Designing a Cognitive Case-Based Planning Framework for Home Service Robots

Yuchul Jung, Hogun Park, Yoonjung Choi, and Sung-Hyon Myaeng

School of Engineering, Information and Communications University
119, Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
{enthusia77, gsgphg, choiyj35, myaeng} @icu.ac.kr

Abstract— **Home-service robots are expected to perform a wide range of tasks commonly encountered in a household environment. For autonomous operations robots should be able to plan their actions to carry out these tasks in advance and they should at least have the ability to plan for additional tasks during their operation. Because of the variability and uncertainty in the environment, it is best to endow robots with a learning-based task planning capability that rests on Human-Robot Interaction (HRI). We take a case-based reasoning (CBR) approach to home-service-robot learning and incorporate the cognitive HRI paradigm that includes four cognitive models (needs, task, interaction, and user model) for case adaptations to the given situation. Given a new command from user, a robot finds the closest task case from already existing tasks to start with a plan and modifies it (i.e. action sequences) to adapt to the given situation based on the cognitive models. In order to promote the reusability and flexibility of task cases used in our CBR approach, a Robot Task Description Language (RTDL) is designed to represent tasks using an Atomic Action Taxonomy [1]. The proposed approach is applied to a "*Bring me a coke*" scenario and implemented in our robot system called IDRO.**

I. Introduction

Robots are expected to become a close friend or a helper to make people's lives more comfortable and enjoyable. For such a role, a home-service robot should have basic capabilities: bringing something (a book, cup, coke, etc), vacuuming, guiding a person to a place, and so on. That is, robots are expected to perform a variety of tasks consisting of actions and sub-tasks. However, predicting and representing in advance all tasks to be executed is almost impossible. Storing all enumerated tasks is not realistic because of the need for real-time control of robotic systems.

The efforts for designing an intelligent home-service robot have led to the development of humanoid robots like ASIMO [2]. This is because household environments are designed for comfort and convenience of human beings with an average physique. However, currently only simple household services such as carrying objects, vacuuming, and operating household appliances are possible in a carefully controlled environment.

Robots are likely to become a ubiquitous part of our daily life, similar to technologies in communication, automobiles, and transportation. In that sense, in order to be ubiquitous, robots must function reasonably and autonomously under a variety of conditions, while adapting to environmental changes and continually pursuing their goals. For this adaptability, we need to minimize the planning time by using prior knowledge about and experience in tasks. Applying classical planning methodologies [3], without proper prior knowledge about household environments, is too complex to make a good plan. The case-based reasoning (CBR) approach is a natural choice, especially given the high complexity of the plans to be made for household environments.

The existing CBR-based work mostly concentrates on retrieval processes [4, 5]. In short, finding the most relevant prior case is the main job. Retrieving one relevant case from the case-base is not sufficient; there is a need to adapt it to the given context [5-7]. Although performing a meaningful adaptation requires rich context information, it is not sufficient, either. We need cognitive models for needs, tasks, interactions, and users that are understood by both the human and the robot in order to handle the ambiguities arisen from incomplete information.

Prior to applying the CBR approach to the robot's task planning, a well-designed case structure for the robot's task is required for planning a task, storing a number of cases, changing sequences of a task, and issuing a final action sequence. For this requirement, we have carefully designed a robot's task case structure based on Robot Task Description Language (RTDL). Based on the RTDL, our cognitive CBR framework manages the given task by undergoing appropriate adaptations with the cognitive models.

The rest of this paper is organized as follows. Section II describes the existing approaches about CBR, robot languages, the robot task with atomic action, and the robot's task planning. Section III illustrates how a robot's task case can be represented using RTDL structure. In Section IV, the derived CBR scheme is explained with the used cognitive models in retrieval and adaptation aspects. A system implementation and the experiment of detailed adaptation procedures with cognitive considerations are described in Section V. Finally, conclusion is appeared in Section VI.

