

Designing Robot Services with Ontology and Learning

Chiung-Hon Leon Lee

Department of Computer Science and Information
Engineering
Nanhua University
Chia Yi, 622, Taiwan
chlee@mail.nhu.edu.tw

Alan Liu

Department of Electrical Engineering and
Center for Telecommunication Research
National Chung Cheng University
Chia Yi, 621, Taiwan
aliu@ee.ccu.edu.tw

Abstract—Considering user preferences in defining robot services is important. How to interact with service robots varies from user to user, and providing a user friendly way for the users to interact with robots becomes necessary since the users are usually non-technical people when dealing with service robots. This article focuses on defining robot services that can suit user's preferences. Service robots are typically designed to provide one set of services not targeted to a particular user. However, ways of controlling robots may differ depending on users. Learning from past experiences enables a robot to adapt to the user's specific needs and interacting style. In this research, we use ontology as the design tool for defining robot services and uses case-based reasoning as a means of learning in storing previous interaction experiences as cases. These cases are reused the next time when similar requests are made.

Keywords—user preferences, ontology, case-based reasoning, learning, service robots

I. INTRODUCTION

As service robots are planned to go into homes in a daily life setting, providing an easy way for the users to interact with robots becomes important since the users are usually non-technical people. The methods of interacting with service robots vary from user to user. In this ongoing research, we are studying how to improve the interaction between the users and robots through personalization and learning from pervious experiences. The research is targeted to a smart home environment where we use a multiagent system to coordinate robots and various devices.

An intelligent multiagent system for a smart home consists of many different missions and different platforms of systems, such as intelligent assistance systems, wireless home network systems, security systems, intelligent appliances, intelligent interfaces, etc. With the needs of controlling various systems and devices, how to design a highly flexible and robust architecture raises a challenge. Traditional sequential or layered architectures do not meet such requirements easily. Therefore, we have focused our research on a multiagent architecture.

In addition, a multi-user environment needs to provide an interface that is capable of satisfying individual needs. As an example, in a place like nursing home, elderly people may have different needs and constraints, and they all may have their

own needs. The system is required to satisfy all their requests suitable for their ways of interacting with the system.

Besides controlling devices in a smart room, as a service robot is introduced in the room to provide services, another important research issue arises to design an intelligent service environment by integrating robots to the environment. Leveraging our previous experiences in user intention analysis and personalization, we are using ontology to define services and case-based reasoning (CBR) to record repeatable usages.

In order to provide a more user friendly interaction, some researchers use different ways of detecting user's needs by integrating different devices for providing different views such as using more sensors to get more information [1]. Others use speech as a means of more natural interaction, but it depends on other devices to provide context to understand what the user means since speech recognition and natural language processing still has their own challenges [2]. From a service providing point of view, timely assistant is also a concern when proactive services are considered [3]. This timing issue lets us to consider how to minimize searching and reasoning time. Thus, for our research we use agents to provide proactive assistant while using CBR to remember individual users' ways of interaction and personal needs.

The rest of this article is organized as follows. The multiagent architecture that we use is introduced in Section 2. In Section 3, we explain the concept of learning through CBR while using ontology as the design method for defining robot services. It is then followed by discussion on personalization in Section 4. Implementation with prototypes is explained in Section 5, and finally, the conclusions are given in Section 6.

II. MULTIAGENT ARCHITECTURE

In our smart home setting, we have agents controlling different devices, because one of the advantages of using agents as the representatives of different devices is that they have the capability of joining or leaving the environment without changing the coordination mechanism. Even though the infrastructure is there, this may still cause problems with ontology alignment since there exist new agents with different missions joining dynamically. Users are not supposed to know all these details. Instead, they only need to know that they can

communicate with the room, and the room is going to take care of their requests.

As shown in Fig.1, besides an agent assigned to interact with the user, we use agents to represent different appliances and devices like vision system or even a robot. A world model may be viewed as information about the current states. These agents form a multiagent system and they can interact with each other and even cooperate with each other. Agent cooperation is an important issue where we have gained some experiences [4][5], but the focus of this article is to let the agent serve the user by acquiring user's request and to remember useful information.

