Scene Understanding by Means of Knowledge and Model-Based 3D Vision for Autonomous Service Robot

Haruki Ueno and Hiroshi Negoto
Department of Computers and Systems Engineering,
College of Science and Engineering,
Tokyo Denki University,
Ishizaka, Hatoyama-cho, Saitama 350-03
E-mail: ueno@k.dendai.ac.jp
JAPAN

Working Group V/1

KEY WORDS: Knowledge-Base, 3D Vision, Robotic Vision, Cognitive Science, Service Robot, Scene Understanding

Abstract

This report presents the concepts and methodology of the knowledge-based 3D model vision system for human-type autonomous service robotic arm HARIS. The HARIS robot consists of 3D vision, intelligent scheduler, computer-based arm/hand controller, HARIS arm and human interface, and aims to serve aged and disabled persons on desk-top 3D object handling as an all-in-one system. Next-generation industrial robot is important application target as well. The 3D vision system recognizes 3D objects in a scene, measures distances among them and generates a scene description within a world model, under natural lightning condition within a room. A combination of 3D vision technologies is used for HARIS to achieve higher level scene understanding, which includes knowledge-based technologies, model-based technologies and 2D-primitives recognition technologies, in addition to usual image processing methodology. The usage of knowledge and model have resulted in flexibility in generating a scene as the result of imaging from partial and unclear image data. The world model plays a key role in integrating a variety of distributed functions as an autonomy in HARIS. The integration has been achieved by means of a generalized frame-based knowledge engineering environment ZERO++ as a software platform which was implemented in C++ for a real-time distributed computer environment via LAN. The vision system is working under an experimental environment in our robotic system.

1. INTRODUCTION

Autonomous service robots are required for the aged and disabled according to increasing of average ages year by year. By autonomous robots we mean that a robot achieves a higher level request by itself [Ueno94]. This type of abilities could be realized by an integration of knowledge-based AI technologies[Barr92, Ueno86], vision technologies[Shirai84] and robotic arm technologies[Onishi96]. The HARIS project aims to make a break through in the field of mainly service robots by demonstrating the high performance and flexibility of an autonomous human-type robotic arm/hand system. The HARIS robot system consists of a model-based 3D vision, an intelligent task and motion scheduler, an arm controller and a human-like robotic arm as an all-in-one system.

Although many trials have been done in the field of robotic vision over more than last twenty years there still exists a variety of problems to solve. In this paper we propose an effective robotic vision methodology for service robots which would be used to assist the disabled and aged in such a situation of desk-top object manipulations. In our approach a combination of knowledge-based reasoning, model-based 3D vision technology and Cognitive Science approach is used. The concepts and methodology of World Model is applied to integrate a variety of functions to achieve the autonomy.

We have developed a highly flexible human-like robotic arm for the project [Onishi96]. The design policy is to realize a human-like arm/hand functions in manipulating a variety of 3D objects for assisting human beings. The structure and functions of human arms, hands and fingers have been analyzed and evaluated in this direction. The robotic arm unit has single turnable body, a three-joint flexible arm with 8 d.o.f. and a five-three-joint-finger hand with 17 d.o.f. The fingers have 178 tactile sensors on their surfaces, which are connected to an embedded micro-processor for detecting objects and controlling finger motions in holding them. In addition, slipping motions can be detected. Usage of multiple tactile sensors results in higher flexibility in manipulating a variety of objects with simpler control mechanisms.

Cognitive science is considered in designing the software architecture of HARIS, i.e., term-oriented task scheduling and semantic network in representing an image of a scene [Ueno94, 96, 96b]. We think that the term-oriented task scheduling is similar to human's way of task scheduling in manipulating his/her own arms in achieving ordinary tasks in daily life. We are trying to mimic a human's way in realizing an autonomous human-type robot arm as much as possible. This is because a human friendly interface should be one of key issues in realizing service robots for serving human beings and our approach would be quite natural for a human who needs robot's services.

A concept of a mental image in representing a scene is also applied to express the world model. We suppose that a human being might create a mental image in his mind to express a scene as the result of image understanding, which should be used in scheduling tasks by means of combining transitive verbs, i.e., terms, such as 'approach', 'hold', 'carry', 'put' and 'release'. In addition, typical holding types are pre-defined to control the fingers, also in terms, such as 'grip', 'grasp' and 'nip'. By this method the task and motion scheduling can simply be done by symbol-level reasoning just as human beings should do in mind.

All the functions in integrating the HARIS software system have been realized in the generalized frame-based knowledge engineering environment ZERO++[Ueno95], which is a reimplemented version of ZERO[Ito86] in C++ to improve performance for the HARIS project. It must be noted that a powerful platform like ZERO++ is absolutely important in developing integrated intelligent robots under distributed computing environment with LAN.

