Robotica (1998) volume 16, pp. 509–516. Printed in the United Kingdom © 1998 Cambridge University Press

Knowledge Driven Planning and Multimodal Control of a Telerobot

Zunaid Kazi and Richard Foulds

Applied Science and Engineering Laboratories, duPont Hospital for Children and the University of Delaware, P.O. Box 269, Wilmington DE 19899 (USA), http://www.asel.udel.edu/robotics/musiic/

SUMMARY

The Multimodal User Supervised Interface and Intelligent Control (MUSIIC) project focuses on a multimodal human-machine interface which addresses user need to manipulate familiar objects in an unstructured environment. The control of a robot by individuals with significant physical limitations presents a challenging problem of telemanipulation. This is addressed by a unique user-interface integrating the user's command (speech) and gestures (pointing) with autonomous planning techniques (knowledge-bases and 3-D vision). The resultant test-bed offers the opportunity to study telemanipulation by individuals with physical disabilities, and can be generalized to an effective technique for other, including remote and time-delayed, telemanipulation. This paper focuses on the knowledge-driven planning mechanism that is central to the MUSIIC system.

KEYWORDS: Knowledge-driven planning; Human-computer interface; Telerobot; Multimodal interface; Supervisory control.

1. BACKGROUND

Several studies have demonstrated the need for the development of a general purpose manipulation aid for use by people with disabilities to perform everyday activities[1,2]. Rehabilitation robotics research literature describes many demonstrations of the use of assistive robotic devices[3–5]. Prototype interfaces have generally taken two approaches; some are command oriented where the user activates the robot to perform pre-programmed tasks while others are control oriented where the user directly controls all the movements of the manipulator much like a prosthesis.

While direct control allows the user to operate in an unstructured environment, problems such as physical and cognitive load on the user, the requirement of good motor dexterity of the user and many other real-time perceptual and motor requirements are hindrances towards the development of an efficient and useful assistive robot.

Command based systems also pose significant problems. While modern speech recognizers provide access to large numbers of stored commands, effective command of a robot will require use of more commands than is reasonable for the user to remember.

At the other extreme of robot control are autonomous systems, the long elusive goal of the Artificial Intelligence *(AI)*, robotics and machine vision communities. Although many important incremental advances have been made in the past decades, this goal seems far from being realizable at this point. Furthermore, absolute automation poses a set of problems such as incomplete *a priori* knowledge about the environment, hazards, insufficient sensory information, and inherent inaccuracy of the robotic devices themselves.

Another critical aspect of delineating the degree of control is philosophical. Will a human user be comfortable with abrogating all responsibilities to the machine? With increased automation, there is a resultant alienation of the human supervisor not only spatially, but also cognitively. This alienation process naturally may lead to frustration and resentment. While automation may assist in relieving the more mundane and repetitive chores, it may not necessarily be welcomed by human users who wish to remain in control. Therefore, while absolute automation might be realizable in the future, a system where the user exerts a degree of control might be preferred over autonomous systems.

This discussion necessarily leads to considering a supervisory control paradigm, which is defined by Ferrel and Sheridan[6,7] as a “control system where one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial end effectors and sensors to the controlled process or task environment.” This allows the system to blend the flexible decision making and error-handling abilities of the human with the pre-programmed autonomous sub-goal satisfying ability of the machine.

Sheridan[6] lists a set of motivations for development of such supervisory control mechanisms:

- to achieve the accuracy and reliability of the machine without sacrificing the cognitive capability and adaptability of the human,
- to make control faster and unconstrained by the limited pace of the continuous human sensorimotor capability,
- to make control easier by letting the operator give instructions in terms of objects to be moved and goals to be met, rather than instruments to be used and control signals to be sent,
- to minimize the demand for continuous human attention and reduce the operator's work load,
- to make control possible even when there are time delays in communication between human and teleoperator,
- to provide a “fail-stop” capability when failure of the operator's direct control would prove catastrophic,

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

- to save lives and reduce cost by eliminating the need for the operator to be present in a hazardous environment and the need for life support required to send the operator there.