II. Related Work

A. Case-Based Reasoning (CBR)

The basic idea of CBR [4] is to solve new problems by comparing them with problems already solved; that is to say, CBR is a process that uses similar solution that were already solved previously to solve the current problem. Although it can be said that CBR system has 4 steps (retrieve, reuse, re-

vise, and retain) [4, 6], the critical step is to find and retrieve a relevant case from the Case-Base. Most CBR Systems were developed for retrieval of the most relevant cases. Representative systems are JULIA (meals), KRITIK (devices), and CADRE (building). However, to make the CBR Systems more intelligent, adaptation process should be included in its implementation. Because a stored case contains a solution, it can be adapted by modifying sequences or parameters of the old problem to suit the new situation resulting in a proposed solution. After the solution is tested, it can be added to the Case-Base if found successful. Adaptation (including planning) procedures and their policies were mentioned in [4-6]; e.g. CHEF (recipes) and SIMMS (robot control). In addition, knowledge acquisition is easier in CBR because of the granularity of the knowledge. To maximize the utilization of the case-base, hybrid approaches that combine the case-base retrieval and rule-based adaptation mechanism is recommendable. The hybrid method can bring out several benefits in some fields like adaptation speed and accuracy [7]

B. Robot Language

A number of task-level programming languages have been developed for robots, such as Task Definition Language (TDL) [9], Robot Sensor Language (RSL) [10], Robotic Markup Language (RoboML) [11], etc. While those languages are powerful and convenient for expressing the information related to their respective applications, a simultaneous utilization of a robot programming language in a single application, they could create significant difficulty when applied to our CBR approach. For example, the RoboML [11] is a markup language for HRI, and its capability is limited as enabling navigation by describing sender, receiver, and robot's wheel rpm. Hence, a new XML-based markup language, which can flexibly support various tasks of home-service robot in atomic action level, is defined in this paper.

C. The Robot's Task and Atomic Actions

For a robot, generally speaking, a task can be said as a sequence of detailed actions to be done or procedural behaviors for completing a goal by satisfying a human's command or react automatically coping with the current conditions. However, the robot's task that human expects can be somewhat different depending on the situation faced, human's intention, and its domain. If the coverage has narrowed down within home-service domain, we can enumerate house chores, such as bring something, opening door, vacuuming some place, and guiding guest.

However, those tasks need to be arranged into more manageable units that can be handled in the robot's task planning. *Kim et al.* [1] defined and classified atomic actions based on the kind of sensor algorithms used during atomic execution in respect of vision verification, sound verification, and force/tactile verification. Based on that, each task in a set of relatively complex tasks for home-service robots was decomposed into a set of atomic actions that can be executed by the robot without further calculations or interpretations. Hence, the atomic actions will be used as basic units in our Cognitive CBR approach.

D. The Robot's Task Planning

There has been a variety of research to accomplish several home-service robot tasks in the robot's task planning and its taxonomy until now [14-18]. In order to make a complete action sequence for a task, there is a need to combine already mentioned tasks such as motion planning which includes grasping [19-21], navigation that contains path-planning and movement [22-24], vision recognition, voice recognition, and so on.

In this line of context, we could develop a firm belief that the robot's task planning should be organized under the consideration of taxonomy about vision, sound, and force & tactile verification. Each of above task planning approaches may complete for themselves, however, due to the limited coverage, there are some difficulties to accomplish a task by integrating with each other intelligently in home-service environment. Although hierarchical task network planning [25] has very similar features compared to our approach in terms of hierarchical task decomposition and sub-task search satisfying the given conditions, there exists clear differences; reusing user's prior experience with the given context information, updating the case-base with newly user-driven cases (or sub-tasks), task structure and its flexibility & extensibility.

III. ROBOT TASK REPRESENTATION

A. Robot Task Description Language (RTDL)

We have deigned Robot Task Description Language (RTDL) for a robot's task structure to be used in our framework. Although the task structure is a 3-level hierarchy (case (or task) → sub-tasks → atomic actions) as shown Fig. 1, sub-task (or atomic action) *per se* can be a task case depending on situations. Thus, multi-level cases can exist in our robot task case database. We assume that all objects are already known to the robot and that every atomic action is complete by itself. Additionally, perception actions run constantly to check the completion status of each step, so they can run in parallel to the effector actions, and their *COMPLETION* or *FAILURE* will decide on the next action to be performed by the robot.