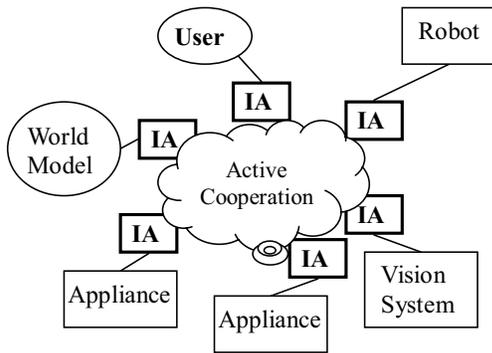


Figure 1. Multiagent infrastructure for a smart home

One of our target applications in the smart home is elderly care. Within such environment as an example, once simple commands from the user are issued, the virtual caretaker is to understand the real needs of the user and organize a mission to be carried out through the cooperation of participating systems. The mission is to be decomposed to behaviors of a robot in terms of a sequence of simple commands, commands to collect vital signals for analysis, or request for sending processed information to the external medical systems.

III. LEARNING CAPABILITY WITH CASE-BASED REASONING

The concept of CBR systems is to use past experience as cases and collect those cases to establish a case base. Using the case base, a CBR system will retrieve a case, reuse the case, revise the case and retain the case [6][7]. As shown in Fig.2, according to features of the new problem, we select similar cases from the case base (case retrieval). A system may find one or more cases that have similar features given. From those similar cases, the system selects one case according to some similarity criteria. The next step, case adaptation, is to tailor the proposed solution to be applied to the new problem. In the adaptation procedure, there exist adaptation strategies and heuristics, along with adaptation knowledge. For example, an adaptation procedure must understand which part in the retrieved case can be replaced to fit a new case.

When having obtained a solution, it can solve problems in the real world by using evaluation method. There are many methods to evaluate difference between adapted cases and new case. After the case adaptation, a learned case is appropriately

stored into the case base for reuse in the future. When we store the case, we must consider how to update the case base and the case index. This process then is considered as a learning capability.

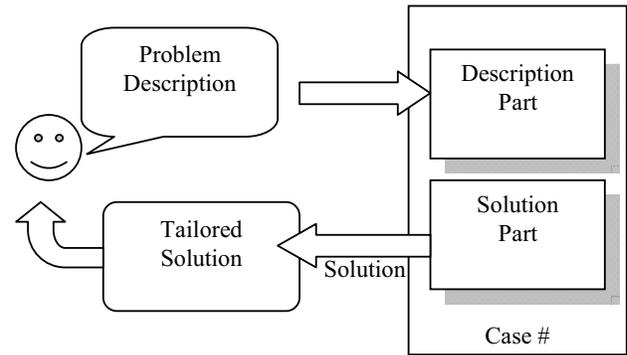


Figure 2. Case based reasoning procedure

IV. PERSONALIZATION WITH ONTOLOGY

In order to design a service robot which is highly personalized for caring a user, such system should be adaptable to individual needs. As illustration, when considering the health status and ability of elderly people, the robot should not only be capable of meeting medication agenda or monitoring the physical conditions of the elderly people, but also be responsible for providing different user interfaces. It is difficult for conventional architectures to satisfy the needs of such highly heterogeneous tasks. Some researchers suggest that using mobile agent is a feasible method for this scenario [8], because different mobile agents can move to a robot and transform that to a personalized platform to carry out heterogeneous tasks. What really important for an agent whether mobile or stationary is to have the knowledge to distinguish different users.

A. Personalization

Besides the service robot that may be interacting with the user, there is a need for devices to collect data in the room. The most challenging task in this research is how to make a robot acceptable to the user. We target our robot to be not only an assistant, but to become a companion or even a caretaker. From the companion point of view, the robot and other devices are designed as a source of data based on observation. In other words, the user is not aware of being monitored to reduce stress. Furthermore, the robot and other devices should be designed as tools for receiving user's commands, always attentive to the user's needs. A service robot should be equipped with interface to the user, but it alone cannot capture everything from the user. Therefore, combining the observation from different devices will be an important task.

To illustrate this point, we consider that the user tries to communicate with the robot through dialog. The user is also doing his gesture at the same time. The vision system can interpret the gesturing message and send the result to the robot which is processing the dialog. As for giving a message to the user, the robot may give a sentence through its built-in speaker

and the virtual steward in the screen may be able to show the message in written words.

