Reactive Model-based Programming of Mobile Scouts

Eric Timmons
Model-based Embedded and Robotic Systems Group
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology

Introduction

There is a growing desire in many scientific and exploration communities to use cheap, small, autonomous vehicles to perform information gathering tasks. As the number of vehicles used for information gathering tasks increases from a single large, capable, often remotely operated vehicle to a collection of smaller, more specialized, autonomous vehicles, so too does the complexity of the planning problem and burden on the scientists and operators increase. When multiple vehicles are involved, there are often coordination and safety constraints that need to be accounted for and the resulting plan is frequently more complex as the vehicles need to work together.

We envision a system that would allow these information gathering agents, or scouts, to be goal-directed and interact with their human operators at a cognitive level. Such a system should have several properties. First, it should allow the operators to specify the desired behavior of the scouts directly in terms of goals on hidden state, e.g. "goal=have picture of building A", and let the scout worry about making the plan to achieve that goal with a high level of certainty. Second, it should allow operators to specify constraints at a high-level, such as "do not enter area B after 1500" or "land within one hour of launch". Third, it should allow operators to have an adjustable level of control over the system. In some circumstances the operator may know exactly what the scout should do, but in other situations, especially in situations where the operator cannot see live data from the scout, it would be better for the operator to describe a set of strategies (as macro actions) and let the scout decide when and where to apply the strategies. The ability to describe macro actions relevant to the task at hand would have the dual benefit of decreasing planning complexity and increasing operator trust in scout.

Scouts using such a system would have the potential to be indispensable tools for a wide variety of human activities. For the physical sciences, mobile robots could be used to map pollution levels in the atmosphere or sea while automatically responding to interesting or unexpected readings. During time critical missions, devices with this technology could focus their efforts on gathering the information most pertinent to mission success. For example, a UAV tasked to assist fire fighters during a forest fire could initially work on mapping the entire fire. But, if the fire fighters' escape route is cut off, the UAV's mission would change to finding a new escape route and it would accordingly change its focus to mapping the areas of the region that show the best promise for a safe path.

In this paper, we first describe a recent real-world deployment of autonomous underwater vehicles as a concrete motivating example. Then, we briefly describe how temporal plan networks can be used to model much of the problems of interest.

Scott Reef Demonstration

In March and April 2015, a team of researchers from MIT's Computer Science and Artificial Intelligence Laboratory (including the author), the Woods Hole Oceanographic Institution, the Australian Center for Field Robotics, the University of Hawaii, and the University of Rhode Island participated in a scientific cruise at Scott Reef off the coast of Australia to explore the issues that would arise, and strategies to deal with using multiple heterogeneous autonomous underwater vehicles (AUVs) in close proximity in real-world operations. Six vehicles were used for this cruise: two Ivers (Anderson and Crowell 2005) (AUV), one Sirius (AUV), one Lagrangian float (Schwithal and Roman 2009) (AUV), one Slocum glider (Jones et al. 2005) (AUV), and one wave glider (Hine et al. 2009) (autonomous surface vehicle). At the peak of operations, five vehicles were in the water at the same time, typically within 1-2 km of the crewed research vessel.

Each class of vehicle used on this cruise was equipped with a unique combination of movement and sensor package. The author worked primarily with the Slocum glider, a high endurance vehicle that moves by "flying" up and down the water column without thrusters. The other AUVs were equipped with sensors that required staying within several meters of the seabed. Every vehicle except the Slocum glider carried an acoustic modem, allowing them to be re-tasked on short notice. The Slocum glider on the other hand could only be re-tasked when it surfaced (approximately every 30 minutes).

The first half of the cruise focused on providing a risk-aware path planning capability for the glider (not discussed further in this paper). The second half was focused on moving away from the stock script based method of controlling the glider by providing both decision making capabilities and elevating the level of at which the operator could specify the mission goals and constraints.

First, the area of the reef being explored was broken down into 500m by 500m grid cells. Then, areas of interest in each grid cell were decided upon by the glider operators, along with a ranking of each area. Next, any temporal constraints needed by the operators were determined, such as an area of interest must be visited before a certain time of day. Coordination with the other vehicles was done by using the planned schedule for each of the other vehicles, provided by their respective operators, to encode further temporal constraints on when the glider could occupy specific grid cells.

