we illustrate a scenario of a healthcare service. With these examples, important points are examined to cope with user's needs. Note that not every detail is illustrated due to the limit of the paper.

3.1 Providing Media Services

The first example in ubiquitous service environment is to provide media services seamlessly.

Example 1. Mike drives his car towards home after work. He is a little late after an urgent task and the Evening News that he regularly watches has already started. To watch the program while driving, Mike commands 'Show me the Evening News.' His agent responses to the command and sets the goal that Mike needs to watch the Evening News. To achieve this goal, the agent searches suitable devices and decidesthat a mobile video monitor in the car is the best choice and plays the program using the monitor. Mike arrives around his house and parks his car in a public parking lot and gets out. At this very moment, Mike's agent realizes that the monitor in the car is not Mike's visual field any more. Therefore it invalidates the goal and strives to re-achieve it. The agent starts again by searching suitable devices and plays the program using the PDA that is currently in Mike's pocket and turns off the monitor. Likewise, when Mike enters his house, the agent decides a TV in a living room is a more appropriatedevice and plays the program using the TV. After the Evening News has finished, Mike says 'Turn TV off' and the agent deletes the goal that Mike needs to watch the Evening News from his goal list.

This example is plain but effectively shows essential points of deciding and acting of an intelligent agent for satisfying a user's need in a ubiquitous computing environment. First of all, an agent needs to be capable of making and managing goals from a user's request. For example, from Mike's command, 'Show me the Evening News', he infers the goal that 'Mike needs to watch the Evening News,' and plans repeatedly every time when the goal is invalidated or a more suitable method is possible. When a goal is not necessary any more, the agent should perceive the situation and delete the goal in order not to bother the user.

In the above scenario, the agent goes through several steps to achieve the user's goal: searching appropriate devices, selecting the best device, turning it on, and tuning it through a specific channel. Although it looks very simple, all these behaviors of the agent are the results of dynamic service composition. We can conclude that service composition is necessary even with the simple scenario such as this one. One thing we need to point out is that the agent does notconsider every

possibility from the beginning. That is, when playing the program on the monitor in the car, the agent does not consider the situation that Mike gets out of the car. Instead of investigating every possibility, the agent only plans and executes the way of achieving the goal at that very specific moment. For coping with changing situations it manages the goal's condition, thus it can provide appropriate services to the user continuously. If the agent wants to make a static plan meeting every possible situation, it should consider the situations of getting out the car and entering the house and so on. The numbers of cases are actually infinite so it is impossible for the agent to make any plan.

In addition, to actually implement the environment stated above, we should consider not only authorities and securities that are generally important not reflected in the example, but also the physical computing resources on which software agents are operating. For example, when Mike commands 'show me the Evening News,' one agent in the car and another agent in his PDA can take the order simultaneously. In this case there must be either a mechanism that deals with the situation or a negotiation protocol for the agents to provide consistent service. Another important aspect is that there needs to be a scheme of translating a user's command to a user's goal, modifying it and removing existing goals from relevantrequests because all have different forms in syntax and semantics. We bravely omit these problems because they are all beyond the scope of this paper. However we can say that when ubiquitous services are provided in an enterprise's environment, the services and relevant goals are given more formally in advance than in a generic environment. Thus difficult points in management of goals can be reduced to some degree.

3.2 Providing Healthcare Services

The next example is a typical situation that occurs when an enterpriseprovides healthcare services. Through this example we will see that a ubiquitous computing with an enterprise's services is not greatly different from the general case when an enterprise restructures its back-end functions and integrates its legacy services to the new environment.

4 Goal-Based Service Agent

In the previous section, we have learned that defining, managing and maintaining the goals in order to provide the necessary enterprise services are the main points in ubiquitous computing environment. With this in mind, in this section, we describe the high level logic for defining, managing, and maintaining goals.

The user's goal should be able to be managed appropriately according to user's request or context. In the Evening News example, it shows that the goal is managed by request of the user while in Mike's example the goal is managed through the context of the user. When the goal is received by the agent, it needs to realize the specific state of the goal whether it is *incomplete*, *pending*, *completed*, or *terminated*.

In the state of incompletion, a plan needs to be determined in order to complete the goal. The pending state is when the service is actually in motion to complete the goal. In this state, service is continually in motion until the goal is completed or an outside event interferes with the current state. While the goal is in a pending state it can end in terminated state before reaching a completed or incomplete state. The goal in its pending state alters into incomplete state again, either when achieving the goal ends in failure due to an exception, or when new opportunities appear to attain the goal in a more appropriate way. And whenever a goal turns into an incomplete state, an agent builds plans to complete the goal as soon as possible. Likewise, when an agent decides the goal in the complete state is invalid because of outside events, the goal turns into an incomplete state. When a one time applicable goal is completed or when a certain goal is no longer

preferable by a user any more, the goal is altered into terminated state.