In this paper, the outlines of HARIS arm and system are discussed in chapter 2 and 3 respectively. In chapter 4 knowledge-based 3D model vision is outlined, followed by intelligent scheduling and control in chapter 5. Concluding remarks are discussed in chapter 6.

2. HARIS ARM

Figure 1 is a photograph of the HARIS arm unit. The arm has three jointed-arms with 8 d.o.f. and a hand with 17 d.o.f.. Arm 1 and 2 consist of three bar-link mechanisms to obtain high torque and are connected by universal joints for easy positioning. Third arm is connected by rotation of forearm and flexion, extension and rotation with differential gear mechanism like wrist. The hand has five fingers like a human hand. The embedded arm control system achieves efficient and flexible manipulations.

For holding an object we have established a tactile sensing system introduced with new recognition method for multi-tactile sensors [Onishi96]. The hand for this project has advanced mechanism of the prior one[Ooshima90]. Every finger surface has a set of 4 to 10 tactile sensors in a diamond array distribution, in total 128 sensors are installed in the finger surfaces. In addition, in order to nip an object with pointing and middle fingers each side face has 7-10 sensors, 50 in total. Therefore, 178 tactile sensors are installed in total. Every sensor is connected to a 8-bit microcomputer chip installed within the hand unit, which recognizes an object by detecting a pattern of contacts with sensing switches. Each finger is able to detect slippage by detecting the deformation of the skin within a time unit.

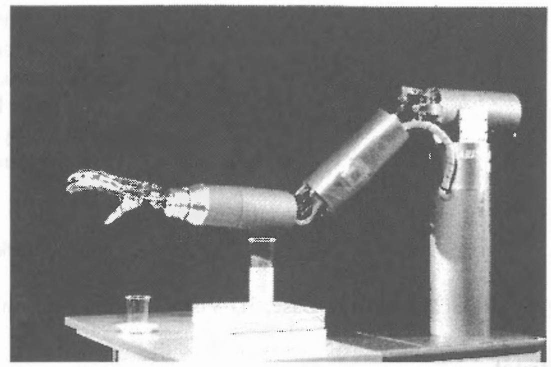


Fig.1 A whole view of the human-type robot arm.

A human hand is capable of performing a variety of holding actions. Typical holding actions are named as grasping, nipping, picking and so on. Our robot hand is designed to achieve something human-like actions in holding an object by means of sophisticated finger structures, motion controls and tactile sensors. The three-jointed plastic finger is designed to twist an object by single micro-motor. The feedback signals from the tactile sensors are used to recognize an objects, stop the actions of holding it and lift it with least torque. By this mechanism our robot is able to grasp an egg, hold a paper cup and lift a glass cup without troubles by simple control commands.

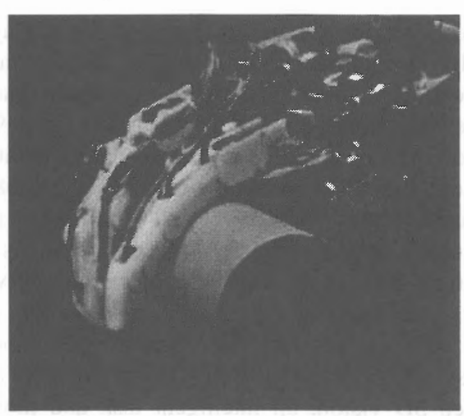


Fig. 2 An example of object holding by HARIS hand: Gripping.

In addition, in order to achieve sophisticated manipulations by simpler actions some fingers have special mechanisms as follows. Pointing and middle fingers are able to move in side-to-side directions up to 20 degrees and they can nip a small pipe as humans do. The thumb is actuated with a differential gear mechanism with a micromotor and an electromagnetic clutch, so that it can outrotate and flexion/extension. The thumb rotates first, then twists until it meets the object. Change of motion is controlled mechanically. An example of object holdings by the HARIS hand is shown in Figure 2.

3. OUTLINE OF HARIS SYSTEM

Figure 3 shows an overall image of the experimental HARIS robotic system environment during task execution and an example of partial scene descriptions in the world model. As shown in the figure the environment includes the robotic arm itself, a desk, cups, a tray on the desk, and so on. The HARIS robotic system includes two color CCD cameras for stereo vision. In an experimental system every potential object within the scene is known, i.e., pre-defined as a class frame within the knowledge base of the world model. We think this is a reasonable condition for current goal of the development, since we are focusing to serve higher level tasks for desk-top object manipulations.

