

**N89 - 15559****PLANNING ACTIONS IN ROBOT AUTOMATED OPERATIONS**

A. Das  
Computer and Info. Sci. Dept.  
Alabama A&M University  
Normal, AL 35762

**ABSTRACT**

Action planning in robot automated operations requires intelligent task level programming. Invoking intelligence necessitates a typical blackboard based architecture, where, a plan is a vector between the start frame and the goal frame. This vector is composed of partially ordered bases. A partial ordering of bases presents good and bad sides in action planning. Partial ordering demands nonmonotonic reasoning via default reasoning. This demands the use of a temporal data base management system.

**INTRODUCTION**

Advanced technology for the space station and the US economy necessitates substantial use of general purpose automation and robotics requiring new generation machine intelligence and robotics technology. Three years ago, on this issue, NASA's Advanced Technolog Advisory Committee published a set of 13-point recommendations[1]. Intelligent plan adoption by robots is one major vital curriculum. Its ultimate purpose is to imply multi-level environment perception and modelling, decisional autonomy ranging from general planning to specific task operating, autonomous mobility capacity, sophisticated high level man-machine interface and efficient execution control systems. However, vagaries of the real world, its geometry, inexactness and noise pose large practical problems to the researcher and this forces investigations to have excercised on a handful of toy examples.

Recently, an attempt has been made to create an architecture for simulating intelligence in robot automated assembly operation[2]. In this architecture the reasoning system works with a two-dimensional system configuration, a task level configuration (embeded to high level plans and common sense reasoning) and a robot level configuration (numeric activities to live with ordinary geometric world). In programming robots the task level operations are specified according to their expected effects on objects, detailed kinetics of motion even as functions of inputs are not considered directly. In the process of task level programming every new robot level task(geometric world) is seen as a clusture of incremental planning at the task level(plans and reasons). Any problem in this incremental planning working with this two-dimensional system configuration is resolved by a blackboard based problem solver[6]. In this blackboard, all classes of temporal, spatial and event class

relationships invoked by the task structure are examined, and suitable prescriptions generated. The current architecture looks at task level plans vectors connecting the start frame to the goal frame and are composed of bases  $\langle R, T \rangle$ s, where R represents robot level parameters and T represents task level plan configuration operative in the geometric world given by R.  $\langle R, T \rangle$ s are seen to form a partially ordered set to admit mutations with time.

This paper gives a closer look at the preconditions existing in action planning for a robot at the task level. The implicability of partial ordering of  $\langle R, T \rangle$ s is questioned and associated problems are formulated with common sense reasoning. It is observed that this leads to reasoning by default[9] bringing task level planning in the paradigm of nonmonotonic reasoning. A demand for a temporal data base system[3] for action planning seems inevitable.

#### PLAN VECTOR ON A BLACKBOARD

In ref.2 a plan is a vector in the robot level task level configuration space:

$$C_1 \langle R_i, T_j \rangle + C_2 \langle R_m, T_n \rangle + \dots + C_n \langle R_y, T_z \rangle \quad (\text{Eq.1})$$

where  $C_1, C_2, \dots, C_n$  are real or complex numbers.  $\langle R, T \rangle$ s are bases that mutate with time, also. They comprise a partially ordered set. A blackboard is a problem solver[3] enriched with highly domain specific heuristic knowledge. Ref.2 discusses the diagnosis of the simple case of the movement of the robot arm on a blackboard. Since the robot does not do only one single job, and since most of the tasks require repetition of the same subtasks several time, a comparative diagnostics is attributable on the black board. Two or more plans are compared side by side. Searching "plan-invoking macros" becomes more effective. There are problems, however.

#### WHY PARTIAL ORDER ?

In eq.1  $\langle R, T \rangle$ s are partially ordered bases. This gives freedom in forming the plan vector intelligently. The total geometric space configuration assumed in performing a job may then be seen as composed of a series of subtasks. Each of these subtasks was constructed out of realizable  $\langle R, T \rangle$ s. Thus two plan vectors designed for totally two different jobs may be known as differing by additions, subtractions and modifications of some  $\langle R, T \rangle$ s. One plan, say, is going to the grocery store and another, going to the doctor's office. These two are implemented in this way:

##### Grocery

Go 2 miles straight.  
Turn left, go straight.  
Turn left, go straight.  
Turn right, go straight.