One more motivation can be added to Sheridan's list:

- to assist a disabled user in manipulating her environment through some mode of interface with the device.

2. RESEARCH OBJECTIVE AND THE MUSIIC SOLUTION

Continuing in the spirit of supervisory control, and constrained by the limitations of current state of the art in *AI*, vision and natural language processing, we need a synergistic integration of the best abilities of both humans and machines to achieve a real-world telerobotic control mechanism. The goal of this paper is to demonstrate that by combining current state of the art in natural language processing, robotics, computer vision, planning, machine learning, and human-computer interaction, it is possible to build a practical robotic assistant for people with disabilities without having to solve the major problems in each of these fields, i.e. full text understanding, autonomous robot-arm control, real-time object recognition in an unconstrained environment, planning for all contingencies and levels of problem difficulty, speedy supervised and unsupervised learning, and intelligent human-computer interfaces. This is possible because current solutions to each of these problems, when combined with each other and with the intelligence of the user, can compensate for the inadequacies that each solution has individually.

We claim that the symbiosis of the high level cognitive abilities of the human, such as object recognition, high level planning, and event driven reactivity, with the native skills of a robot can result in a human-robot system that will function better than both traditional robotic assistive systems and current autonomous systems. We describe a system that can exploit the low-level machine perceptual and motor skills and excellent *AI* planning tools that are currently achievable, while allowing the user to concentrate on handling the problems that they are best suited for, namely high-level problem solving, object recognition, error handling and error recovery. By doing so, the cognitive load on the user is decreased, the system becomes more flexible and less fatiguing, and is ultimately a more effective assistant.

The Multimodal User Supervised Interface and Intelligent Control (MUSIIC) project uses a novel multimodal (speech and gesture) human-machine interface built on top of a reactive and intelligent knowledge-driven planner that allows people with disabilities to perform a variety of novel manipulatory tasks on everyday objects in an unstructured environment. MUSIIC has three main subsystems that required extensive software development; these subsystems deal with planning, vision and the human-computer interface. Here is an overview of these subsystems.

2.1 Knowledge Driven Planning Subsystem

The planning subsystem is built upon an object-oriented knowledge base of objects where objects are represented in an increasingly specialized sequence of object classes in an inheritance hierarchy. There is also a user extendible library of plans ranging from primitive robot handling tasks to more complex actions and purposeful actions. Our multimodal human-machine interface and object-oriented representation allows the user to interact with the planning system at any level of the planning hierarchy, from low-level motion and grasp planning to high-level planning of complex tasks. Synthesized plans are supplemented by user intervention whenever incomplete information or uncertain situations prevent the development of correct plans. The user does this by taking over control of the planning process or by providing necessary information to the knowledge bases to facilitate the development of a plan capable of handling a new or uncertain situation. Furthermore, incomplete sensory information may be supplemented by user input, enabling the planner to develop plans from its plan library without the need for extensive user intervention during plan execution.

2.2 Vision Subsystem

The vision subsystem[8] allows MUSIIC to determine the three-dimensional shape, pose and location of objects in the domain. no object recognition is performed. The vision requirement is to provide the knowledge-based planning system with the parameterized shape, pose, orientation and location information of the objects in the immediate environment. This information can then be used to fill slots in the object-oriented representation and to support planning activities. The vision system plays a role in the user-interface as well. A pointing gesture, using a laser light pointer allows a user of MUSIIC to indicate her focus of interest. The vision subsystem identifies the spot of laser light and processes information relating to the object of choice.

2.3 Human-Computer Interface Subsystem

The HCI subsystem employs a multimodal input schema where users of our system point to indicate locations and speak commands to identify objects and specific actions. The combination of spoken language with pointing performs a critical disambiguation function. It binds the spoken words in terms of nouns and actions to a locus in the physical work space. The spoken input supplants the need for a general purpose object-recognition module in the system. Instead, 3-D shape information is augmented by the user's spoken word, which may also invoke the appropriate inheritance of object properties using MUSIIC's hierarchical object-oriented representation scheme. More detail is available in reference 9.