The software system consists of a 3D vision system, intelligent task and motion schedulers, command generator, robotic interface, a monitor system, a human interface and two knowledge bases which are a world model and a task model, as shown in Figure 4.

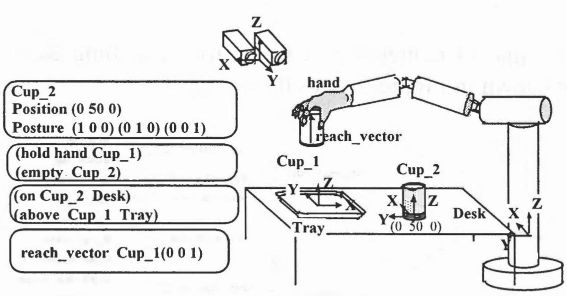


Fig. 3 An image of a scene and its description in the world model.

The world model is a knowledge base to represent a scene by a combination of a shape model, a functional object model and a spatial model. The world model plays a key role as the communication channels among the software modules of the system. The shape model is used to identify an object in 3D using the shape-related attributes of each object within a scene. The functional model, which is used for task scheduling and motion controls, represents functional attributes of each object. The spatial model describes the spatial relationships among the objects in a semantic network by means of relational operators and distance descriptors. The spatial model is used for both task and motion schedulings.

The user interface receives a user's request in a natural language sentense, such as "Put a red cup onto a white tray", understands it and sets a goal task. In this example "put" is the target task.

The scheduling model is another knowledge base to represent the knowledge about tasks and motions. The task scheduler receives the goal as a request from the user interface, makes a sequence of primitive tasks to achieves the goal by means of the task model as a sequence of transitive verbs, such as, "reach", "hold", "lift", "move", and so on. Since the HARIS system is designed based on the concepts of Cognitive Science a human-robot interaction could be achieved smoothly by means of verbal terms. Then the scheduler makes a sequence of motion primitives to achieve the tasks by means of the motion model, and generates a set of commands to execute the motions by the command generator, which are sent to the robot interface to control the robotic arm. The path plans are generated as a part of the motion scheduling using the knowledge stored in the functional model and spatial model.

The vision system, i.e., a model-based 3D scene understander, receives stereo color images through two CCD color cameras, identifies objects, measures distances, and generates the scene description in the world model. Since the knowledge of every candidate object is pre-defined as a template in a class frame the scene understanding can be achieved by generating instances for identified objects using partial information of them.

The robot interface receives the commands one by one in a sequence from the command generator, and executes them in a real-time mode. During the executions situations are monitored by the monitor system which is a combination of computation-based virtual monitor and a real-time image recognizer.

4. VISION SYSTEM

A combination of knowledge-base and model-base is applied to the 3D vision and scene understanding in the HARIS system. It might be well known that the model-based vision technologies have both strong points and weak points [Hasegawa92, Shirai84]. A typical strong point is that object recognition could be made from partial information of potential objects by means of the knowledge of the model. By contrast, this type of systems will not be able to recognize any other non-predefined objects. In our system the strong points of this approach should be quite useful and the limitations are negligible, since HARIS is supposed to be used for desk-top object manipulations in a controlled situation. This is a reason why we have chosen the model-based approach. The vision system includes a knowledge base, a 3D object understander, a distance measurer and a scene understander.