Fig. 1 represents a tree model of our RTDL in XML schema level. Basically, a task can be divided into a number of sub-tasks for performing a final goal (i.e. user-command's ultimate purpose), and each sub-task is composed of several atomic actions for fulfilling a sub-goal.

A task case can have a unique ID (Case ID) as an attribute. In addition, a case can have CTASKCATEGORY (Case's category), CHANDLINGOBJ ECT (which object is handled), COBJECTSTATE (state of the object), CONTAINER (what container is need to handle the object), CONTEXTINFORM ATION (what are environmental constraints surrounding the robot), and NUMST (total number of sub-tasks) as the building blocks. In term of sub-task, it has ID (Sub-task Id)

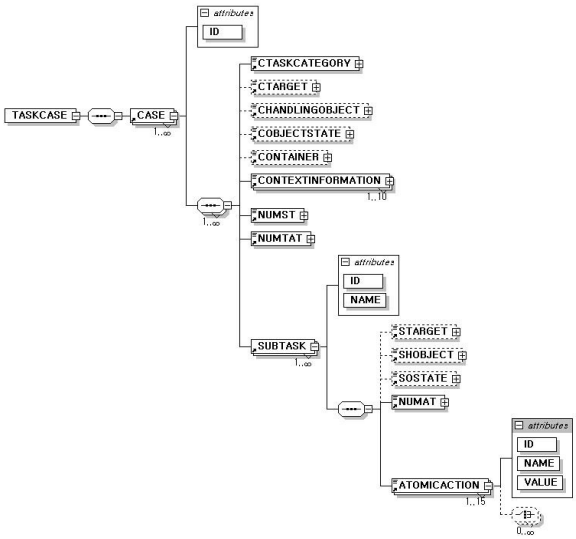

Fig. 1 A Tree Representation of RTDL

and NAME (Sub-task name). The sub-task can maintain STARGET (what is target location of this sub-task), SHOBJECT (what is the object that this sub-task handles), NUMAT (total number of atomic actions which are used in this sub-task), and atomic actions as elements. The sub-task has several atomic actions to fulfill its own sub-goal. Each atomic action has its own ID, name, and value, but the value can be adopted considering the surrounding context.

TASK: A robot task is defined to be a specific piece of work done, whether requested by a human or generated by the robot itself. The number of tasks that robots can do will increase as technologies developed, making it a moving target to define prototypical tasks. For example, *Bring me an object, Throw an object into a wastebasket* and *Vacuuming the floor* are examples of the robot task. In Fig. 1, *CASE* is considered as a robot task.

Sub-task: All robot tasks are simple for humans but are complex multi-step operations for robots. A sub-task is a quite natural procedural process for a human being rather than a robot. It is of such granularity as *Find_object, Grasp_obect, Handover_object, Move_to_location,* and so on. They usually involve a single object or location parameter, and their execution requires coordination of sensing, planning and action modules of the robotic system. Table 1 shows the number of sub-tasks required for each of the several household tasks.

Atomic Action: We have defined 45 atomic actions that are required to complete roughly 10 household tasks [Table 2]. They are organized and classified in respect of vision, sound, and force/tactile verification based on *Kim et al.*'s work [1]. Categorizations such as movement, grasping, etc are done for easy understanding about each atomic action. The number of atomic actions can be increased according to the future needs and the increase of available tasks.

Table 1. List of Sub-tasks

Sub-tasks	***Description***
Find_object	Find the location of an object
Move_to_location	Navigate to the location
Grasp_object	Grasp an object
Throw_object	Release an object while moving arm
Open_door	Open a door
Handover_object	Hand over an object
Wait	Wait for a moment
Get_command (from user)	Get a command from the user
Push_object	Push an object
Pull_object	Pull an object
..	..