With these subsystems, MUSIIC first determines what the user wants and then makes plans to accomplish the task. As a consequence of insufficient information, uncertainty, advent of new information, or failure of a plan, the system either tries autonomously to replan or engages in a dialogue

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

with the user that enables the planner to revise its plans and actions.

3. KNOWLEDGE DRIVEN PLANNING AND CONTROL

In this section we focus on the central underpinning of the MUSIIC system, namely the knowledge-driven planning system that is the heart of MUSIIC.

3.1 Background

Faced with the overwhelming complexities of planning, a practical planning system must balance between being as expressive as practical and making sufficiently restrictive assumptions, so that a viable, efficient implementation can be realized. In our research, we have tried to strike the same balance. We believe that the integration of an object-oriented planning mechanism with a knowledge-based planner allows us to reduce planning complexity, while still providing us with sufficient expressiveness that this planning system can perform under the relaxed assumptions that need to be made about the real world. Classical planning systems strove to maintain this balance, as well as provide four of the capabilities identified by Wilkins[10]. These capabilities include:

- **Non linear planning**–A plan is non-linear if it contains actions that are unordered with respect to each other. This allows the planner to represent and reason about plans without committing to a course of action until more information about plan ordering has been gathered.
- **Hierarchical planning**–In a complex domain it is crucial that the planner is able to plan at different levels of abstractions. If a planner starts planning at a detailed level it would be overwhelmed by choices available. Therefore, a planning system must be able to support planning at different levels of abstraction.
- **Domain independence**–Planners must be domain independent so that their techniques are readily available for new problems.
- **Replanning ability**–The planner must also be capable of restarting after an unexpected occurrence.

Sacerdoti's NOAH[11], and Fikes and Nilsson's STRIPS[12] mark the beginning of this approach. However, classical *AI* planning systems have worked under a set of restrictive assumptions. All objects and their properties are assumed to be known *a priori*. Spatial and temporal dimensions are discretized. All relevant facts about the problem to be solved are known in advance. The system must solve the problem without any external help, no matter how complicated the problem. Once the plan is constructed, the system is done; some other system will carry out the plan.

However, in order to make *AI* planning systems practical in the real world, some of these assumptions must be relaxed. Especially in the domain of robotics, we have to make different assumptions about the world:

- Not all objects that will be encountered will be known, nor will all their properties be known in advance
- Spatial and temporal dimensions in the real world are continuous.
- In some applications, e.g. the rehabilitation robot application, the system can get external help from the user, and efficient handling of common problems may be more useful than the ability to solve more theoretically challenging problems unaided.
- Only some of the relevant facts about the problem to be solved will be known in advance, which raises issues of plan execution, error detection and recovery.

Since the mid 1980s, the realization was made that many of the assumptions underlying the classical framework could not hold in real-world applications. This has led to new directions for planning systems that have attempted to work their way out of the limitations of classical planning assumption. Research has been focused on hierarchical planning, as in SIPE[10], NONLIN[13], and DEVISER[14], and learning and reuse, such as [15–17].

3.2 The Planner's Knowledge Structures

The MUSIIC planning architecture is based on an *Object-Oriented* paradigm. The planner uses two object-oriented knowledge bases:

- A knowledge base of objects: WorldBase
- A knowledge base of plans: PlanBase

In addition, there is a world database, DomainBase, where the work-space information is stored, and a planner that uses that object knowledge base, world database, and action knowledge base to synthesize robot plans.

3.2.1 WorldBase (World Knowledge Base). The WorldBase is a knowledge base of objects organized in a four-tiered abstraction hierarchy that is used by the planner for plan synthesis. We first describe the knowledge representation of objects in the WorldBase and then describe the structure of the abstraction hierarchy.

3.2.2 Object Representation. Each object, depending on the degree of generalization, has a set of attributes that assists the planner in developing correct plans

3.2.3 The Object Inheritance Hierarchy. Objects are represented in an increasingly specialized sequence of object classes in an inheritance hierarchy. We have developed a four tiered hierarchy, where object classes become increasingly specialized from the top level class to the bottom level class as shown in Figure 1. The four classes are described below.

