IN-63-TM

91489

P.10

A Design Rationale for NASA TileWorld

ANDREW PHILIPS
KEITH SWANSON
MARK DRUMMOND
JOHN BRESINA

AI RESEARCH BRANCH, MAIL STOP 269-2
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035
(415) 604-6527

(NASA-TM-107882) A DESIGN RATIONALE FOR NASA TILEWORLD (NASA) 10 p

N92-26627

Unclas
G3/63 0091489

NASA Ames Research Center

Artificial Intelligence Research Branch

Technical Report FIA-91-04

April, 1991

A Design Rationale for NASA TileWorld

Andrew B. Philips*, **Keith J. Swanson**,
Mark E. Drummond*, **John L. Bresina***†

NASA Ames Research Center
Mail Stop: 244-17
Moffett Field, CA 94035

Abstract

Automated systems that can operate in unrestricted real-world domains are still well beyond current computational capabilities. This paper argues that isolating essential problem characteristics found in real-world domains allows for a careful study of how particular control systems operate. By isolating essential problem characteristics and studying their impact on autonomous system performance, we should be able to more quickly deliver systems for practical real-world problems. For our research on planning, scheduling, and control we have selected three particular domain attributes to study: *exogenous events*, *uncertain action outcome*, and *metric time*. We are not suggesting that studies of these attributes in isolation are sufficient to guarantee the obvious goals of good methodology, brilliant architectures, or first-class results; however, we *are* suggesting that such isolation *facilitates* the achievement of these goals. To study these three attributes, we have developed the *NASA TileWorld*. In this document, we describe the NASA TileWorld simulator in general terms, present an example NASA TileWorld problem, and discuss some of our motivations and concerns for NASA TileWorld.

*Affiliated with Sterling Federal Systems
†Also affiliated with the Computer Science Department at Rutgers University

Introduction

The world around us is replete with activities that may be characterized by such terms as "dynamic", "resource limited", "unpredictable", "time critical", etc. These problem characteristics are intrinsic to many task domains, such as automated remote exploration and process control. Robust, reliable, intelligent automated systems that can operate in such complicated domains are still well beyond current computational capabilities.

Isolating essential problem characteristics or *attributes* found in real-world domains allows for a careful study of how particular control systems operate. For our research on planning, scheduling, and control we have selected three particular domain attributes to study: *exogenous events*, *uncertain action outcome*, and *metric time*. *Exogenous events* are those events not under the system's direct control. By *uncertain action outcome*, we mean that the effects of an action taken by the system can not be identified uniquely. By *metric time* we refer to temporal properties of the domain or task. We are not suggesting that studies of these attributes in isolation are sufficient to guarantee the obvious goals of good methodology, brilliant architectures, or first-class results; however, we *are* suggesting that such isolation *facilitates* the achievement of these goals. Working on a real-world problem has obvious benefits, but to understand the systems that we build, we must isolate attributes and carry out systematic experimentation.

To study these three attributes, we have developed the *NASA TileWorld* [2]. These attributes have been captured in NASA TileWorld with simple and parsimonious domain semantics. In this document, we describe the NASA TileWorld in general terms, present an example NASA TileWorld problem and discuss some of our motivations and concerns for NASA TileWorld.

Domain Attributes

During a brain storming session at the 1990 Benchmarks and Metrics Workshop [1], numerous problem and domain attributes were suggested by the participants, such as multiple-agency, time stress, exogenous events, predictability, optimality, incomplete domain knowledge, informability, versatility, geometric reasoning, sensor/effector reliability, opportunities for learning, inter-agent communication, etc. The workshop discussion amply demonstrated that there is no coherent terminology for precisely describing these attributes.

Lack of a precise language aggravates the already difficult task of attempting to characterize what makes problems hard, and thus, on what area to focus our research. We offer no general solution to this attribute description problem. Our approach is to take a first cut at what appear to be good descriptive terms for domain attributes and to refine our definitions with experience. In service of this approach, NASA TileWorld allows an experimenter to study exogenous events, action outcome uncertainty, and metric temporal properties. Certainly, these first definitions provided by NASA TileWorld will undergo significant change; however, we see no other way to eventually settle on a coherent terminology: generate-and-test appears to be our only search strategy at this early stage.