Table 2. Atomic actions for home-service robots

Category	***Atomic Actions***
Movement	GoToward, InvokePathPlanner, Navigation, GoBetween, GoRight, GoLeft, GoFoward, GoBackward, Stop, Wait, GoAlong, …
Grasping	InvokeArmMotionPlanner, MoveArm, CalculateGrapType, HandOver, Grasp, …
User Interaction	GetVoiceCommandFromUser, GetTextCommandFromUser, PromptToUserVoice, …
Vision Capability	CheckQuantityWater, CheckStability, FindLocationFromVision, FindLocationFromKB, …

B. Task Representation using Sub-tasks and Atomic Actions

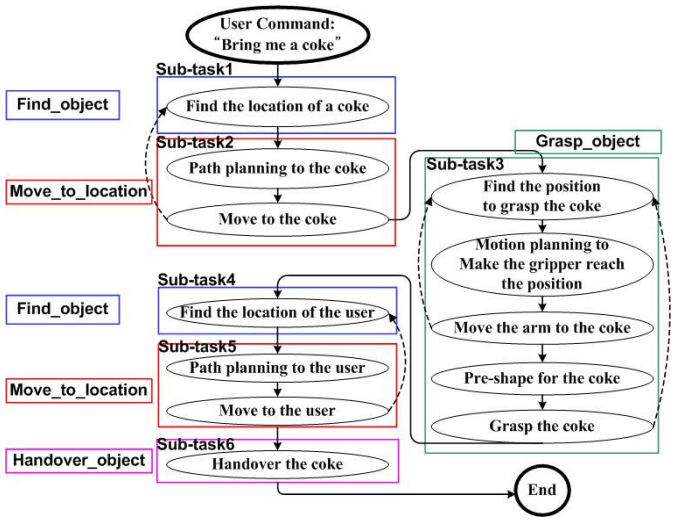

Fig. 2 Example: "*Brine me a coke*"

Fig. 2 shows the flow chart for the "*bring me a coke*" task in detail. It has six sub-tasks. The solid arrows denote the normal flow of operations while the dashed arrows show routes for possible error handling and recovery. This flow chart in fact represents a piece of knowledge in our robot task case database, and it can be hard-coded initially in the robot

or may be taught by HRI on the fly.

The six phases are designed for the task, but there can be variations according to the situations. For example, the number of sub-tasks can be increased or decreased to satisfy the problem requirement and the context conditions. That is, adding a sub-task (or atomic action) and undoing the given sub-task (or atomic action) for error-recovery can happen. In other words, two instances of the sub-task "*Move_to_location*" do not always entail the same atomic actions as its members. Even the number of atomic actions can be different, depending on the needs (e.g. avoiding unexpected obstacle or emergency)

IV. CBR Approach for the Cognitive Robot Task Planning

A. Overview

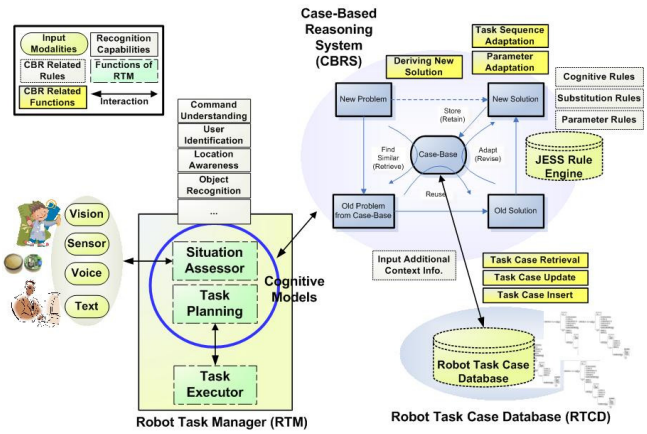

Fig. 3 Overall Architecture for our CBR Approach

In Fig. 3, the Robot Task Manager (RTM) analyzes various attributes of the given task and searches for similar cases by exploring Robot Task Case Database (RTCD). If there exists a matching task with enough similarity to the given task, the pre-planned action sequence of the matching task can be used to carry out the given task after some modifications. If there is no similar task, the robot asks the user to demonstrate how to perform the task (using text input at present) and stores the action sequence taught by the user. However, this is the worst case example. We assume that RTCD manages enough number of cases that can cover user's command in a household environment because we already manually edited several task cases for each of different task goals. Our RTM can start with at least one relevant task case retrieved from the RTCD.