- **Generic Level**–This is the top level tier in the hierarchy of objects. This class is invoked during the planning process when nothing is known a priori about the object except the information obtained from the vision system, which includes object location, orientation, height, thickness, width and color.
- **ShapeBased Level**–This class of objects is the second tier in the class hierarchy. Since the domain of this research is robotic manipulation, the primary low level robot action

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

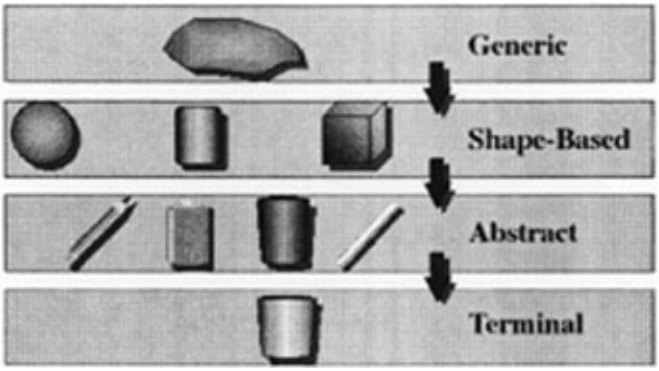


Fig. 1. Knowledge Structure

of grasping is directly affected by the shape of the object. The approach point and the grasp point of objects can be specified more accurately if some idea of the shape is available. Currently the following shape-based classes are available at this level: *Cylindrical, Cuboid, Spherical*, and *Generic.*

- **Abstract-Object Level**–The third tier constitutes general representation of commonly used everyday objects, such as a cup, can or box. These objects are derived from the Shape-Based classes. Approach points and grasp points are further specialized at this level. Constraints are added at this level to restrict the mode of manipulation. For example, the default orientation of a cup is upright, and when the robot arm picks up the cup, this orientation must be maintained. Furthermore, the constraint on the cup object is considered to be permanent; i.e., the robot arm will maintain the default orientation in all cases of manipulation. In case of a straw that may be lying horizontal on the work-space, the default orientation is vertical. When the robot arm picks up the straw, it will then rotate the straw to a vertical position. However, this default constraint on the orientation of the straw is not permanent. This implies that the arm may orient the object to any orientation during the course of manipulation but will then bring the object to its default orientation at the end of the plan execution. In the WorldBase, the actual height, width or thickness of an object may not be specified. These attributes are filled from information obtained by the vision system if manipulation is being done at this level.
- **Terminal Level**–This is the fourth and final tier of the abstraction hierarchy. These represent actual instantiations of objects that are in the user's domain. These objects are derived from the Abstract-Object classes, and the object attributes are fully instantiated. Information from the vision system as well as user input may be used to instantiate these objects. All attributes are fully specified, and this allows a more accurate execution of user intentions by the robot arm. This tier refers to specific objects which can be measured uniquely such as the user's own cup, a particular book, or the user’s favorite pen.

3.2.4 DomainBase (Specific Domain Knowledge). In addition to the knowledge base of objects, the system also maintains a knowledge base of the objects that it actually sees in the domain called the DomainBase. The attributes of the objects in this knowledge base, such as color, height, width and thickness, are instantiated from information obtained either from the vision system or from the user. Essentially, this knowledge base is a subset of the WorldBase with attributes augmented from the vision system, and user input.

3.2.5 The PlanBase. The plan knowledge base, PlanBase, is a collection of STRIPS-like plans[11,18], and the planner is based on a modified STRIPS-like planning mechanism. The main difference between conventional STRIPS-like planning and our system is that we take full advantage of the underlying object-oriented representation of the domain objects, which drives the planning mechanism. Plans in this model are considered to be general templates of actions, where plan parameters are instantiated from both the WorldBase and the DomainBase during the planning process. For example, the constraints slot for a Move action might contain the slot *Object-constraints*. This implies that this slot parameter is going to be filled up from the constraints field of the object on which the action is being invoked. In the case of the cup example previously illustrated, the constraint that the cup must be maintained in a certain orientation is used to instantiate the *constraint* slot of the *Move* action. The constraints instantiated from the object in question are added to the set of constraints already present. Sometimes some of the constraints obtained from the objects themselves may be in direct contradiction to constraints already present in the action being invoked. When that happens, the constraints obtained from the object override default constraints in the action body.