The interaction with the glider operators (everything before encoding the planned schedule of the other vehicles) was done using RMPL, the Reactive Model-based Programming Language (Willams and Gupta 1999; Ingham, Ragno, and Williams 2001; Kim, Williams, and Abramson 2001). RMPL was originally developed in order to enable users to program autonomous spacecraft in a familiar way — using Java-like object oriented programming. RMPL combines plant model specification and a control program within a single program. A subset of RMPL's control program syntax can be compiled into a temporal plan network (described in the next section). In the TPN generated from the operator's specification, the decision variables represented the order in which to visit the sites of interest.

The TPN resulting from augmenting the operator TPN with the schedules of the other vehicles was then solved and the resulting plan encoded as a script executable by the glider. Every time the glider surfaced, the predicted schedules of the other vehicles (and corresponding temporal constraints) were updated and a new script generated for execution.

Temporal Plan Networks

For this work, temporal plan networks (TPNs for short) (Kim, Williams, and Abramson 2001) are used to encode the set of possible plans that can be executed. A TPN couples together a simple temporal network (Dechter, Meiri, and Pearl 1991) with a set of discrete-valued decision variables that enable and disable constraints and events. A TPN is a tuple $tpn = \langle Ev, SV, DV, Ep \rangle$.

- Ev is the set of *events*. Each event $e \in Ev$ is atomic, and has no other attributes except for its identity. Each event e is also associated with a guard condition $guard(e)$.
- SV is a set of *state variables*. Each state variable $sv \in SV$ is associated with a domain $dom(sv)$.
- DV is a set of *decision variables*, which is disjoint from SV. Each decision variable $dv \in DV$ is associated with a *finite* domain $dom(dv)$, and a guard condition $guard(dv)$.
- Ep is a set of *episodes*. An episode ep is a tuple $\langle fromE, toE, dc, sc, gc \rangle$, where:
  - $fromE$ and toE are events (referred to as the from event and the to event, respectively).
  - dc is a constraint on the duration of the episode. Formally, dc is a Boolean function from $\mathbb{R}$. We assume that all duration constraints are simple temporal constraints (Dechter, Meiri, and Pearl 1991), of the form $toE - fromE \in [lb, ub]$, for $lb, ub \in timedom$.
  - sc is a state constraint. sc describes feasible state trajectories during episode ep. Formally, sc is a Boolean function from $SV \times \mathbb{R}_{\geq 0}$.
    If sc is trivially true, then this episode is called a *temporal constraint*. Otherwise, we require that the duration is non-negative, that is $dc(x)$ is false for all $x < 0$.
  - gc is a guard condition, also referred to as $guard(ep)$, as described below.

A *guard condition*, $guard$, is a Boolean expression composed using arbitrary Boolean combinators (and, or, not) of expressions of the form $dv = v$ where $dv \in DV$ is a decision variable, and $v \in dom(dv)$.

A *candidate solution* for a TPN $tpn = \langle Ev, SV, DV, Ep \rangle$ is a pair $sol = \langle s, d \rangle$ where:

- $s : Ev \rightarrow \mathbb{R}_{\geq 0} \cup \{\perp\}$ is a partial schedule, assigning times to some of the events (where $s(e) \neq \perp$) and not scheduling other events (where $s(e) = \perp$), and
- d is a partial assignment to decision variables, assigning to each $dv \in DV$ either some value in $dom(dv)$ or $\perp$.

A candidate solution is a valid solution if all activated events and episodes are temporally consistent and a trajectory for the state variables exists that satisfies all state constraints.

References

Anderson, B., and Crowell, J. 2005. Workhorse auv - a cost-sensible new autonomous underwater vehicle for surveys/soundings, search amp; rescue, and research. In *OCEANS, 2005. Proceedings of MTS/IEEE*, 1–6.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.

Hine, R.; Willcox, S.; Hine, G.; and Richardson, T. 2009. The wave glider: A wave-powered autonomous marine vehicle. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, 1–6.

Ingham, M.; Ragno, R.; and Williams, B. 2001. A reactive model-based programming language for robotic space explorers.

Jones, C.; Creed, E.; Glenn, S.; Kerfoot, J.; Kohut, J.; Mudgal, C.; and Schofield, O. 2005. Slocum glidersa component of operational oceanography. In *Proc. 14th Int. Symp. on Unmanned Untethered Submersible Technology*.

Kim, P.; Williams, B. C.; and Abramson, M. 2001. Executing reactive, model-based programs through graph-based temporal planning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-01)*, 487–493.

Schwithal, A., and Roman, C. 2009. *Development of a new lagrangian float for studying coastal marine ecosystems*. IEEE.

Willams, B., and Gupta, V. 1999. Unifying model-based and reactive programming in a model-based executive. In *Proceedings of the 10th International Workshop on Principles of Diagnosis*.