

# Rational-Based Visual Planning Monitors

Zohreh Alavi

Wright State University

alavi.3@wright.edu

## 1 Overview

The ability to act and respond to exogenous events in dynamic environments is crucial for robust autonomy. In dynamic environments, external changes may occur that prevent an agent from reaching its goal(s). I am interested in the design of reasoning and planning components operating in environments that undergo changes in real time. My goal is to develop a framework for fully integrated planning, execution and vision in dynamic environments.

In the initial phase of this work, we have concentrated on the problem of enabling a planning system to deal with relevant changes in the environment during planning time. We introduce a new system for planning in a world under continuous change in an agent with visual perception. Our main contribution is to make vision sensitive to relevant changes in the environment that can affect an agent plans. We applied a rational-based monitor technique [Veloso *et al.*, 1998] to the SHOP Hierarchical Task Network (HTN) planner [Nau *et al.*, 1999]. We modified SHOP to generate plan monitors to interact with a vision system and react only to those environmental changes that bear on current planning decisions. Thus when the monitors detect any relevant changes, corresponding plan transformation are executed as needed.

Rationale-based monitors provide a means of focusing visual attention on features of the world likely to affect the plan. When a feature being monitored changes, and the change is detected, we say that the monitor fires. Deliberation can then be performed to decide whether the plan under construction should be changed. If the planner decided to attend to the detected changes of the world state, it will perform a plan transformation. In particular, parts of the plan may be deleted because they have become unnecessary; new tasks may be added and current ones refined, and prior decisions about how to achieve particular goals may be revisited. Originally monitors were implemented in the state space planner Prodigy [Veloso *et al.*, 1998], our work differs in using these monitors in the SHOP HTN planner.

We have added our extended SHOP planner in the planning phase of a cognitive architecture named MIDCA [Cox *et al.*, 2016]. The meta-cognitive, integrated dual-cycle architecture (MIDCA) consists of "action-perception" cycles at both cognitive level and the meta-cognitive level. A cycle selects a goal and commits to achieving it. The agent then creates a plan to achieve the goal and subsequently executes

the planned actions to make the domain match the goal state.

MIDCA communicates with a Baxter humanoid robot to accomplish a goal in a dynamic environment using the monitors to focus vision and adapt plans. We have added an interface to MIDCA to communicate with ROS and the Baxter. It is responsible for sending messages to ROS as requested by MIDCA, and for placing messages received in appropriate queues for MIDCA to process. During the perceive phase, these messages will be accessed and stored in MIDCA's main memory. The interpret phase is responsible to reason about these messages and also create world states which are represented symbolically as logical predicates. Each monitor hires a perception node that is running asynchronously, which guides vision to focus on a specific features of the world.

Additionally, we have run experiments in the blockworlds domain. Initial results show that planning with rationale-based monitors can reduce the total planning time when the world changes.

## 2 Implementation

We have implemented rational-based monitors within the SHOP planner. Algorithm 1 shows the overall procedure. SHOP is an HTN planning algorithm which creates plans by recursively decomposing tasks into smaller subtasks until it reaches the primitive tasks which can be accomplished directly. SHOP uses methods and operators. An operator specifies a way to perform a primitive task, and a method specifies a way to decompose a non-primitive task into a set of subtasks.

To integrate with rational-based monitors, two parts are added to the SHOP planner. First, the monitors are generated when a primitive subtask is added to the set of subtasks (step 14 in the algorithm). Second, at each cycle of planning sensing is performed to see if any monitor fires and if so, a plan transformation is done in response (step 2 in the algorithm).

### 2.1 Monitor Generation in SHOP

New monitors can be generated whenever a primitive subtask is added to the list of tasks to accomplish. The monitors are generated to observe those features of the world that led us to choose that subtask.

At each planning cycle, the SHOP planner checks for fired monitors. If a monitor fires, the planner goes back to the point which the related subtask was chosen and refines the plan.