Another way in which the object-oriented paradigm has extended the classical STRIPS planning mechanism is described here. As has been mentioned previously, the body of an action may contain further subactions into which the actions may be decomposed. This facilitates hierarchical planning, one of the essential features of a planning system.

However, certain tasks that can be handled generally for most objects may not be applicable to certain objects in the real world. Suppose we have an appliance that is used often in the domain of the user. The instrument has a peculiar shape and must be picked up from a specific point. To approach the grasp point, it may not be possible to just simply specify a certain approach point and assume that the robotic arm will then be able to pick up that appliance. The approach path may be convoluted, and there must be some way to specify such an atypical case in our planning system. This is done by the use of the plan-fragment associated with an object. In a manner similar to the way *action* slots are filled, depending on the object on which the actions are invoked, *subaction* slots are also filled, if so specified, from the object's *plan-fragments* slot.

Thus, we see that this integration of knowledge-driven planning with an object-oriented approach allows us to use general plans whenever we can. Additionally, this method

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

allows us to develop plans for specific objects that are peculiar to the domain without the need to perform computationally expensive operations. The object hierarchy allows us to abstract out the general features of an action and invoke them on objects about which the knowledge bases might not have any information. It also allows us to view an action as a single template that is applicable to many kinds of objects instead of as a set of actions, each applicable to only one specific object.

3.2.6 Plan Representation. Plans in the plan library have the following general format:

- **Name**: This slot defines the name of the plan as well as the parameters with which this plan is invoked. Given the manipulative domain of this planner, these parameters are either an object name or a location name or both. These parameters are global variables, which are initialized by the planner at the beginning of the plan synthesis process, and the initialized values are used by the plan and any subactions of the plan.
- **Plan Type**: This slot identifies the type of the plan. As will be explained later, plans can be of three types; primitive, complex, and user defined.
- **Preconditions**: The preconditions represent what must be true before the plan can be invoked. During the interpretation of user instructions, the planner makes no attempt to make a precondition true. This restriction is lifted during the replanning and plan adaptation process, which shall be elaborated upon in Section 5.6.
- **Plan Body**: The plan body contains the subactions that are executed to make true the goals of the plan. This structure allows the MUSIIC planner to apply hierarchical planning.
- **Goals**: The goals specify what this action is intended to achieve.

In STRIPS, the basic assumption was that no predicate will change its truth value when something occurs, unless the operator explicitly states the change of truth values. In STRIPS, this is done by explicitly listing predicates in its add and delete lists. The predicates on these lists were then added to or deleted from the world model. Unlike STRIPS, our approach does not use add and delete lists. Domain knowledge is contained within the DomainBase. As was described earlier, objects have location and orientation attributes. Each primitive updates the state of the world by changing and updating the location and orientation parameters, depending upon what each action is supposed to be doing. For example, an action that will move an object to a certain location will update the location attribute of the object being moved after successful completion of the task.

The plans in the PlanBase are of three types: user defined, primitive, and complex. User-defined plans are those that the user teaches the robot through both off-line and on-line instruction. Primitive and complex plans are system encoded plans that are defined a priori. Primitive plans are those that are executed directly as a robot operation (i.e., their plan body contains only one subaction). Complex plans are plans whose plan body contains a list of plans, both primitive and complex.

3.2.7 Planning and Execution. Our hierarchical human-machine interface and object oriented representation allows the user to interact with the planning system at any level of the planning hierarchy, from low-level motion and grasp planning to high-level task planning of complex tasks such as feeding. The generic plans and specialized plans are supplemented by user interaction whenever incomplete information precludes the development of correct plans by taking over control of the planning mechanism or providing information to the knowledge bases to facilitate the development of a plan capable of handling a new or uncertain situation. Furthermore, incomplete sensory information may be supplemented by user input, enabling the planner to develop plans from its plan library without the need for extensive user intervention.

