Case-Based Reasoning Approach to Task Planning of Home-Service Robots

Yuchul Jung, Yong K. Hwang, Manjai Lee
Digital Media Laboratory
Information and Communications University
Seoul, Korea
{enthusia77, ykhwang, manjai}@icu.ac.kr

Abstract

Home-service robots are expected to perform a wide range of tasks commonly encountered in a household environment. For autonomous operation robots should possess intelligence to plan their actions to carry out these tasks from the beginning, or they should at least have the ability to learn to plan for more tasks during their operation. Since it is impossible to predict all tasks in advance and write programs for robots to perform the tasks, it is best to endow robots with a learning capability. We use a case-based reasoning approach to home-service-robot learning because of the richness and diversity of information needed for task planning. Given a new task, a robot finds a closest task among the tasks it knows how to plan for, and it modifies the plan to adapt to the new task. The “task metric” measures the distance between tasks based on robot action type, object involved in the task, and the context in which the task is ordered. With an appropriate task metric, the robot’s task planner finds similar tasks in the knowledge base and adapts the corresponding action plans to generate a plan for a newly given task.

Key words: Home-Service Robot, Task Planning, Case-Based Reasoning (CBR), Task Metric

1. Introduction

The efforts for designing an intelligent home-service robot have led to the development of humanoid robots like ASIMO [1]. This is because household environments are designed for the comfort and convenience of humans with an average physique. Currently only simple household services like carrying objects, vacuuming, and operating household appliances are possible in a carefully controlled environment. To make a robot system that deals with real, complex scenes for household tasks, a set of target tasks and services need to be specified before automatic planning algorithms can be developed for the robot to perform the tasks.

A task planner devises a plan for a complex task as a sequence of simple actions called “atomic actions”, which the robot knows how to execute without further planning [2]. For example, if a human gives an order to a robot such as “bring me a Coke,” then the robot’s task planner should decompose the task as follows 1) find the location of a can of Coke, 2) plan a path and navigate to the can, 3) grasp the can, 4) plan a path and navigate to the user, 5) hand over the Coke to the user. Although a general-purpose inference engine could be used to solve this problem, it requires a huge body of knowledge before it can plan action sequences for a variety of household tasks. We instead take a procedural-knowledge approach to planning a set of frequently-used commands such as “bring an object to a destination,” and store the action sequences explicitly in a knowledge base. We do this for as many tasks as we can manage. For the tasks not in the knowledge base, we use an instance-based learning algorithm to find the most similar tasks and generate a plan by modifying the plans for the similar tasks. Because of the wide range of household tasks and the variety of contextual and environmental conditions changing over time, we have chosen a case-based reasoning (CBR) algorithm for the learning mechanism.

Two things must be done for our algorithm to succeed. First, a set of household tasks must be classified into meaningful groups so that a representative task from each group collectively forms a good coverage of all the tasks. Second, the task metric, which measures the closeness between two tasks, must be defined so the CBR algorithm can find similar tasks in the knowledge base and generate plans for a given task. These two steps are essential to expedite the planning process and increase the validity of resulting plans. Our algorithm is designed by carefully coordinating results from three areas of research: 1) automated hierarchical planning systems developed in artificial intelligence field [3, 4], 2) knowledge management (KM), which advocates reusing previous problem solving and decision making experiences to improve organizational processes [5], and 3) a case-based reasoning system, which can reuse past problem-solving experiences and learn from solving new problems [6]. Building on this foundation, we are developing an automatic task planner for a general-purpose home-service robot. This paper in particular presents the household task classification, the task metric to be used by the CBR algorithm, the architecture of the task planner (an integrated set of methodologies including hierarchical generation), case-based reasoning (including *k*-nearest-neighborhood algorithm), and case storing/retrieval for reusing experiences to support task planning.

2. Case-Based Learning for Task Planning

After a user's input command, the robot's task planner analyzes various attributes of the given task and searches similar cases by exploring the knowledge base. If there exists a matching task with enough similarity to the given task, the preplanned action sequence of the matching task is appropriately modified to generate a plan to carry out the given task. If the new plan has enough novelty with respect to all the existing plans, it is added to the knowledge base which constitutes a learning process of the task planner. When 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. This constitutes the second learning process of our task planner (Figure 1).

