

Multiple Script-based Task Model and Decision/Interaction Model for Fetch-and-carry Robot

Yo C. Kim¹, Wan C. Yoon², Hyuk T. Kwon³, and Geun Y. Kwon⁴

^{1,2,3,4} Intelligent Systems Laboratory, Industrial Engineering Department,
Korea Advanced Institute of Science and Technology, Daejeon, Korea,
e-mail:(nkmissio, wcyoon, jehdeiah, eulory)@kaist.ac.kr

Abstract— The characteristics of the behaviors of home-service robots are different from those of other intelligent systems. Based on these characteristics, we developed a task model that can both support meaningful interactions and improve task performance. The task model is able to generate the robot's diverse actions and aid its interaction management. A decision/interaction model was established to decide autonomy of decision problems that are embedded on the task model. When Interactions with humans are required, it also manages timing of the interactions. We demonstrated the applicability of the developed system with a computer simulation.

I. INTRODUCTION

Humans tend to endow interactive systems such as service robots with personality [1]. Appropriate interactions encourage task performance of the robot system [2][3]. Consequently, this report includes a research study of a system that plans and manages high-level robot actions based on the interactions between a service robot and a human.

In traditional artificial intelligence (AI) fields, there has been a significant amount of research regarding processes to devise plans (see [4] for an overview of active research in AI planning and [5] describes the paradigms of AI planning). However, since home-service robots share cognitive and physical domains with humans, the robot's actions have to be considered in the following matters. Firstly, the interactions between the human and the robot should enhance the level of task performance [2]. It does not just mean that interactions need to exist. Interactions should also be managed for the optimization of tasks [6]. Specifically, home-service robots should grasp the user's intention by using interaction. Secondly, the robot has to show similar actions to those of humans. For achieving that, an approach that generates action by a cognitive process similar to those of humans will be useful [7]. Finally, by possessing diverse action styles for a given task, the robot is able to work in different situations flexibly.

Human knowledge can be divided into declarative and procedural knowledge [8]. Declarative knowledge is explicit knowledge of which we are conscious and can declare. On the other hand, procedural knowledge is the implicit knowledge about how to perform various activities. Procedural knowledge can be represented as scripts [9], event schemas. In a fetch and carry errand, the human has several scripts about how to deliver a given

object, and (s)he deals with this by implicit process. But, for decision making problems, for example what object (s)he delivers and what auxiliary tool are needed, (s)he goes through explicit and concrete cognition.

This paper is organized as follows. In the next section, we explain the configuration of a developed system based on the properties of knowledge. Section 3 and 4 show a task model and decision/interaction model used in the system. Section 5 describes the simulation to which the system applied. Section 6 reviews related works, and Section 5 contains our conclusions and future works.

II. OVERVIEW OF SYSTEM

Because it is one of the most typical home services and it is easy to show cognitive scientific issues, the fetch-and-carry errand is chosen for the target task category.

The proposed system contains roles of planning the robot's action, requesting each action to each related module, dealing with cognitive action or decision, and managing and executing interactions with humans. The robot's physical action, the following result, and robot knowledge concerning environment are not involved in this research target. They are implemented by virtual codes for simulation.

Fig.1 shows the system configuration. The light colored blocks are the models of which the system consists, and the deep colored blocks are the virtual models set for the simulation. The Task Manager, in accordance with the robot's mission, composes *Script Sets*. The script sets are composed by using information stored in the Script Database (Script DB). Cognitive actions described in script sets induce decision making with the Knowledge Database (Knowledge DB). Physical actions described in script sets are worked by each Physical Action Executor. In addition, the executed results are delivered to the Task Manager. Information that is produced during the execution of the Task Manager is managed by Knowledge DB.

