Integrating Human-Computer Interaction with Reactive Planning for a Telerobotic System

Zunaid Kazi

May 18, 1995

1 Introduction

1.1 Basis

Telerobots are robotic devices that extend a person's sensing and manipulation capability to a location remote from the person. In a dynamic and unstructured environment, tasks that need to be performed are sufficiently nonrepetative and unpredictable such that human intervention is necessary. This disallows the use of autonomous preprogrammed systems. Consequently, human perception, planning and control are still required. This research proposes a novel approach to an intelligent telerobotic system for such an environment: **speech-gesture control complemented by a knowledge-base driven reactive planning mechanism.**

In a teleoperated master-slave manipulator system, the operation of the telerobot is under complete control of the human supervisor. Work sites that are hazardous or are remote from the user, such as outer space, nuclear, biologically or chemically toxic environments, mines, construction sites etc., lead us to consider telemanipulation as a mode of controlling a robotic device. Also, such a telemanipulative robot can also be an assistive device used by a physically disabled user manipulating her environment which she may have been unable to do independently. However, such direct human control poses a set of problems arising from operation in an unstructured environment:

- incomplete a priori knowledge about the domain
- mode of operation
- hazards
- strategies of exploration of the environment they operate upon
- insufficient sensory information
- inherent inaccuracy in controlling the remote device with sufficient precision

Absolute control has the disadvantage of necessitating extensive physical load on the user while on the other hand, total autonomy suffers from the fact that current AI research is still lagging behind in developing a system capable of operating autonomously in real-world environments. Therefore, for such a robotic system to be practical in the real-world, the complementary capabilities of a human and robotic system must be integrated for improved system performance. Therefore, what is required is an intelligent telerobotic system where the user is an integral part of the system but the human sensing, intelligence and control is augmented by artificial sensing, intelligence and control.

Sheridan [She92] terms such control "supervisory control" and lists a set of motivation for development of such 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 workload
- 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
- to save lives and and reduce cost by eliminating the need for the operator to be present in hazardous environment, and for life support needed to send the operator there

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

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

In essence an intelligent telerobotic system in an unstructured domain must have behavioral adaptability and flexibility, cognitive, perceptual and motor skills to work cooperatively with humans. Schenker [Sch92] identifies the following functions and resources necessary:

- Robust perception: The system must have a rich sensory capability to make as accurate a model as possible of the work space.

- Autonomous task planning: Robot planning and reasoning must be sufficiently advanced to reduce the cognitive workload of the human operator and provide necessary checks and diagnostic procedures. In remote time-delayed work sites, autonomous planning is necessary to perform a major part of the tasks.

- Reactive/adaptive control: The robot must be sufficiently flexible to adapt and react to inherently uncertain and physically unpredictable situations [NSF88].

- Flexible human machine interface: The human supervising operator must be able to interact in an efficient and flexible manner.

In addition to the above features, a telerobotic planning system must relax some of the assumptions under which classical AI planning systems were constrained to work. The first attempts at planning in AI, such as STRIPS and NOAH developed at SRI International during the late sixties, made some fundamental assumptions which dominated most planning research up to the mid-eighties, and also have been the basis of a vast majority of such research. The fundamental assumption of a classical AI planning paradigm is the state-based representation of the world. As such the world is represented by taking "snap-shots" at a particular instance of time and the world is described in terms of the situation that existed at the time the "snap-shot" is taken. The world representation is usually formalized in terms of a formal language that described what was true in the "snap-shot."

Classiacal 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:

1. Not all objects that will be encountered will be known, nor will all their properties be known in advance
2. Spatial and temporal dimensions in the real world are continuous
3. 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.
4. 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.

We must again emphasize one significant assumption of classical planning systems: *the world does not change while the planning is being carried out.* That is, the planner constructs a plan for later execution without taking into consideration any changes that might have taken place in the world during the planning process. Hence, one has to distinguish between planning time and execution time. A system that can react in an *acceptable* amount of time to any changes that occur in the world while the system is planning is called a reactive planning system.

1.2 The Proposal

Given the discussion in the previous section, the realm of telerobots to perform satisfactorily in the real world, we need a system that has three essential components:

1. A sensory mechanism, including a vision system, with which the system should be able to sense the environment it is operating in.

2. A Human-Machine interface by means of which the system will interact with a human, whose cognitive abilities will be used to manipulate the environment.

3. A reactive planning mechanism that would allow planning in an unstructured real world environment fusing sensory data as well as human intervention.

This research proposes the design of an intuitive and efficient interface between the user and the manipulator as well as a reactive planning mechanism that conforms to the enumeration above. We propose a new hybrid interface strategy which combines command and control approaches to provide for user direction of the robot through the use of multiple modes of interface in conjunction with sophisticated capabilities of the machine. Users of this system use gestures (pointing) to indicate locations, and spoken commands to identify objects and actions. The use of multiple modes of control and command allows the user to operate the robot in a manner which more closely matches the user's needs.

This multi-modal approach is based on the assumption that the user's world is unstructured, but that objects within that world are reasonably predictable. Our work reflects this arrangement by providing a means of determining the three-dimensional contours of objects and surfaces which are in the immediate environment (regardless of the actual objects), and an object-oriented knowledge base and planning system which superimposes information about common objects on the three-dimensional world.

The unique aspect of this research is the development of a reactive planning mechanism that incorporates the user in the planning loop by means of a novel Man-Machine Interface, as well as providing for flexibility of operation in a real world where we have to relax some of the assumptions that classical planners were constrained to work under.

We propose an architecture for task planning which incorporates a novel reactive planning mechanism where the user is an integral component of the planning mechanism. In this model, the user information, in the form of both verbal command and pointing, is part of the environment to which the robot reacts. We are proposing a novel human computer interface viewed as a reactive planner that integrates sensory information, including the interactive guidance of the user.

The planning mechanism is based on an object oriented knowledge base incorporating in it the relaxed assumptions about the world that are essential for the mechanism to be practical in the real world and facilitating human-computer interaction as a means of providing reactive and re-planning capabilities.