4.0 MUSIIC HARDWARE ARCHITECTURE OVERVIEW

We now describe the test-bed system that has been set up to validate this research. The major components are distributed over a number of machines to facilitate programming and development of the individual components. The physical entities comprising the set-up include the vision subsystem (RoboEye), the robot and robot controller sub-system (RoboArm), the speech recognition subsystem (RoboEar), and the planning and knowledge-base sub-system (Robo-Mind). This is illustrated in Figure 2 and Figure 3. Details of the implementation will be provided in successive chapters.

4.1 RoboArm: The Robot Controller Subsystem

The manipulator device used for the test-bed is the Zebra ZERO robot manufactured by Integrated Motions, Incorporated (IMI). The ZERO is a six-degree-of-freedom robot arm and is supplied with force control mode, which allows it to exert a specified force on the environment as well as sense forces exerted on various joints. The software provided by the manufacturer allows control of manipulator movements through joint control.

4.2 RoboEar: Speech Recognition

The speech recognition system used in the test-bed is DragonDictate 1.0, which is a large-vocabulary, isolated-



Fig. 2. Set–up A

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

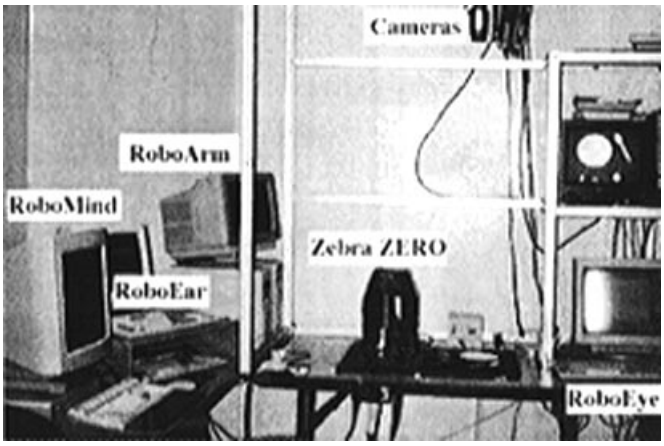


Fig. 3. Set–up B

word dictation package developed by Dragon Systems. The package has been installed in a 486 machine running Windows for Workgroups 3.11. The SM10A microphone manufactured by SHURE is used with a stereo M-Audio Capture and Playback Adapter (M-ACPA) card, which receives input from the microphone, digitizes the signal, and stores it in the computer.

4.3 RoboEye: The Vision Hardware

The backbone computing machine for the vision interface is an SGI XS-24 IRIS Indigo computer. Images are captured by two CCD color cameras. Each camera is equipped with a motorized zoom lens. Camera outputs are digitized using an SGI Galileo graphics board. The Noesis Visilog-4 software package installed on the SGI machine is used as an image processing engine.

4.4 RoboMind: The Planning Subsystem

The planner and the knowledge bases reside in a Sparc 5 workstation. The code has been developed in C++ for portability. The long term goal is for all the sub-systems to be combined and ported into one "black-box" for mounting on a wheel chair for use as an assistive robot.

4.5 RoboSpeech: System Output

The system responds to user instructions in a number of ways. It generates a number of on-screen diagnostic messages during processing. It also uses a text-to-speech package developed at the Applied Science and Engineering Laboratories to provide audio output

4.6 RoboNervous System: RPC Communication

Given that the subsystems reside in different host machines, we devised an RPC (Remote Procedure Calls) communication mechanism based on a client-server model that allows the different systems to communicate with each other. Each of the RoboEar, RoboArm, RoboMind, and RoboEye subsystems run a client procedure. A single server runs on the SparcStation that plays host to the planner. Inter-process communication is achieved by simple message passing of strings between the client procedures through the server. Messages are sent by a client through a "send" call and are buffered by the server until the recipient invokes a "receive" call.

5.0 A SIMPLE ILLUSTRATION OF CAPABILITIES

The operation of MUSIIC is illustrated through two annotated task scenarios. These scenarios involves the task of inserting a straw into a cup and bringing the cup to the user. The workspace contains a cup and a straw and the World Base contains entries for "straws" and "cups".