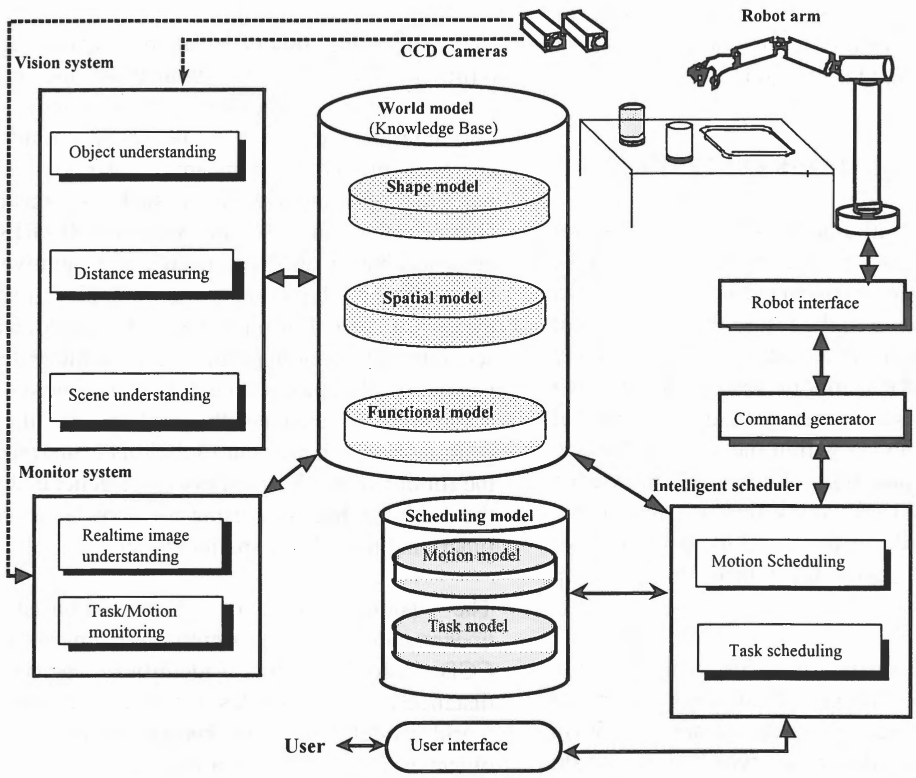


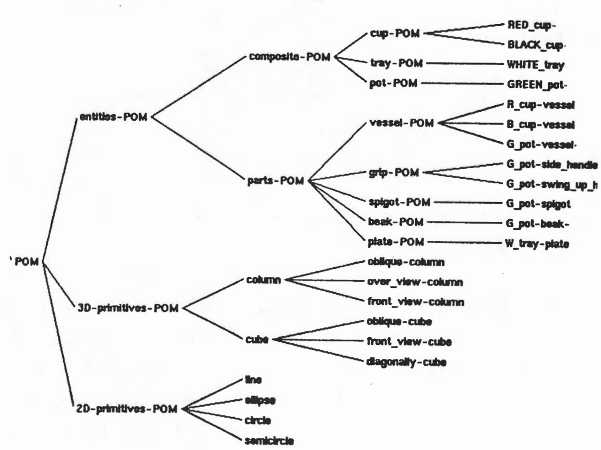
Fig.4 Overall image of knowledge-based autonomous robot system

4.1 Frame-Based Knowledge Descriptions

The knowledge on the objects and the scene created as the result of the 3D vision are represented in a frame-based knowledge representation framework in the ZERO formalism [Ito86].

As outlined before the world model consists of three kinds of sub-models to handle different attributes about the scene, which are the shape model, functional object model and spatial model. The shape model represents the attributes about shapes of objects for identifying specific object within the scene from partial information. The functional object model represents attributes about functions of objects, which is mainly used to intelligent task scheduling. The spatial model represents spatial relationships among the objects within the scene. This model is generated as the result of scene understanding and used for both intelligent task scheduling and intelligent motion scheduling. Every candidate object to be seen within the environment are pre-defined as a class frame in a frame hierarchy. As the result of scene understanding a set of instance frames is generated under associated class frames.

Figure 5 shows the ISA (A-KIND-OF) hierarchy to represent template knowledge about potential objects for describing the shape model(POM). As in the figure class frames are represented in a hierarchy so that descriptions be simpler. Each node in the hierarchy represents a frame. An example of contents of a frame for describing Red-cup is shown in Figure 5 as well.



Frame name: Red-cup

A-kind-of	Cup
Color	Red
Has-parts	(R_Cup_Vessel)
External_diameter	7
External_height	12

....

Fig. 5 A frame hierarchy to represent the shape model and an example of frame contents for Red-cup.

4.2 Method of Scene Understanding

It should be noted that there exists the basic difference between computer vision and robotic vision since this difference is very important in the discussion of our approach in scene understanding. In the computer vision precise generation of a scene from image data is the issue. Computation time is not considered. While, in the robotic vision simplification is more important than preciseness within limited time. Real time object recognition should be an ideal goal in a dynamic environment. We are trying to achieve faster scene understander in a limited situation by means of the model of objects and knowledge-based reasoning with partial information from the scene. This is characterizing our approach which is outlined as follows.

At first, color image data through two CCD cameras are preprocessed with the image processor to minimize noises. Then object regions are localized by means of color information such as red, black and white. By color-oriented localization object recognition time could be extremely decreased. Since specific color attribute is attached to each knowledge of every candidate object within the knowledge base at class frames, top-down based object understanding can be applied. For example, if the color attribute of region "p" is identified as "red", then "a red cup" is supposed to exist there. In this case, only a couple of parallel lines is recognized by means of straight line identifier, which is one of 2D-primitive identifiers. This can be simply decided using the knowledge stored in the shape model of the knowledge base, i.e., World Model. By means of color "Red" and "a parallel lines" the system easily understands this object as a red cup. (If two or more red colored objects would be possible within a scene additional attributes have to be used. In our situation of desk-top object manipulations for serving disabled persons color-based identification of objects are reasonable.)

