Software Product Assurance for Autonomy On-board Spacecraft

Contact Author: JP. Blanquart (Astrium SAS)

ASTRIUM SAS, 31, avenue des Cosmonautes – F-31402 Toulouse Cedex 4 (France)
Phone: +33 5 62 19 69 56, Fax: +33 5 62 19 78 97, E-mail: jean-paul.blanquart@astrium-space.com

Co-authors: Maria Hernek (ESA/ESTEC), Christophe Honvault (Astrium), Jean-Clair Poncet, Nelly Strady-Lécubin (Axlog Ingénierie), David Powell, Pascale Thévenod, Félix Ingrand, Sara Fleury (LAAS-CNRS)

1 INTRODUCTION

The increase of autonomy is an important trend in space systems, taking advantage of the increase of the on-board processing power to enable new or more efficient complex missions. This is particularly useful when the ground cannot react in real-time due to the communication delays, non-visibility periods, complexity or variability of the context. This raises new challenges for mission reliability and safety, due to both the criticality of the autonomous on-board software components, and to their complexity and the context variability. The former leads to strong software dependability and safety requirements, while the latter makes more difficult to fulfil such requirements. These peculiarities imply especially adequate software product assurance methodology and software dependability techniques.

SPAAS (Software Product Assurance for Autonomy on-board Spacecraft) is an ESA project (contract ESTEC 14898/01/NL/JA), granted to a consortium led by Astrium SAS with Axlog Ingénierie and LAAS-CNRS [1]. The objectives of the project are to investigate dedicated software product assurance measures to support autonomous functions both for nominal spacecraft operations and for fault detection, identification and recovery management, i.e., how to ensure safety and dependability of autonomous space software and especially of software in charge of autonomous functions dedicated to the spacecraft safety and dependability management. Special attention is put on software product assurance for advanced autonomy techniques (artificial intelligence, self-learning techniques, etc.).

The project is split in two phases. The first phase investigates the lessons learnt from autonomous non-space applications, the software product assurance requirements and then methods, tools and procedures, for autonomous space systems. Special autonomy software safety aspects are then investigated and an implementation plan is proposed for the second phase. This second phase is dedicated to the definition of software functions (on-board and in the ground system) for the safety of spacecraft with autonomy, and to their implementation and assessment through a pilot application.

2 STANDARDS AND PRACTICES

This section analyses the various methods for software dependability and safety, as recommended in standards and norms, or used in industrial practice. Seven standards and norms were analysed, US Department of Standards MIL-STD-498 and 882D, the IEC 61508 standard on programmable safety related systems, the Cenelec EN 50126/8/9 series of standards for railway applications, the UK Ministry of Defence MoD 00-55/6 standards for safety related software, the civilian aircraft DO 178B/ED12B standard, and the IEC 14598 standard on the evaluation of information technology products.

In addition, industrial practices were analysed, from former ESA studies on software dependability and safety (PASCON WO12) and from advanced autonomy projects for airborne, waterborne and terrestrial systems.

This survey shows that most safety-related software standards pay little explicit attention to autonomy and to the particular advanced software technologies for system autonomy. However the recommended set of techniques and methods for safety-related software may not be easily applicable considering, e.g., the size and complexity of the software and of the input and state domains, the dependency of the software

behaviour on knowledge bases, etc. [2] This is confirmed by the available reports and studies on advanced autonomy systems, as discussed for instance in a recent specific workshop that addressed the verification and validation of autonomous and adaptive systems [3].

3 SOFTWARE FOR AUTONOMY

This section will describe the various software autonomy techniques, and discusses the applicability of software dependability methods and of the clauses of the software product assurance standards for space systems (European Cooperation for Space Standardization, ECSS [4]).

This survey addresses rule-based systems, case-based reasoning, constraint programming, genetic algorithms, fuzzy logic, artificial neural networks, probabilistic networks, Markov decision processes, agent and multi-agent systems.

Each technique is described according to its mathematical and algorithmic definition, impact on space architecture and functions, and applicability of current software product assurance standards. A focus is put on issues of interest for autonomy in space systems, including planning and scheduling, diagnosis, and on the notion of on-board control procedures.

