Commentary on
“Maximally Expressive Modeling”

Tim Grant

Department of Computer Science, University of Pretoria, South Africa
tgrant@cs.up.ac.za
Atos Origin Nederland b.v., Papendorpseweg 93, Utrecht, Netherlands
Tim.Grant@atosorigin.com

Abstract. This short paper comments on J. Jaap, E. Davis, and L. Richardson, *Maximally Expressive Modeling*. Proceedings, International Workshop on Planning and Scheduling for Space, 2004.

1 Approach

I have two affiliations: one academic and the other commercial. I chose a commercial approach to my commentary because the paper proposes a representation for planning and scheduling of the operation of spacecraft payloads based on experience with previous payload scheduling systems. This indicates that the representation is likely to have gained some maturity driven more by user needs than by academic or technical considerations. My primary aim was to see if this expectation was borne out.

The question of commercial maturity has wider importance for Intelligent Planning & Scheduling (IP&S) technology. Taking STRIPS as the first practical planning system, the technology is now over 30 years old. Given the rule-of-thumb that 40 years is needed to bring a technology from invention to viable commercial exploitation, one might expect IP&S technology to be emerging in Commercial Off-The-Shelf (COTS) products. The applications orientation of this workshop series and the advent of companies such as ILOG, Cosytec, APsolve, Visopt, and OnTime is encouraging. However, the roadmap of the European Union’s Network of Excellence in AI Planning (PLANET) shows that considerable R&D is still needed; see http://www.planet-noe.org/. My secondary aim was to see how far the commercialization process had come.

My views are based on over 16 years experience with Atos Origin, a leading French-Dutch global IT services provider (http://www.atosorigin.com/). As a principal consultant in the Technical Automation business unit, I specialize in human supervisory control, mostly in the aerospace, defence and motorway traffic control domains. Disciplined software development using modern design notations and project management methods is essential for both technical and business success. It is from this viewpoint that I reviewed Jaap, Davis and Richardson’s paper. Having analyzed the paper, I exchanged emails with the lead author. This short paper documents the key issues arising.

2 Outline of paper

The aim of Jaap, Davis and Richardson’s paper is to propose a representation – the modeling schema – for planning and scheduling the operation of spacecraft payloads. The associated reasoning functionality is under development in the form of a Scheduling Engine. In terms of its Figure 2, the paper focuses on the representation of Task Knowledge, i.e. representing procedural knowledge. The representation of the System & Hardware Knowledge – the domain model – is briefly covered in Section 2.5 in terms of modeling equipment modes. Initial ideas for the Scheduling Engine are outlined in Appendix A.

3 User populations

The intended user populations are the starting point for commercial software development.

In the context of the current scheduling paradigm, the paper mentions three user populations: scientists as the task experts, vehicle and hardware experts, and scheduling engine experts. The paper implicitly argues that the design of a suitably expressive modeling representation should make the “scheduling cadre” unnecessary. A streamlined paradigm is presented (Figure 2) in which the vehicle and hardware experts enter the system and hardware constraints and the scientists - the “Principal Investigator” (PI) - enter task knowledge. The paper claims that the two user populations can work independently and in parallel with one another.

Many COTS products – not just in the space domain – enforce the entry of hardware knowledge before task knowledge. This has a number of advantages:

- When the user wants to specify an activity, he/she can first select the desired hardware item from a pull-down menu and then select one of the permissible mode changes for that item.
- Verification and validation of the scheduling knowledge is improved. Non-existent hardware items and impermissible modes cannot be selected. Users do not

> have to remember the item's name, and the danger of mistyping names is avoided.

The lead author's response was as follows:

> Usually, long before tasks modeling commences, the hardware has [to] be delivered, tested, and configured into the space vehicle. And the hardware models (modes) have been (pre)defined. When building his task models, the PI selects the equipment from a list and then selects a predefined mode. The "in parallel" situation arise when the PI discovers that he needs a new mode. He then contacts the system/hardware experts who builds the new mode. The PI then continues with task modeling.

4 Capturing user needs

User-centred design is the favoured approach in commercial software development. User needs can be captured, formalized and managed systematically in a variety of ways, e.g. questionnaires, interviews, user workshops, discourse analysis, including users in the development team, traceability matrices, etc.

I enquired whether the representation was based on an analysis of the users' needs and what methodology had been followed in capturing them. The lead author's response was as follows:

> The analysis began in 1978 to support SpaceLab missions. While developing the scheduling system used at MSFC/NASA, Elizabeth Davis (co-author) and myself enhanced the modeling schema of a predecessor P&S system. This schema was used at MSFC for 16 SpaceLab missions, by JSC for 2 SpaceLab missions and by DLR for the D-1 SpaceLab mission. E. and I also worked as "timeline engineers" for the SpaceLab-1 missions. As such we were users of the system. … When the International Space Station (ISS) came along, JSC created a new and different modeling schema … . This schema was somewhat better that what we used for SpaceLab, but could not capture all the requirements. MSFC won the P&S job for ISS payloads. Again the scheduling cadre was required to enter the models as defined by the PI (now called PD: Payload Developer). E. and I were assigned to develop a web-based method to collect the PD's requirements. This program is known as iURC. We took the knowledge we had from SpaceLab and enhanced JSC's schema to produce a schema that would let the PDs describe their requirements. THEY STILL WRITE NOTES to provide all the details. E. and I have access to every one of the PD submissions and by looking at how they are using iURC and what notes they write (captured by iURC), we can deduce what improvements are still needed and have included them in the proposed modeling schema. In the summer of 2001, Dr. John Usher, head of the Industrial Engineering Department at Mississippi State University, spent 10 weeks with us. He analyzed what we had done and made suggestions for improvements. These are included in the proposed modeling schema.

5 Implementation

Implementation is the key test of the workability of the user needs and the software design. To reduce development time, costs and risks, it is advisable to integrate existing COTS products where possible.

I asked whether the representation had been prototyped, e.g. using a COTS product like Microsoft Visio, compared against existing notations like UML or Harel state-charts or existing tools like MATLAB and Matrix$_X$. The lead author replied that MATLAB had been cursorily reviewed. The editor had been implemented initially in Java, but was now implemented in J#. It was demonstrated at SpaceOps'04 in Montreal in mid-May 2004.

6 Commercialization outside space domain

I asked whether commercialization of the representation outside the space domain had been considered. The lead author's response was as follows:

> Dr. Usher … thought that it is probably suited to large scale-building projects (Queen Mary 2) and large-scale product development. He also felt that it might be suited to operating room (hospital) scheduling and to airline crew/equipment scheduling. ...

7 Scheduling Engine performance

Scalability to large-scale, real-world problems is vital for commercial success. I asked whether the scalability of the Scheduling Engine's algorithm had been analyzed. The lead author's response was as follows:

> We are in the process of developing the scheduling engine, but cannot yet assess the performance issues. Of course, in this day, any performance issue can be solved by buying next year's computer. So far, we are pleased with the performance we are seeing.

8 Conclusions

Development of the proposed representation based on analysis of the documented shortcomings of the previous schema in real space missions is undoubtedly its strength, although it would be useful to include PIs and hardware experts in the development team. No evidence is offered for the modeling schema being maximally expressive. I look forward to trying out the editor myself, and I share Dr. Usher's view that the representation has a commercialization potential outside space. Scalability of the scheduling algorithm remains to be seen.