5.1 Scenario One

Instruct the system to load in the plan library.

User: "Load plans"
MUSIIC: "Plan loading complete"

Instruct the system to synchronize the various system components.

User: "Synchronize"
MUSIIC: "Synchronization complete"
User: "Home"

The robot then moves to its home configuration.

MUSIIC: "Home successful"
User: "Scan"

The vision system generates object size, position, orientation and color information.

MUSIIC: "Scanning complete"

Instruct the vision system to transfer the information to the planning subsystem to build up the DomainBase.

User: "Load domain"
MUSIIC: "Domain loading complete"

The user points to the straw while simultaneously saying the word "straw".

User: "That's a straw"
MUSIIC: "Looking for the straw"

MUSIIC searches the WorldBase for the "straw".
MUSIIC: "I found the straw"

User points to the cup and identifies it to the system.

User: "That's a cup"
MUSIIC: "Looking for the cup"
MUSIIC: "I found the cup"

Instruct the robot to insert the straw into the cup.

User: "Insert the straw into the cup"
MUSIIC inserts the straw into the cup. On success:
MUSIIC: "I am ready"

Instruct the robot to bring the cup to the user.

User: "Bring the cup"

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

The arm approaches the cup and grasps it by the rim. It then brings the cup to a predetermined position that is accessible to the user.

5.2 Scenario Two

User: "Bring that"

Here we present an alternate scenario where the user did not explicitly identify the cup. The user points to the cup while simultaneously saying the verbal deictic "that". RoboEye continuously records any identified spot along with time-stamp values that mark the time when the spot was recorded. The speech system also time-stamps utterances and these values are used to determine the location of the spot that was generated when a verbal deictic such as "that" is spoken. The system then finds the object that is in that specific location and invokes the "bring" task. Since the object has not been identified specifically, planning is based on general principles. Instead of grasping it by the rim, the arm grasps the cup along its width and brings it to the user.

6.0 FUTURE WORK AND FINAL WORDS

The development of the MUSIIC test-bed is now complete. Work is now progressing on integrating the different components into one machine for further actual user testing. A panel of users with varying degrees of manipulation disabilities has been selected. We are currently drawing up a set of scenario tasks that users will perform with the help of the MUSIIC system. A qualitative survey is being developed to then gauge user satisfaction with a system such as MUSIIC. The survey is going to attempt to determine whether the multi-modal input schema enhanced by artificial intelligence can be efficiently usable by the test group. We will then undertake an iterative process where user input will be used to further fine-tune the MUSIIC interface mechanism. In parallel with the determination of user satisfaction, we also are focusing on further technical enhancements to the MUSIIC system, including a more natural language like interface, unsupervised learning of new tasks, as well as an enhanced vision processing mechanism that is more robust.

Much work has been done in trajectory planning and plan synthesis in the robotics and *AI* planning communities, but practical, rapid, and highly autonomous systems are still far from being realized. Limitations of processor speeds and the absence of accurate and real-time object recognition algorithms are other obstacles in the path of real-world telemanipulation. Furthermore, developments in natural language understanding still lag behind the need for a natural user interface to a robot. The solution that this research proposes has not been in trying to solve the AI problem, or the vision problem, or the natural language understanding problem. Instead, the key contribution of this research has been the development of a telerobotic control architecture that is able to work around these limitations.

Our philosophy is based on the symbiosis of human and machine. This symbiosis is achieved through a novel framework for integrating a multimodal human-computer interface with a knowledge-driven planner. MUSIIC brings about a synergistic integration of the best abilities of human and machine, resulting in a telemanipulation mechanism that is realizable in an unstructured domain. The MUSIIC system is capable of exploiting both the low level perceptual and motor skills of the machine and excellent AI planning tools that are currently available, while allowing the user to concentrate on handling the problems for which she is best suited: high-level problem solving, object recognition, error handling, and error recovery. We claim that the symbiosis of the high-level cognitive abilities of the human, such as object recognition, high-level planning, and event-driven reactivity, with the native skills of a robot, results in a human-robot system that will perform in an unstructured domain.