Figure 6 shows an example of generated instance frame to describe a identified line. As the result of object understanding an instance frame is created under the class frame for the red cup. For every region within the scene similar process is applied to understand each object.

```
FRAME NAME: R_cup-vessel-POM          FRAME TYPE: INSTANCE
A-KIND-OF:          FRAME      vessel-POM
DDESCENDANTS:       NIL
DESCRIPTION:        STRING     *UNDEFINED*
CREATED-BY:         STRLIST    ("miyasaka" "Tue Nov 15 16:16:45 1
MODIFIED-BY:        STRLIST    ("someone" "Fri Feb  2 18:00:06 19
     SLOTNAME: Color
        FROM:                  entities-POM
        DATA TYPE:             ATOM
        VALUE:                 Red
        DEFAULT:               *UNDEFINED*
     SLOTNAME: PART_OF
        FROM:                  components-POM
        DATA TYPE:             FLIST
        VALUE:                 (RED_cup-POM)
        DEFAULT:               (RED_cup-POM)
     SLOTNAME: PROJECTED_ON
        FROM:                  components-POM
        DATA TYPE:             FLIST
        VALUE:                 (oblique-cylinder1 oblique-cylinder3)
        DEFAULT:               (cylinder)
     SLOTNAME: Diameter
        FROM:                  components-POM
        DATA TYPE:             INTEGER
        VALUE:                 *UNDEFINED*
        DEFAULT:               7
     SLOTNAME: Height
```

Fig. 6 An example of contents of a frame describing identified "line".

The trigonometric measurement method is applied next in order to both assure the objects and measure distances. Next, spatial descriptions are created in the spatial model using the results of distance measurement and the shape model. At last, the functional object model is generated by instanciating associated class frames. Automated calibration for distance measuring and color detection is done beforehand. Distance recognition errors of current system are between 5 and 10 millimeters under the condition where a distance between two cameras is 30 cm and between the cameras and the target area is about 200 cm.

In representing objects in the scene two coordinate systems are used, which are local coordinate systems and a world coordinate system as shown in Figure 3. Each coordinate system is expressed by X, Y and Z axes. A local coordinate system is used to represent the geometric relations of the components of each object within an object description. While, the world coordinate system is used to represent geometric relations of the objects in the whole scene. The origin of the local coordinate system for each object is the central and bottom point of it. The origin of the world coordinate system is the central and surface level of the robot arm, i.e., a surface level of the desk. Since the local coordinate values for each object are pre-measured and stored within the shape model as attributes of it the role of the world modeler, i.e., the scene understander, is to represent compete descriptions of the objects in the world coordinate system. The spatial model is represented by the combination of the semantic network representations and the geometric representation of the scene by means of the world coordinate descriptions of the objects within it.

According to the results of the 3D image understanding the functional model is generated by instanciating associated class frames for identified objects. This is very simple since the attributes of every candidate object within the environment is pre-defined within the slots of the associated frame. Figures 7 shows an simplified image of a part of the red cup frame created in the functional model. Major attributes are components, state, roles, acceptable tasks, acceptable holding types, default holding type, and so on.

Frame: Cup-1	
a-kind-of	Cup
has-parts	(container_1 handle_1)
state	empty
roles	(contain_drink)
color	red
accept_tasks	(reach hold lift carry put release leave)
accept_htypes	(hook grasp nip)
default_htype	hook

Fig.7 An example of a frame to describe "Red cup" in the functional object model.

Using the results of the distance measurement geometric positions of the objects and spatial relations among them are recognized, which are attached to associated descriptors of the associated frame of the spatial model. The spatial relations among the objects are represented by means of the semantic network model, and the geometric positions of the objects are described by the world coordinate system. In order to represent the spatial relations we have introduced nine description primitives which are 'left-of', 'right-of', 'in-front-of', 'behind-of', 'on', 'under', 'above', 'below' and 'contact-with'. Figure 8 a part of the spatial model to be generated as the results of the model-based 3D image understanding respectively.

```
Frame: Spatial-description
a-kind-of        Spatial-model
hasparts         (Red_cup_1   Black_cup_1 Tray_1)
Red_cup_1        ([150 110 0] ([100] [010] [001]))
Black_cup_1      ([250 310 0] ([100] [010] [001]))
Tray_1           ([-200 250 0] ([100] [010] [001]))

Frame: Objects-relations
a-kind-of        Spatial-model
Red_cup_1        ((left-of Tray_1) (left-of    Black_cup_1))
Black_cup_1      ((Right-of Tray_1) (Right-of Red_cup_1))
Tray_1           ((right-of Red_cup_1) (left-of Tray_1))
```

Fig.8 An example of spatial descriptions of a scene in Spatial model.