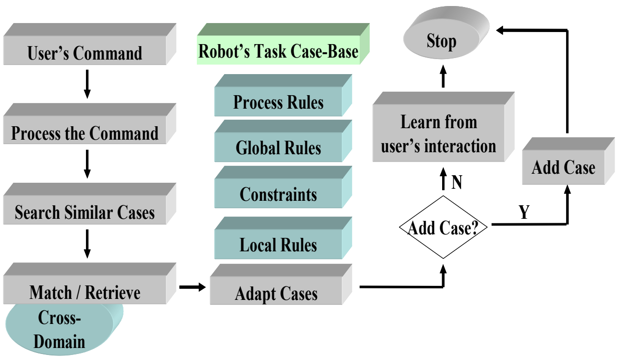

Figure 1. Case-Based Reasoning for Robot Task Planning

There are a number of algorithmic issues that need to be addressed for our task planner. To quickly access a set of cases from the case-base database that are similar to the current task, an efficient two-level indexing scheme similar to [7] is used. The first index set uses a reduced set of task attributes to identify tasks in the knowledge base that are similar to the current task with a minimal amount of computation time. The reduced feature set includes the types and arguments of the user commands which includes the main command verb, object involved in the task, and most of all the preplanned action sequences. Each action in the preplanned sequence is represented as a domain as shown in Figure 2, and the tasks with actions that are relevant to the current task are assigned a high matching score. Once a set of similar tasks are selected, a full set of task attributes are used to identify the most similar task or tasks. Because a reduced set of task attributes does not perform well for the nearest-neighbor problem in high-dimensional space, it can be more effective and simple to scan the entire data set rather than using sophisticated data structure [8].

The full feature set includes the detailed properties of objects involved in the task, user preference when the user command allows options, and locations when the task is robot navigation or object transport. To satisfy the user's present intention and the context in which the task is given, the full feature set also includes the present background information.

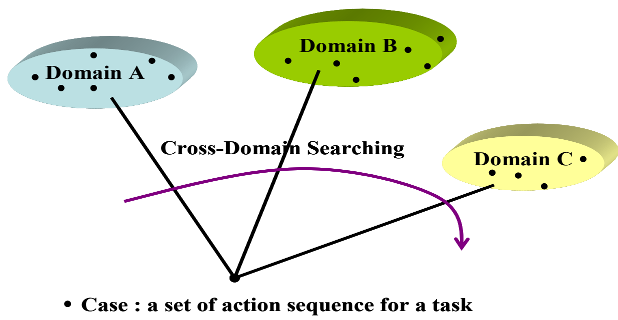

Figure 2. Cross-Domain Searching

The task metric, which is the main contribution of this paper, is a distance measure between two tasks. It can be computed based on robot action type, object involved in the task, and the context in which the task is ordered. Once the task metric is defined, we use the *k*-Nearest-Neighborhood Matching Algorithm (KNNM) [9] to select a small number of similar tasks with pre-computed plans. The KNNM compares the attribute value of each feature of each case in the set of similar cases to every corresponding feature's attribute of the current case, calculates the comparison values and then sums them for each case to get a total comparison value.

Just retrieving one relevant case from the case-base is not complete. A retrieved case is sometimes exactly the same as the current task, but mostly the retrieved case is only a similar one. Thus corresponding solutions should be modified carefully to fit the current situation and satisfy the current task's requirements. Figure 3 shows the case-adaptation process. When a case for a task is created, a series of rules is defined for adapting cases and it is applied to new cases whenever necessary.

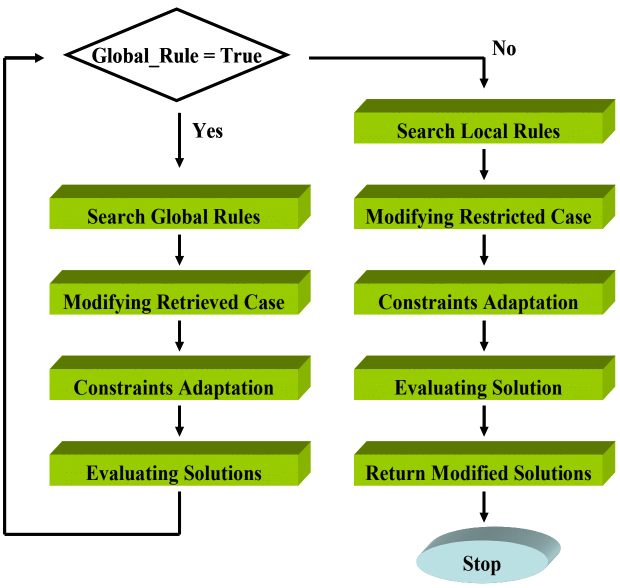

Figure 3. Case Adaptation Process

Adaptation rules are divided into global rules and local rules. First, global rules examine the problem fields and solution fields of the retrieved case. The rules are used to adapt the action sequence of the retrieved case that can check constraint satisfaction conditions specified in the knowledge base. The rules are solution fields of the retrieved case itself. If there are any constraint conflicts, the task planner provides a new problem-solving proposal. Otherwise, they adapt the solution of the retrieved case to the new problem, i.e., user's input. Second, after the global rules are applied, it immediately checks the local rules defined in the retrieved case. It applies these local rules to the retrieved case to perform local adaptation (i.e., unique to this case). In addition, to extract optimal action sequences for the task in the given situation, atomic actions that are imported from other cases in different domains are combined with those of the currently most similar case.

The task planning loop continues until the system finds at most *K* cases satisfying the task specification, or announces a failure. When the system finds a set of retrieved cases and performed successful adaptation with some of the *K* cases, it automatically updates the case-base and returns the adapted case. The task planner resorts to human assistance when it cannot find a similar case [10].

3. Experimental

Our task planner based on CBR has been applied to the tasks listed in Table 1. These tasks by no means cover all the household tasks, but they are the most common tasks expected to be performed by a home-service robot in the near future. To facilitate task comparison, adaptation, and learning based on case-based reasoning, all the tasks are converted to a hierarchical task network according to the way humans solve the tasks. Each task has its own sequence of atomic actions as illustrated in Figure 4.

Table 1. Various Household Tasks for a Robot

Category	Various Tasks	
Carrying	Bring me a (Coke, glass, fork, milk, ..)	*Eating*
	Bring me a (USB memory, pencil, ..)	*Work*
	Bring me a (radio, mp3 player, ..)	*Leisure*
	Bring me a (soap, shampoo, ..)	*Shower*
	Bring me a (ruster, vacuum, ..)	*Clean*
	Bring me a (watch, sack, shoes, ..)	*Etc*
Navigation	Go to the (bedroom, kitchen, living room, ..)	
Reading / Informing	Reading (book, magazine, newspaper, news, memo), Telling (News, weather, temperature, schedule)	

Suppose that user's command is "Bring me a glass of Coke." The cross-domain search process would then find two cases "Bring me a Coke" and "Bring me a glass." A meaningful adaptation for the request would be a combination of these two cases including additional atomic actions like "Pour Coke from the Coke-can into the glass." There could be various combinations of sequences for the task, but to guarantee the validity of the solution, adaptation rules are defined according to the criteria such as the task's objective, constraints, the object's type, cost-effectiveness, available tools, and the user's preference. The sample adaptation for a task "Bring me a glass of Coke" is illustrated in Figure 5.

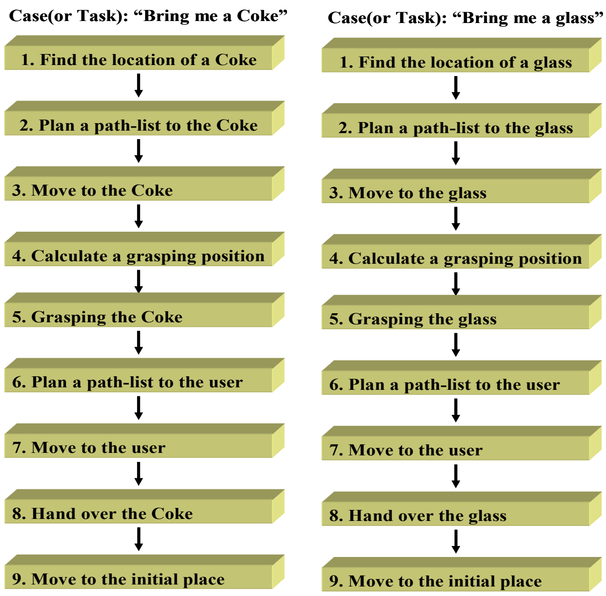

Figure 4. Example of Hierarchical Task Network

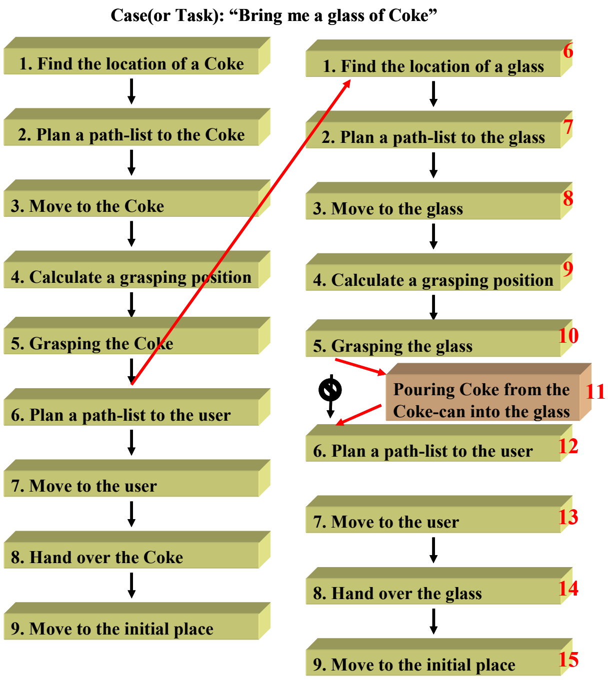

Figure 5. Task Adaptation Procedure for "Bring me a glass of Coke"

To adapt a case properly in the given context in advance, the task planner has a conceptualized behavior model which includes necessary sub-tasks for satisfying the user's command. The bottom part of Figure 6 shows 4 staged sub-tasks for "Carrying" task except the user's command/stop phase.