ACKNOWLEDGMENTS

Work on this project is supported by the Rehabilitation Engineering Research Center on Rehabilitation Robotics, National Institute on Disabilities and Rehabilitation Research Grant H133E30013 (Department of Education).

References

1 C.A Stanger, C. Anglin, W.S. Harwin, D.P. Romilly. "Devices for assisting manipulation: A summary of user task priorities" *IEEE Transactions on Rehabilitation Engineering*; **2**(4);256–265 (1994).

2 A. Batavia and G. Hammer. "Consumer criteria for evaluating assistive devices: Implications for technology transfer" *12th Annual Conferrence on Rehabilitation Technology.* Washington DC: RESNA Press (1989) pp. 194–195.

3 R.A. Foulds, (ed.) *Interactive Robotics Aids–One Option for Independent Living: An International Perspective*. Monograph 37 (World Rehabilitation Fund, 1986)

4 D.C. Bacon, T. Rahman and W.S. Harwin, (eds.) *Fourth International Conference on Rehabilitation Robotics*: (A.I. duPont Institute, Wilmington, Delaware, USA, 1994. Applied Science and Engineering Laboratories).

5 M.V. der Loos, J. Hammel, D. Lees, D Chang and D. Schwant. "Design of a vocational assistant robot workstation". *Annual report*. (Palo Alto VA Medical Center, Palo Alto, CA: Rehabilitation Research and Development Center, 1990).

6 T.B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. (The MIT Press, Cambridge, Mass., 1992).

7 W.R. Ferrel and T.B. Sheridan. "Supervisory control of remote manipulation". *IEEE Spectrum*. **4**(10) 81–88 (Oct, 1967).

8 S. Chen. "World information through color images in task planning for a rehabilitation robot". *Technical Report ROB9605*. (A.I. duPont Institute and the University of Delaware, Wilmington, DE 19899: Applied Science and Engineering Laboratories, 1996).

9 Z. Kazi. "Integrating Human–Computer Interaction with Planning for a Telerobotic System". *PhD thesis* (Department of Computer and Information Sciences, University of Delaware, 1997); Chen, S. "World information through color images in task planning for a rehabilitation robot". *Technical Report ROB9605*. (A.I. duPont Institute and the University of Delaware, Wilmington, DE 19899: Applied Science and Engineering Laboratories, 1996).

10 D.E. Wilkins *Practical Planning: Extending the Classical AI Planning Paradigm.* (Morgan–Kauffman, San Mateo, California, 1988).

11 E.D. Sacerdoti. *A Structure for Plans and Behaviour.* (American Elsevier, New York, 1977).

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press

12 R.E. Fikes and N.J. Nilsson. “STRIPS: A new approach to the application of theorem proving to problemsolving”. *Artificial Intelligence* **2** 189–208 (1971).
13 A. Tate. “Generating project networks. In: (J. Allen and J. Hendler A. Tate, eds Readings in Planning. (Morgan Kaufmann Publishers, San Mateo, California, 1990) pp. 888–893.
14 S. Vere. “Planning in time: Windows and durations for activities and goals”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 250–271 (1981).
15 I. Mina. “KMPR: An experimental knowledge–based modeling prototype for robots”. In: *1987 IEEE International Conference on Robotics and Automation*. Raleigh, North Caroline: IEEE Press, Piscataway, New Jersey, (1987), pp. 2011–2020.
16 K.J. Hammond. “Explaining and repairing plans that fail”. *Artificial Intelligence.* **45**(1–2), 173–228 (Sept., 1990)..
17 J.L. Kolodner. “Improving human decision making through case–based decision aiding”. *AI Magazine*, **12**(2) 52–68 (Summer, 1991).
18 E.D. Sacerdoti. “The non–linear nature of plans”. In: *International Joint Conference on Artificial Intelligence*. Tbilisi, USSR, (1975) pp. 1123–1145.

https://doi.org/10.1017/S0263574798000666 Published online by Cambridge University Press