Commentary on “Argumentation for Coordinating Shared Activities”

Commentary Author: Mario Merri
European Space Agency, Robert-Bosch-strasse 5, Darmastadt, Germany,
Mario.Merri@esa.int

Paper Authors: Bradley J. Clement, Anthony C. Barrett, and Steven R. Schaffer
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
{fbclement, barrett, srschaffg}@aig.jpl.nasa.gov

Abstract. This is the commentary on the paper “Argumentation for Coordinating Shared Activities”.

1 Introduction

In the following, the main findings of the commentary process are reported. Firstly, I briefly introduce the understanding of the paper assigned to me. Subsequently, I summarise the main issues identified.

2 Understanding of the Paper

The paper presents an innovative approach to allow self-interested entities to negotiate a decision. As in most of the most successful application, the idea is that of mimicking human interactions. In this case, a number of entities have to make a decision within a certain time in a constrained environment. The proposed approach, called Shared Activity Coordination (SHAC) by the authors, is that these entities initiate a collaborative dialogue among themselves with the goal to reach a decision within the time limit and within the constraints.

The field of application of the above technique is clearly mainly concerned with deep space missions, where it is difficult or impossible to have the “man in the loop”. For these missions, the operations plan can no longer be a static, pre-defined sequence of activities, but it unavoidably becomes dynamic with fast re-planning based on “what just happened”. In this context, I feel that we are more in the field of on-board autonomous system than in that of mission planning proper, although the boundary between them is clearly difficult to define.

Additionally, if the technology proves to be successful, it could also be adopted for other types of missions with the aim of reducing operational costs.

The core part of the work is the SHAC algorithm whose purpose is to negotiate the scheduling and parameters of shared activities until consensus is reached. Clearly, if not properly handled, this negotiation could lead to a big mess (thrashing) if the various entities involved undo each others contributions to solving the problem. Four protocols are introduced by the authors to address this problem:

- Chaos protocol, which is laying the foundation of all other protocols below by means of inheritance. The name is already indicative of the fact that this protocol, alone, will not work.
- Master/Slave protocol, whereby only one entity (master) is given permission to modify an activity. Slave entities must trust the master and are thus cooperative.
- Round Robin protocol, whereby the consensus is established on a shared activity by rotating a master role by changing permission constraints.
- Asynchronous Weak Commitment protocol, whereby entities have associated dynamically established priorities on each decision to be made. These priorities are then used to achieve consensus.

Finally, the authors evaluate the four protocols by prototyping and presenting preliminary results.

3. Commentary

I have found the topic of the paper very interesting and certainly tackling an important issue that is mandatory for current/future space missions. The paper is also clear and well written.

However, a number of important issues should be addressed which have not necessarily been considered. Most of these issues relate to the “next phase”, that is the deployment of this technology in actual operations. I believe that they should be taken on-board by the researches as soon as possible in order to ease the adoption of their technology. The issues are:

1.) Testing
2.) Safety-Critical Operations
3.) Ground Observability of Plan and Timeline Execution Statuses

4.) Autonomy Valid Only for Simple Planning Problems

5.) Problem Complexity and Ease to Converge to a Solution.

3.1 Testing

One of the known critical issue for these types of autonomous system is the complexity of testing. In fact, as the system will have to make decisions in isolation, it is clearly extremely important to have the system undergoing an extensive testing campaign. It is also well know that testing a mission planning system might be extremely complicated as it is often difficult to cover all possible cases. In this context, the authors should provide details on how the testing had being planned, set up and performed in their case.

3.2 Safety-Critical Operations

There are today two distinguished schools of thought regarding the handling of safety-critical operations: on one hand, operational staff would like as much as possible to be in control of their execution; on the other hand, researchers as well as more challenging mission profiles offer solutions where autonomy is either unavoidable or plays a strong role.

More specifically, my question to the authors is if, in their opinion, they felt that their system was adequate also for the execution of safety-critical operations or if their concept foresaw that such operations were only handled under strict supervision from ground. If the latter applies, how would your concept fit with the ground control?

3.3 Ground Observability of Plan and Timeline Execution Statuses

Another issue is the dualism between fully autonomous systems versus ground controlled ones. I feel that an autonomous system would stand more chances to be accepted even by “conservative” operational staff if it had the capability to provide operational staff with relevant information on the plan and on the execution of the autonomous timeline. In this context, I would like to know if any mission intending to make use of their autonomous planning tool had put observability requirements on the execution status of the timeline. That is to say, if missions have specified mandatory information to be downlinked to ground so that operational staff could know exactly what is going on on-board. Furthermore, in case of positive answer to the previous question, how do you handle these observability requirements?

3.4 Autonomy Valid Only for Simple Planning Problems

The complexity of the planning problem clearly plays a fundamental role in the outcome of the planning system. Furthermore, complex planning problems require complex planning systems that are clearly more prone to errors also considering that it is more difficult to exhaustively testing them. My observation was that, for simple planning problems, the behaviour of the planning system, and thus the status of the spacecraft, is mainly deterministic (see also observability point above). Instead, if the mission planning problem is a complex one, the overall status of the spacecraft is somehow more stochastic, which, in case of problems, might bring to unrecoverable situations. Has your technology been used also with reasonably complicated problems? With what results?

3.5 Problem Complexity and Ease to Converge to a Solution

While reading the paper, the same question came to my mind over and over again: “what if the algorithm does not converge?” This seems a very superficial question, but, in my opinion, it contains the key issue. Once the network of agents, constraints, argumentation protocols, etc. becomes extremely wide, the complexity of the problem increases extraordinarily and the risk exists that it is difficult to converge to a solution in useful time. How can we have a feeling of what the limits are and how can we guarantee that a decision is always made in useful time.

Finally, Sometime people say: “a wrong decision is always better than no decision. Is this applicable to your case as well? How does it relate with the famous Apollo 13 quote “failure is not an option” hinting that in our business a wrong decision might be the end of the mission?