

THE EXECUTION OF PLANS IN AN INDEPENDENT DYNAMIC MICROWORLD

Gordon McCalla
Department of Computational Science
University of Saskatchewan
Saskatoon, CANADA S7N 0W0

Peter F. Schneider
Department of Computer Science
University of Toronto
Toronto, CANADA M5S 1A7

The problem of executing plans in an independent dynamic microworld is discussed. The microworld is a simulated city that not only contains permanent features (e.g. streets) but also transient features (e.g. other cars). A simulated taxi driver (ELMER) must plan and execute routes through this city. Among the interesting features of ELMER are the use of hierarchical rather than linear plans; the representation of knowledge by routes (similar to and possibly built from plans); and planning by splicing together these routes.

1 INTRODUCTION

One of the more productive areas of AI research has been the study of producing plans to solve problems in a microworld. Usually such worlds (e.g. the blocks world) are static. We are interested in planning for microworlds which can change continuously and independently of the planning agent. Such microworlds have received some attention [1] [2].

The microworld here is an abstraction of the environment a taxi driver might encounter. The microworld and the taxi driver (ELMER) are simulated as asynchronous processes where ELMER informs the microworld of speed and direction changes and the microworld provides ELMER with a "window" of what he can "see" at any given instant. The three central components of ELMER are the Planner that creates a travel plan from a customer's request; the Executor that carries out the plan; and the Map that maintains ELMER's developing knowledge of the city.

ELMER exists in Simon City (Figure 1). The microworld component of the system must simulate this city, including not only the permanent street arrangement, but also transient features such as traffic lights or other cars. It must also maintain the window and keep track of ELMER. The microworld is not yet implemented, but an implementation strategy based on something like Hendrix's scenarios [3] may be appropriate.

2 THE EXECUTOR

The Executor undertakes plans produced by the

Planner. Plans are represented as hierarchies of ever more detailed sub-plans similar to the plans of Miller *et al* [4], but in contrast to the usual linear sequences of actions (e.g. HACKER [5] or STRIPS [6]). Figure 2 shows a plan for going from Lenat Lane at Kuipers Crescent to Reiter Road at Schubert Street in Simon City. The numbered "boxes" are the core of the plan; the other boxes are demon-like secondary plans controlling ELMER's speed and direction and looking out for obstructions such