---

**Algorithm 1** SHOP with Rationale-based Monitors

---

```
1: procedure SHOP(s, T, D)
2:   check for fired monitors if any, keep the plan and
   backtrack to the the level the first fired monitor was gen-
   erated
3:   if T = nil then
4:     return nil
5:   end if
6:   t ← the first task in T
7:   T ← the remaining tasks in T
8:   if t is primitive (i.e. there is an operator for t) then
9:     nondeterministically choose an operator o for t
10:    P ← SHOP(o(s), T, D)
11:    if P = Fail then
12:      return Fail
13:    else
14:      generate monitors for precond(o)
15:    end if
16:    return cons(o, P)
17:  else if there is a method applicable to whose precon-
  ditions can all be inferred from S then
18:    nondeterministically let m be such a method
19:    return SHOP(s, append(m(t, S), U), D)
20:  end if
21: end procedure
```

---

### 3 Experiment

To evaluate the performance of our approach in planning, we ran the system in a modified blocksworld domain. The goal of this experiment is to examine the benefit from using monitors to improve planning in a dynamic environment. In this experiment, we changed the world state that made the planner jump to a different partial plan.

We added the possibility that blocks could catch on fire and before any block was picked up, the fire should first be extinguished. To use an extinguisher, it first needed to be taken out of the box, which is considered a block. If the box was not clear, the planner generated a plan to make the box clear. Furthermore, there were additional actions available to our agent allowing it to deal with these refinements. The three new types of actions are as follows:

**put-out-fire**( $B_a$ ) If  $B_a$  is on fire, extinguish  $B_a$

**get-extinguisher**(*ext*,  $B_b$ ) if the  $B_b$  is clear, take out the extinguisher *ext* from  $B_b$

**make-box-clear**( $B_b$ ) if  $B_b$  is not clear, unstack all blocks on top of  $B_b$

In each planning problem we set the initial state to be one with a block,  $B_a$ , on fire, a separate tower with  $B_b$  as its bottommost block, and a fire extinguisher, *ext*, inside  $B_b$ . The goal is to holding( $B_a$ ). By varying the height of the tower, we can vary the complexity and length of the solution. For example, if the height of tower is 5, the planner has to unstack and putdown 4 blocks in order to obtain the fire extinguisher from block  $B_b$  and use it on block  $B_a$  before it can pick up  $B_a$ . If the fire goes out during the planning process, the planner will jump to pickup( $B_a$ ). Here, the purpose of monitoring is

to observe such a change as the fire going out, and suggest a jump to the shorter plan.

In this experiment, we varied the height of tower,  $n$ , from 4 to 26. During planning, the monitor which observes the state of onfire( $B_a$ ) detects the change and lead to plan refinement. We vary the time at which this monitor fires during the planning process, namely after 0, 10, 30, 50 planning steps.

Our initial result shows when the environment does not change, the amount of time increases with  $n$ . However, with the rationale-based monitors, the planner can react to the state changes and find a plan faster. As would be expected, when the changes occur later, the savings benefit of the planner is reduced, because it has already performed significant planning.

### 4 Future Work

There are many promising avenues for future work. First, we plan to examine the benefit of using these monitors during the act (execution) phase. This approach could allow the agent to respond to unexpected changes during execution (i.e., after planning has already succeeded). This helps the agent to focus only on what is important.

In our current work, we detect changes during planning time using rationale-based monitors. In doing so, we interleave perception and planning. This suggests that cognitive tasks may benefit from calling other cognitive tasks and/or changing the order of in which cognitive processes operate. Exploring this idea further is another avenue of future work. In this modified cognitive architecture, different phases will be able to call each other as needed. We also suggest that interpreter should be running in parallel with all phases. Whenever a new change is detected, the interpreter will perform reasoning to make decisions on how to react to detected change. These changes may result in the consideration of new plans or alter the agent's intentions regarding its own reasoning processes. Some changes may cause the system to replan many times, but the better solution might be to change focus completely and pursue new goals. The interpreter will decide which solution should be taken in response to the new changes.

**Acknowledgments:** This work was supported in part by grants NSF 1217888 and ONR N00014-15-1-2080

### References

- [Cox *et al.*, 2016] Michael T Cox, Zohreh Alavi, Dustin Dannenhauer, Vahid Eyorokon, and Hector Munoz-Avila. Midca: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In *AAAI*, 2016.
- [Nau *et al.*, 1999] Dana Nau, Yue Cao, Amnon Lotem, and Héctor Muñoz-Avila. SHOP: Simple hierarchical ordered planner. In *Proceedings of the 16th IJCAI-Vol. 2*, pages 968–973. Morgan Kaufmann, 1999.
- [Veloso *et al.*, 1998] Manuela M Veloso, Martha E Pollack, and Michael T Cox. Rationale-based monitoring for planning in dynamic environments. In *AIPS*, pages 171–180, 1998.