For performing such cooperation between robots and other devices, we can use agents to integrate such information. The important task is to know when and what to integrate. For this we start from the research of ontology for understanding and defining the vocabularies used and situations that the user may encounter. For this purpose, we study the user's needs and the concept to understanding the context well [9][10]. Our approach uses six types of concept to associate domains and user needs as listed below:

- Domain Independent Concept (DIC). A concept may appear in different domains, but the meaning of the concept is the same despite of different domains.
- Domain Dependent Concept (DDC). The meaning of a concept differs in different domains.
- Domain Specific Concept (DSC). A concept may be used only in a specific domain.
- User Independent Concept (UIC). The meaning of a concept is agreed by all users in the same domain.
- User Dependent Concept (UDC). The meaning of a concept is different according to different users.
- User Specific Concept (USC). A concept is only meaningful to a specific user.

Using the definition above, we can determine nine different kinds of concept which is shown in Table I. The table shows how domain dependencies and user dependencies can be combined together. These categories can be used in designing ontology.

TABLE I. CATEGORIES OF CONCEPTS

User Aspect \ Domain Aspect	User Independent	User Dependent	User Specific
Domain Independent	User Independent & Domain Independent	User Dependent & Domain Independent	User Specific & Domain Independent
Domain Dependent	User Independent & Domain Dependent	User Dependent & Domain Dependent	User Specific & Domain Dependent
Domain Specific	User Independent & Domain Specific	User Dependent & Domain Specific	User Specific & Domain Specific

B. Ontology for Designing Robot Services

AI planning is used commonly for providing a sequence of operations to control a robot. The robot operations may be primitive as ‘move forward,’ ‘stop,’ ‘turn right,’ etc. If the user is using these operations to communicate with the robot, then it is just like using a remote control device. It is more natural for the user to give high-level or abstract command instead of these primitive operations. When defining tasks performed by robots,

a hierarchical definition can serve as the description of abstract behaviors. Simmons and the colleagues introduced a task description language and used a task tree to represent the hierarchical nature of the robot behavior [11]. For illustration, they defined a mail delivering robot with a high-level behavior, ‘deliver mail’ which becomes the root of a task tree. Under the root, there are four nodes, ‘navigate to location,’ ‘center on door,’ ‘speak,’ and ‘monitor pickup.’ The definition goes on under each node. Techniques of using these decomposable behavior definitions were widely carried out with task definition, behavior definition, and plan definition [12]. A typical definition of behavior may look like Fig.3, which we use Protégé [13] in defining them. Within this definition, a ‘bring something’ operation consists of ‘find something,’ ‘move to location,’ ‘grab,’ and ‘deliver’ along with some slots being defined.

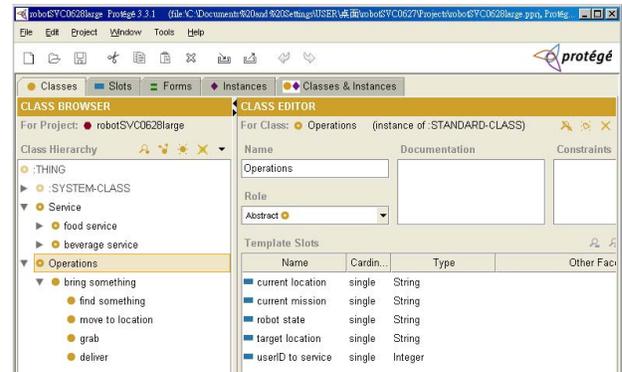


Figure 3. Typical behavior definition

Besides the definition of operations, anything associated with them should be considered. Our work is similar to the research done by Jung and the colleagues [14][15]. They defined the taxonomy to action definition and applied their methods to digital home environment. The main difference is that we use ontology as the means of service design, where not only behaviors are defined, but the associated objects and events are also put into consideration. Basic operations can be used by a planner to compose a more complicated operation. In addition, we use CBR to record the successful behavior definition without re-calculating or composing a complicated behavior as defined before.

Following the above discussion in designing services of a robot, we start from defining the operations performed by a robot and also the objects associated. Then, we consider the preferences of users and different situations in which the operations are carried out. The categories listed below are considered in defining services of a robot:

- Operations. This category includes the definition of functions carried out by a robot. The operations may be in different abstraction levels like ‘deliver a newspaper’ or a primitive one like ‘move forward.’
- Objects. Anything that is required by the operations mentioned above is defined along with its properties and functionalities.

