Integrating Information, Planning, and Execution Monitoring Agents

Manuela Veloso Tucker Balch Scott Lenser
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891

1 Introduction

In this paper, we report new work illustrating the integration of information agents, such as Web agents, with planning, and execution monitoring of multiple physical agents. Information agents, external to the planner and to the execution agents, specify the planning mission. The planner generates a plan which is then executed. Web agents again are requested to monitor the plan-relevant features of the world. Replanning must occur dynamically when a failure is encountered. These techniques are combined in the CMUExpress architecture. The CMUExpress architecture demonstrates a solution to the integration of planning with real information and execution agents an *Interaction Manager*, that effectively maintains necessary links of communication and monitoring between the different sets of agents.

We consider in particular multiple execution agents that must react to unforseen events while operating in the real world. Hence, in contrast to most information tasks where the world may not change while a query is being processed, we assume that the real world changes *while* we are solving the problem. In approaching such non-trivial real world problems, we recognize that desirable universal planning solutions may be impossible to reach as the real world is impossible to model completely. Replanning is inevitable, even to support probabilistic and conditional planning. We provide an approach to replanning that allows for the incorporation of guidance to minimally disturb the plan to be refined.

2 The Scenario and the Agent Environment

The specific scenario for this experiment is a Non-combatant Evacuation Operation (NEO). Several NEO operations per year are carried out by the United States government. They usually involve extracting citizens from a crisis situation either domestically or in international locations. To test our system, a ficitious NEO scenario is used as a testbed.

Figure 1: Events in "information space" force plan revision and changes in real-world execution.

In the scenario more than 150 civilians must be located, picked up by multiple rescue vehicles and brought to a safe location. This task presents a number of challenges, including information gathering, logistics, planning and execution monitoring, and very importantly, integration and communication. The first phase of the task, locating the civilians, is handled by Ariadne, an information-gathering agent developed at ISI. We focus on the logistic planning for extracting the civilians once their locations are known. This consists of planning routes for rescue assets (passenger vans), monitoring their progress and reacting appropriately to dynamic changes in the world (e.g. road blockages).

Approximately 20 agents are integrated in the agent community developed for this experiment; agents for database lookup, webpage monitoring, translation, speech recognition, brokering and so on. Of particular intest in this system: MMM - a user interface agent developed at SRI; Ariadne - an information agent developed at ISI; CMUExpress - our suite of planning, execution and monitoring agents.

SRI's Multi-Modal Map (MMM) serves as the user interface to the system. MMM provides multimodal input, the user can interact using handwritten, vocal and gestural commands. The user's requests are processed by a distributed community of software agents (e.g. speech recognizers and information agents like Ariadne). Feedback is graphical (e.g. route maps) and textual.

To provide realism, the scenario includes potential traffic blockages as a complicating factor. Once execution of the plan begins, it is likely that one of the routes used by the passenger vans may become blocked. The system must recognize the problem, and route the rescue assets around the blockage (Figure 1). In an actual NEO mision, information regarding route blockages might be available through satellite monitoring, by other assets on the scene, or via the web. In anticipation of the move towards online resources we developed a traffic website for use in our experiment (Figure 1, left). The website is monitored by Ariadne during the execution of an evacuation plan. Blockages in routes that are being used trigger a revision in the plan.

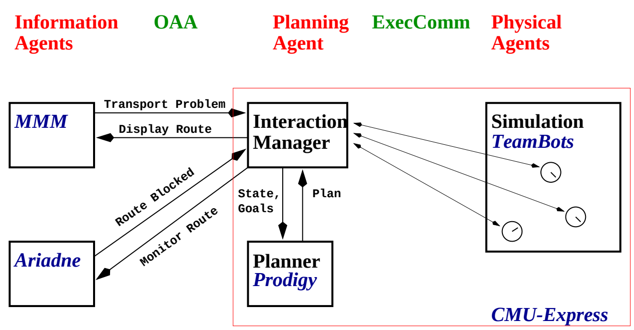

Figure 2: MMM and Ariadne, the information agents, and CMUExpress connected through OAA

We model the passenger vans as agents capable of navigating autonomously to a waypoint. The waypoints are transmitted to each van in sequence. When a waypoint is reached, we assume the agent transmits a "success" message and proceeds to the next waypoint it is told to go to. Present technology would enable this to be implemented by equipping vans with embedded computing consoles and communication equipment. Additionally, the vans may have a limited capacity, so plans for them must account for periodic trips to the destination for dropping the passengers off.

Communication between CMUExpress and the information agents used in this experiment is implemented using the Open Agent Architecture (developed at SRI). In OAA, each agent intially registers its capabilities with the OAA *facilitator*. When an agent needs a request satisfied (e.g. to find the location of a person), it sends the request to the facilitator. The facilitator then matches a request from one agent to the capabilities of another agent and automatically routes messages to the correct receiving agent. In our case, for instance, CMUExpress registers its capability to transport people from one location to another.

3 The CMUExpress Architecture

CMUExpress is composed of three main components, namely the Interaction Manager, the planner, and the simulator. Figure 2 shows these components and how CMUExpress is integrated via OAA with the other agents in the system. We address the integration problem by developing a specific component, namely the Interaction Manager, which plays the crucial role of bridging the information agents to the planner, as well as the planner to the execution agents.

In a nutshell, CMUExpress addresses the integration of planning, execution, monitoring and replanning, through the three concrete components:

- An *Interaction Manager* that acts as go-between for information agents, the PRODIGY planning agent, and physical agents that execute PRODIGY plans in the TeamBots simulator.
- PRODIGY, a generative planner and an associated NEO domain description. The formalized description of the problem enables NEO scenarios to be easily input to PRODIGY for planning.
- The TeamBots simulator for executing plans realistically. The simulation includes rescue vans, people to pickup and modelling of major traffic routes in Kuwait City, Kuwait.

An example use of the complete system illustrates the functionality of the architecture. The mission commander, using MMM begins by querying the system for the locations of civilians in Kuwait City. MMM passes the query to Ariadne, which searches its Web information sources for the answer. Once the locations of the people are ascertained, the commander can indicate (with pen-based gestures on the MMM interface) where each group of people is located. At that point, the problem is sent to CMUExpress for planning and execution.

4 Conclusion

CMUExpress demonstrates an integration of real information agents, and planning and execution agents. The architecture contributes a modular decomposition of the task with a separate Interaction Manager to handle the communication between the multiple agents. A module similar to the Interaction Manager is needed for many real problems to integrate information sources and capabilities provided by different agents. Real-world problems are automatically translated from information agents to the planning agents. Execution of multiple physical agents is monitored and the replanning occurs with domain-guided minimal plan disturbance when failures in the world are detected.