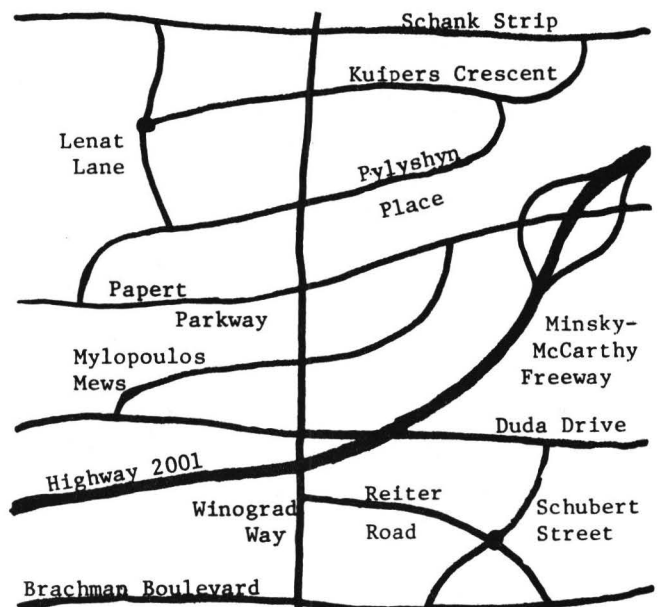
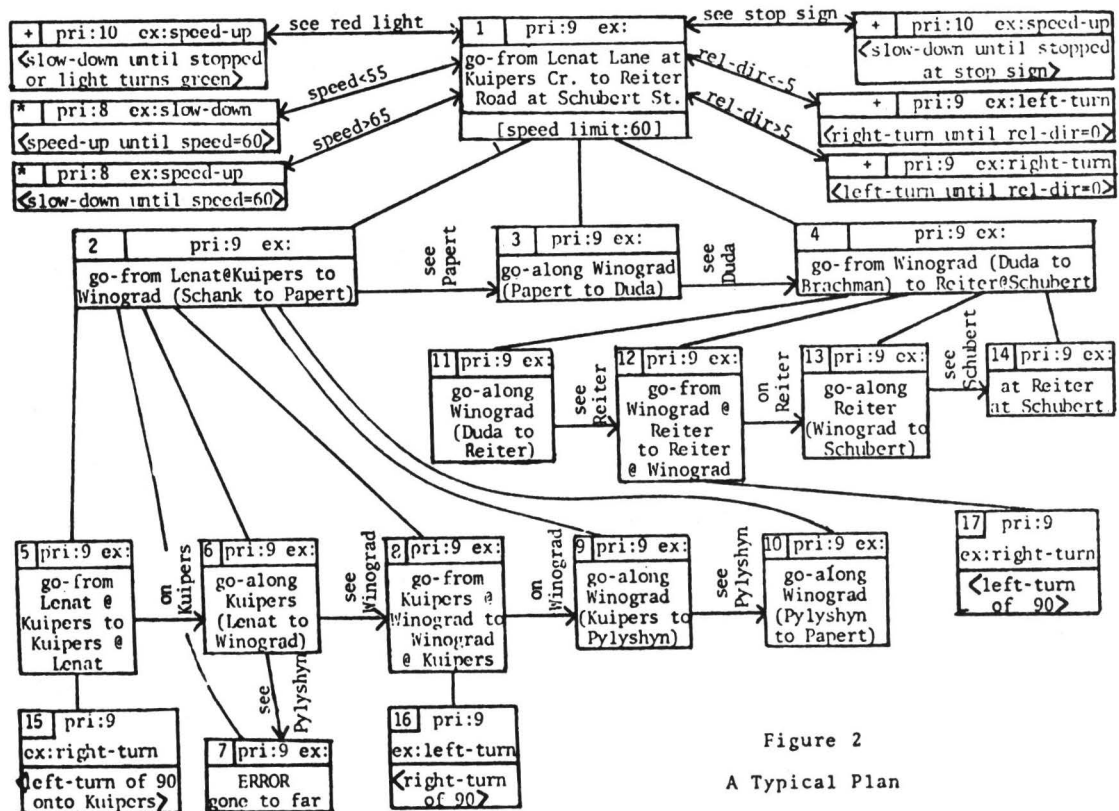


Figure 1 - Simon City

are activated. Note the similarity of a plan to a TOTE (Test-Operate-Test-Exit) [4] and to a DOWHEN, DOWHAT, DOUNTIL triple [3]. Of course, the labels and actions here are more limited than the arbitrary procedural capabilities of these other approaches.

Attached to the Planner-generated core at a level of detail appropriate to their action are Executor-generated secondary plans. These fall into two categories: (i) plans (indicated by + in Fig. 2) to handle normal driving perils such as stop signs and (ii) plans (indicated by *) to handle Planner suggestions on speed limits, etc. A secondary plan is activated in the same way as a core plan, except the transition is parallel in that the old plan continues to execute. Parallel transitions allow ELMER to do several things at once (e.g. slow for a stop sign and continue undertaking the core plan). During parallel execution, one plan can suggest an action which contradicts the action of another plan (e.g. both maintaining speed and stopping at a stop sign). To overcome such conflicts, a priority and an exclusion list are associated with each plan. The exclusion list indicates other plan types which cannot be executed in parallel with the plan; the priority determines whether another plan has



the authority to exclude the plan (it does if it has higher priority). In the example, the stop sign would exclude speed maintenance. When a plan is excluded by another plan, it and all of its active descendants are suspended until the excluding plan is de-activated or is itself suspended. A suspended plan can send no commands to the microworld, but transitions from it will still be made (allowing a "you-are-here" pointer (a la Kuipers [7]) to be kept even if some parallel action has suspended active processing of the core).

