SIMULATION OF EXECUTING ROBOTS IN UNCERTAIN ENVIRONMENTS*

by

L. Siklóssy & J. Dreussi

TR-16

May 1973

ABSTRACT

A simulated robot solves tasks in an environment which she knows only approximately. The robot is given a description of the uncertain environment and of her capabilities. From the latter, she generates procedures that are evaluated to solve tasks. As tasks are solved, the robot improves her knowledge of the environment and the efficiency with which she can solve problems.

The design used, that of an executing robot, is contrasted to the design of planning robots. It is shown that performing robots are more efficient than planning robots. In uncertain environments, planning robots are inadequate.

*Work partially supported by NSF Grant GJ-34736.

1. Introduction - Robot Simulation.

The past two years have seen an increased activity in robot simulation work[1-5]. Given an axiomatization of the capabilities of the robot, and a description of initial and final states of the world, a solution to the robot problem is a sequence of elementary moves by the robot which would transform the initial state of the robot into the final state, or into a state satisfying some terminating conditions. However, until now the work has been restricted to robots which know all about their capabilities, and know everything about their environment. To make robot problem solving more realistic, it is necessary to lift at least the last restriction: in this article, we shall be interested in robots trying to solve problems in worlds which they do not know perfectly.

Section 2 describes past approaches to robot problem solving--<u>robot planning</u>--in known environments, and serves as an all too sketchy introduction into the area. The deficiencies of robot planning in insufficiently known environments are exhibited in section 3. Section 4 describes an alternative design for a robot problem solving system, which we have called an <u>executing robot</u>. This alternate design has some intrinsic advantages over robot planning even in perfectly known worlds, and lends itself readily to an adaptation to robot problem solving in environments that are not known perfectly. The design is implemented in a computer program called AMINA. Aspects of her implementation are described in section 5; some of her performances are mentioned in section 6.

2. Robot Planning - Some Past Approaches.

As in references [1] to [5], we consider worlds consisting of sets of predicates such as AT(BOX1 A1). The world can be changed by applying to it an operator. An operator can be applied to a world only if the world satisfies the preconditions of the operator. The changes of the world as a result of the application of the operator result from deleting from the world the delete set of the operator, then adding to the resultant world the add set of the operator. For example, to water a plant, the operator IRRIGATE may be used. To IRRIGATE(TERRARIUM), the robot must be near the TERRARIUM, and have with her a pail full of water. After she has watered the TERRARIUM, the statement (FULL PAIL) is deleted from the representation of the world, and the statement (EMPTY PAIL) is added. (The above is not a complete description.)

Two conflicting approaches to robot planning, those that emphasize generality[1] at the cost of performance-when programs are written to solve problems in various robot worlds- and those that emphasize performance at the cost of generality-when specialized programs are written to solve problems in a particular robot world, as in [6]- have recently been synthesized in a robot -LAWALY[5]- which combines the strengths of both approaches. As in the first approach, LAWALY accepts descriptions of certain types of robots and their worlds, and generates specialized programs similar to those that a human programmer would write in the second approach. The design of AMINA owes much to the work on LAWALY.

3. Robot Planners in Imperfectly Known Worlds.

Given a problem to solve, LAWALY (or another robot planner), if successful, finds a sequence of moves that she must perform to accomplish the task. However, this sequence of tasks may be of no value if the solution is based on incorrect information. For example, if LAWALY must water a terrarium, and bases her solution on the existence of the pail in some room, while in reality it is in some other room, the solution found does not solve the actual problem. Furthermore, if LAWALY does not know where the pail is at all, she cannot find a solution. The above examples illustrate two cases of insufficiently known worlds:

a) the knowledge about some fact is incorrect.

b) the knowledge about some fact is nonexistent.

In addition, it is possible that:

c) the knowledge about some fact is correct.

Of course, the robot does not know whether the positive knowledge she has (about the truth of some fact in her present world) is correct or not; nor does she know whether the negative knowledge she has (the absence of some fact in her present world) is correct or not.

4. The Design of AMINA.

4.1 Design in Perfectly Known Worlds.

Planning in a robot problem solver such as LAWALY may be viewed as a Gedanken experiment where a twin of the robot progresses along some path, realizing at various stages that some alternative paths could have been taken. If the robot reaches a dead-end, she instantly "goes back"- i.e. backtracks-to a previous alternative. She does not have to retrace her steps since no steps were really taken! In fact, steps are executed only after a total plan has been obtained. As we have seen, the execution of the plan may fail shortly after execution has begun if the plan was based on imperfect information. We notice further that some history of states must be maintained to allow the instant jump back into the Gedanken past.