In addition, the RTM controls the flow of task sequences by observing the completion of a sub-task (or an atomic action) and external sensory information. Although the RTM initiates the first action from the retrieved task case, it undergoes appropriate adaptations based on the four cognitive models as described in Section 4.3. Actually, it resolves ambiguities using the related rules in the JESS [27] with regard to the suitability of the given sub-task (or atomic action) in the given context information. If the newly adapted case has enough novelty with respect to all the existing robot task cases, it is added to the RTCD.

B. Retrieval and Similarity Measures

The case retrieval, a main part of our framework, assesses the similarity of a given query to the cases in the RTCD

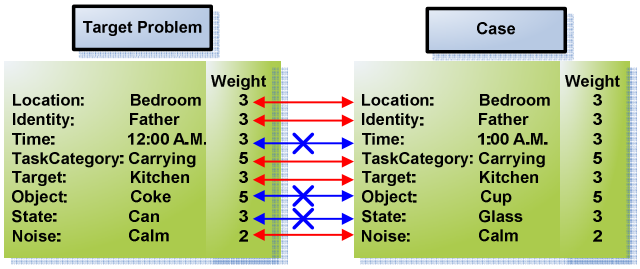

Fig. 4 Example of a Simple Similarity Comparison

Fig. 4 shows a matching process between the target problem and a case, which is chosen from the Case DB. It finds the closest match among the cases in the Case DB. Each case consists of a predefined set of features that are defined by a name and a data type, which may be any of the string, float, integer, and Boolean types. The closest match is calculated by using the weighted Euclid distance like the Pythagoras theorem in n-dimensions. That is based on the k-nearest neighbor (k-NN) method which assumes all instances that correspond to points in the n-dimensional space [28, 29]. The distance between the search and a case is a floating point number between 0 and 1 and is calculated as follows:

$$case\_dist = \sqrt{weight_1 \times dist_1^2 + weight_2 \times dist_2^2 + \cdots weight_n \times dist_n^2}$$

For numerical attributes, we simply calculate the difference between the query value and the case value and normalize the result to the interval. In case of string data types (e.g. target, location, object, etc), a synonym word list and canonicalization are used to help in dealing with different notations that have similar meanings. Additionally, time representation can be transformed into arithmetically computable form for the distance measurement. The weights w_i can be assigned heuristically by a domain expert.

C. Cognitive Models for Task Adaptations

Performing a meaningful adaptation can be done by using an appropriate reasoning procedure [4, 26] on the basis of rich context information. However, to manage the efficient and flexible task planning, a set of cognitive models need to be shared or possessed by the human and the robot. The most essential models are those of the task and the interaction [30].

In the cognitive models, task characteristics (e.g., task category, sub-task, atomic action, target, object, etc), innate capabilities of the robot (e.g., wheel-based movement, grasping capability with two grippers, distance measurement with 12 ultrasonic sensors, etc), and the context information (location, user, time, etc) are considered for the robot to appropriately deal with the communication issues and task sequence adaptations while managing the interaction process in a friendly manner. Especially, task sequence substitution and

parameter adaptation have been performed to handle ambiguities arisen from incomplete information that is allowed to the robot in natural but limited communication.

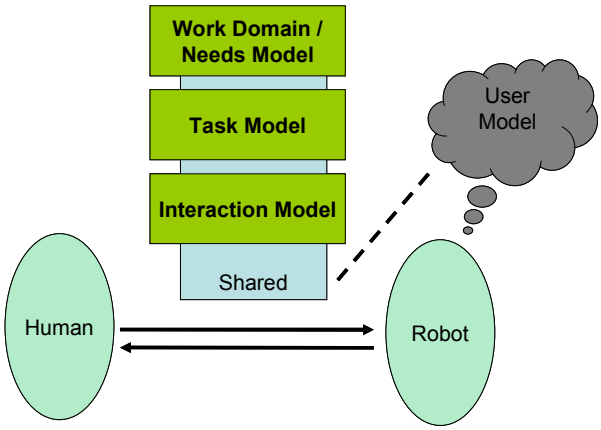

Fig. 5 A Model of Cognitive Interaction [30]