Though monitoring outcome events are indispensableto manage the states of relevant goals, it can impose high computational loads to an agent. This is because in order to decide the need of altering the states of goals, the agent is required to inspectcorrelations between the event and every goal in a list. Besides, if events and goals are represented following certain ontology which includes inheritance relationship, computational costs increase dramatically. Therefore an agent needs the mechanism of filtering events for a goal to monitor outside events efficiently. Fig. 1 shows the approach of our Goal-Based Agent.

```
function GOAL-BASED-AGENT(event) static:
        KB, a konwledge base
        goal_list, a list of goals
        new_goal  MAKE_GOAL(event, KB)
if new_goal is not null
then append new_goal to goal_list
for each goal in goal_list
if filter of goal approves event
then
EXECUTE-AND-UPDATE-GOAL-STATE
(goal, event)
if state of goal is incomplete
then     plan  KNOWLEDGE-BASED-PLANNER
(goal, KB) set plan for goal
else if state of goal is completed
then remove goal from goal_list
return
```

Fig. 1. An agent that manages goals based on the assumption that each goal is independent to each other

The Goal-Based Agent uses Knowledge-Based Planner as a subroutine to plan for a goal. The Planner is explained in the next section. In addition, the Goal-Based Agent assumes the general form of the Replanning Agent and Continuous Planning Agent of AI research [7].

5 Knowledge-Based Service Composition

Most of the conventional methods for semantic service composition follow the approach mentioned below. First, it defines a unit service, and a user's request and goal are given. After, it searches unit services that has output or effect that is related to the user's goal. This unit services' input or precondition isset as new sub-goals. It also searches the user's request that is related to the new sub-goals. If not, it searches unit services that has output or effect that is related to new sub-goals. This procedure is repeated. If all services' input and precondition is satisfied with the user's request or services' output and effect it finally decides to complete the service process. Next it selects the appropriate one among these completed service processes. Namely, this approach directly connect user's goal to unit service's output. So the goal is achieved by using only the combination of unit services.

Such conventional approach has many limits as we refer to in Section 1 and 3. Limits are caused by characters of ubiquitous computing environment. Ubiquitous computing environment is an environment that provides appropriate services according to the user's goal with the user as the central figure. This environment has enormous amount of users and services, and they are rapidly created and expired. As well as users and services dynamically change state such as location and etc. the user's goal is presented complexly, ambiguously, and focusing on the users and not the actual services. Therefore it is impossible to directly match user's goal to service's output and connect them. It is also impossible to composite all service processes that can cope with all situations and compare these processes. The reason is due to the following. Ubiquitous computing environment is a dynamic environment as we describe above. We can't imagine situations where the goal is entered and the service is actually executed. Also if we consider all situations, terribly, lots of services will affectthat terribly huge service process. In the experimental environment that contains suitable amount of services, it will be enabled that all

situations are considered, but in the actual environment, it will be impossible to do that. Finally sometimes we cannot search service that is needed to composite. In all cases that are described above, conventional approach cannot provide service flexibly.

In order toovercome limits that are described above, we will fill the missing link between services using not just conventional approach that directly combine input, output, precondition and effect but also approach that refine the goal. We grasp the meaning of a goal that is complex, various and user-centric by reasoning through knowledge-base, and refine the goal using methods that refine the goal. Because of this, the meaning of goal can be more precise and explicit. Through this, we can achieve to provide a more flexible service by using an appropriate service.

In this section, in order to refine the goal more precisely and explicitly, we explain and describe knowledge-base that is organized by the connection of concept that has meaning and also represent elementsof knowledge-base and service composition algorithm, utilizing this for ubiquitous computing environment.

6 Conclusions and Future Work

In order to verify the methods that are represented in our research, we embody the prototype system. We construct domain ontology that is related to diabetes and minimum common sense ontology for service composition using the RDF Schema, and knowledge-base is represented according to this RDF Schema as RDF. We simulated company's back-end function to provide healthcare servicethrough the use of simple expert systems. We also simulated devices and agents to interact with the user using MS Windows environment. Knowledge-based reasoning that is based in ontology is executed using SWI-Prolog engine through Horn logic, and we connected virtual components to the reasoning engine using C language interface of the SWI-Prolog. Virtual device, application and enterprise service communicate with each other using RDF messages.

Experiments with limited number of services in virtual environment show that the approach of our research is feasible. However in order to cope with various services and unpredictable scenarios, enormous common sense knowledge will be needed. This is not easy similar to the limitation of the CYC [6]. In order to rationallycope with the actual situation, decisions in uncertain situations and considerations of the feasibility of devices and services are also needed.

References

1. Aversano, L., Canfora, G., Ciampi, A.: An algorithm for Web service discovery through their composition. In: Proceedings of the IEEE international Conference on Web services, (2004) 332-339
2. Brown, S.: Fueling Innovation in U-Health: Healthcare Transformation in the United States. In: u-Healthcare 2004, Seoul (2004)
3. Choi, K.-S., Kim, J.-H., Miyazaki, M., Goto, J., Kim, Y.-B.: Question-Answering Based on Virtually Integrated Lexical Knowledge Base. In : Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, Sapporo (2003) 168-175
4. Cooper, M.C.: Semantic Distance Measures. Computational Intelligence. 16 (2000) 79-94
5. Fujii, K., Suda, T.: Dynamic Service Composition Using Semantic Information. In: 2nd International Conference on Service Oriented Computing, New York (2004)
6. Lenat, D. B.: CYC: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM, 38 (1995) 33-38
7. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd edn.Pearson Education, Inc., New Jersey

(2003)

8. Shin, H., Koehler, S.: A Knowledge-Based Fact Database: Acquisition to Application. In: Proceedings of the International Conference (KBCS2000), Mumbai (2000)
9. Sycara, K., Paolucci, M., Ankolekar, A., and Srinivasan, N.: Automated discovery, interaction and composition of Semantic Web services. Web Semantics: Science, Services and Agents on the World Wide Web. 1 (2003) 27-46
10. Weiser, M.: The Computer for 21stCentury. Scientific American. 265 (1991) 94-104