The basic architecture in brief is composed of a knowledge base of two parts: A knowledge base of objects (Object Base) and a knowledge base of actions (Plan Base). The knowledge base is object-oriented where objects and actions are themselves classed into an object hierarchy of monotonically increasing specialization. In addition there is a world data base where the workspace information is stored and a planner that uses the Object Base, Plan Base, a Data Base about the workspace, and user and sensor provided feedback, to construct robot plans. For the whole system to work coherently we also require a domain theory that contains information regarding both temporal and spatial relationships between objects. This domain theory is basically a set of axioms describing temporal and spatial relationships between objects in a domain and general axioms that govern object interaction.

The novel human-machine interface that we propose is one where user interfacing is achieved by speech and gestures. Gestures identify to the system objects that need to be manipulated and locations that need to be reached. Speech is used to convey tasks, interact with the system to provide information that the system requires to perform the task and which it may not be able to obtain from the knowledge bases, and to also provide a feedback for imparting reactivity.

Reactivity is achieved in two ways. An autonomous runtime reactivity is obtained through sensor fusion. Sensory information from the vision system, force sensors, etc will be used by the planner to obtain information for not only task planning but also to react to environment changes. However, there are two fundamental limitations to having a totally sensor based reactive planning system:

1. Uncertainty: Sensors can only return limited or incomplete information about the environment. Decisions as to what action to perform on what object are then made under conditions of uncertainty leading to possible failures.

2. Computational Limitations: Sensor interaction is bounded by computational limitations. The most effective actions are those which are sufficient for the problem at hand and require as little effort as possible.

Therefore, we propose a hierarchical human-machine interface, where the user is able 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.

Given this underlying architecture, the system 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 engages in a dialogue with the user which enables the planner to revise its plans and actions.

The rest of this proposal is organized as follows. We begin in section 2, where we review relevant research and literature. Section 3 describes the proposed model, presents preliminary research, and discusses the contributions that will be made by this work. Finally, in section 4, we summarize.

2 Review

In this section we review research and work that are relevant to our research. We first review in general basic AI planning that is cogent to our work and then in more detail the literature that is relevant to the multifarious aspects of our proposed research.

2.1 Planning in AI

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 the 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 [Wil88]. In particular, a classical AI planning system is domain independent, permits plan variables, supports hierarchical planning, and permits non-linear plans.

Sacerdoti's **NOAH** [Sac77] [Sac75], and Fikes and Nilsson's **STRIPS** [FN71] mark the beginning of this approach. 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 which have attempted to work their way out of the morass of classical planning assumptions. Research has been focused on hierarchical planning as in **SIPE** [Wil88], **NONLIN** [Tat77], and **DEVISER** [Ver81], learning and reuse such as [Min87], [Ham90], and [Kol91], and generalized planning if we look at classical AI literature.

2.2 Knowledge based and Object Oriented Planning Mechanisms

More recently, a lot of work has been done in robot planning systems which have tried to extend the classical AI planning paradigm along the direction that we have indicated we must go in order to have practical systems which can perform in the real world. The research has mainly focused along two directions, a neural network approach such as that of Takanori et al. [TT93] [TFK+91], and along the standard symbolic AI path. We describe some of the approaches that have used knowledge bases as well as tried to incorporate the object oriented paradigm in their planning process.

Knowledge based planning systems have been around for a while such as the early **ISIS** system [FS84] and more recently in assembly planning such as Mina [Min87], Koehne and Welz [KW91], Huang and Lee [HL91] and others. The major thrust of knowledge based systems has been in assembly planning such as Huang and Lee [HL91], Delchambre [Del92], Rondeau and El-Maraghy [RM87], and Kak [GK90] amongst others. The most common restrictions to most of these systems was the fact that these systems were restricted to a specific work cell and thus violated one of the major assumptions for a planning system that we have specified in section 1. The use of knowledge bases simply facilitated the planning process and did not in any way provide the flexibility that is necessary.

An object oriented approach has also been used in robot planning, such as those by Chang et al. [CKW91], Todd and Green [MG13] and others. In these systems the OOP paradigm was used to abstract information about objects from a knowledge base into families and was used to pattern match an unfamiliar object with one that exists in the knowledge base.

There have been recent works where researchers have attempted to integrate the object oriented paradigm with knowledge based planning. We outline here some of the more recent relevant research along this direction.

Ryder, Webster, and Dooley [RWD91] describe the application of an object oriented (OO), knowledge based system to the management of the heart rehabilitation patient. The OO model parallels the typical departmental hospital organization. The extensive knowledge base satisfies daily requirements while supporting epidemiological studies. A rule-based application monitors patient risk category and a planning model is provided to satisfy the goals of the four-phase rehabilitation program.

Chang, Kannan and Wong [CKW91] describe the design of an object oriented system for manufacturing planning and control of a job-shop. Job-shop entities such as cells, machines, jobs, parts, and schedules are modeled as objects. The architecture of the system is derived from knowledge based systems and computer simulation.

Sheu and Kashyap [SK87] propose to use the object data model as the basis for automatic process planning. With the representation power provided by logic, the authors propose to represent the description of parts in a knowledge base, and abstract similar parts into families. When a part is given to manufacture, its description is matched against the descriptions of the existing parts (families) and, if a match is found, the associated manufacturing knowledge can be used. In case there is no match, the knowledge base provides a set of high-level operators for the user such that the manufacturing process can be described without worrying about the details.

Bagchi and Kawamura (1992) describes the use of distributed and object oriented concepts in the design and implementation of a single manipulator robotic system called ISAC. They have taken an object oriented approach, but in a way quite dissimilar to how we envisage taking advantage of object oriented knowledge bases and planning systems. They model each agent and activity as an encapsulated object with methods that define actions on these objects. They also have an inheritance hierarchy for objects where the class hierarchy is used for problem decomposition. Their main use of having an object oriented system is to facilitate inter-module communication from client to server resource objects. The distributed modules send or receive tasks through a task blackboard. Their main focus of search has been to develop a distributed planning system, with the advantages of being reliable and flexible.

2.3 Autonomous Planning Vs. Complete Control

Robotics research has either focused on total autonomous planning systems, with the associated limitations described in section 1 [Sch87], [LH91], or towards complete user control.

We describe two seminal works on reactive planning that highlight the limitations in reactivity in a fully autonomous system, which further justifies our novel approach integrating the Human user in a shared control mechanism to enhance reactivity.