- Preferences. Users may prefer the robots to carry out missions in different ways according to individual needs. Asking for a drink, one user likes a glass of cold water while another prefers a cup of hot tea.
- Situations. Timing or additional requirements may affect what services to provide and how services are carried out. Moving across a room straight and fast may be efficient, but it may be safer to follow the wall and go slower when there is a person in the room. This can be extended to ‘politeness’ to behave quieter and slower when there is a guest around.

The main advantage of defining ontology is that we can add more related concepts easily. In addition, modification to the definition can be performed systemically. Fig.4 shows a sample of designing a service consisting of not only operations, but also objects. Objects, in turn, contain information, such as basic properties, functionalities, and other objects relating to them. A ‘food service’ is defined by ‘bring food operation’ which is based on the ‘bring something’ operation and the ‘food to bring’ information which is the content to deliver for particular purposes like ‘snack,’ ‘meal,’ and ‘fruits’ to satisfy the user’s needs. Within the items, instances are defined with some properties like ‘calories,’ ‘size,’ ‘utensils,’ etc. The instances to ‘fruits’ may be ‘apple,’ ‘applepear,’ ‘grapes,’ etc.

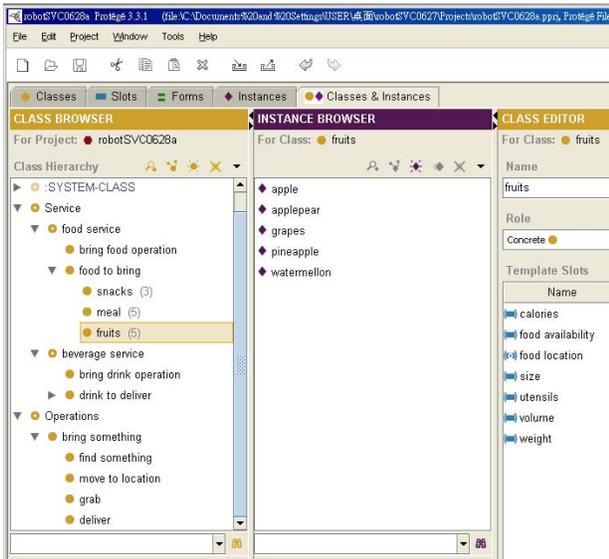


Figure 4. Service definition

V. IMPLEMENTATION

A. CBR System

In our previous work, we have used jCOLIBRI [16] in developing a BDI agent [17]. In that work, our target was to perform case retaining for storing experiences. Extending that approach, in this research, we integrate another CBR tool, myCBR [18], with the ontology tool, Protégé, for helping the designer of a service robot define the robot services.

For the smart home applications, Fig.5 shows our system for processing user requests. The ‘User Interface’ component processes user input and provide system output. The ‘Behavior Planner’ component produces a plan to control the robot and devices in the smart home. The information for planning is based on the ‘Intension and Context Reasoner’ component which uses ontology and past experiences to find a suitable solution to the user.

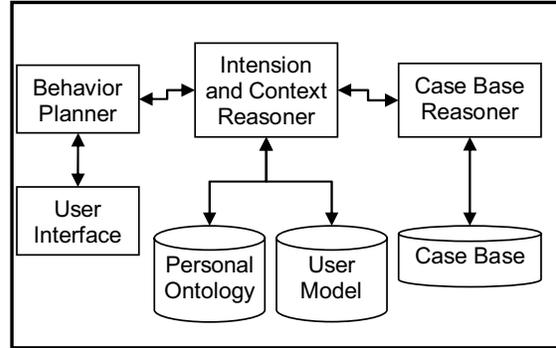


Figure 5. User interaction agent

As discussed in the previous section, designing a service many involve definition of other associated items. One of the challenging tasks is the user’s preferences and experiences. We show here that we can analyze observed user reactions to produce the ontology to users needs. Table II shows nine observed situations involving users’ needs in different stages. In this list, we can see that ‘alan’ prefers ‘sandwich’ when he is ‘hungry’ while ‘ken’ likes ‘steaks’ and ‘leon’ ‘meatloaf.’ When they are ‘thirsty,’ ‘alan’ prefers ‘pops,’ ‘ken’ ‘juice,’ and ‘leon’ ‘water.’ It is also observed that when ‘alan’ is ‘happy’ he enjoys ‘music’ while drinking ‘pops’ if he is also thirsty. When they are happy, ‘ken’ likes to listen to music while drinking some juice, but ‘leon’ reads his novel while drinking some water.