By contrast, AMINA is an <u>executing</u> robot, in the sense that she should be viewed as executing every step that she takes. To return to a previous world configuration, AMINA must execute an appropriate sequence of moves. An example will illustrate the differences.

Suppose that boxes BOX1, BOX2, and BOX3 are at three different places initially, the robot being in a fourth place. The robot can <u>go</u> next to a box, and if she is next to a box, she can <u>push</u> it next to another box. (We disregard problems of sizes of boxes, etc.) Our goal is the state: NEXTTO(BOX1 BOX2)$\wedge$NEXTTO(BOX2 BOX3).

The solution by LAWALY would be the following: to the initial world W0 apply go(BOX1), to obtain world W1. Then push(BOX1 BOX2), achieving world W2. Further, to obtain NEXTTO(BOX2 BOX3), she wants to do push(BOX2 BOX3), and realizes that an alternative would be push (BOX3 BOX2). Along the first line, she does a go(BOX2) to be next to BOX2-she was only next to BOX1 before-and obtains world W3. She is then ready to push(BOX2 BOX3), but that would undo NEXTTO(BOX1 BOX2). Hence, she instantly goes back to world W2 to take the alternative line push (BOX3 BOX2). From W2 she moves to W3'by a go(BOX3), and finishes the task in world W4' obtained from W3' by push(BOX3 BOX2).

By contrast, AMINA only uses one copy of the world. Reaching W3 as above, she could return to W2 by doing go(BOX1), but that would be an extra step. Instead, she continues with a go(BOX3), followed by push(BOX3 BOX2).

LAWALY's solution has four steps: go(BOX1), push(BOX1 BOX2), go(BOX3), push(BOX3 BOX2). AMINA's solution has five steps: go(BOX1), push(BOX1 BOX2), go(BOX2), go(BOX3), push(BOX3 BOX2). Nevertheless, AMINA's solution was obtained faster: LAWALY needs to keep copies of or recreate past worlds, and thereby pays a heavy cost in overhead. Usually, AMINA solves problems about 25% to 50% faster than LAWALY.

4.2 Design in Imperfectly Known Worlds.

Since AMINA executes her moves immediately, she does not run the danger of following along a path that is inconsistent with the actual world. To simulate AMINA's behavior in imperfectly known worlds, we make the following assumptions: AMINA has direct access to her <u>model of the world</u>. This model may differ from the <u>real world</u>. As moves are executed, both copies of the world are updated. AMINA bases her actions on her model of the world. That is, to make a certain move, she will try to realize all the preconditions to this move in her model. However, if the preconditions are not realized in the real world, the move cannot be taken, and AMINA is interrupted. At that point, she explores her environment to find out what went wrong when she tried to make her move. She also explores her environment if she realizes that she is ignorant of some aspect of it. This realization is achieved in the course of problem solving through self-awareness, without resorting to additional mechanisms. Hence, the different kinds of knowledge mentioned in section 3 are treated in an entirely uniform manner! AMINA's excellent performance is probably a consequence of her unified design. A few examples will further describe her behavior.

6. Examples of Behaviour.

In this example we describe two problems solved by AMINA. Figure 1 shows the typical world in which she has operated, a world of rooms with doors, broom, pail, water faucet, etc.

6.1. Task 1: NEXTTO(BOX1 BOX2).

AMINA is asked to obtain NEXTTO(BOX1 BOX2). In her model, there is no knowledge of the positions of either box, which are in reality in two different rooms, the robot being in a third room.

To achieve the goal, AMINA decides to push BOX1 next to BOX2: push(BOX1 BOX2). To do that, she must know where BOX2 is, i.e. bind the variable rm in (INROOM BOX2 rm). Looking at her model, she finds no instance of (INROOM BOX2 x), so decides to accomplish the subtask (INROOM BOX2 rm), which can be done by carrying BOX2 to rm, i.e. carryob(BOX2 rm). To carry BOX2, she must be (HOLDING BOX2), which is not true in the world. Hence, she decides to pickup(BOX2). As before, she hopes to accomplish this result by carryob(BOX2 rm), but at that point realizes that she is in a <u>loop</u>: to be able to do something, she must do the identical thing! (Loops could also be detected on trying to achieve identical preconditions.)