The Knowledge DB plays the role of working memory in the robot system, and the information in the DB should be accessible by the developed models at any time. The information is divided into four types. The first is Object Ontology and involves properties of target fetch-and-carry objects. The second is the Environment

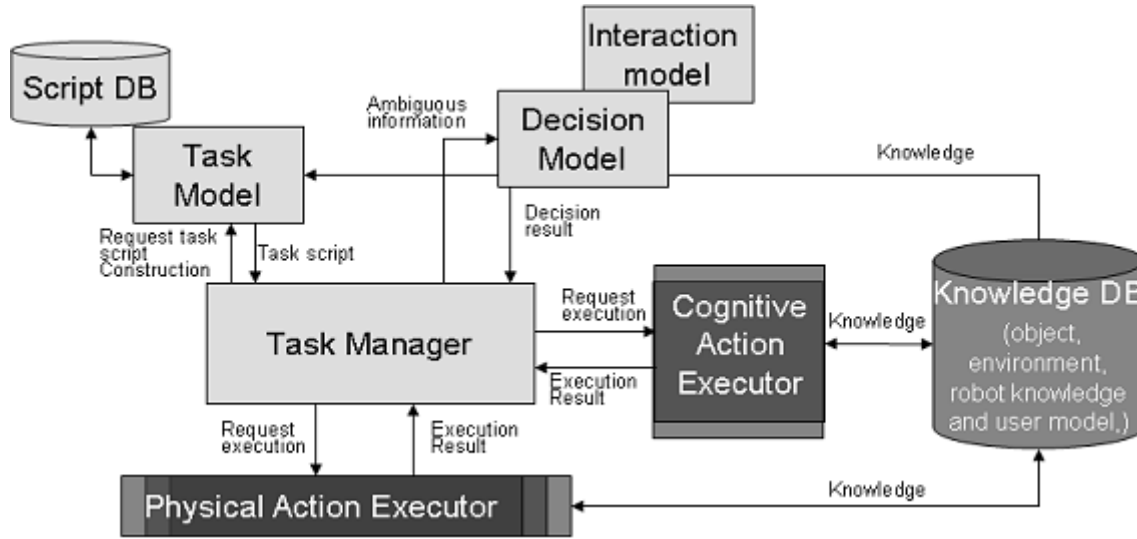


Fig. 1 System configuration diagram of the developed system

Information of the properties of each environmental object that is set up virtually. The third is a User Model, which describes user preferences (lists and ratios) regarding target objects. The fourth is Robot Knowledge, which means information about target objects identified by the results of interactions, inference, and actions.

Each Physical Action Executor is worked by each request from the Task Manager. Examples of these executors' names are MoveToLocationOfBeverage, UncoverTargetLocation, PickUpTargetBeverage, CoverTargetLocation, etc. The names are not only the atomic actions defined in script DB, but also the goals of each executor.

III. MULTIPLE SCRIPT BASED TASK MODEL

The task model consists of scripts that represent processes by which humans deliver objects. Scripts are produced by engineers with a consistent form. Why does the robot not plan or compose its own actions? In addition, why should they make robot's actions by using scripts? In answer to the questions, there are two benefits. Firstly, since the robot acts in accordance with scripts that humans describe, its actions are predictable and reliable. Secondly, we can arrange inducing points of interactions for proper positions. Therefore, it will be possible to interact with humans in a timely and meaningfully manner.

The task model has a process illustrated in Fig. 2.

A. Multiple scripts

The previously mentioned script can be described by two methods. The first method is that one script includes every branch of several situations. By using this method, the robot can predict what will occur while performing specific actions. It is additionally assured that the robot shows coherent actions in task performance. Nevertheless, one script is limited to expressing many different kinds of scripts, so it is hard to show the flexibility of tasks. The

second method is that each script is written for each set of actions. It is called multiple scripts and we can describe diverse actions through them.