TABLE II. OBSERVED USER PREFERENCES

ID	name	state	mood	food	drink	entertainment
123	alan	hungry		sandwich		
234	ken	hungry		steaks		
345	leon	hungry		meatloaf		
123	alan	thirsty			pops	
234	ken	thirsty			juice	
345	leon	thirsty			water	
123	alan	thirsty	happy		pops	music
234	ken		happy		juice	music
345	leon		happy		water	novel

We record these observation results into a CSV file and uses myCBR as the plug-in feature of Protégé to import this file. The tool processes each observation as a case and uses the entry of the attributes to construct value ranges while building the slots of the ontology. Fig.6 shows the CSV file being

imported to the system, and Fig.7 shows the resulting ontology definition to 'Users' with seven slots and different values defined. Fig.8 shows the case retrieval process with nine cases. Along with this process, we can use the 'Similarity Measure Editor' to modify the functions for calculating similarities. This helps the developer understand the importance of attributes and find more attributes or cases. It acts as a prototyping tool for defining ontology.

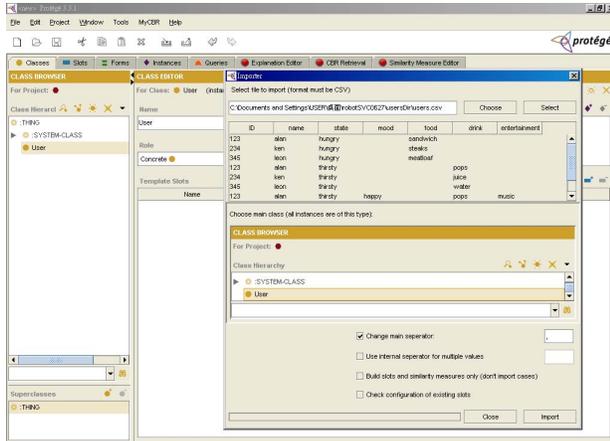


Figure 6. Importing observed cases for ontology definition

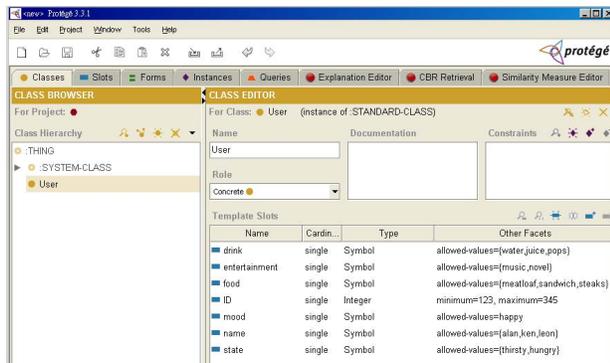


Figure 7. Generated user definition with slots and slot values

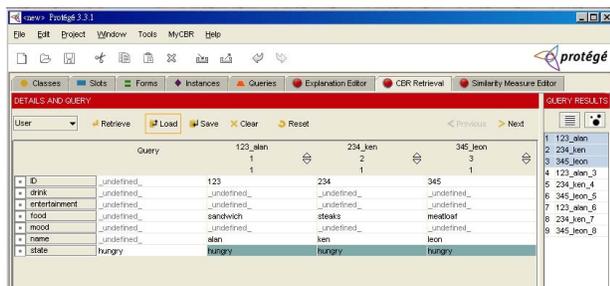


Figure 8. Case retrieval from nice cases

As the understanding to the prototype grows, we may observe more facts. There may even be new information to

add to the observed list. Table III shows some addition to the original list, where the users do something different because of events. For 'dating', 'ken' goes to 'movies,' and 'alan' is having a piece of 'cake' with 'tea' for his 'birthday.' We can repeat the definition again by importing this new list, and the system will prepare a new ontology definition along with an updated case base. Fig.9 shows eight slots and more values from the new CSV file. Though not shown, there are now twelve cases in the case base.