4 AUTONOMY SOFTWARE DEPENDABILITY AND SAFETY

It appears that autonomous systems and especially those based on advanced autonomy technologies and artificial intelligence (AI) pose some significant challenges regarding software product assurance. They are a relatively new trend in real-world critical embedded applications, particularly in space systems, and there have been few studies aimed specifically at defining appropriate assurance techniques. However, several tentative conclusions may be drawn [5]:

- The problem of verifying and validating knowledge-independent components of an AI-based system (e.g., inference mechanisms) is similar to that of classical software engineering.
- Separate knowledge representation is one key aspect that makes verification and validation of AI-based systems different to that of classical software engineering. Checking the consistency and completeness of the knowledge representation has thus received deserved attention. Note, however, that several authors underline the advantages, from a product assurance viewpoint, of having domain-specific knowledge represented separately from procedural mechanisms making use of it, since domain experts may more readily check it. Moreover, inference mechanisms based on logic may allow formal proof of correctness properties.
- Learning systems, whose function emerges from training examples or during operation, prove to be quite robust in practice. Nevertheless, they are less amenable to dependability and safety arguments than those whose knowledge and inference mechanisms are determined *a priori* by the designer.
- The most significant challenge in the use of AI-based techniques for autonomy is that of unanticipated and complex situations in which the system is nevertheless expected to act sensibly. There are only two apparent (complementary) ways to address this challenge:
    - Use extensive simulation testing to increase statistical confidence that the autonomous system will behave as expected. For really extensive simulation testing, some form of automated oracle should be envisaged.
    - Use on-line assurance techniques, such as the safety-bag or safety supervisor approach to ensure that catastrophic failures are avoided, which implies some form of graceful degradation [6]. The generalization of the safety bag concept towards “active safety management” is also an interesting direction for future research [2].

- Although autonomous systems are required to operate for extensive periods of time without human intervention, it is important that autonomous systems also support human intervention when necessary.
- When humans and AI-based systems are to interact synergistically, new human factor risks may be introduced.
- Autonomous operation can significantly impact software development in that domain-specific knowledge needs to be encoded early on. An evolutionary program development strategy should facilitate a progressive refinement approach in which critical autonomous system capabilities may be addressed first.

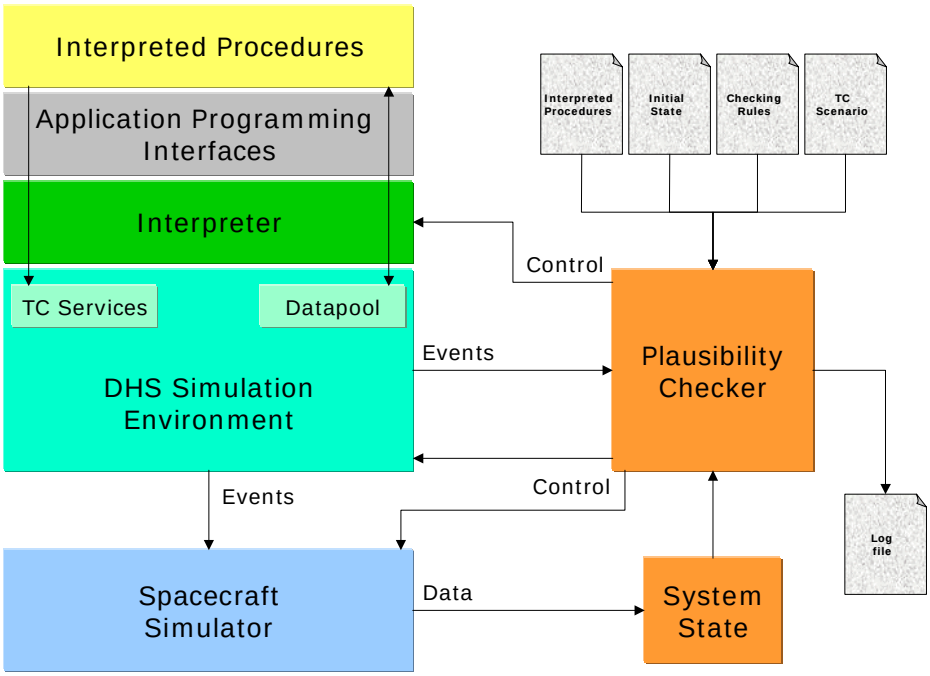
5 COMPONENTS FOR SAFE AUTONOMOUS SPACECRAFT

The survey of dependability and safety software issues for autonomy in space systems especially highlights:

- the importance of the verification activities, which must be supported by various approaches and tools to widen the coverage for systems with such large spaces of states, inputs and possible behaviours,
- the fact that despite intensive verification and validation activities, there may remain design faults, as well as contexts and events leading to insufficiently specified and possibly inappropriate behaviours; consequently it is necessary that mechanisms be provided to monitor possible anomalous situations and inappropriate behaviours when they occur, with the capability to maintain as much as possible the desired properties, especially safety properties.

This leads to the definition of two kinds of software components:

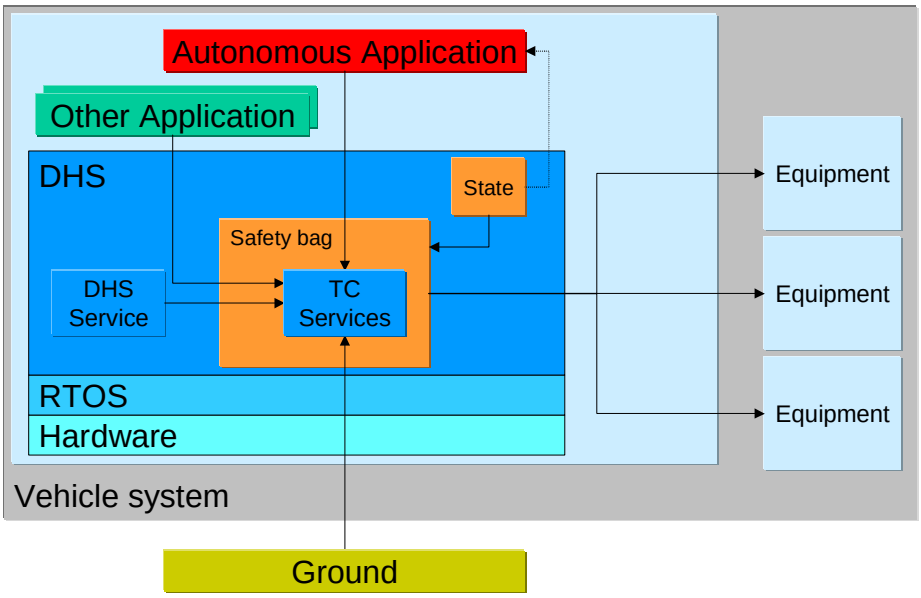
- A ground-based "plausibility checker" to support and complement the ground validation of autonomy software, and especially the on-board control procedures before upload and actual execution:



(DHS: Data Handling System; TC: Telecommand.)

Figure 1: Plausibility Checker architecture and situation

- An on-board “safety bag” to monitor on-line a set of safety properties so as to authorise or not the execution of commands to the spacecraft elaborated by the autonomous software applications:



(DHS: Data Handling System; RTOS: Real-Time Operating System; TC: Telecommand.)

Figure 2: Safety bag architecture and situation

The SPAAS project includes the elaboration of these two software components, safety bag and plausibility checker, as generic components to be instantiated and used in various real space projects with as few adaptations as possible, so as to support their dependability and safety.

6 EXPERIMENTATION AND ASSESSMENT

This section will describe the practical experimentation of the safety bag and plausibility checker components through a three-month pilot application on hardware, software and safety properties from real space projects. It presents and discusses the results of the experimentation in terms of performances, overheads and efficiency. A discussion of the generic features of the components, of the instantiation process and of the definition of the safety and plausibility properties to check, is provided.

7 CONCLUSION

The study reported in this paper addressed the software dependability and safety issues for autonomous spacecraft, with focus on software product assurance approaches applicable to autonomy software.

The survey of software safety and dependability methods, standards and industrial practice highlighted the needs both to complement the verification of autonomy software through intensive simulation and assessment of plausibility properties, and to monitor on-line at least the most important safety-related spacecraft properties. This led to the definition, development, validation and experimentation of generic software components to support dependability and safety of autonomous spacecraft: an on-board safety-bag and a ground-based autonomous procedures plausibility checker, to be used in future autonomous space projects.

8 REFERENCES

[1] SPAAS project (Software Product Assurance for Autonomy on-board Spacecraft). Contract ESTEC 14898/01/NL/JA. SPAAS technical notes are available at: ftp://ftp.estec.esa.nl/pub/tos-qq/qqs/SPAAS/StudyOutputs

[2] J. Fox and S. Das, Safe and Sound - Artificial Intelligence in Hazardous Applications, AIAA Press / The MIT Press, 2000.

[3] RIACS Workshop on the Verification and Validation of Autonomous and Adaptive Systems, 5-7 Dec. 2000, Asilomar Conference Center, Pacific Grove, CA: http://ase.arc.nasa.gov/vv2000/

[4] European Cooperation for Space Standardization (ECSS). *Space Engineering — Software*, ECSS-E-40B (draft 1), 29-5-2002, *Space Product Assurance — Software Product Assurance*, ECSS-Q-80B (draft 1), 29-5-2002.

[5] D. Powell & P. Thévenod-Fosse, "Dependability Issues in AI-Based Autonomous Systems for Space Applications", 2nd IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments, October 7-8 2002, Toulouse, France, pp.163-177.

[6] P. Klein, "The Safety Bag Expert System in the Electronic Railway Interlocking System ELEKTRA", *Expert Systems with Applications*, 3 (4), pp.499-560, 1991.