NASA TileWorld

NASA TileWorld represents a spectrum of domains involving a grid of cells, a set of tiles, and a set of agents which can grasp and move tiles. Points along the spectrum vary in terms of tile characteristics, agent capabilities, single agent vs. multi-agent, grid topology, and the underlying physics of the grid.

NASA TileWorld is historically related to the sliding tile domain developed by N.S. Sridharan, C.F. Schmidt, and J.L. Bresina (reported in [5]). In the Summer of 1989, Bresina sketched the initial design of the NASA TileWorld domain; this sketch was refined by Bresina, Philips, Mark Drummond, and Mark Boddy to form the simulator specifications. The implementation of the specifications was carried out by Philips. Around this same time period, other related simulated domains were developed independently; e.g., the tileworld at SRI [4] and Sutton's gridworld [6]. Though similar in name, these three simulated domains are rather different in nature.

The NASA TileWorld simulator[‡] encodes a particular range in the NASA TileWorld domain. The simulator is a two-dimensional grid of cells populated with movable tiles and a single mobile agent (see Figure 2). The grid is oriented with North as up, East to the right, etc. The agent can grasp tiles in adjacent cells in the four compass directions, release a grasped tile, and move one cell at a time in a given compass direction. The agent can sense its location, determine whether it is grasping a tile, sense the contents of any cell regardless of distance or line-of-sight "obstructions", and request the current world time. The table in Figure 1 summarizes the agent commands.

The simulator has three types of commands: *interaction*, *display*, and *customization*. The first type is for agent control, and the latter two types are for experimenter control. The interaction commands allow an agent controller to sense the world state and operate the agent. Display commands allow the experimenter to have access to and modify presentation of the NASA TileWorld display. Customization commands allow the experimenter to create a NASA TileWorld problem instance, tune simulator parameters (e.g., agent movement speed), adjust the behavior of exogenous events (e.g., the frequency and other characteristics of the winds), and introduce action outcome uncertainties in the behavior of the agent's effector actions (e.g., to make the agent sometimes "veer" off course or "drop" a tile).

As mentioned above, NASA TileWorld has been created to permit study of three specific domain attributes: *exogenous events*, *uncertain action outcome*, and *metric time*. The *ex-*

[‡]NASA TileWorld is written in Franz Allegro Common Lisp and is available for public use. Email requests for copies of the code or manual should be sent to "tileworld@ptolemy.arc.nasa.gov".

Effectors		Sensors	
`grasp`	*compass-direction*	`attached`	*compass-direction*
`release`	*compass-direction*	`in`	x,y
`move-agent`	*compass-direction*	`my-location`	
		`world-time`	

Figure 1: NASA TileWorld Agent Actions

ogenous events in NASA TileWorld are gusts of wind which blow from the borders towards the interior. A wind acts on a single column or row, has a range, and has a period. A wind blows a tile along a clear path and that tile stops when it either encounters another object or is blown to the limit of that wind's range (see the next section for an example).

Uncertain action outcome is realized in NASA TileWorld in two ways. First, a probabilistic model of alternative action outcomes can be specified; for instance, it is possible to say that when the agent attempts to grasp a tile, 80% of the time the tile is grasped, 19% of the time the command has no effect, and 1% of the time any other tiles being grasped are dropped. Second, because calls to the effectors return no information about the success, failure, or duration of an action, the agent controller needs to sense the world to determine if (or when) the action has achieved the intended effects.

Metric time is an aspect of the simulator's operation as well as the domain tasks that can be posed. Since the simulator characterizes the evolution of an environment over time, an experimenter should be able to influence its metric temporal properties. In NASA TileWorld, an experimenter can tune the following metric temporal properties of the simulator: a wind's velocity, a wind's period, and an agent's velocity.[§] In our study of the temporal properties of tasks, we have concentrated on *goals with temporal extent*. Examples of such goals include: maintain (or prevent) some property over an interval of time, and achieve (or destroy) some property by a time deadline; these goals are in contrast with the goals of "achievement", without deadlines, typically used in classical AI planning. To allow a controller to evaluate its progress with respect to temporal goals, the simulator provides a "clock" sensor.

NASA TileWorld provides a simple but effective means for testing the behavior of controllers or problem-solvers on "real-time" problems. There is considerable debate on what constitutes a real-time problem, and we do not propose to resolve the issue here. However, it seems clear that exogenous events, action outcome uncertainty, and temporal dependence all play an important role in real-time problems.

[§] There is currently no facility for making the errors of action execution vary as a function of time.