Having detected the loop, AMINA decides that she should explore her environment to locate (INROOM BOX2 rm). She decides to explore the various rooms. These subgoals are solved by recursive calls to the problem-solver!

Once in a new room she "discovers" whether BOX2 is there: in the simulation, this is accomplished by allowing her access to the real world (i.e. simulation of seeing, etc. facts about BOX2). The access to the real world can be viewed as a simulation of some sensory device that would allow the robot to "see" inside a room.

Eventually, AMINA finds BOX2. When she wants to push BOX1 next to BOX2, she discovers , in the same manner as above, that she needs to explore the world to locate (INROOM BOX1 rm). She succeeds, and finally accomplishes the task, which, in a particular configuration required 18 steps and was solved in 14.8 seconds in interpreted LISP on the CDC 6600.

6.2. Task 2: WATERED(TERRARIUMA).

To irrigate TERRARIUMA, AMINA needs a full pail. An empty pail can be filled at the faucet. AMINA thinks the pail to be full, while in fact, it is empty*; she also thinks the faucet is in one room, while in reality it is in another.

AMINA goes to the pail, carries it to the TERRARIUMA and thinks she is ready to water it since in her model all conditions for watering are satisfied. But since they are not satisfied in the real world, the watering process is interrupted! AMINA then realizes that she cannot water, and therefore that some of the conditions necessary for watering are not <u>really</u> satisfied. She then explores the world to verify each of the conditions to watering. The exploration mechanism is identical to the one mentioned in Task 1: when a fact is not certain, it is assumed not known, and an attempt is made to find out about it in the real world. AMINA finds out that the pail is not full**,

---

*The pail is made of lead, has a cover, and is so heavy that the weight of the water would not be noticeable.

**Perhaps by lifting the heavy lead lid, who knows?

and decides to fill it at the faucet. Carrying the pail to the faucet, through a maze of doors, she is ready to place the pail under the faucet, but is interrupted again: the faucet is not there! She does eventually find the faucet after further exploration, fills the pail, takes it to TERRARIUMA - which, luckily enough, is still where it was! - waters TERRARIUMA, and joins a union. In a particular configuration, the total solution required 42 steps, and was solved in 32.7 seconds.

AMINA assumes that, if perhaps her knowledge of the world is imperfect, her moves have known results. She can then mark as "known for sure" facts in her model which are the direct result of some move, or were preconditions to an operator which was executed. In her exploratory behaviour to find the cause of an interrupt, AMINA does not verify the facts which she knows for sure, thereby diminishing the amount of work she has to do.

The problem of a robot which does not know exactly what she does appears significantly more difficult than the problem we have described here, and is presently under investigation.

7. Conclusions.

AMINA is a robot simulation program based on the design of an executing robot. She can operate in both perfectly and imperfectly known worlds. From an axiomatic description of her capabilities, she generates procedures which she will evaluate to solve given tasks. At any one time, AMINA only stores one copy of the world.

In imperfectly known worlds, AMINA handles true, false, incorrectly known and unknown facts about the world in a uniform manner. She determines when she needs to explore the world to discover more about it. The exploration is simulated by giving her partial access to the real world. The simulation takes into account the impossibility of making some move in reality, even though AMINA may think that the move is feasible. As AMINA is given tasks in an environment, her knowledge of the environment grows, both in quantity and in quality - "quality" here being the certainty with which she knows facts about the environment. -

8. References.

[1]Fikes, R. E. and Nilsson, N. J. "STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving," Artificial Intelligence, 2, 189-208, 1971. (Review: Computing Reviews, 13, 5, 216-217, 1972.)

[2]Fikes, R. E. et al., Learning and Executing Generalized Robot Plans. Technical Report 70, Stanford Research Institute, July 1972.

[3]Siklóssy, L. Modelled Exploration by Robot. Technical Report 1, Computer Sciences Department, University of Texas, Austin, 1972.

[4]Siklóssy, L. and Roach J. "Proving the Impossible is Impossible is Possible: Disproofs based on Hereditary Partitions," Proc. Third International Joint Conference on Artificial Intelligence, Palo Alto, California, 1973.