Suspension can be tricky since the suspension of a plan which is itself suspending other plans may allow the re-activation of some of these plans. A suspension rule is given in McCalla *et al* [8] that guarantees a unique set of suspensions providing a plan has priority no greater than the priorities of its ancestors. Since most plans having ancestors are core plans which usually don't exclude one another, this isn't a serious restriction.

4 THE MAP AND THE PLANNER

The Map contains ELMER's knowledge of the city represented mostly as routes (unlike the multiple representations of [7]). Such routes are similar in structure to plans, differing mainly in that associated with each route is the region ELMER would travel if he traversed the route. The containment relations of such regions are integrated into a region hierarchy. We feel that "instantiated" old plans will provide the main source of new Map routes. This would correspond nicely with Miller *et al*'s intuition that "probably the major source of new plans is old plans" [4, p. 177].

The Planner uses Map routes to produce a plan to solve a particular problem. If the problem is to go from region b to region d, the Planner essentially reasons as follows: if there is a Map route which connects super-region B $\supseteq$ b (in the region hierarchy) to super-region D $\supseteq$ d, use this route as the skeleton for the desired plan; else look for a route connecting B to some region C and another route connecting C to D and splice them together to form the skeleton; else give up. Converting a skeletal plan into an actual plan may involve the further elaboration of parts of the skeleton (as well as adding recommendations for Executor *-type plans). Such elaboration is done on the earliest portions of the plan first so that ELMER can start executing the plan as quickly as possible (contrast this to Sacerdoti's [9] planner which produces the whole plan before any execution is considered). More details of the Map and Planner can be found in [8].

5 CONCLUSION

The two main contributions of this research are the ability to handle a dynamic world (via secondary plans attached to the core plan) and the uniformity of representation of plans and routes which not only allows a route splicing planning methodology but may also allow ELMER to add new routes built from old plans. This latter aspect is currently under investigation as are problems of error recovery and the need to discover the limits of this approach.

ACKNOWLEDGEMENTS

We would like to thank Robin Cohen and Hector Levesque for their contributions to this research. The support of the Natural Sciences and Engineering Research Council of Canada through grant A3323 is also acknowledged.

REFERENCES

- [1] Hayes, P. J. "A Representation for Robot Plans", *Proc. IJCAI-75*. Tbilisi, U.S.S.R., 1975, pp. 181-188.
- [2] Chien, R. T. and S. Weissman. "Planning and Execution in Incompletely Specified Environments", *Proc. IJCAI-75*. Tbilisi, U.S.S.R., August, 1975, pp. 169-174.
- [3] Hendrix, G. G. "Modeling Simultaneous Actions and Continuous Processes." *Artif. Intell.* 4 (1973) 145-180.
- [4] Miller, G. A., E. Galanter, and K. H. Pribaum. *Plans and the Structure of Behaviour*. New York, N.Y.: Holt, Rinehart & Winston, 1960.
- [5] Sussman, G. J. "A Computational Model of Skill Acquisition." AI Lab. AI-TR-297, MIT, Cambridge, Mass., 1973.
- [6] Fikes, R. E. and N. J. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving." *Artif. Intell.* 2 (1971) 189-208.
- [7] Kuipers, B. J. "Representing Knowledge of Large Scale Space." AI Lab. AI-TR-418, MIT, Cambridge, Mass., 1977.
- [8] McCalla, G. I., P. Schneider, R. Cohen, and H. Levesque. "Investigations into Planning and Executing in an Independent and Continuously Changing Microworld." Dept. of Comp. Sci. AI Memo 78-2, U. of Toronto, Toronto, Ont., 1978.
- [9] Sacerdoti, E. *A Structure for Plans and Behaviour*. New York, N. Y.: Elsevier, 1977.