5. OUTLINE OF INTELLIGENT SCHEDULER

The role of the task and motion scheduler, which is another knowledge-based module of the robot system, is to generate a sequence of primitive tasks for achieving the goal and converts them to motion primitives. From the motion primitives robot commands are generated to control the HARIS arm to achieve the request. Prior to receiving the request the robot has already the image of the scene, i.e., the world model as mentioned above. In the design of the task scheduling system we have considered human's way of a task scheduling as a mental model.

The primitive tasks and motions are pre-defined as the templates and stored in the task knowledge base, i.e., task model. The attributes for a task template include such as an agent, an instrument, a manipulation object, a destination, prior tasks, next tasks, pre-conditions, post-conditions and acting constraints. These attributes are stored in associated slots of associated task frame in the task model. The knowledge on prior tasks and next tasks is used to generate a reasonable task sequence in scheduling. The attribute values of initial tasks and terminal tasks are "nil", which terminate the process of the task scheduling automatically. As the result of the task scheduling the sequence of associated tasks is instanciated from the templates. The pre-conditions and post-conditions give the initial and terminal conditions for the motion of the task, which are used in the robot control.

The primitive tasks consist of basic manipulations of the robot arm at verbal level such as 'reach' the arm for an object and 'grasp' an object. The current task knowledge base stores the task-related knowledge for seven primitive verbs including such as 'reach', 'hold' and 'lift' in class frames. While, six primitive motions are defined which include such as ‘hform (hand_form)’, ‘approach’ and ‘leave’. Some primitive motions are same to primitive tasks, since these tasks are one-to-one relations to associated motions. Table 1 shows the relations among tasks, motions and robotic commands (Currently the robotic commands are those for Mitsubishi's MoveMaster).

In order to simplify both task/motion scheduling and robot hand control, object holding types by the robotic hand are standardized to five typical patterns which are 'grasp (tsukamu)' with five fingers, 'grip (nigiru)' with five fingers, 'nip (hasamu)' with two or three fingers, 'pinch (tsumamu)' with two or three fingers and 'hook (tsurusu)' with single, two or four fingers. For examples, nip with three fingers means nipping an object with thumb, pointing and middle fingers, and nip with two fingers is with pointing and middle fingers as a human does. The knowledge on the holding types are maintained by class frames and instanciated in generating the task schedule as well. It should be noted since acceptable tasks and holding types are attached to associated functional object model HARIS is able to deny improper tasks and to choose suitable holding procedures. In case of an example ‘hook’ is chosen for holding a red cup. Based on the primitive tasks associated motion schedule to control the robot arm is generated. The motion schedule consists of assigning primitive motions to associated tasks and path planning for approach and move motions. The path is used to move the robot hand from the initial position to the destination successfully.

The basic strategy of task scheduling is a kind of term-based goal-oriented. That is, a target task is identified as a goal from a request sentence first. The target task is easily extracted from a user's request using the Case grammar-based method of natural sentence analysis. Then a sequence of primitive tasks is generated as a sequence of terms, each of which is directly connected to specific primitive task, in order to achieve the goal. In the current system we are dealing with simple request such as "Put a red cup onto a white tray". In this case the target task, i.e., goal, is "put".

Generation of a task sequence starts with the "put" task, i.e., goal task, which results in generation of the goal frame. For examples, the "put" task needs the "carry" task as the prior task, which needs the "lift" task as its prior task, and so on. This sequence should terminate with the "reach" task as the first task since the value of the prior task slot of the "reach" task frame is "nil". A sequence of the following tasks is generated in a similar way using the values stored in the next task slots. In cases where plural tasks are allowed for prior or next tasks the shortest sequence of tasks should be assigned for a proper task schedule.

```
Task  "put":
pre_condition:    ((freep white_tray[on
                        [qprism [0 0 0] +-210 +-150 +5]])
                    (abovep red_cup white_tray)
                    (hand_inp red_cup[nip
                        [globe [0 0 50] +-10 +-10 +-10]]
                      MMEX_hand[nip
                      [globe [0 0 0] +-100 +-100 +-100]])
                    (posturep MMEX_hand[nip[0 1 0]]
                       red_cup[nip[0 0 1]] 5)
                    (liftp MMEX_hand red_cup)
                    (inp red_cup[globe [0 0 0] +-35 +-35 0]]
                        white_tray[on
                        [qprism [0 0 50] +-210 +-150 +5]])
                      (posturep white_tray[#[0 0 1]]
                        red_cup[#0 0 1]] 5)
                      (posturep white_tray[#[0 0 1]]
                         world[#[0 0 1]] 5))
post_condition:     ((onp red_cup white_tray)
                      (hand_inp
                      red_cup[nip
                      [globe [0 0 50] +-10 +-10 +-10]]
                      MMEX_hand[nip
                      [globe [0 0 0] +-100 +-100 +-100]])
                      (posturep MMEX_hand[nip[0 1 0]]
                        red_cup[nip[0 0 1]] 5)
                      (holdp MMEX_hand red_cup)
                      (inp
                        red_cup[globe [0 0 0] +-35 +-35 0]]
                        white_tray[on
                        [qprism [0 0 50] +-210 +-150 +5]])
                        (posturep white_tray[#[0 0 1]]
                        red_cup[#0 0 1]] 5)
                        (posturep white_tray[#[0 0 1]]
                         world[#[0 0 1]] 5))
constraints:        ((no-slip MMEX_hand red_cup))
```