In terms of reactive planning in unpredictable environments, the work of Schopper [Sch87] is considered a landmark one. In this work, Schoppers describes a new kind of plan, which he terms the "Universal Plan," which can be synthesized automatically, and yet generates appropriate behavior in unpredictable environments. As opposed to the classical AI paradigm where problems were constructed with unique initial and goal states, his approach specifies only one goal condition. Consequently, the planner is unable to commit to any specific future course of events but must specify appropriate reactions for anticipated situations.

Schoppers states that the core problem of classical planning techniques is over-commitment in an environment that is unpredictable and/or dynamic. He states that the use of abstract plans, of partial plans, and of "reactive procedures" side-steps the early commitment problem by the interleaving of planning and plan execution. In this paper [Sch87] Schoppers focuses on the plan representation and the interpreter that he has developed.

He argues that the novelty of his reactive planning approach is due to the fact that the representation he used to encode robot behaviors can specify appropriate reactions to every possible scenario in a given domain, and yet can be cheaply and automatically synthesized. However, his "Universal Plan" development can get exponentially expensive in terms of computation in a complex domain, and that even though the environment changes may be unpredictable, agents and objects in his domain are known completely *a priori*. However, the fundamental limitation of his "Universal Plans" was the fact that the world must be completely known in advance. Only then, is his system capable of reacting to changing situations.

As another example in reactive planning systems, Lyons and Hendriks [LH91] cast planning as the ongoing adaptation of a reactive system to suit changes in goals or the environment. Their approach is the development of a real time system that continually interacts with the environment, and where the planner is a separate and concurrent system. This eliminates the problem of reactive scheduling and the problems of hierarchical planner-reactive system approach. The reactor is a system containing a network of sensory-motor linkages called reactors. It's two key properties are:

1. It can produce action at any time, and

2. It is a real-time system.

The reactor is initialized with a number of useful behaviors already in place. The planner is separate from the reactor but is concurrent with it. It continually cycles, determining if the reactor's behavior in the current environment will conform to the planner's objectives. Again, even though the planning system has a reactive capability, it is restricted to the current environment, and is limited by the amount of information obtained through sensory data.

Almost all reactive planning systems have been focused on autonomous systems and reactivity was obtained through feedback provided by various sensory mechanisms. And, as was previously emphasized, this approach suffers from the two fundamental limitations of **uncertainty** and **computational limitations**, a fact that has been highlighted by the two works described above.

2.4 Command and Control

Control strategy for semi-autonomous telemanipulation systems has also been a focus of research in robotics. Rehabilitation robotics research literature describes many demonstrations of the use of robotic devices by individuals with disabilities. These are reviewed in [Fou86] and [GT90]. In general, the existing interface strategies have not met the desires of the user community.

In general, the prototype interfaces have taken two approaches to achieving effective use by users. Many have commands which are issued by the user and activate the robot to perform pre-programmed tasks. Others have sought to give the user control of the manipulator's motions. These are analogous to a person riding an elevator where a single command (the selection of a floor) executes a predefined action, as opposed to riding a bicycle where every movement (including steering) is under the control of the user.

2.4.1 Command Based Interface

The command based interface is one in which the robot is programmed to execute pre-defined movements. The early work on the APL/JHU [SS86] Robotic Arm Work Station describes a system which uses a mechanical arm under user command. The concept was to place items in fixed locations on the surface of the work-space and to use pre-stored trajectories and operations to carry out desired functions.

The users entered their selection of a command through a mechanical keypad. In addition to the commands, the system offered modest control of the joints of the robot. A proportional control of the movement of individual joints was possible in order to move the robot to locations that were different from the pre-programmed locations.

Similar types of pre-programmed commands were employed in vocational workstations [GF87] [Fu86]. Considerable work has also been published on the robot workstation designed at Stanford University and the Palo Alto VA Medical Center [HdlP91] [dlHL+90a] [dlHL+90b].

2.4.2 Control Based Interface

In contrast to the command oriented telerobotic systems, there have been a number of systems in which the user directly controls the movements of the manipulator much like a prosthesis. Zeelenberg [Zee86] describes a small robot whose movements are controlled by the position of a track ball. The user has complete control of all movements. None is pre-programmed. Earlier work on the Spartacus Project and the Manus Project [Kwe86] [KTC+92], shows the potential for multiple degrees of freedom control over the robot. The movements of the elements of the robots are controlled with proportional devices, like joysticks and track balls.

This approach offers tremendous flexibility since there are no restrictions to a preset number of commands, a structured environment, or machine *knowledge* of the objects in the world, but is likely to be too demanding of many prospective users. An interface with some built-in intelligence is needed to lighten the cognitive and physical load on prospective users.

2.4.3 Integrating Command and Control

The limitations of a command-based interface were discussed in Michalowski et al. [MCL87]. While modern speech recognizers provide access to large numbers of stored commands, these investigators present the case that effective command of a robot will require use of more commands than is reasonable for the user to remember. As the number of possible commands grows, the human/machine interface becomes increasingly unmanageable. They propose greatly expanding the capability of the robot to not only recognize spoken words, but also understand spoken English sentences.

In a continuation of this work, Crangle [CLSB88] provided an example where the user spoke the sentence,

Move the red book from the table to the shelf.

The proposed system would recognize the spoken sentence and understand the meaning of the sentence. The system would have a knowledge of the immediate world so that the robot knew the locations of the table and shelf, as well as the placement of the book on the table. The knowledge base must also know a great deal about the items in the environment. It must understand what books and tables are. It must also know that the book is red as opposed to blue, and red as opposed to read. Additionally, this system must know that the book has not been moved to a different location.

While the use of such *natural language* interfaces is extremely interesting, and would offer great benefit, the limitations are many. The requirement that the world be entirely structured so that the robot knows precisely where every item is, is likely to be too demanding. No consumer lives in such a world. In addition, the inclusion of a vision system to accommodate a less structured environment will require the ability to perform object recognition. Both the object recognition and the natural language understanding capabilities required by this effort are the subjects of large scale computer science and artificial intelligence research. Until such time as this research progresses to the point where applications are possible, such a sophisticated system is not likely to meet user needs.