Table 3. Roles of Cognitive Models

Cognitive Models	*Roles*
Task Model (TM)	It generates the details of task procedures and modifies them according to the ongoing communication with the human. A task procedure can be modified by the known preference of the user and disambiguated information due to communication or situational knowledge acquisition while conducting the task.
Interaction Model (IM)	Concerns in issuing of questions and suggestions to the user. It is also used to build anticipation for the possible communication initiated by the human
Needs Model (NM)	The needs model contains the conditions and requirements that determine the boundaries of the robot's behavior that are not specified explicitly in the task model. It may represent commonsense restrictions or the robot's limitation or safety-related constraints.
User Model (UM)	Based on the information of the user's preference, it could produce adaptive behavior for individual users.

V. Implementation

In our implementation, to imitate cognitive models' functions, we use the JESS [27], a rule engine. In addition, to perform and test appropriate adaptation processes about the "*Bring me a coke*" task, we manually crafted the required task cases and cognitive rules. For realistic testing, our approach has been embedded in our robot system called IDRO. Fig. 6 and Fig. 7 show detailed steps needed during the task execution and their adaptations. From the start, the task model engages the whole task processes. Although the robot starts with a retrieved case, it can ask the user for a clear preference on the coke type based on the interaction model. In addition, if the robot fails to grasp the coke (or navigation to some-

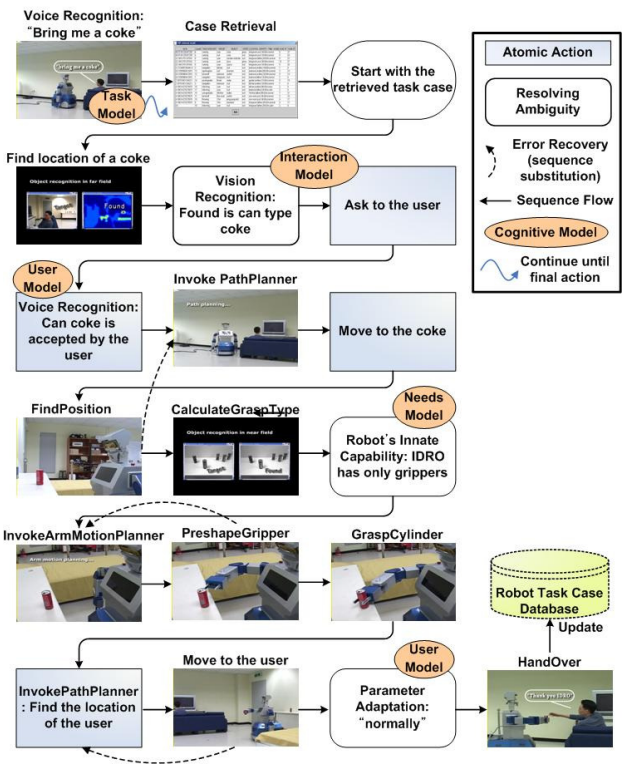

Fig. 6 Execution Flow for "*Bring me a coke*"

1) Get a command "Bring me a coke" from the user
2) Retrieve the most relevant task case from RTCD. However, this case is for a glass of coke.
3) Initiate actions with the retrieved robot task case.
4) To find the location of a coke, the robot calculates the location in far field with vision recognition capability.
5) However, the found is a can of coke. So, to solve the arisen ambiguity, the robot asks the user's preference.
6) The user says "A can of coke is also O.K."
7) The ambiguity above is solved.
8) The robot invokes path planner to find the shortest path to get to the coke
9) Using the planned path, the robot starts navigation to the coke.
10) If the robot fails in identifying the location of the coke can in near field, then go back to 7th sequence for the error recovery. If succeed, move to 11st sequence.
11) To grasp the coke, the robot calculates a grasp type, especially, cylinder type.
12) Introspecting the robot's innate capability, only two grippers are available to grasp the coke.
13) The robot invokes arm motion planner to calculate motion paths of arm and gripper.
14) Based on the arm motion planning, preshape the coke with the gripper observing with the vision (PreshapeGripper action is a hybrid function of arm motion control and vision recognition). If the preshaping fails, go to 13rd sequence.
15) If the preshaping succeeds, the robot grasps the can of coke carefully measuring its completion status with vision. (GraspCylinder action is also a hybrid one like PreshapeGripper).
16) After grasping the coke, the robot invokes path planner to get the shortest path to move to the user.
17) Start navigation to the user with the gotten path. If it fails in approaching to the user, go back to 16th sequence.
18) When the robot stands in front of the user, consideration of user's profile should be preceded before handed over the coke. In this case, because the user is 20 years old, polite manner is not required.
19) Hand over the coke to the user normally.
20) Finally, update the RTCD with the currently performed task case (a set of action sequences) if it has novelty comparing with the existing cases in RTCD.

Fig. 7 Step-by-step explanations: "*Bring me a coke*"

thing), it modifies the action sequences upon the task model. Finally, he selects a handover pattern (e.g. "politely" or "normally") because the user model participates. Those situational adaptations come from the communications

among the cognitive models (task, interaction, user, and needs model).

VI. Conclusion

This paper presents a cognitive CBR framework towards a task planning of home-service robots. Because the knowledge for the robot's task planning is extremely incomplete and dynamic, it is very difficult to formalize general rules to solve problems (i.e. planning tasks for a robot) automatically. However, in general, the CBR approach can integrate knowledge acquisition, reasoning, storage and learning in one platform. Hence, a system using CBR approach can add newly derived cases without changing the fundamental system structure, and the newly derived ones can be inserted into the case base for future usage. In that sense, we designed a cognitive CBR based on the four cognitive models to resolve ambiguities arisen during action execution. Additionally, our Robot Task Description Language (RTDL) – a new task structure for home-service robot – is designed to be used efficiently in planning a task, storing a number of cases, and changing action sequences for our Cognitive CBR framework. The framework and RTDL support interactions with the user to acquire insufficient information to perform a given task completely by taking into consideration the context information and the given task's characteristics.

To test our system's applicability, we have implemented the "*Bring me a coke*" scenario based on our cognitive CBR approach, and found that it helps HRI-based task planning in terms of ambiguity resolution and subsequently required action sequence adaptations. Although a home-service task, which is the focus of this paper, could cover very limited number of services compared to a number of household chores that are done by housewives daily, we believe that the cognitive CBR approach to the robot's task planning for home-service above can be used very substantially, especially, when reusing new tasks and resolving ambiguities for the task completion.

VII. Acknowledgment

This research (paper) was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

VIII. References

[1] H. S. Kim, Y. C. Jung, and Y. K. Hwang, "Taxonomy of Atomic Actions for Home-Service Robots," *Journal of Advanced Computational Intelligence and Intelligence Informatics* (JACIII), 2005.

[2] Honda's ASIMO: http://asimo.honda.com/

[3] S. Russell and P.Norvig, *Artificial Intelligence A Modern Approach, Prentice Hall*, (2nd ed.), pp 375-416, 2003.

[4] I. Watson and F. Marir, "Case-Based Reasoning: A Review," *AI-CBR*, University of Salford, UK, Available: http://www.ai-cbr.org/classroom /cbr- review.html

[5] M. Balázs, S. Imre, and K. Lajos, "Adaptation Methods in Case-Based Reasoning," *microCAD'99*, Hungary, 1999.

[6] W.Wilke and R. Bergmann, "Techniques and Knowledge used for Adaptation during Case-Based Problem Solving," *IEA/AIE*, Vol. 2, 1998.

[7] A. R. Golding and R. S. Rosenbloom, "Improving accuracy by combining rule-based and case-based reasoning," MITSUBISH ELECTRIC RESEARCH LABORATORY, TR-94-19a, 1995.

