Commentary, regarding "Automatic Planning for Autonomous Spacecraft Constellations"

Mark Boddy
Honeywell Technology Center
MN 65-2200
3660 Technology Drive
Minneapolis, MN 55418
mark.boddy@honeywell.com

1 Summary of paper

In the paper "Automatic Planning for Autonomous Spacecraft Constellations," the authors discuss a specific approach to the problem of integrating relevant lower-level concerns (primarily continuous dynamics of various sorts) with higher-level planning functions for spacecraft constellations. Despite the title, the paper does not describe a planning system, discussing instead a specific proposal for the kinds of models on which planning should be done.

The main point of the paper is that using constraint-based models, in particular CLP, for the representation and solution of high-level autonomous control problems—"planning under operational constraints," in their words—has specific benefits for spacecraft constellations. In particular, this approach permits the integrated, explicit representation of low-level, continuous constraints derived from orbital dynamics (for example), mixed discrete/continous representation for various types of resources (power, data storage space, etc.), and discrete constraints representing specific goals, or the assignment of discrete domain elements (individual spacecraft, e.g.) to satisfy some need. Furthermore, the ability to integrate these different types of constraints, coupled with the ability to compose constraint problems, means that a local solution of the planning problem for one spacecraft is simply a subset of the overall solution. Finally, there are all the usual advantages claimed for CLP approaches (modelling generality, declarative semantics, separating search from problem statement, and so forth).

This work does not involve "planning" at all, in the classical AI sense of the term. The models constructed reflect constraints upon valid plans, but there is no search through the space of valid plans, rather an assumption that any solution of the CSP will be acceptable. I do not regard that as a problem, rather as a potential misunderstanding to be cleared up, and as a place where further work is needed. The main contribution I see in this paper is that it argues for the use of CLP methods within multi-agent systems, and provides some examples to validate the proposed approach.

The authors provide three examples of constraint classes. The first class covers the expression of the dynamics of individual spacecraft, specifically motion planning in an orbital environment, with timing and positional constraints applied to specific actions (observations). The second class of constraints discussed are for coordinating multiple spacecraft within a formation, for example in coordinated orbital manouvers. The third class, called a "coordination model" by the authors, covers changes in operational modes both for single spacecraft and synchronized across multiple spacecraft. This last class looks very much like a special case of a more general expression of resource usage and interdependent operational states.

2 Related issues

There are three areas of current work which intersect with this in interesting ways, illuminating or extending some aspect of the proposed approach.

2.1 Hybrid systems

The way in which I am most sympathetic to this work is that it has as a basic thesis the notion that hybrid constraint representations should form the basis for constructing, analyzing, and executing models of the operations of complex systems.

Necessary as abstraction is, there are many places in the real world where an abstraction that moves from continuous to discrete with increasing abstraction is not helpful. Of course, there are historical reasons, especially within AI, why there has been a bias in term of abstracting towards something like a propositional representation. Furthermore, these abstractions worked, sort of, for robots wandering down hallways, because the problems were stated such that all the relevant continuous detail could be buried in the lower level, with the occasional exception of duration. While it does matter whether the robot can fit through a given door, or navigate successfully down a corridor, precise clearances are not important. Locations can frequently be abstracted to a graph of discrete locations.

For spacecraft, as in many other domains, these simplifications will cause problems. Setting a state variable corresponding to whether a spacecraft is "at" Jupiter or not is not very helpful. It would be much better to know the orbital elements, especially for planning out some sequence of operations (close enounters with natural satellites, for example). Similar arguments can be made for decision-making involving transition times (rotating an instrument platform), actions with their own continuous dynamics, or resource models involving material or energy flows.

In domains where continuous dynamics are important in this way, there can be an interesting inversion, in which the most sophisticated continuous models are actually employed at the highest level of abstraction. My imperfect understanding of mission planning for spacecraft is that the orbital models used in preflight mission planning are considerably better than any used for enroute autonomous navigation. The same inversion applies in other domains. In chemical manufacturing, for example, medium-range production planning (including planning the purchase of raw materials) usually involves the use of very detailed kinetic models of the reactions involved in chemical processes. These models are frequently tailored to individual pieces of equipment in a given plant. This level of planning abstracts away individual activities and operations (the discrete part of the problem), and concentrates on getting right the continous dynamics and resulting material balances. Once that has been done, the planning and scheduling of individual operations can be accomplished (within their own sets of continuous constraints). Finally, the individual activities are turned over to operations and executed using control models that are rarely more complex than is required to take a target setpoint (temperature, pressure, flowrate) and determine a gradient and the resulting control action.

Over the past few years, work on hybrid systems, particularly within the areas of automatic control and real-time systems, has grown and evolved to the point that it makes sense to talk about a "hybrid systems community." The results from this community have primarily been in the areas of synthesizing or analyzing controllers (or "reactive systems," depending on your background) for particular problem domains. While in no way belittling the importance or relevance of this work, I will at the same time claim that it is of little use for high-level decision-making, in particular the kinds of decisions generally made by or made using systems doing what we are used to calling planning and scheduling. The basic problem is that the explicit, relatively long-term predictive simulation models employed for planning and scheduling require different techniques than those used either for traditional control theory, which only requires a gradient (or at most a limited predictive horizon), or for system verification, in which the system's behavior can be verified in a forward simulation. The property required for effective planning and scheduling is something Tom Dean once (about 15 years ago) called reasoning about time "from the side:" the ability to view and manipulate an entire chronicle at once.

2.2 Planning for real-time systems

There are two requirements for planning for real-time systems where failures are possible. First, timing constraints must be taken into account within nominal plans. CLP approaches, among others, should have no problem with this. Second, there are timing constraints imposed on the system's response to failure. One shortcoming of the straight CLP approach as presented here is that the system generates a plan, which it assumes can be executed. In practice, the course of events is rarely that predictable. The answer proposed in this paper is essentially a "safing" behavior, where the constellation as a whole goes into a safe mode, waiting for a new plan to be generated.

As has been repeatedly observed, there are situations in which this form of fault detection and recovery will have consequences ranging from inconvenient to disastrous. One instance is orbit insertion (one of the Cassini scenarios for autonomous planning). Another arose recently on Galileo, where safing was temporarily disabled during the IO fly-by.

There are several possible approaches to this problem, most of which have been investigated previously, none of which is a clear favorite, or for which the last chapter has been written.

- High level planning with low-level reactive behaviors. This approach involves conventional classical planning on top of an abstract action representation. These actions are further decomposed within a reactive planning and execution architecture such as RAPS or TCA, assuming that both execution failures and the pixels-to-predicates problem can largely be handled at the lower level. This approach has the problems with abstraction discussed above.

- State-based planning with explicit contingencies. In these systems, plans consist of graphs (or tables) of states with timed transitions corresponding to either actions or other events. Correctness is defined in terms of avoiding failure, and in a secondary sense in terms of eventually achieving a specified goal. The essentially Markov nature of these representations present difficulties for goal-directed behavior (as opposed to proving that a goal is reachable), and to the explicit management of resources over time. The state-based representation itself is potentially subject to combinatorial explosion and does not lend itself to the expression of continuous state parameters. All of these shortcomings are currently being addressed by various research groups, but I don't think any of them have been finally and generally resolved.

- Contingent "classical" "planning." The scare quotes around "classical" are because this category could also include systems like C-Buridan, which employs a probabilistic action semantics. Those around "planning" are because it is possible as well to build an explicitly contingent scheduler, as Bresina and others did for the APT, for example. The problem with this approach is the need to reason about (and generate a plan representing) all relevant contingencies.

- More complex architectures. Remote Agent, as far as I can tell, employs a complex combination of these techniques. HSTS constructs plans on an abstract model which still includes some continuous information. The executive takes the abstract actions in that plan and expands them, assuming that that expansion will not introduce infeasibilities into the plan (it does, sometimes). Finally, execution failures are detected and local responses to those failures are generated in Livingstone, which attempts to keep the system in a safe state until some more general recovery action can be planned for.

Something like the set of tradeoffs involved in RA will be required of any autonomous embedded system. The simpler architectures do not provide enough opportunity to exploit useful domain structure. Some questions raised by the approach proposed in this paper include whether the abstractions employed by RA are the right ones, and how easily RA can be adapted for coordinated multi-agent operations.

2.3 Multi-agent systems

The authors argue that constraint-based representations form a good basis for multi-agent systems. The system they actually describe involves one agent generating a plan for an entire formation, within the overall constellation. Coordination between formations is discussed in terms of a "collaborative policy," with little indication that there is any active coordination between formations. Well and good, and it is already clear in this situation why a hybrid constraint representation makes sense, and why the compositionality inherent in a CSP formulation is a good thing.

However, there are two ways in which I do not believe the authors have pushed this as far as they could and should. First, as discussed above, their "planning" isn't really planning. A more complex form of search over something more like a classical planning representation will add additional requirements and complexity, in ways that would make the current argument stronger. Second, the authors have taken a restrictive view of what it might mean to have a constellation of autonomous spacecraft. In particular, the level of autonomy provided to the individual spacecraft in a formation is quite limited. This is somewhat like describing your hand as a multi-agent system because you have five fingers, each equipped with their own sensors, actuators, and reflexes.

It would be more interesting, and consistent as an extension to the approach proposed here, to have the spacecraft negotiating cooperatively in the execution of some task, by exchanging commitments expressed as constraints, possibly within some global plan. In private communication, one of the authors has framed this more general view as a problem of distributed solution of the global CSP, i.e.,. coming up in a distributed framework with an equivalent answer to the one that would be found if the entire CSP was solved in one centralized model. This misses the point, to a large extent. The interesting questions for distributed systems involve situations in which, for reasons having to do with (for example) communication bandwidth or divisions of authority, individual agents are attempting to come to some kind of global solution that satisfies their own local constraints (possibly optimizes a local objective function), within some global set of constraints on behavior (related to a global plan, for example).

3 CLP and its limitations

One note of caution regarding what is proposed here: I do not, personally, believe that CLP solvers are ready for the kind of industrial-strength use that is required in a production environment involving broad application and extensive re-use and adaptation. There are two obstacles to be overcome before that happens. First, these systems must become sufficiently stable. CPLEX (an LP package now owned by Ilog) is a good example of a solver that is sufficiently mature to be generally useful in a production environment. While I expect some argument on the point, I do not believe that any more general CLP/CP system is, as of yet. I do believe that they eventually will be, though in what form is still uncertain.

Second, these systems must be sufficiently *and appropriately* expressive. CPLEX satisfies this requirement, though in an odd way. There is a large community of people who are very, very good at rendering real problems in the sometimes very unnatural form of a linear program. This translation (and for that matter, the interpretation of the solver's output) is essentially a black art, but one for which a graduate curriculum has been developed.

There is no such curriculum broadly available for model translation and interpretation of solver results for more general CPs. Such a curriculum is even more necessary for CPs than for LPs. The range of modelling choices is considerably larger, rendering the choice of an appropriate model more difficult and more critical. That flexibility is exploitable in other ways, for example in the construction of combinations of solvers and modelling primitives designed for particular classes of applications (e.g., Ilog Schedule). Such systems are even less mature, and their application to real problems is still a black art. Despite claims to the contrary, applying such tools to a complex real-world problem still requires considerable sophistication in understanding both the underlying machinery and how to work around their expressive and computational limitations. This is the same set of difficulties on which "expert system shells" foundered, more than a decade ago.

None of this should be taking as detraction from the point of view that constraint-based representations are a fruitful avenue to explore. My qualification has to do with the assertions that these representations obviate the modelling problem, or that the current set of available tools are mature enough for general use. Nor am I asserting that the authors claim either of these things, though others certainly have.

4 What next?

This paper provides a useful first step in considering the application of CLP/CP representations to multi-agent systems in general and spacecraft constellations in particular. There is considerable additional work required, some of which it appears the authors are actively pursuing but did not have room to put in the paper. I look forward to further developments.