When using multiple scripts, it is important to realize what action should be executed, and what problems should be resolved. Besides, it is critical to keep coherency of action courses during a specific task's performance. For that reason, in multiple scripts, a script that leads interactions and actions is essential. It is defined as a *dominant script*. The dominant script is selected in multiple scripts and it serves as a task agenda. The dominant script includes a characteristic, which features the ability to eliminate different scripts from itself while a task progresses.

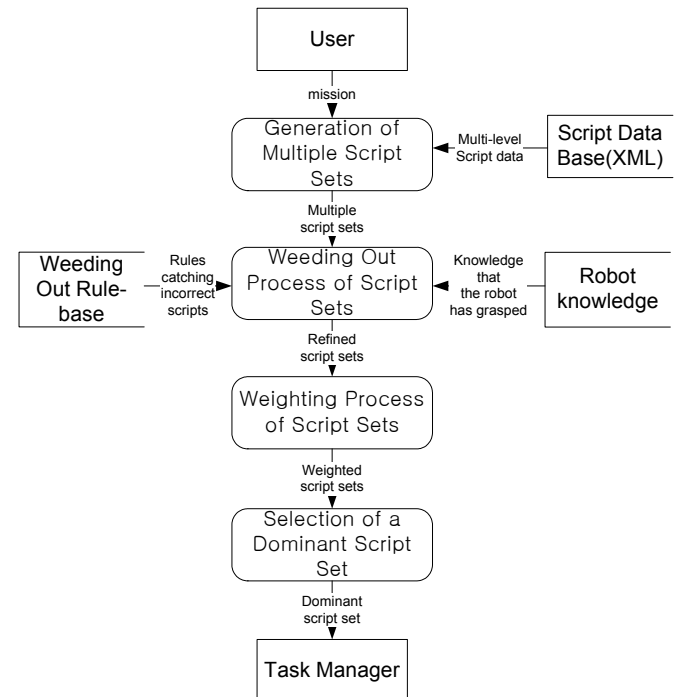


Fig. 2 Processes and data flows in the task model

B. Scripts of mixed abstraction and Script sets

Humans can describe action scripts with various levels of abstraction [10]. We express every action as either an atomic or a concrete action, or as an abstract action. Furthermore, it is possible to use mixed expressions in varying levels of abstraction. For example in the fetch-and-carry beverage action, one can consider the process of obtaining a beverage as a concrete action, while thinking of the process of obtaining a cup as an abstract action.

Scripts of mixed abstraction regarding fetch-and-carry tasks are stored in Script DB. Script DB is composed by Extensible Markup Language (XML). Each element of XML contains a higher action that decomposes into several lower actions in order of achievement. So, one higher action can be expressed by different lower actions. The lower actions can be either an abstract type or a concrete one. But, abstract types of the lower actions should be defined by other actions in the database. With this structure, a goal of a fetch-and-carry mission can be expressed by a set of concrete actions.

Script DB includes the information whether the action is physical or cognitive. The cognitive type of actions makes questions to humans or inferences of knowledge. Moreover, the physical types of actions link to the Physical Action Executor. Script DB also allows expression of the related properties, when specific actions can be changed with other actions. Finally, Script DB includes estimated utility information of each action. The following XML codes show an example of script DB.

```

1 <script goal="FetchAndCarryBeverage"
  ScriptID="FCB037" utility="1">
2   <action decomposetype="abstract"
    acttype="physical">BeverageAcquisition</action>
3   <action decomposetype="abstract"
    acttype="physical">ContainerAcquisition</action>
4   <action decomposetype="abstract"
    acttype="physical">Drinking</action>
5 </script>
6 <script goal="BeverageAcquisition"
  ScriptID="FCB039" utility="1">
7   <action decomposetype="concrete"
    acttype="cognitive">DecideTargetBeverage</action>
8   <action decomposetype="concrete"
    acttype="cognitive">IdentifyLocation</action>
9   <action decomposetype="concrete"
    acttype="physical">MoveToLoc</action>
10  <action decomposetype="concrete"
    acttype="physical"
    BranchableProperty="InvisibilityOfBeverage:yes">UncoverT...
11  <action decomposetype="concrete"
    acttype="physical">PickUpTargetBeverage</action>