[8] W3C's XML specification: http://www.w3.org/XML

[9] R. Simmons and D. Apfelbaum, "A Task Description Language for Robot Control," *Proc. Of the IEEE/RSJ Int. Conf. on Intelligence Robotics and Systems (IROS'98)*, vol.3, pp. 1931-1937, Canada, 1998.

[10] S. Leak, "Robot Sensor Language," *Proc. of the Goddard Conf. on Space Applications of Artificial Intelligence and Robotics*, 1987.

[11] Robotic Markup Language Project: http://www.roboml.org/

[12] G. Boudol, "Atomic Actions," *EATCS*, vol. 38, pp. 136-144, 1989. Available: http://www.sop.inria.fr/meije/personnel/Gerard.Boudol.htm

[13] Y. K. Hwang, M. J. Lee, and D. M. Lee, "Robots' Role in Ubiquitous Computing Household Environments," *Int. Symposium on Robotics*, France, 2003.

[14] M. Ehrenmann, R. Zollner, O. Rogalla, and R. Dillmann, "Programming Service Tasks in Household Environments by Human Demonstration," *IEEE Int. Workshop on Robot and Human Interactive Communication*, pp. 460-457, 2002.

[15] L. Peterson, D. Austin, and D. Kragic, "High-level Control of a Mobile Manipulator for Door Opening," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 2333-2338, 2000.

[16] R. J. Firby, P. N. Rpokopwicz, and M. R. Swain, "Plan Representations for Picking Up Trash," *Int. Conf. on Tools with Artificial Intelligence*, pp. 496-497, 1995.

[17] K. Nagatani and S. Yuta, "An Experiment on Opening-Door-Behavior by an Autonomous Mobile Robot with a Manipulator," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 45-50, 1995.

[18] U. D. Hanebeck, C. Fischer, and G. Schmidt, "ROMAN: A Mobile Robotic Assistant for Indoor Service Applications," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2. pp. 518-525, 1997.

[19] M. R. Cutkosky, "On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no 3, pp. 269-279, 1989.

[20] S. A. Stansfield, "Representing Generic Objects for Exploration and Recognition," *IEEE Conference on Robotics and Automation*, vol. 2, pp. 1090-1095, 1988.

[21] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," *IEEE Conf. on Robotics and Automation*, pp. 1824-1829, 2003.

[22] M. Kruusmaa and B. Svensson. "Using Case-Based Reasoning for Mobile Robot Path Planning," *Proc. of the 6th German Workshop on Case-Based Reasoning,* Berlin, 1998.

[23] E. J. Perez, C. Urdiales, F. Sandoval, and J. Vázquez-Salceda. "A CBR strategy for autonomous reactive navigation learning," *10th Int. Symposium on Robotics and Applications (ISORA 2004),* Spain, 2004.

[24] PRODIGY: http://ai.eecs.umich.edu/cogarch0/prodigy/ and http://www.intelligent-systems.com.ar/intsyst/prodigy.htm

[25] D.S. Nau, S.J.J. Smith, and K. Erol, "Control Strategies in HTN Planning: Theory versus Practice," *Proc. of AAAI-98/IAAI-98*, pp. 1127-1133, 1998.

[26] F. Gebhardt, A. Vob, W. Grather, and B. Schmidt-Belz, *Reasoning with Complex Cases*, Kluwer Academic Publishers, (1st ed.), 1997.

[27] Sandia National Labortory's JESS: http://herzberg.ca.sandia.gov/jess/

[28] Classification Problem: k-Nearest Neighborhood algorithm: http://www.xs4all.nl/~dpsol/data-machine/nmtutorial/classificationproblemknearestneighboralgorithm.htm

[29] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, pp.176~192, Willey-Interscience Publication, (2nd ed.), 2001.

[30] K.W. Lee, H.R. Kim, W.C. Yoon, Y.S. Yoon, and D.S. Kwon, "Designing A Human-Robot Interaction Framework For Home Service Robot." *IEEE Int. Workshop on Robots and Human Interactive Communication*, ROMAN 2005, pp. 286-293, 2005.