A different approach to command-based robot operation was proposed by Harwin et al [Har86]. A vision system viewed the robot's workspace and was programmed to recognize bar codes that were printed on each object. By reading the barcodes and calculating the size and orientation of the barcode, the robot knew the location and orientation of every item.

This was successful within a very limited and structured environment. This system did not easily lend itself to a variety of locations and may not be flexible enough to accommodate the needs of users. It did, however, demonstrate the dramatic reduction in *machine intelligence* that came by eliminating the need for the robot to perform object recognition and language understanding.

2.5 Multi-Modal Interfacing

Researchers have proposed a number of systems which investigate alternate modes of human-computer interaction in addition to the speech and vision based ones described in the previous section. Work has been carried out in using gestures and hand pointing as a mode of man-machine interface. In some systems [WG89], [COK92] researchers have required the users to use hand gloves, while others [FMS92] require calibration for each individual's hand shapes and gestures. Cipolla et el. [CHH] report in a preliminary work on gesture-based interface for robot control. Their system requires no physical contact with the operator, but uses un-calibrated stereo vision with active contours to track the position and pointing direction of a hand. Based on a ground plane constraint, their system is then capable of finding the indicated position in the robot's workspace.

Some researches have also attempted to extend this concept by using multiple modes of man-machine interfacing. The concept of multi-modal interfacing has been discussed extensively by Richard Bolt of the MIT Media Laboratory [Bol80]. Bolt introduced the expression *put that there* in describing his work in optimizing the interface between a user and a large graphical display. The system combined human supervisory control (visual selection and feed-back) with voice commands and gestural pointing.

Cannon at Stanford extended this concept to three dimensional robot operation [Can92]. Cannon's system has worked quite well in laboratory trials. However, it presents problems when being considered as an interface for rehabilitation robotics. The requirement that the user control two video cameras acting as a manually operated range-finder makes this less than desirable for an individual with disabilities.

3 Reactive planning integrating the user in an object-oriented Knowledge-Base model

The introductory section emphasised the features that a telerobotic system must have to perform satisfactorally in a real-world unstructured domain. We saw the necessity for the man-machine interfacing as well as the need for an intelligent reactive planning mechanism.

3.1 The Proposed Model

We have devised a novel reactive planning mechanism that takes a holistic yet simplified approach to planning in a real-world scenario. This novel approach is based on integrating the Human user in the planning process, by means of a gesture-speech driven multi-modal interface, which provides complementary reactive and re-planning capability to the semi-autonomous system, and by means of devising a planning mechanism based on integrating the *object oriented paradigm* with knowledge based planning. We contend, as we shall describe in a later section, that this approach is a novel way to handle the robot planning problem, given the assumptions and capabilities that a planning system must have in order to function in the real world. We propose a new way of achieving the degree of flexibility required, with added advantages that we derive from having an object-oriented model. We shall elaborate on this in a later section.

In this section we describe in brief the architecture of the planning system and will show that this model addresses the new assumptions about the world that need to be made in order to have a system that is practical.

3.2 Characterizing the Robot World

The Robot world is characterized by three physical components:

The physical **objects** themselves, which are the collection of objects that the robot encounters in its environments and which it must manipulate. An object is a rigid solid piece. A piece is always supported by another object, the workspace, or the robot arm. In the actual domain, each object may be supported by more than one support.

The objects and relevant information about them are stored in an object oriented knowledge base. The relations and interactions between objects are specified by a domain theory based on predicate calculus.

The **workspace** provides a surface on which the pieces must rest. The workspace for our domain is essentially the table top on which the objects rest. It may support more than one object, which in turn may support more objects.

The **Robot Arm** is the manipulator capable of moving about the workspace and manipulating the pieces.

3.3 The Architecture

Based on the above premises, we propose an architecture for task planning which is based on an **Object Oriented Knowledge Base**. The knowledge base is essentially composed of two parts:

1. A Knowledge Base of objects
2. A Knowledge Base of actions.

In addition there is a *World Database* where the workspace information is stored and a planner that uses the object knowledge base, the world database, and the action knowledge base to construct robot plans. In order for the whole system to work coherently, we also require a domain theory that contains information regarding temporal and spatial relationships between objects.

3.4 The Object knowledge base

Objects are represented in an increasingly specialized sequence of objects in an inheritance hierarchy. At the top level, we start with a generic abstract object and at the bottom we end up with specific objects whose attributes are fully specified. From the abstract top level objects, we derive objects with intermediate levels of specializations; the choice of these intermediate classes of objects is dependent on the kind of general objects that the system might encounter and the set of tasks that the system might be called on to perform on these objects.

Each object, depending on the degree of generalization, has a set of attributes that will assist the planner in developing correct plans. An initial investigation into the kind of tasks the robot might be called on to undertake prompts us to visualize a set of attributes which include, *shape*, *size*, *dimensions*, *weight*, *approach point*, *grasp points*, *constraints* and *plan fragments*. The *constraints* and *plan fragments* attributes need to be described in a little more detail to explain the working of our model.

3.4.1 Constraints

Constraints may be placed on objects which further constrain approaching, grasping, and moving sub-tasks. For example, we may place a constraint on a cup such that the cup is moved in a specific orientation in order to prevent spillage. These constraints are dependent on which action is being invoked upon the object. For example, in the case of the cup, the constraint about the fixed orientation must be over-ridden if the action involves pouring something out of the cup. Thus the representations of constraints is further qualified by which actions these constraints are applicable to.

3.4.2 Plan Fragments

Another needed component would be plan fragments that are going to be incorporated into plans formed by the planner. Certain tasks may be specific to an object, and those plan fragments may be associated with the object in question in order to facilitate correct planning.

3.5 World Data Base

In addition to the knowledge base of objects, the system also maintains a data base of objects that it sees in the domain, called the *World Data Base.* The objects in the domain contain additional attributes which get instantiated after objects have been identified by the system. Currently, the attributes considered necessary are location and orientation, and attachments to other objects and the workspace.

3.6 Example

A very simple example of the object hierarchy is iluustrated here. Prior to interaction with the user, the system sets up the domain-base as a collection of *blobs* of different sizes and shapes, with only the position with respect to the world origin being known. The *blob* world image is obtained from the vision system and size and location parameters are instantiated in the world data base from the information obtained by the vision system. We are not at this point envisaging the system doing any sort of object recognition. Based on the premise that the user is in the loop, the user points to a *blob* and identifies it to the system. For example, she may point to a specific blob and inform the system that this is a *cup.* The system then updates the attribute slots of the *blob* with attributes that it obtains from the knowledge base. The user may also identify the blob as a specific object, such as *my-cup*; in such a case, the system is aware of a specific object in the knowledge base which is known as *my-cup* and the blob in its domain-base is replaced by the exact *my-cup* that the system knows, and the attributes of *my-cup* in the domain-base is set up from the knowledge base and information derived from the snapshot of the world. It is entirely possible that the user may not have identified any specific blob, and the system then is only aware of the general shape, and the blob is identified at a certain degree of generalization, such as *long-narrow* etc.

3.7 Action Knowledge Base

At this moment we are considering a **STRIPS**-like planning mechanism, where plans have the following general format:

- Action
- Preconditions:
- Constraints:
- Sub-Tasks:
- Goals:
- Effects:

Preconditions must be true before the action can be executed and constraints must remain true during the execution of the action. The top level task may be broken down into more primitive sub tasks, each of which itself is represented as **STRIPS**-like plans. Goals are the primary effect of invoking the action, and effects are the changes that occur in the world as a result of execution of the plan.

The main difference between conventional **STRIPS**-like planning and the proposed 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 as general templates of actions, where plan parameters are instantiated from the object knowledge base during the planning process. For example, the constraint 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 constraint 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 over-ride default constraints in the action body. All plan slots may be instantiated from information obtained from *objects* on which they are invoked in a similar manner.

Another way in which the object oriented paradigm has extended the classical **STRIPS** planning mechanism is illustrated below. As 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. Certain tasks which can be generally handled for most objects may not be applicable to certain objects in the real world.

Suppose we have an instrument that is used often in the domain of the user. The instrument has a peculiar shape and must be picked from a specific point. In order 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 instrument. The approach path may be convoluted and hence 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 of an object. In a manner similar to the way action slots are filled depending on the object on which the actions are invoked, subtask slots are also filled, if so specified, from the object's *plan-fragment*. This is illustrated with an example as shown below.

Suppose the generic *Move* action is specified as follows.

Move *object to-location*

This action may be hierarchically decomposed into sub-tasks,

Open (d1)

Approach [*object.plan-fragment.approach*][*object at-location*]

Close (d2)

MoveTo (*to-location*)

The approach sub-task has a default general plan (the general approach action which will be invoked with the parameters specified in the second set of square brackets), as well as an object based plan (specified inside the first set of square brackets). If the object on which this *Move* plan is being invoked does have an approach plan in its plan-fragment field, then that plan is invoked to satisfy the goal. In the case of the example of an instrument given previously, the plan-fragment field will contain an approach sub-slot where a specific plan for approaching the instrument may be fully specified.

Thus we see that this integration of knowledge base planning with an object oriented approach allows us to use general plans whenever we can but as well allow us to develop plans for specific objects peculiar to the domain without the need to perform computationally expensive operations.

Moreover, each action has a generalized version and specialized versions that are invoked according to knowledge about the object. This allows us to abstract out the general features of an action and invoke them on objects about which the knowledge base might not have any information. It also allows us to view an action as a single action that is applicable to many kinds of objects instead of as a set of actions, each applicable to only one kind of objects as is done in other **STRIPS**-like systems.

However, a conventional **STRIPS**-like planning system is not reactive as of itself. Reactivity is achieved by means of both sensor fusion and human intervention. In our proposed system, the user is actively involved in the planning process, both as a reactive and re-planning component, supplementing the stand-alone planning capability of the model. In addition to the data obtained from sensors, we propose a hierarchical human-machine interface, where the user is able 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.

Given this underlying architecture, the system 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 engages in a dialogue with the user which enables the planner to revise its plans and actions.

Reactivity under uncertainty in a real-world unstructured domain is necessary under two general scenarios, one dealing with minor plan failures and the other with catastrophic failures. **STRIPS**-like planning systems generate a sequence of sub-tasks which need to be performed in order to achieve the top level goal as specified by the user. Each of these sub-tasks have sub-goals that need to be met before subsequent sub-tasks can be executed. During the execution phase, our reactive planner needs to determine after each sub-task execution whether the desired sub-goal has been achieved or not. Due to the inherent uncertainty of the domain, it is quite probable that the desired goal may not have been met satisfactorily. For example, slippages of both the motor gears and the object being grasped by the manipulator, and the inherent inaccuracy of the vision system might result in an object being moved to a position where it was not intended. When such discrepancies in goal-satisfaction occur during each sub-task execution, our planner then needs to correct the situation. This is to be done by plans which are generated on-the-fly from information gathered from the sensors. Such plans might be general enough to be applicable under different situations or they might be more specific to the task that is being executed. Knowing the probable errors in executing such a task, such a plan might also have pointers to a set of plans from which the planner choses one or more based on sensory information so as to be able to correct the error in execution. On the other hand, catastrophic changes in the environment may prevent the planner from generating rectifying plans to succesfully achieve the desired user goal. Under such conditions, the planner terminates its execution and engages in a dialogue with the user. This dialog might be necessary to update the planner's knowledge base to allow it to replan or react to the catastrophic change or even to formulate new plans that take into account the changed situation. Furthermore, during plan execution, the user may need to intervene when the plan that is being executed appears not to lead to a goal desired by the user. Under such scenarios, the planner again needs to interact with the user so as to obtain new goals or new information that is necessary to achieve the desired or changed goal of the user.

With an enhanced ability to reason about complex and unstructured environments and the associated increase in the number and complexity of actions available to a system comes a concomitant increase in the amount and complexity of information required to make intelligent and informed decisions. However, current information processing limitations put a severe constraint on the amount of information that can perceived, stored, retrieved and processed in a given time-span, whereas a reactive system must be capable of responding to uncertain and changing environments in an **acceptable** time-span.

3.8 Expected results

In this section we describe a set of scenarios that illustrate how our proposed system is going to function in a real-world scenario.

3.8.1 Preliminaries

A partial description of the contents of the knowledge base is provided to facilitate the description. The system has in its knowledge base knowledge about top level objects such as cylinders, cubes, flat objects etc. In its knowledge base exists information about a specific cup *my-cup* that the user often uses. The *my-cup* object is hence fully instantiated. The exact dimensions, approach points, grasp points amongst others are fully specified. This *my-cup* object is derived from a more generic *cup* object. At this level of generalization, dimensions are given in terms of ranges and possible values. Furthermore, a constraint is placed on the *cup* object, which states that the cup must always be kept at a certain orientation. However, this constraint must be over-ridden if the user wants to pour something from the cup and such constraints can be over-ridden by the action being invoked on the object. Furthermore, in the plan library are top level actions such as *pour*, as well as more primitive actions such as *pick*, *put*, and *move*. Associated with each object are plan fragments that affect the actions which can be performed on them. These are placed in the plan fragment slots of the object attributes. This is so because not all plans can be made generic enough. For example, in the case of *my-cup*, the *pick* operation for a cup is very specific. The cup must be grasped by the handle and such an action is specific to only *cup* objects. So when the *pick* action is invoked on a *cup* object, the plan derives a sub-task from the object itself which determines how exactly the object might be picked. Given the object oriented nature of both the pieces and actions in our domain, this is easily facilitated.

3.8.2 A General Illustration

The user approaches a table on which there are a bottle and a glass, both of which are in the knowledge base. The user points to the bottle, and says, *bottle*. From this, the system knows how to approach the bottle. The user points to the glass and says *glass*, *above there*, indicating that the object is a glass, and the final location of the bottle is above the glass.The knowledge base provides additional information. The system, knowing the two height dimensions, computes the distance needed to be *above* the glass. The bottle is moved to its desired destination.

The user says, *pour*, which initiates pouring by rotating the bottle in a pouring motion directly over the glass. The user monitors the process and stops the pouring by saying, *stop*. A command, *return*, restores the bottle to its original location. The command, *home*, moves the robot out of the way.

3.8.3 Illustration of the approach (known objects)

In this scenario, the user approaches his/her own desk where objects are routinely used and are familiar. These objects may be present in the knowledge base. The vision system surveys the scene, and computes a three dimensional surface (the orientation of this surface will be dependent upon the position of the user's wheelchair). The user points to the user's cup and identifies it as a known object with the word *my-cup*. This tells the system about its weight, its dimensions, and the approach path to be taken by the robot. The user says *move*. The user then points to a surface on the table and says *there*. From the information that the planner derives from the *my-cup*'s parent class, the planner will then calculate the path that needs to be followed to place the *my-cup there*, while maintaining the constraint that the cup must be kept at a certain orientation.

3.8.4 Illustration of the approach (unfamiliar environment)

The user in a wheelchair equipped with a portable robot and its vision system approaches a desk. There are objects on the desk with which the system is not completely familiar. After the vision processing, the user points to a cup on the desk and identifies it as a cup. The system instantiates its world base from the knowledge about the *cup* object. Now if the user then gives the same instruction as in the previous illustration, the planner would be able to plan the correct path on which the cup must be moved.

3.8.5 Illustration of the approach (plan adaptation or unknown object)

In this modified scenario the user again approaches an unfamiliar environment. After the vision processing, the user points to a mug and tells the system that this is a *mug* object. The system was previously unaware of a *mug* object in its knowledge base. If the user wants to now pick and move the mug she can do one of four things.

She can load up the knowledge base with information regarding the *mug* object so that the system is able to handle operations on the mug.

Secondly, she can inform the system that a *mug* is a *cup*-like object and that it derives from a *cylinder* type object. When the user invokes a *pick* command on the mug, the system then generalizes the *pick* command applicable to a *cup* object and uses it on the mug. From the knowledge base the system is able to infer that a cup must be picked up from the handle, and the system then attempts to determine the location of the handle for the mug in order to ascertain what the approach points and the grasp points are going to be. Based on the vision system information and the generalization of the *pick*, the system then instantiates the parameters for the *mug* object so that next time it has no need to generalize. Further attributes for the object, such as *weight* etc., can be added during the actual process of executing the *pick* operation.

Thirdly, the user simply informs the system that the object is a mug and instructs it to pick the mug up. This time the system uses information gathered from the vision system to determine a suitable approach point and grasp point (this may not necessarily be the mug handle) and initiates the action with the gripper open wide enough to grasp, and uses the force sensors in its fingers to grasp the mug. This is an example of the most abstracted example of a *pick* operation in our plan knowledge base. During the process the system as usual updates as much as it is able to the attributes for the mug object.

Fourthly, the user may directly control the arm in approaching and grasping the object. During the operation the system again instantiates the attributes for the object for later reuse.

3.8.6 Illustration of the approach (reactivity and user interaction)

In a similar scenario as described in the preceding sub-sections the user instructs the arm to pick a cup and place it on the table at a certain location. At the end of the execution phase, the system ascertains that the cup which is still in its grasp is not quite touching the surface of the table. This determination is made through sensory information such as force feedback. When this occurs, the planner then generates a plan which will allow the cup to be placed on the table, while continually receiving force feedback to determine the succesful completion. In a more catastrophic scenario, the cup falls out of the grasp of the manipulator. The fact that the cup has fallen can be detected by the arm, but further execution of the plan is impossible and the system then needs to interact with the human user. The user may then determine whether the goal needs to be satisfied given the catastrophic event. In the event that the goal still needs to be satisfied, the system has to rescan the work space and the user needs to identify the new location of the dropped cup so that the planner can generate a new plan.

3.9 Discussion

The results of the our research will be applied to the development of a multi-modal controlled rehabilitation robot. Work on this project is supported by the Rehabilitation Engineering Research Center in Rehabilitation Robotics, NIDRR Grant H133E30013, and Nemours Research Programs. The test-bed in this project will be established in the robotics laboratory of the ASEL.

This project addresses the integration of command-based and control-based interfaces to provide an effective user/robot system that can be used in a variety of settings and in both structured and unstructured environments. The project will employ multi-modal user inputs including voice recognition for commands and gestures (pointing) for locations (end points) and movement paths (trajectories). This user direction (a combination of command and control) will provide for rapid operation of the manipulator. It will employ the power of predefined commands in conjunction with the flexibility of user control. The robot may be attached to the user's wheelchair and the user should be free to use the arm in any environment in which he/she travels. No particular structure will be imposed on the environment.

In the conceptualized scene shown in Figure 1, an individual with a disability uses an electric wheelchair and a portable robot arm. The user wishes to move the pen, which is on the desk, to the box. The user (in this example using a head pointer), points to the pen and says, "move." The user then points to the box, and says, "there." The combination of the initial pointing accompanied by the command "move" tells the robot to pick up an object at a specific location. The combination of the subsequent pointing and the command "there," tells the robot where to move the object.

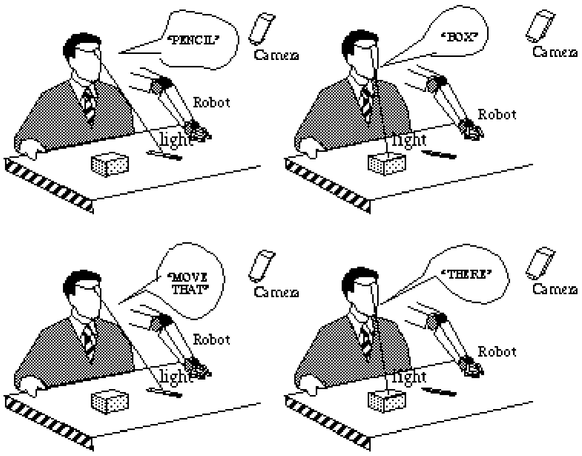

Figure 1: Multi Modal Control

An experimental test-bed with a Zebra ZERO six degree of freedom manipulator has been developed in the laboratory. The vision system work currently in progress by other researchers is being done on a Silicon Graphics Iris Indigo machine. The robot control software and voice input will be performed on a 486 machine. Our work dealing with the integration of the various input data from the vision system as well as voice input is being implemented in a Sun workstation. In order to validate the research, we intend to test the system in a simulated office environment, where the user is going to manipulate a subset of objects she may encounter in such an environment. Currently we have developed a prototype knowledge-base and the interfacing mechanism between the input devices and the robot. Further work will be done to enhance this for the domain we have in mind as well as expanding the knowledge base and implementing the reactive-planner.

3.10 Summary

The main thrust of our research has been to propose a robot planning system that is able to work in the real world where we need to relax a set of assumptions that most early planners were constrained to work under. We have proposed a novel way of integrating the user as a component in a reactive planning schema based on an architecture which integrates knowledge based planning with the object oriented paradigm. As a consequence we have a novel reactive planning / re-planning mechanism which also satisfies the capabilities of a planning system that Wilkins [Wil88] stated is necessary.

As mentioned in the introduction, human intervention as well as an intelligent planning mechanism are essential features of a telerobotic system. In our research, we propose a new model of robot planning system that will be practical in the real world where we need to relax some strong assumptions about the domain made by classical planning systems. This is achieved by integrating a novel gesture-speech driven human interface to a reactive planning mechanism. The desired features of a planning mechanism is further enhanced by sensor-fusion and having the underlying planning mechanism based on integrating knowledge based planning with the *object oriented* paradigm.

We believe, that this novel approach of gesture-speech based human-machine interfacing enables our system to make realistic plans in a domain where we have to deal with incomplete knowledge and plans and where spatial and temporal dimensions are not discrete as well as react to uncertain and changing environments. We believe our system will be flexible enough to work in disparate environments as well as be able to deal with issues such as plan re-use, adaptation and recovery.

We contend that our system not only has all the capabilities of a planning system that Wilkins identified as being essential, in addition it has capabilities that enable it to function in the real world, namely:

- domain independence
- operating in unstructured environments
- operating under conditions of continuous spatial and temporal dimensions
- operating under the conditions of incomplete information
- reactive planning capability

Therefore, we contend that the approach we have taken breaks new grounds in telerobotic research where the novel integration of multi-modal human machine interfacing and the reactive planning mechanism

References

[Bol80] R. A. Bolt. Put that there: Voice and gesture at the graphics interface. *Computer Graphics*, 14(3):262–270, 1980.

[Can92] D. Cannon. *Point and Direct Telerobotics: Object Level Strategic Supervision in Unstructured Human-Machine Interaction.* PhD thesis, Stanford University, Department of Mechanical Engineering, 1992.

[CHH] R. Cipolla, P. Hadfield, and N. Hollinghurst. Uncalibrated stereo vision with pointing for a man-machine interface. To appear in MVA'94.

[CKW91] A. Chang, P. Kannan, and B. Wong. Design of an object-oriented system for manufacturing planning and control. *Proceedings of the Rensselaer's 2nd International Conference on Computer Integrated Manufacturing*, 1991.

[CLSB88] C. Crangle, L. Liang, P. Suppes, and M Barlow. Using english to instruct a robotic aid: An experiment in an office-like environment. In *Proc. of the International Conf. of the Association for the Advancement of Rehabilitation Technology*, pages 466–467, 1988.

[COK92] R. Cipolla, Y. Okamoto, and Y. Kuno. Qualitative visual interpretation of hand gestures using motion parallex. *Proc. IAPR Workshop on Machine Vision Applications*, pages 477–482, 1992.

[Del92] A. Delchambre. Knowledge-based process planning and line design in robotized assembly. *Proceedings of SPIE*, 1468(1):367–378, 1992.

[dlHL+90a] M. Van der loos, J. Hammel, D. Lees, D. Chang, I. Perkash, and L. Leifer. Voice controlled robot systems as a quadriplegic programmer's assistant. *In Proceedings of the 13th Annual RESNA Conf. Washington, DC*, 1990.

[dlHL+90b] M. Van der loos, J. Hammel, D. Lees, D. Chang, and D Schwant. Design of a vocational assistant robot workstation. Annual report, Rehabilitation Research and Development Center, Palo Alto VA Medical Center, Palo Alto, CA, 1990.

[FMS92] M. Fukimoto, K. Mase, and Y. Suenga. Realtime detection of pointing action of a glove free interface. *Proc. IAPR Workshop on Machine Vision Applications*, pages 473–476, 1992.

[FN71] 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.