TABLE III. OBSERVED USER PREFERENCES

ID	name	state	mood	food	drink	Entertainment	event
123	alan	hungry		sandwich			
234	ken	hungry		steaks			
345	leon	hungry		meatloaf			
123	alan	thirsty			pops		
234	ken	thirsty			juice		
345	leon	thirsty			water		
123	alan	thirsty	happy		pops	Music	
234	ken		happy		juice	Music	
345	leon		happy		water	Novel	
123	alan		tired		coffee	Music	
234	ken		happy			Movie	dating
123	alan		happy	cake	tea	Music	birthday

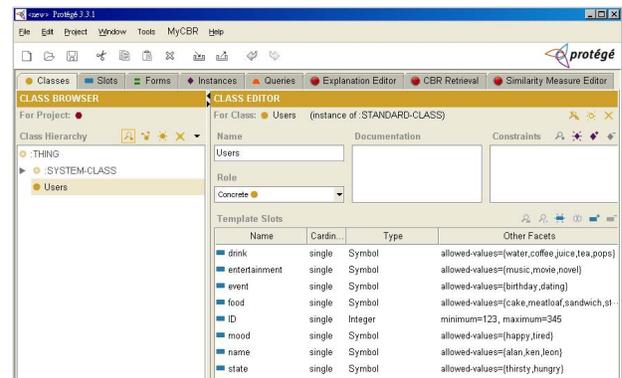


Figure 9. Newly observed cases for ontology definition

B. Robot and Devices Integration

In a smart home setting, the components which are most frequently and directly associated in communicating to each other are the home gateway and the robot. Any addition or removal of devices is not supposed to affect the interaction between such devices and the robot since all information is processed before it is sent from home gateway to the robot. However, if an additional robot is introduced in the environment, then the home gateways needs to identify the robot as a new addition and to provide services and control to that new robot.

As mentioned before, this is why we consider using a multiagent system. The cooperation among agents is not trivial since it involves different robots and devices in heterogeneous platforms. Robot cooperation can be analyzed from two perspectives. One is cooperation mechanisms processed at the

physical robots as mentioned in the previous section. The other is at the agent level where autonomous agents cooperating in a deliberative manner. We have incorporated CBR in agents in order not only to interact with users, but also for robots to interact with each other and also with devices. Currently, it is only at the simulation stage, but interacting with the physical robot is next.

Our agent uses CBR techniques for learning the user behaviors and issues orders to the robot accordingly. Our next step is to implement our agents with a physical robot. Fig.10 shows a hybrid architecture, in which the physical robot may be seen as a reactive module which has its own action to take; however, it may be blended with the deliberative module to deliver more abstract and advanced behaviors.

For coordinating such robots, we consider cooperation policies and strategies as cases. Thus, the task in knowledge acquisition is to analyze robot behaviors based on its basic commands and organize those behaviors as cases to be stored in a case base. At the same time, the user's preferences will also be recorded in the case base for future use.

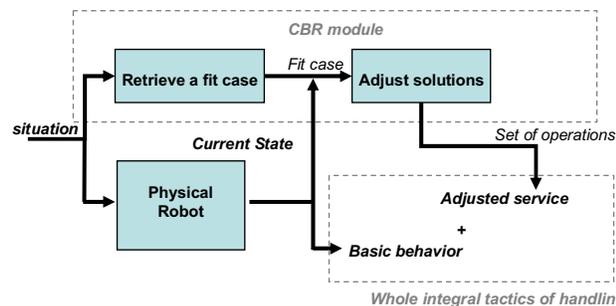


Figure 10. Task execution procedure

VI. CONCLUSIONS

Designing robot services to suit individual users is a challenging task. In designing a service, not only the robot operations need to be defined, but other information like objects, situations, and preferences need to be considered. We have used ontology as a tool to record the definitions. By continuing our previous research in CBR, we also noticed that using cases as the basis of ontology definition to be an effective way for design. Using open source tools also helped us obtaining prototypes quickly and gained valuable information relating to the ontology. With the understanding in defining high level services, we can introduce such knowledge into agents through CBR to aide the agents to serve the user better.

ACKNOWLEDGMENT

This research was partially supported by the National Science Council in Taiwan through Grant numbers NSC-97-2221-E-194-048-MY3 and Grant NSC-97-2752-E-008-001-PAE.