[5]Siklóssy, L. and Dreussi, J. "An Efficient Robot Planner which Generates its Own Procedures," Proc. Third International Joint Conference on Artificial Intelligence, Palo Alto, California, 1973.

[6]Winograd, T. Procedures as Representation for Data in a Computer Program for Understanding Natural Language, Technical Report AI TR-17, M.I.T., 1971.

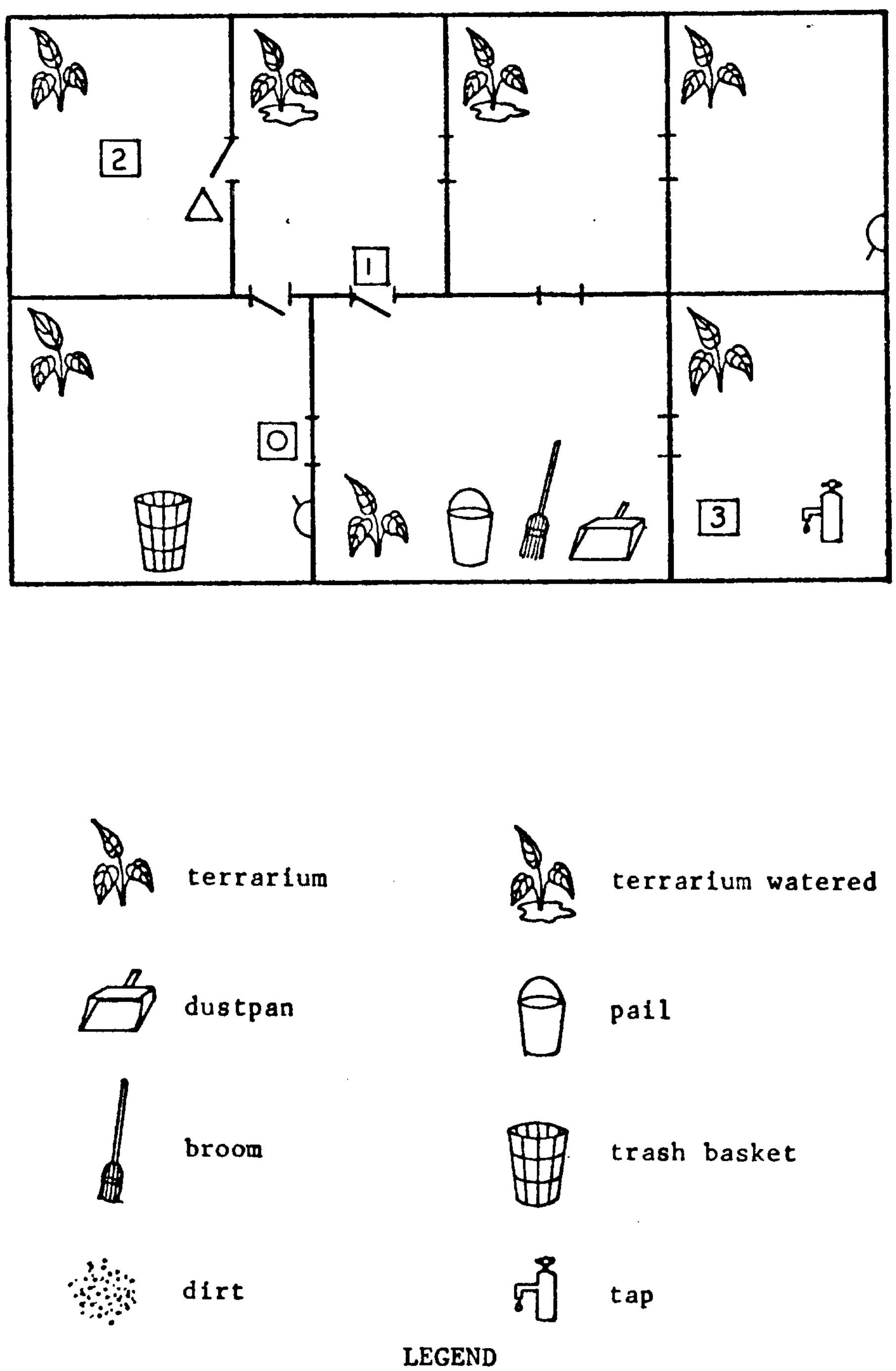
2
1
O
3
terrarium
terrarium watered
dustpan
pail
broom
trash basket
dirt
tap
LEGEND

Figure 1