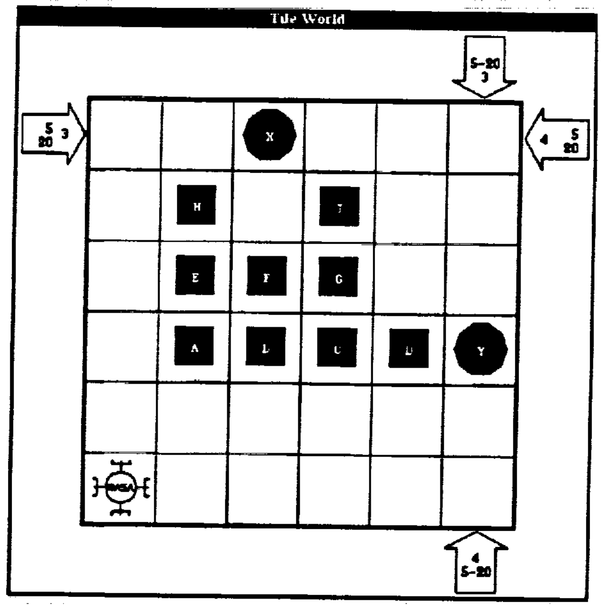

Figure 2: The Windy Maze

The Windy Maze

Figure 2 shows a NASA TileWorld graphics window which contains a sample problem we call the Windy Maze. The agent is located in the lower-left corner of the grid; tiles are distributed throughout the grid. The arrows positioned along the borders of the image describe winds that can blow tiles along the row or column to which they are pointing.

Consider the arrow in the upper-left corner. The 3 indicates that the wind has a range of three from the border cell and moves a tile to the fourth cell in the top row, provided that there are no obstructions. So, a tile in the upper-left corner of the grid would be blown 3 grid cells to the right by that wind. Notice that the octagon marked X would only be blown one cell to the right by that wind due to the wind range limit of 3. A wind's *period* is the time in seconds between successive gusts. In this example, the period of the wind fluctuates randomly within the [5..20] interval.

In the Windy Maze problem, the agent must move from the lower left corner $(0,0)$ to the upper right corner $(5,5)$ without using the grasp or release commands to move any tiles. At first glance it may appear to be impossible to achieve this goal, since there is no clear route from the agent's current location to the goal. Once the winds are taken into account, one can see that two different "paths" are possible. When the winds are set in motion the octagon marked X slides back and forth in the top row between the cell above square H and the cell above square I. Also, the octagon Y slides up and down the far right column going no lower than its current position and no higher than one row below the top. Given these exogenous events, there are two possible paths by which the agent can move to the upper-right corner. The first option is to move around the D tile when Y slides out of the way. The second option is to temporarily park between tiles H and I when X is above I. Once tile X has blown to the west, the agent has a clear route through to the goal.

Our experience with this specific problem has indicated that it is not a "puzzle" in the classical sense of the word. When a person is given the problem, they almost immediately bring common sense knowledge to bear and reason that they must interleave their actions with the occurrence of exogenous events. While this problem is apparently simple for the average person, we feel that it is beyond the capability of most current "real-time" reasoning systems.

For example, let us assume the controller has been given the goal of having the agent at location $(5, 5)$ by 8:27:00, and the problem is presented at 8:22:49. No matter how a given controller "solves" the problem, there is only four minutes and eleven seconds available. Some of this time is inevitably used by the actions needed to move the agent from $(0, 0)$ to $(5, 5)$. The exact amount of time depends on the speed of the agent and on the specific behavior of the winds during this given problem instance. The controller is free to allocate the available time in whatever manner it deems fit. Many types of agent controllers and problem solving methods can be imagined. But, the specific agent controller and method it uses to accomplish the given goal is not important. What *is* important is how the controller's performance is evaluated.

How can controllers be compared? By what metric can we judge the performance of a controller? One possible way to judge performance is to see if the agent is in cell $(5, 5)$ at 8:27:00. If not, the agent controller simply fails to satisfy the goal. Although this metric is easy to evaluate, its binary distinction does not supply very much information about how "well" the controller performed. Clearly, the specific metric used reflects an experimenter's scientific objectives. The selection of metrics is a complicated and interesting issue and a possible source of future work.