REFERENCES

- [1] J.F. Maas, T. Spexard, J. Fritsch, B. Wrede, G. Sagerer, "BIRON, what's the topic? A Multi-Modal Topic Tracker for improved Human-Robot Interaction," The 15th IEEE International Symposium on Robot and Human Interactive Communication, 2006. ROMAN 2006. Volume , Issue , 6-8 Sept. 2006 pp.26 – 32
- [2] S. Huwel; B. Wrede; G. Sagerer, "Robust Speech Understanding for Multi-Modal Human-Robot Communication The 15th IEEE International Symposium on Robot and Human Interactive Communication, 2006. ROMAN 2006. pp.:45 – 50
- [3] O.C. Schrempf, U.D. Hanebeck, A.J. Schmid, H.Worn, "A novel approach to proactive human-robot cooperation," 05. IEEE International Workshop on Robot and Human Interactive Communication, 2005. ROMAN 20, pp.555 – 560
- [4] H.C.H. Hsu and A. Liu, "A Flexible Architecture for Navigation Control of a Mobile Robot," IEEE Transactions on Systems, Men, and Cybernetics – Part A, Vol.37, No.3, May, 2007, pp.310-318.
- [5] C.H.L. Lee, W.S. Tseng, C.M. Chen, A. Liu, S.M. Huang, and H.B. Huang, " Designing an Intelligent Agent for Information Appliances," Journal of Computers, Vol.18, No.1, April, 2007, pp.61-70.
- [6] Agnar Aamodt and Enric Plaza, "Case-base Reasoning: Foundation Issues, Methodological Variations, and System Approach", AI Communication, IOS press. Vol. 7, No. 1, 1994, pp. 39-59.
- [7] R. Bergmann, J. Kolodner and Enric Plaza, "Representation in case-based reasoning," The Knowledge Engineering Review, Volume 20, Issue 3, 2005, pp. 209-213.
- [8] L. M. Camarinha-Matos and W. Vieira, "Intelligent mobile agents in elderly care," Robotics and Autonomous Systems, vol.27, 1999, pp.59-75.
- [9] C.H.L Lee and A. Liu "User intention satisfaction for agent-based semantic Web services systems," Proc. of The 12th Asia-Pacific Software Engineering Conference (APSEC'05)
- [10] Jeng-Shin Hung and Alan Liu, "Using Personal Ontology in Evaluating Service Quality," Proceedings of the IEEE International Conference on Services Computing (SCC 2007), pp.332-339.
- [11] R. Simmons and D. Apfelbaum, "A task description language for robot control," , Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998, pp.1931-1937.
- [12] S. Knoop, M. Pardowitz, and R. Dillmann, "From Abstract Task Knowledge to Executable Robot Programs," Journal of Intelligent Robotics Systems Vol.52, Nos.3-4, 2008, pp.343-362.
- [13] Protégé: the Ontology Editor and Knowledge Acquisition System <http://protege.stanford.edu/>
- [14] Y. Jung, H. Park, Y. Choi, S.H. Myaeng, "Designing a Cognitive Case-Based Planning Framework for Home Service Robots," Proceedings of the 16th IEEE International Symposium on Robot and Human interactive Communication, (RO-MAN 2007), 2007, pp.827-832.
- [15] H. Kim, Y. Jung, and Y. K. Hwang, "Taxonomy of Atomic Actions for Home-Service Robots," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.9, No.2 pp. 114-120, 2005
- [16] J. A. R. Garcia, A. Sanchez, B. Diaz-Agudo and P. A. Gonzalez-Calero, jCOLIBRI 1.0 in a Nutshell: A Software tool for designing CBR systems, Proceedings of the 10th UK Workshop on Case Based Reasoning, 2005, pp. 20-28.
- [17] K.Y.R. Cheng, C.H.L. Lee, and A. Liu, "Applying a Case-Based Reasoning System Development Tool in the Design of BDI Agents," Journal of Internet Technology, Vol.9, No.4, October, 2008, pp. 339-346.
- [18] A. Stahl and T.R. Roth-Berghofer, "Rapid Prototyping of CBR Applications with the Open Source Tool myCBR," Proceedings of the 9th European Conference on Advances in Case-Based Reasoning, 2008, pp.615-629.