Fig. 9 Examples of pre/post-condition and acting constraints for the put task in the task model.

Pre- and post-conditions for each task are generated during the process by means of the information stored in the pre- and post-conditions slots of an associated task template, the information of the associated objects in the functional model, the information stored in the spatial model and the knowledge to instantiate task frames. The task scheduling procedure is outlined in [Ueno96b].

(1) At first, a request given in a natural sentence is analyzed by the Case grammar-based method and a transitive verb is identified as a target task. A class frame is generated to define the target task, i.e., goal task. By using the target task a sequence of primitive task frames are generated in a top-down way by means of the information attached to pre- and post-task slot as mentioned above.

(2) The pre- and post-conditions for the task are generated next. Since the templates for the pre- and post-conditions for the task are pre-defined at the task template frame in the task knowledge base in expressions, the generation of the conditions is to replace variables by values.

(3) Next, a sequence of primitive motions are generated from the sequence of tasks by means of attached motion attributes to associated task frames one-by-one in a real time mode (see Table 1). For each motion(i.e., action) a path plan is made by means of the knowledge attached to it. For the current experimental system the path planning is quite simple since the initial condition as well as the environment is assumed not to be changed during execution.

(4) The command sequence is generated next to control a manipulator. Since the HARIS hand is under tuning-up for tactile sensors and grasping controls, a Movemaster which is a commercially available robotic arm is used for feasibility study. Since the Movemaster has own control interface to connect to another computer and has functions to accept relatively higher level robotic commands such as "open hand", "close hand" and "contain object in hand", it is easier to control by simply sending commands in sequence. However, "nip" is only available in holding an object. Table 1 shows relationships between primitive tasks, primitive actions and Movemaster commands. Figure 9 shows an example of a part of the generated "Put" task frame.

Table 1 Relationships among primitive tasks, primitive actions and robotic commands.

Primitive task	Primitive action	Robotic commands	
reach	hform	GO	open hand
	approach	PD MO MT	assin destination reach hand for object contain object in hand
hold	hold	GC	close hand
lift	lift	DW	lift object
carry	move	PD MO	assin destination reach hand for object
put	move	PD MO	assin destination reach hand for object
release	hform	GO	open hand
leave	leave	PD MT OG	assin destination contain object in hand return hand

6. CONCLUDING REMARKS

1) Scene understanding by means of knowledge- and model-based 3D vision technologies has resulted in reasonable performance in an experimental situation for the HARIS robot on desk-top 3D objects manipulations. We have to solve such problems as occlusion and shade deletion to achieve higher flexibility.

2) Current system deals with very simple objects such as cups without a handle and boxy tray. Sophisticated objects like coffee bottles and spoons are next targets.
3) Vision-based monitor system is not working yet under dynamic situations in a real-time mode. We are planning to develop monitoring system by combining tactile sensors, computer-based virtual imaging and dynamic image processing technologies. Human-robot communication/collaboration would help reasonable monitoring by simpler technology.
4) An integration of knowledge-based AI technologies, model-based 3D vision technologies and mechatronic arm technologies has resulted in a possibility of highly flexible human-like autonomous service robot HARIS for assisting the aged and disabled in desk-top object handlings. This technology should also be applicable to next generation industrial robots in handling complex objects.
5) By introducing Cognitive Science the integration of scene understanding and task/motion scheduling has been smoothly achieved. This approach would be useful for a human-robot interface in realizing service robots for the handicapped[Saito94]. For example, we will be able to imagine a future situation, where autonomous service robots and a handicapped person would exist to collaborate each other. In such a situation a robot should serve him/her by inferring his/her intentions from body actions and/or simple requests in terms during execution. This is true for communications between human beings as well.
6) Usage of frame-based software platform ZERO++ in conjunction with the concept of World Model plays undoubtedly important roles in realizing integrated autonomy for the HARIS robotic system. Module-based software development and maintenance have been easily accomplished by this approach. This is very helpful in such a university-like development environment, where students are joining and leaving every year, while the system must be progressive continuously. We are planning to develop distributed system via LAN and Internet for further flexibility and performance.

Acknowledgment

The HARIS project is a collaborative work between us and Prof. Y. Saito of Department of Industrial Mechanical Engineering. This study is supported in part by the Research Promotion Foundation of Tokyo Denki University, and by Japan Ministry of Education for international collaborative research between our group and Intelligent Robotics Laboratory at Vanderbilt University (Prof. K. Kawamura).

References

[Bagchi96] S. Bagchi, G. Biswas and K. Kawamura : Interactive Task Planning Under Certainty and Goal Changes, Journal of Robotics and Automation Systems, Vol.18, pp.157-167, 1996.

[Barr92] Barr, A. and Feigenbaum, E., The Handbook of Artificial Intelligence, Addison-Wesley, 1977.

[Hasegawa92] Hasegawa, T. Model-based Vision for Robotic Manipulation, Journal of the Robotic Society of Japan, Vol.10, No.2, pp. 153-158, 1992.

[Ito86] Ito,H. and Ueno,H., "ZERO: Frame + Prolog", Logic Programming '85, LNCS221, pp.10-17, 1986

[Kawamura95] Kawamura, K., Baguchi, S., Iskarous, M. and Bishay, M., Intelligent Robotic Systems in Service of the Disabled, IEEE Transactions on Rehabilitation Engineering, Vol. 3, pp. 14-21, 1995.

[Kawamura95b] Kawamura, K. and Thomas, E. D.(eds.), Proceedings of International Workshop on Biorobotics: Human-Robot Symbiosis, Tsukuba, Japan, 1995.

[Ohnishi96] Ohnishi, K., Saito, Y., Ihashi, T. and Ueno, H., Human-type Autonomous Service Robot Arm: HARIS, Proc. 3rd France-Japan Congress on Mechatronics, pp.849-854, 1996.

[Ooshima90] Ooshima, T, Higashibara, T., Saito, Y. and Kubo, S., Holding Features of Multiple Tactile Sensors for Artificial Arm, Abstract of 11th Annual Conference of Japanese Society for Biomechanism, pp.5-6, 1990.

[Asaad96] S. Asaad, M. Bishay, D.M. Wilkes and K. Kawamura : A Low-Cost, DSP-Based Intelligent Vision system for Robotic Applications, Proc. Int. Conf. on Robotics and Automation, pp.105-111, 1996.

[Saito89] Saito, Y., Itoh, H., Kawashima, T., Maeda, A., Haneyoshi, T. and Fujii, S., A Development of Torque Equilibation Type Deburring Robot System, Proc. 20th IS on Industrial Robots, pp.901-908, 1989.

[Saito94] Saito, Y. et al, A New Approach in Robot Control Systems for Home Nursing, Journal of Robotics and Mechatronics, Vol. 4, No. 4, pp. 279-287, 1994.

[Shirai84] Shirai, Y., Model-based Vision, Journal of the Robotic Society of Japan, Vol.2, No.6, pp. 593-598, 1984.

[Ueno86] Ueno, H., Object Model for a Deep Knowledge System, Springer-Verlag, Lecture Note in Economics and Mathematical Systems 286, pp.11-19, 1986.

[Ueno94] Ueno, H. and Saito, Y., An Autonomous Robot Arm with Visual and Tactile Sensors, Proc. Mechatronics '94, pp.448-451, 1994.

[Ueno95] Ueno, H., Kogure, S. and Ookuma, Y., Implementation of Frame-based Knowledge Engineering Environment ZERO in C++, Lecture Note on Software 15, pp.101-110, Kindaikagaku-sha, 1995.

[Ueno96] Ueno, H. and Saito, Y., Model-based Vision and Intelligent Task Scheduling for Autonomous Human-Type Robot Arm, Special issue on Robotics and Autonomous Systems 18, pp.195-205, Elsevier Science Jopurnal, 1996.

[Ueno96b] Ueno, H., Intelligent Task and Motion Scheduling for Autonomous Human-Type Service Robot Arm, Artificial Intelligence: Methodology, Systems, Applications (Ed., A.M. Ramsay), Frontieers in Artificial Intelligence and Applications, Vol. 35, pp.239-248, IOS Press, 1996.

[Yamada96] Yamada, T., Intelligent Task Planning and Intelligent Action Control for Human-Type Intelligent Service Robot Arm HARIS, Master thesis, Tokyo Denki University, 1996.