```

```

12  <action decomposetype="concrete"
    acttype="physical">CoverTargetLoc</action>
13 </script>

```

As explained above, the abstraction actions for a task is decomposed into concrete actions in regular sequence. It is called a script set. The script set forms from the information provided in script DB. In other words, every sequence of lower actions, which can be decomposed from higher action, is enumerated for each script set. In the enumeration process, the irregular or inexecutable script sets can be generated. The incorrect script sets are eliminated by the rules of weeding out process, and the rest of the script sets are used for the robot's performance and interactions. Examples of script sets follow:

- 1 DecideTargetBeverage[CognitiveAction]→DecideNecOfContainer[CognitiveAction]→;NecessityOfContainer:no;IdentifyLocationOfBeverage[CognitiveAction]→MoveToLocationOfBeverage→;InvisibilityOfBeverage:yes;UncoverTargetLoc→PickUpTargetBeverage→CoverTargetLoc→Drinking(MoveToDrink→DrinkBeverage)
- 2 DecideTargetBeverage[CognitiveAction]→DecideNecOfContainer[CognitiveAction]→;NecessityOfContainer:no;IdentifyLocationOfBeverage[CognitiveAction]→MoveToLocationOfBeverage→;InvisibilityOfBeverage:no;PickUpTargetBeverage→Drinking(MoveToDrink→DrinkBeverage)
- 3 BeverageAcquisition(DecideTargetBeverage[CognitiveAction]→IdentifyLocationOfBeverage[CognitiveAction]→MoveToLocationOfBeverage→;InvisibilityOfBeverage:yes;UncoverTargetLoc→PickUpTargetBeverage→CoverTargetLoc)→DecideNecOfContainer[CognitiveAction]→;NecessityOfContainer:no;Drinking(MoveToDrink→DrinkBeverage)

C. Dominant script selection and Replacement

A dominant script is selected from the script sets by a weighting process. Each script set is evaluated by three measures: efficiency, reliability, and utility. Efficiency is represented by:

$$efficiency = 1/(\# \text{ of actions}) \quad (1)$$

The equation is based on an assumption that the user generally prefers simple procedures which come to mind easiest for selection of script sets. For reliability, it is important to avoid performing unintentional actions. The reliable script sets involve as many cognitive actions as possible, so that they admit margins for replacements of a dominant script even if the script sets become the dominant script. Consequently, reliability is expressed by the number of cognitive actions. Utility values are calculated by the multiplication of each utility of the related script.

Each weight of script sets equals to:

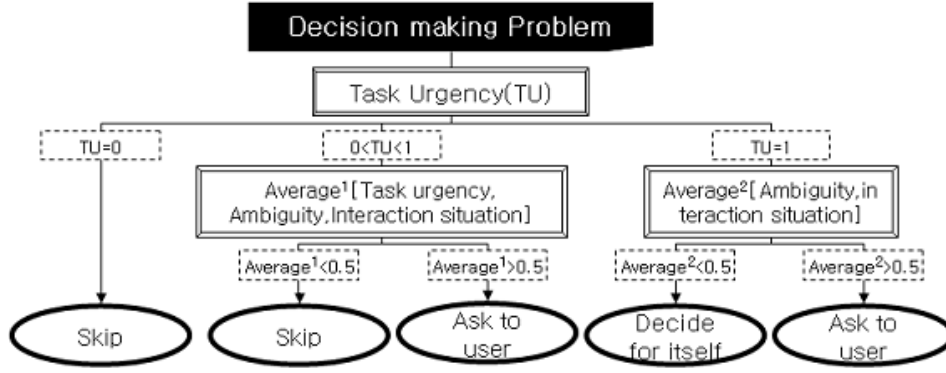


Fig. 3 Tree for selection of the decision strategy

$$weight = efficiency \times reliability \times utility \quad (2)$$

The dominant script is selected by weighted random selection based on the calculated weights. The weighted random selecting method ensures flexibility of the robot's action procedures.

Whenever each action of the dominant script is executed; script sets that are different with the executed node of the dominant script are eliminated. After the robot makes a decision by cognitive action, script sets that are not accorded with the result of the decision-making are also eliminated. Through this process, the total number of script sets converges on one. If the dominant script is not accorded with the result of decision-making, then the current dominant script is discarded. New dominant scripts are selected by the same selecting method described above from the extant script sets.

D. Utility reinforcement

Utilities expressed in each element of Script DB influence selection of the dominant script. These utilities are reinforced by the results of the robot's performance. After a task is performed, the finally determined dominant script has more probability than other script sets to succeed at the task. Hence, the utilities of the elements in the Script DB that forms the dominant script increase. But, if the robot receives a cancel command from a user during the execution of an action, then each related element's utilities will decrease. Via continuous task performance, the robot shapes its own principal behavior styles [11].

IV. DECISION/INTERACTION MODEL

There are some cognitive problems to solve during the delivery of an object. For example, the agent has to grasp what kind of the thing (s)he wants, where it is, or whether (s)he needs auxiliary tools. These decision-making processes are defined as cognitive actions in the script sets. Furthermore, Cognitive Action Executors extract which candidates are for the cognitive problem. However, discovering how and when these problems should be resolved is a quota of a decision/interaction model.

The cognitive problems of service robots can be resolved by self-decision or interaction with the user. By utilizing these two methods appropriately, the agent can maximize effectiveness of the decision-making. In the situation that the interactions are required; it is needed to manage the points of time of the question. This is because too frequent questions annoy the user, and it is good to ask as many of the queries as possible at the same time. Therefore, according to each situation, the actions should be performed with consideration of the decision methods and timing of the interactions.

In order to decide the methods and timing of the decisions, three factors should be considered: Ambiguity, Task urgency, and Interaction situation. Ambiguity is the ambiguities of the candidates of decision-making. It is calculated by referring to the entropy of information:

$$Ambiguity = \frac{\sum_{i=1}^k p_i \log_2 \left(\frac{1}{p_i} \right)}{\log_2 \left(\frac{1}{1/k} \right)}, \quad (k \neq 1) \quad (3)$$

$$= 0, \quad (k = 1)$$

Where i is the index of each candidate, k is the number of candidates and p_i is the belief of each candidate.

Even if a cognitive action's turn has not come round yet, the action can be performed ahead of time. In this situation, Task urgency means the urgency of the action's execution. Therefore, it follows how long between the cognitive action's timing and the current progressive action.

Finally, the interaction situation implies how current situation is easy to interaction. Some kinds of knowledge can be taken: user's feeling, busyness, distance between user and robot, and whether conversation is in progress or not.

From the three factors, the strategy of the decision and the interaction is defined in proportion to the tree of Fig. 3. The strategy has three types of result: Skip, Ask to user, and Decide for itself. If the Task urgency of the cognitive action is very high, the action has to be performed now by any strategy. Therefore, the Ambiguity and Interaction situation are additionally required. But, when the decision is not very urgent, it is

resolved by either asking the user or skipping over the action. Even though the cognitive action is skipped over, it can be performed when its Task urgency becomes higher.

The outcomes decided by interactions or inferences are delivered to the Knowledge DB through the Task Manager.

V. SIMULATION

For simulation of the developed model, we suppose the home domain as the one in the following figure. In the supposed environment, a mug and cup are in a tea cabinet, and a bottle of cola, a bottle of juice, and a soda can are in a refrigerator. There is a jar on a range and a cola can on a dressing table. Books are located on a shelf in front of a television. (See Fig. 4)

It is assumed that the robot knows the location or properties of all the objects. Instead, the user's commands can be abstract (Example: "Give me a cup of water from the pot.") as well as concrete (Example: "Give me something to drink."). Cognitive or physical actions represented in the script sets are composed for the simulation environment.

The interface of the simulator is shown in Fig. 5. A button and field at the upper left part is for receiving several of the commands. Script sets for the mission place at the upper center of the window. Each weight of script sets is on the right of the script set's field. The dominant script is written in a field at the middle center part. The bottom fields contain the contents of the knowledge DB mentioned in the second section. For the fetch-and-carry errand, knowledge in the DB is in Table I.

We made 21 script elements for the Script DB. The system generated 58 script sets by them. Then, it fulfills each physical/cognitive action executor and leads the related questionnaire windows.

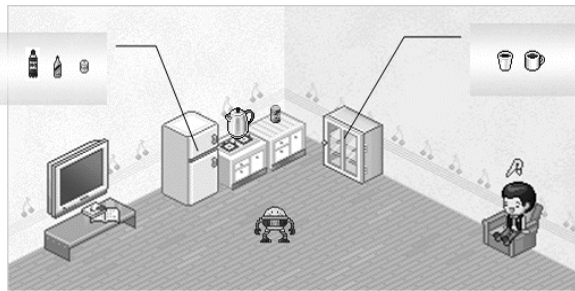


Fig. 4 A home environment for simulation of the proposed system

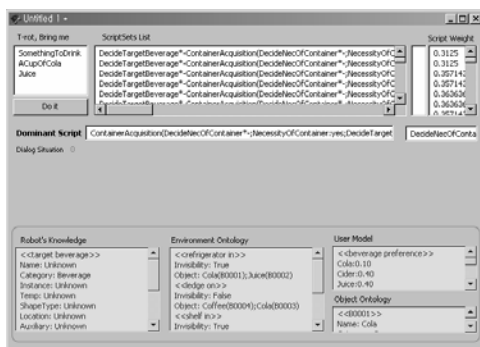


Fig. 5 Interface of the simulator

Table I Information in the knowledge databases

Knowledge types	Categories	Items
Robot knowledge	Target beverage	Name Category Instance Temp ShapeType Location Auxiliary
	Target auxiliary	Name Category Instance ShapeType Location
Environment ontology	Refrigerator in	Invisibility Object
	Ledge on	Invisibility Object
	Tea-cabinet in	Invisibility Object
	Distance	BetweenRobotAndUser
	Location	Robot
User model	beverage preference	<<object's names and its preference ratios>>
	container preference	<<container's names and its preference ratios>>
	user state	Busyness, <ratio> Feeling, <ratio>
Object ontology	<<instances of objects>>	Name Category Temp ShapeType Location

VI. RELATED WORKS

Several research studies regarding task model and interaction have been conducted recently. Systems of [12], [13], etc. instigate interaction with humans by script or task model and achieve the tasks of object recognition or search problem. In addition, [14], [15], etc. studied the interactions about ambiguous information in home-service robots. One distinct advantage of our research is that the robot manages interactions with diverse scripts.

For the most part, the proposed task model predominately follows case based reasoning [16] in terms of concept and structure. Since the reasoning system searches cases from the accumulated case-base as previous experiences and adapts to new situation or problems, the task model retrieves script sets and selects the dominant script as a current ballpark solution. While the robot behaves with the dominant script, it continuously evaluates and changes script sets and the dominant script. In this way, the robot gradually discovers the most appropriate script for the current situation.

VII. CONCLUSION AND FUTURE WORKS

We developed the task model, which instigates interaction with the user, as well as generating and managing flexible robot behaviors in fetch-and-carry errands, a representative home service. These models have a contribution to lead diverse behaviors based on script like human's ones and interactions with humans. We also developed the decision/interaction model on the task model. This model decides autonomy about problem solving, and manages the time of the interactions, so that makes the robot interacts comfortably with the users. The developed system is applicable to various personal service robots as well as fetch-and-carry robots and it is expected to facilitate the development of familiar and effective robots.

The model that is able to make decisions and manage interactions plausibly still has to be developed. Furthermore, applicability of the developed models in a real home service domain needs to be tested by integration with other technologies in a robot platform. By the integration, it is possible to adjust several parameters' values and weights. The developed system will also be improved by elaboration of the model. For example, the method of weighting script sets and reinforcement can be made more real by using measures that are more precise. Additionally, the integration of emotional research intimates, and the user model makes a more plausible interaction model. These elaborations will encourage developing robot systems that resemble human cognition processes.

VIII. ACKNOWLEDGEMENTS

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.

IX. REFERENCE

- [1] K. E. Bumby and K. Dautenhahn, "Investigating children's attitudes towards robots: a case study," in *Proceedings of the Third Cognitive Technology Conference*, 1999, 391-410.
- [2] T.W. Fong, "Collaborative control: A robot-centric model for vehicle teleoperation," Doctoral dissertation, tech. report CMU-RI-TR-01-34, Robotics Institute, Carnegie Mellon University, November, 2001.
- [3] E. Horvitz, "Uncertainty, Action, and Interaction: In Pursuit of Mixed-Initiative Computing," *IEEE Intelligent Systems*, Vol. 14, No. 5., September 1999, 17-20.
- [4] J. Rintanen and J. Hoffman, "An Overview of Recent Algorithms for AI Planning," *Konstliche Intelligenz*, Vol. 2., No. 1., May, 2001, 5-11.
- [5] C. Kohler, "A Hybrid Planning Architecture for Robotic Assistant Systems," Technical Report at the FAW (Research Institute for Applied Knowledge Processing), 2001.
- [6] S. Lauria, G. Bugmann, T. Kyriacou, J. Bos, and E. Klein, "Training personal robots using natural language instruction," *IEEE Intelligent Systems*, Vol. 16, No. 5., September-October 2001, 38-45.
- [7] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. (Second Edition), Prentice Hall, New Jersey: 2003, 1-5.
- [8] J. R. Anderson, *Cognitive Psychology and Its Implications* (third edition), W. H. Freeman, New York: 1994, 217.
- [9] R. Schank and R. Abelson, *Scripts, Plans, Goals and Understanding*, Erlbaum, NJ: 1977.
- [10] W.C. Yoon, J. Park and S.H. Lee, "A Diagrammatic Model for Representing User's Interface Knowledge of Task. Procedure Knowledge of Task Procedures," in *Proceedings of Cognitive Systems Engineering in Process Control*, 1996, 276-285.
- [11] B. F. Skinner, "The shaping of phylogenic behavior," *Journal of the Experimental Analysis of Behavior*, Vol. 24., No. 1., July, 1976, 117-120.
- [12] H. Kimura, T. Horiuchi, K. Ikeuchi, "Task-model based human robot cooperation using vision," *Intelligent Robots and Systems*, 1999, 701-706
- [13] L. E. Barnes, R. R. Murphy, and J. D. Craighead, "Human-robot coordination using scripts," Technical report, Center for Robot-Assisted Search and Rescue, 2005.
- [14] J. H. Hong and Y. S. Song, and S. B. Cho, "A Hierarchical Bayesian Network for Mixed-Initiative Human-Robot Interaction," in *Proceedings of the 2005 IEEE, International Conference on Robotics and Automation*, 2005, 3809-3813.
- [15] M. Jun, K. Iwase, and Y. Shirai, "Interactive Teaching of a Mobile Robot," in *Proceedings of the 2005 IEEE, International Conference on Robotics and Automation*, 2005, 3378- 3383.
- [16] J. L. Kolodner, "An Introduction to Case-Based Reasoning," *Artificial Intelligence Review*, Vol. 6., No. 1., 1992, 3-34.