##### Doctor's Office

Go 2 miles straight.  
Turn left, go straight.  
Turn left, go straight.  
Turn right, go straight.

Turn left, go straight.  
Turn left, go straight.

Turn left, go straight.  
Turn right, go straight.

These two plans differ in their final turn. Bases  $\langle R, T \rangle$ s are partially ordered in both these plans. Therefore, if one plan is implemented successfully and the other is not, a comparative diagnostic measures can be worked out (possibly searching common macros in both the cases). If on the other hand  $\langle R, T \rangle$ s were totally ordered, the two plans differ without any flexibility of having a match between them.

#### POSSIBLE TROUBLES IN PARTIALLY ORDERED $\langle R, T \rangle$ s

The aforesaid example of comparing two plans on the basis of partially ordered bases  $\langle R, T \rangle$ s requires that a strong table management system admonishing context dependent properties of  $\langle R, T \rangle$ s need be present (A comparative diagnostics of a faulty plan, or, an working plan showing bugs later requires all the macros in correct order invoking the two plans). If in the second plan, for example, a fault is observed in the last right turn, then the comparative diagnostics requires an account of the past history in correct order in both the plans. If you have known how to go to the grocery store (Plan 1) then going to the doctor's office (plan 2) needs a little modification in the final phase. Tracing the two plan vectors side by side with  $\langle R, T \rangle$ s implementing them is necessary in case of bugs observed in one (or both) of them. Such tracing of history also requires cataloging and comparing time of occurrence of all subtasks, their duration needs to be noted too. To understand why you could not reach the doctor's office requires answering a question like how long did you take to perform the first two left turns, for example. Temporal ramifications of  $\langle R, T \rangle$ s and managing their order of context dependency causes trouble in working with partially ordered  $\langle R, T \rangle$ s. Obviously, it is a horrendous task.

#### THREE MORE PROBLEMS

Three more problems will arise in comparative diagnostics, infact, in any reasoning about action. These three problems are the frame problem of McCarthy and Hayes[8], qualification problem of McCarthy[7] and the ramification problem of Finger[4].

The frame problem enters into comparative diagnostics when two or more plan vectors are compared basis by basis, or subtasks by subtasks to determine which of them remain invariant in time while the action is taking place. If I succeed in going to the grocery shop but fail to go to the doctor's office, it was necessary to be determined that these two plans differed only in their final turn (as seen before). No interim unwarranted turn is admissible in both the plans, they were framed by preconditions.

This framing of preconditions lead to the second problem called qualification problem. It arises because the number of preconditions are always very large. Imagine all of the

possibilities that prevented me in taking the last right turn while going to the doctor's office. Probably, my car broke down. probably, it was raining heavily and I missed the road sign. Probably, I stopped somewhere before the final turn to buy coffee and while taking off took a different direction. Probably, in the last right turn there was a detour sign forcing direction change. It is very unaffordable to pin point all these worldly possibilities.

The third problem is the ramification problem which is very severe in comparative diagnostics because it is unreasonable to explicitly record all the consequences of actions. In both our examples of plans on going to grocery and doctor's office a great number of possible consequences may occur which may not have any consequence at all. In going to the grocery store after making the first left turn I may see the fish market and buy some fish and after the next turn I may find my sister's home nearby and deliver part of the fish to her. The applicability of these ramifications for one plan will not be the same for the other. Moreover, the comparative diagnostician will not be able to work out which ramifications are supposed to show up any time for any plan vector under investigation. Inference in default logic may be a way out.

INFER  $\langle R_x, T_y \rangle$  FROM THE INABILITY TO INFER  $\langle R_m, T_n \rangle$

All the ramifications of any plan vector must be expressible using bonafide bases  $\langle R, T \rangle$ s. It turns out that if there are  $n$   $\langle R, T \rangle$ s in a plan vector, any one single new  $\langle R, T \rangle$  for admitting a new event or action in the plan vector will require  $n$  verifications for consistency with existing constraints. Therefore, reasoning with default logic automatically sets in: facts persists in the absence of information to the contrary. If I want to compare my faulty plan not leading to the doctor's office with the successful plan leading to the grocery store, all the possible ramifications that may be present in both successful and unsuccessful plans needs to be assumed existing, because they cannot be verified. This is expressed using Reiter's default rules[9]:

$$\frac{\langle R, T \rangle_t : \langle R, T \rangle_{do(a,t)}}{\langle R, T \rangle_{do(a,t)}} \quad , \quad (Eq.2)$$

Which states that if  $\langle R, T \rangle$  is true in time (or situation)  $t$  and  $\langle R, T \rangle$  are still consistent after the action  $a$ , then we can infer  $\langle R, T \rangle$  after the action. Computational problems still persists, though. To determine what is true after an action has been performed, the default frame axiom must be examined once for every fact of interest[5].

#### PLANNING ACTIONS

Within the context of above mentioned observations and

conditions planning actions needs a strong nonmonotonic reasoning system for its support. The three immediately visible reasons for this are: the presence of incomplete information requires default reasoning, a changing world must be described by a changing data base, temporary assumptions about partial solution may be required for generating a complete solution[10]. In the present model the bases  $\langle R, T \rangle$ s chronologically mutate in time between the start frame and the goal frame. This manifests a temporal data base system which is an extension of classical predicate calculus data base. Mutations of  $\langle R, T \rangle$ s also means that temporal information is incomplete, that is, our knowledge on the occurrence of events totally admits to partial ordering of  $\langle R, T \rangle$ s to implement a plan. Such a system can be fruitfully dealt with a data base system called the time map manager or TMM[3]. Such a TMM admits shallow temporal reasoning to be consistent with the default reasoning system, because by default a deductive reasoning system works with a small number of calculations. Shallow reasoning systems provide the TMM with a mechanism for monitoring the continued validity of conditional predictions. Thus, the TMM rearranges tasks to take advantage of existing preconditions and warns of unexpected dangerous interaction between the effects of unrelated tasks.

## CONCLUSION

The comparative diagnostics on a blackboard with the help of a time map manager is akin to visually scanning a massive amount of data organized in the form of a map. This brings a graphical picture to action planning. Default reasoning makes this action planning a nonmonotonic shallow reasoning system. Hints are there that using the time map manager a comparative diagnostics on a blackboard may be able to overcome some classical problems associated with partial ordering of  $\langle R, T \rangle$ s. It is currently under investigation.

## REFERENCES

- [1] Cohen, A. and Erickson, J.D., "Future Uses of Machine Intelligence and Robotics for the Space Station and Implications for the U.S. Economy", IEEE Journ. Robotics & Automation, Vol. RA-1, No.3 (1985) 117-123.
- [2] Das. A. and Saha. H., "Embedding Intelligence In Robot Automated Assembly", Proc. First Int. Conf. on Indus. & Engng. Appln. of A.I & Expert Systems, Tullahoma, Tennessee, June 1-3 (1988) 1083-1088.
- [3] Dean, T.L. and McDermott. D.V., "Temporal Data Base Management", Artif. Intell. Vol. 32, No-1 (1987) 1-55.
- [4] Finger, J.J., "Exploiting Constraints In Design Synthesis", Ph.D. thesis, Stanford Univ., Stanford, CA (1987).
- [5] Ginsberg, M.L. and Smith, D.E., "Reasoning About Actions I:

- A Possible Worlds Approach", Artif. Intell. Vol. 35, No. 2 (1988) 165-195.
- [6] Hayes-Roth. B., "A Blackboard Architecture For Control", Artif. Intell. Vol. 26, No. 3 (1985) 251-321.
- [7] McCarthy. J., "Epistemological Problems of Artificial Intelligence", Proc. IJCAI-77, Cambridge, Mass (1977) 1038-1044.
- [8] McCarthy.J. and Hayes.P.J., "Some Philosophical Problems From The Standpoint of A.I." in Machine Intelligence-4, ed. by Meltzer.B. and Michie. D., American Elseveir, N.Y. (1969) 463-502.
- [9] Reiter. R., " A Logic For Default Reasoning", Artif. Intell. vol. 13, No. 1 (1980) 81-132.
- [10] See for example, Artificial Intelligence by Rich. E., McGraw-Hill Book Co., NY, 1983.