[Fou86] R. A. Foulds. *Interactive Robotics Aids-One Option for Independent Living: an International Perspective*. World Rehabilitation Fund, 1986.

[FS84] M. S. Fox and S. Smith. ISIS: A knowledge-based system for factory scheduling. *Expert Systems*, 1:25–49, 1984.

[Fu86] C. Fu. *Interactive Robotics Aids-One Option for Independent Living: an International Perspective*, volume 1 of *37*, chapter An Independent Vocational Workstation for a Quadriplegic, page 42. World Rehabilitation Fund, 1986.

[GF87] M. Gilbert and R. Foulds. Robotics at the Tufts New England Medical Center. In *Proceedings of the 10th Annual Conference on Rehabilitation Engineering*, pages 778–780, San Jose, 1987.

[GK90] S. N. Gottschlich and A. C. Kak. Assembly knowledge representation for assembly motion planning and execution. *Proceedings of the 5th IEEE International Symposium on Intelligence Control 1990*, pages 948–956, 1990.

[GT90] M. Gilbert and J Trefsager. In *Proceedings of the 1990 International Conference on Rehabilitation Robotics*, Wilmington, DE, 1990.

[Ham90] K. J. Hammond. Explaining and repairing plans that fail. *Artificial Intelligence*, 45(1-2):173–228, Sep 1990.

[HdlP91] J. Hammel, M. Van der loose, and I. Perkash. Evaluation of DeVAR-IV with a quadriplegic employee. Annual report, Rehabilitation Research and Development Center, Palo Alto, CA, 1991.

[HL91] Y. F. Huang and S. G. Lee. An automatic assembly planning system. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 599–604, Piscataway, NJ, USA, 1991. IEEE, IEEE Service Center.

[Kol91] J. L. Kolodner. Improving human decision making through case-based decision aiding. *AI Magazine*, 12(2):52–68, Summer 1991.

[KTC+92] H. Kwee, M. Thonninsen, G. Cremers, J. Duimel, and R. Westgeest. Configuring the Manus system. In *Proceedings of RESNA International'92*, pages 584–587, Washington, DC, 1992. RESNA International, RESNA PRESS.

[KW91] A. Koehne and B. Welz. Wissensbasierte aktionsplanung und konfigurierung. *Ein Ueberblick Ueber den Stand der Technik*, 7(3):169–177, September 1991.

[Kwe86] H. Kwee. *Interactive Robotic Aids-One Option for Independent Living: An International Perspective, volume Mono graph 37*, chapter Spartacus and Manus: Telethesis Developments in France and the Netherlands, pages 7–17. World Rehabilitation Fund, 1986.

[MCL87] S. Michalowski, C. Crangle, and L. Liang. Experimental study of a natural language interface to an instructable robotic aid for the severely disabl. *In Proc. of the 10th Annual Conf. on Rehabilitation Technology*, pages 466–467, 1987.

[MG13] T. Michael and D. G. Green. Object-oriented approach to robotic motion. In *IEEE SOUTHEASTCON 1993*, Piscataway, NJ, USA, 19913. IEEE, IEEE Service Center.

[Min87] I. Mina. KMPR: An experimental knowledge-based modeling prototype for robots. *Proceedings - 1987 IEEE International Conference on Robotics and Automation*, pages 2011–2020, 1987.

[NSF88] NSF. Report of workshop on coordinated multiple robot manipulators: Planing control and applications. *IEEE Journal of Automation and Robotics*, 4(1):91–93, 1988.

[RM87] J. M. Rondeau and H. A. El Maraghy. Development of a knowledge-based robot task planning system for mechanical assembly. *American Society of Mechanical Engineers, Design Engineering Division (Publication)*, 20:23–30, 1987.

[RWD91] R. M. Ryder, W. A. Webster, and R. L. Dooley. An object-oriented, knowledge-based system for the management of the heart rehabilitation patient. *Proceedings of the 13th Annual International Conference of the Engineering in Medicine and Biology Society*, 13(3), 1991.

[Sac75] Earl D. Sacerdoti. The non-linear nature of plans. *Proceedings of IJCAI-75*, 1975.

[Sac77] Earl D. Sacerdoti. *A Structure for Plans and Behaviour*. American Elsevier, New York, 1977.

[Sch92] Paul S. Schenker. *Intelligent Robotic Systems*, chapter Intelligent Robotics for Space Applications, pages 545–591. Marcel Dekker, Inc, 270 Madison Avenue, New York, 1992.

[She92] Thomas B. Sheridan. *Telerobotivs, Automation, and Human Supervisory Control.* The MIT Press, Cambridge, MA, 1992.

[SK87] P. Sheu and R. L. Kashyap. Object-based process planning in automatic manufacturing environments. *Proceedings - 1987 IEEE International Conference on Robotics and Automation*, pages 435–440, 1987.

[SS86] W. Seamone and G. Schmeisser. *Interactive Robotics Aids-One Option for Independent Living: an International Perspective*, chapter Evaluation of the APL/JHU robotic arm workstation, page 51. World Rehabilitation Fund, 1986.

[Tat77] A. Tate. Generating project networks. *Proceedings of IJCAI-77*, pages 888–893, 1977.

[TFK+91] S. Takanori, T. Fukuda, K. Kosuga, F. Arai, and M Tokita nd T. Mitsuoka. Hierarchical hybrid neuromorphic control for robotic motions–sensing, recognition, planning, adaptation, and learning. In *IECON Proceedings (Industrial Electronics Conference)*, pages 1465–1470, Los Alamitos, CA, USA, 1991. IEEE, IEEE, Computer Society.

[TT93] S. Takanori and F. Toshio. Hierarchical hybrid neuromorphic control system for robotic manipulators. *JSME International*, 36(1):100–109, March 1993.

[Ver81] S. Vere. Planning in time: Windows and durations for activities and goals. *IEEE Transactions in on Pattern Analysis and Machine Intelligence*, pages 250–271, 1981.

[WG89] D. Wiemer and S. G. Ganapathy. A synthetic visual environment with hand gesturing and voice input. *Proc CHI'89*, pages 235–240, 1989.

[Wil88] D. E. Wilkins. *Practical Planning: Extending the Classical AI planning Paradigm.* Morgan-Kauffman, California, 1988.