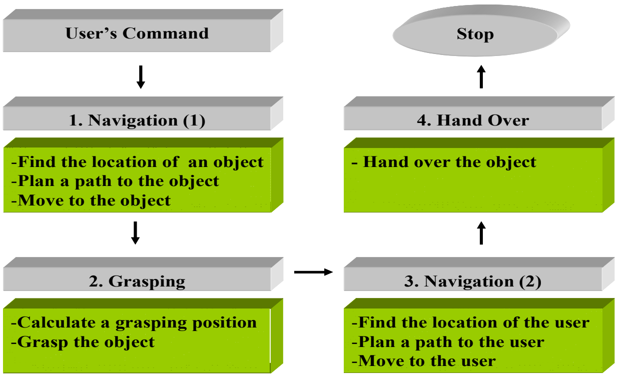

Figure 6. A Behavior Model of "Carrying" task

Each sub-task such as Navigation, Grasping, and Hand over, has its own local rules to adapt subordinating atomic actions. The atomic actions for a household robot are classified according to robot's action property, computation complexity and task completeness [11].

A short summary of atomic actions that are defined for a CBR task planning is shown in Table 2. This table is based on our experimental approach and non-absolute criterions. It could be expanded or modified considering future need and extension.

Table 2. Atomic actions of a household robot

Units	Detailed Atomic Actions
AT 0100	Movement (forward, back, wait, stop, ..)
AT 0200	Grasping (grasp, release, position calculation, ..)
AT 0300	User Interaction (getting command, prompt, ..)
AT 0400	Recognizing external sensors (calculating distance, ..)
AT 0500	Finding Location (from Vision or Knowledge DB)
AT 0600	Humanlike gestures (shaking hands, nodding, ..)
AT 0700	Getting external information (from Web site, ..)

4. Conclusion

This paper presents a case-based reasoning approach for the automatic task planning of a home-service robot. It consists of *k*-nearest-neighborhood matching, a task metric for measuring similarities among tasks, and plan adaptation and learning mechanisms. Our task planner has been applied to 3 categories of about 50 common household tasks, and an example of cross-domain plan adaptation is shown in the experimental section. We are implementing our CBR task planner in a home-service robot called IDRO. There are many issues encountered during our work that need to be addressed in future research. We plan to store all the robot's tasks related to household services in a systematic knowledge database to scale up our task planner to hundreds of objects/tasks. More elaborate representations of contextual information and additional constraints like appropriate etiquette and safety will also be incorporated into our task planner.

References

[1] ASIMO: http://asimo.honda.com/

[2] Muñoz-Avila, H., Aha, D. W., and etc, SiN:Integrating Case-based Reasoning with Task Decomposition," *Int. Joint Conf. on Artificial Intelligence*, Seattle, USA, 2001

[3] Erol, K., Hendler, J., and Nau, D. S., "HTN Planning: Complexity and Expressivity," *Proc. of National Conf. on Artificial Intelligence*, Seattle, USA, 1994

[4] Muñoz-Avila, H., Gupta, K., and etc, "Knowledge-Based Project Planning," *IJCAI-2001 Workshop on Knowledge Management and Organizational Memories*, Seattle, USA, 2001

[5] Liebowitz, J., *Knowledge Management Handbook*, CRC Press, 1999

[6] Janet K., *Case-based Reasoning*, Morgan Kaufmann Publishers, Inc., 1993

[7] Xiaoli Q. and William C. R., "Applying Case-based Reasoning to Mechanical Bearing Design," *Proc. of ASME Design Engineering Technical Conf.*, Baltimore, USA, 2000

[8] Shaft, U., Goldstein, J., and Beyer, K., "Nearest Neighbor Query Performance for Unstable Distributions," Technical Report TR 1388, Depart. of C. S., Univ. of Wisconsin at Madison.

[9] Duda, R. O., Peter E. H., and David G. S., *Pattern Classification*, pp.176~192, Willey-Interscience Publication, 2^{nd} edition, 2001

[10] Jung, Y. C., Hwang, Y. K., and Lee, M. J., "Knowledge Query/Re-query System based on Interaction between Human and Ubiquitous Robot for Personalized Serviced in Household Environment," *HCI 2004*, Pyeongchang, KOREA, 2004

[11] Kim, H. S., Jung, Y. C., and Hwang, Y. K., "Taxonomy of Atomic Actions for Home-Service Robots," *SCIS & ISIS 2004*, Tokyo, JAPAN, 2004