Discussion

Selecting relevant domain attributes and defining a simplified problem containing those attributes facilitates careful analysis of a controller. NASA TileWorld allows precise specification of problems that exhibit a selected subset of what might be considered "interesting" attributes. Simple domain semantics facilitates analysis and discussion of problems while still retaining some of the essential, challenging attributes found in many real-world tasks. The simplicity of the NASA TileWorld simulator facilitates systematic study of the *reasons* for performance. Most real-world problems have so many interconnected attributes that it is hard to isolate and analyze the underlying reasons for success and failure.

While NASA TileWorld provides a simulation environment for studying specific domain attributes, the NASA TileWorld domain can appear overly simplistic. However, a simulated environment that *appears* more realistic, (e.g., a simulation of autonomous agents responding to forest fires [3]) may, in fact, be even more simplistic. Usually, only the designer of the simulator has a solid grasp of its real complexities. Given a description of a system's success with the simulated version of a task, it is dangerously easy for the unwary reader to infer that the system "solves" the simulation's real-world analog, when, in fact, such is not the case.

An additional worry about our approach is that by isolating attributes for scientific study and removing essential information, we might significantly alter the original problem. By removing information, the simplification *may* lead to artificially hard problems; that which was simple in the original task can become problematic in the abstracted version. For instance, one might argue that it is the sterile simplicity of the classical blocks world that makes it so difficult. It can also be argued that by isolating attributes from a given problem domain, those attributes are altered in some significant way and thus, made easier to manage. In this case, what might have been a real problem in the original task can disappear in the abstracted version. Identifying relevant and irrelevant information in a real-world task is a difficult problem, and it is our belief that we will make progress only by exploring some of the myriad possible variations.

Conclusions

There is an important role for simple simulators that allow one to study selected domain attributes while jettisoning irrelevant semantic baggage. We have designed and implemented the NASA TileWorld simulator with this in mind. NASA TileWorld allows one to study problems that involve certain types of *exogenous events*, *action outcome uncertainty*, and *metric time*. There are many problems and domain attributes that cannot be expressed in this simulator. However, we feel that this simulator, viewed as a single element in an array of available tools, represents a simple and useful mechanism for systematically studying the underlying reasons for system performance. The NASA TileWorld domain is easy to describe and modify, and thus, can facilitate communication among researchers. The simulator itself can also help foster more precise empirical comparison between various approaches. We are the first to admit that the NASA TileWorld simulator is not the last word in comprehensive simulation environments. However, we expect that the NASA TileWorld domain, and others like it, will help in the construction of a common vocabulary of problems and domain attributes.

Acknowledgements

Thanks to Smadar Kedar and Rich Levinson for help with the design of NASA TileWorld. Thanks to Smadar, Rich, and John Allen for comments on a draft of this paper.

References

[1] Drummond, M., & Kaelbling, L. 1990. Integrated Agent Architectures: Benchmark Tasks and Evaluation Metrics. *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Diego, CA. pp. 408-411. Morgan-Kaufmann, San Mateo, CA.

[2] Philips, A.B., & Bresina, J.L. 1991. *NASA TileWorld Manual* (NASA Technical Report TR-FIA-91-04). Moffett Field, CA: NASA Ames Research Center, Code FIA.

[3] Cohen, P.R., Greenberg, M.L., Hart, D.M., & Howe, A.E. 1989. Trial By Fire: Understanding the Design Requirements for Agents in Complex Environments. *AI Magazine*, 10(3): 32-48.

[4] Pollack, M.E., & Ringuette, M. 1990. Introducing the Tileworld: Experimentally Evaluating Agent Architectures. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 183–189), Menlo Park, CA: AAAI Press.

[5] Sridharan, N.S., and Bresina, J.L. 1984. *Exploration of Problem Reformulation and Strategy Acquisition - A Proposal* (Rutgers Technical Report RU-LCSR-TR-53; RU-CBM-TR-137). New Brunswick, NJ: Rutgers University, Department of Computer Science.

[6] Sutton, Richard S. 1990. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynmaic Programming. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 216-224), San Mateo, CA: Morgan Kaufmann Publishers.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Dates attached

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

Titles/Authors - Attached

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Code FIA - Artificial Intelligence Research Branch
Information Sciences Division

8. PERFORMING ORGANIZATION REPORT NUMBER

Attached

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Nasa/Ames Research Center
Moffett Field, CA. 94035-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Available for Public Distribution

Peter Friedland 5/14/92 BRANCH CHIEF

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Abstracts ATTACHED

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT