SPACE SYSTEMS SYMPOSIUM (D1)
System Engineering Tools, Processes & Training (3)

Author: Dr. Miriam Alves
Institute for Aeronautics and Space (IAE), Brazil

A FRAMEWORK PROPOSAL FOR FORMAL VERIFICATION AND VALIDATION OF SPACE SOFTWARE SYSTEMS

Abstract

The ability to deal with a high level of complexity in a flexible way makes software an essential and increasing part of space and ground segments products, which makes the context of space software engineering closely related to the general engineering process of space systems [1][2][3]. The complexity of the functional and performance requirements demands special measures and emphasis for software verification and validation, especially for critical software. The flight control software of the Brazilian Satellite Launcher (VLS) is included in this category. Considering this context, this paper proposes a framework for verification and validation of critical software systems that includes the incorporation of ready-to-use formal techniques, allowing the designers to make a more effective and consistent analysis of the software in development. Such framework brings the novelty of putting together different formal verification techniques that can be applied to ensure properties like safety, liveness, timing and deadlock-freeness. The initial efforts are concentrated in the framework conception and definition followed by the construction of a proof-of-concept prototype that will incorporate a formal technique based on concepts of algebraic topology [4] and model-checking techniques [5]. A methodology is also established with this framework, encompassing all the phases of development, ranging from initial verification activities to the final validation of the software, when it is delivered to the systems engineering team for the final acceptance tests. The space software can be verified by one or more techniques, which include: tests, analysis, review of design, inspection and demonstration, whereas the formal verification will be conducted for verifying the consistency of the models according to the type and criticality of the requirement. Not long ago, formal techniques evolved from a mostly theoretical discipline in computer science to practical verification and validation methods, although a few applications of formal verification at industrial scale have been reported in the research literature so far [6] [7]. Some of the results of this work include the uncovering of potential errors, giving the team the opportunity to seek for a better solution rather than makeshift fix to accommodate deadlines; criteria for whether or not to proceed to the next development phase; and incremental preview of the system performance with chance of making early corrections, due to revealed important flaws previously unknown to the designers. Considering that any verification and validation process is inherently incomplete, the use of formal techniques and traditional testing techniques are complementary within the proposed framework.

References

[1] M.C.B. Alves, M.A.D. Abdala, R.B. Silva, “Bringing together space systems engineering and software engineering processes based on standards and best practices”, book chapter of Complex Systems Concurrent Engineering, Springer London , 2007, pp. 159-166. [2] ECSS-10A Space Engineering – System Engineering, ESA Publications Division, The Netherlands, 19 April 1996.

[3] J. Callahan and G. Sabolish, “A Process Improvement Model for Software Verification and Validation”, Journal of Quality Assurance Institute, 1996, Vol. 10, pp. 24-32.

[4] M.C.B. Alves, C.C. Dantas, N.N. Arai, and R.B. Silva, “A topological formal treatment for scenario-based software specification of concurrent real-time systems”, In Proceedings of International Conference on Systems and Software Engineering and their Applications – ICSSEA 2007, Paris, 2007.

[5] Berard, B., M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen, P. Mckenzie, "Systems and Software Verification: Model-Checking Techniques and Tools", Springer-Verlag Berlin Heidelberg, Germany, 2001.

[6] P. Godefroid, R. S. Hanmer, and L. J. Jagadeesan. Model Checking Without a Model: An Analysis of the HeartBeat Monitor of a Telephone Switch using VeriSoft. In Proceedings of ACM SIGSOFT ISSTA'98 (International Symposium on Software Testing and Analysis), pages 124–133, Clearwater Beach, March 1998.

[7] G. J. Holzmann and M. H. Smith. A Practical Method for Verifying EventDriven Software. In Proceedings of the 21st International Conference on Software Engineering, pages 597–607, 1999.