

MODSARE-V: Validation of Dependability and Safety Critical Software Components with Model Based Requirements

Extended Abstract - submission to DASIA 2010, EUROSPACE, Budapest, Hungary

Corresponding Author: *Daniel Tomás de Maia Mozart Silveira*
GMV-SKYSOFT.
daniel.silveira@gmv.com

Tobias Schoofs
GMV-SKYSOFT.
tobias.schoofs@gmv.com

Ana Isabel Rodríguez Rodríguez
GMV-AD[†]
airodriguez@gmv.com

Elena Alaña Salazar
GMV-AD
ealana@gmv.com

Marie-Odile Devic
ESA/ESTEC – Noordwijk, The Netherlands [‡]
Marie-Odile.Devic@esa.int

Abstract

The wide use of RAMS methods and techniques [1] (e.g. SFMECA, SFTA, HAZOP, HA...) in critical software development resulted in the specification of new software requirements, design constraints and other issues such as mandatory coding rules.

Given the large variety of RAMS Requirements and Techniques, different types of Verification and Validation (V&V) [14] are spread over the phases of the software engineering process. As a result, the V&V process becomes complex and the cost and time required for a complete and consistent V&V process is increased.

By introducing the concept of a model based approach to facilitate the RAMS requirements definition process, the V&V may be reduce in time and effort.

MODSARE-V is demonstrates the feasibility of this concept based on case studies applied to ground or on-board software space projects with critical functions/components.

* GMV-SKYSOFT, Torre Fernão de Magalhães, Av. D. João II Lote 1.17.02, 7º Andar 1998 – 025, Lisboa, Portugal, Phone: +35121382366, Fax: +351 21 386 64 93, www.gmv.com.pt

† GMV, Isaac Newton, 11, P.T.M., Tres Cantos, E-28760, Madrid, Phone: +34918072126 Fax: +34918072199, www.gmv.com

‡ ESA - European Space Agency, European Space Research & Technology Centre (ESTEC), TEC-SWS, Keplerlaan 1 Postbus 299, 2200 AG Noordwijk, The Netherlands, Phone: +31 71 5658553, Fax: +31 71 5655420.

This paper describes the approach adopted at MODSARE-V to realize the concept into a prototype and summarizes the results and conclusions met after the prototype application on the case studies.

1 Introduction

The verification and validation of dependable and safety critical software components starts by ensuring that the requirements (RAMS requirements) have been confirmed as complete and consistent.

Model-based technologies [2] have shown potential to ease the requirements definition process and ensure their completeness and consistency.

A tool to analyze RAMS requirements generated through the formalism of modeling resources (tools, languages, meta-models, transformation) would support the preparation of high quality validation test suites.

The works presented in [2] [3] reaches similar conclusions concerning the direction for future research or further work on V&V. Other works already present partial solutions by focusing on model based on a proprietary modeling language [4][5], or by limiting the scope to a subset of the model using formal methods [6].

These works enforce the need to evolve model based technologies towards so that model defined at software design phase can be used at RAMS validation and at the same time having the capacity to cover all kind of requirements and model elements [13].

MODSARE-V prototypes tools provides:

- Improved effectiveness in the verification and validation phase by reducing the time and effort spent on this process;
- Automatically prepared test suits with requirements models support;
- Demonstration of the test suite suitability, increasing the confidence that the product meets the quality targeted.

The feasibility of the prototype was demonstrated on case studies [7][8] selected from existing space projects (ground or on-board software) with critical functions/components.

The case studies show that the prototype supports:

- The modeling of D/S requirements, also known as RAMS requirements, resulting from each definition phase;
- The simulation of these requirements and the quantification of their completeness and consistency;
- The derivation of test objectives/suites to be injected in the verification/validation process;
- The assessment of completeness and relevance of the tests implemented.

In order to obey the objectives of this prototype, the study done for this work was structured into the following tasks:

- Inventory and evaluation of existing models;
- RAMS requirements – typology;
- RAMS requirements - verification/validation mechanisms;
- Relating the requirements and the verification/validation mechanisms in the model(s);

- Application on case studies refinement of the previous concept – Prototype.

2 MODSARE-V Approach

2.1 Requirement Catalogue and Typology

A catalogue containing/listing more than 900 RAMS requirements resulted from the research with 9 different critical space projects. Those requirements were classified by types based upon recommended design and implementation approaches provided by ECSS documentation namely [9], [10] and [11].

The catalogue and the typology classification led to the creation of generalized/prototypical RAMS requirements, which were modeled with the SysML notation, this model is known as the RAMS database.

2.2 V&V Test Scenarios

The RAMS database information was improved with the research for the best V&V strategies, resulting in the definition of V&V Scenarios with regard to the modeled RAMS requirements.

The improvement in the RAMS database was obtained through the addition of know-how for tests from a model automated derivation, execution and results report. This is possible because the collected data allowed the modeling of constraints needed to check the related RAMS correctness/threat in a model object.

2.3 Modeling Guide

The knowledge gathered by MODSARE-V is made available to the user through the modeling guide [12].

This document is designed to be independent from the prototype and self-explanatory. MODSARE-V modeling guide has the primary objective of removing/minimizing RAMS threats from the modeled software and can be applied for any modeling language.

The modeling guide is structured according to common functionalities that are defined in a software model. For each functionality, a set of scenarios and their respective constraints are explained, supporting the user to model avoiding RAMS threats.

The constraints are clearly identified and associated to the RAMS database generic/prototyped RAMS requirements and scenarios. The constraint is applied to verify:

- Keywords that must exist in the model;
- Correct sequence of model objects;
- Detection of model patterns;
- Association between model objects;
- Coherency between static and behavior/dynamic data.

The only action required by the modeling guide from a new software project is the update of this model in order to follow the constraints.

3 RAMSCheck Prototype

The prototype realized by MODSARE-V is called RAMSCheck and it goes beyond the principles Model Based Testing by creating a battery of tests against the diagrams from the model, namely the class, deployment, activity and state machine diagrams.

RAMSCheck verifies the correct application of the Modeling Guide constraints against a model using the knowledge stored in the RAMS database.

The input used by RAMSCheck is OMG XMI 2.1 format file from SysML or UML models. This file format is commonly obtained from SysML or UML modeling tools.

The RAMSCheck session is started and configured by setting what requirements were implemented in the model and what model objects will be analyzed.

The session configuration was designed to have minimum intervention possible from the user, providing single action configuration allowing:

- Complete model and requirements selection
- Selection of high, medium and low level requirements supporting the analysis through all software engineering phases.

RAMSCheck output is a report listing the RAMS threats found in the model. Additional information, such as the failed RAMS requirements is reported and solutions to the threat are proposed as well as test scenarios, requirement description etc...

The analyzed model can then be corrected by the user according to the detected constraint failures stated in the report.

Figure 6 depicts the RAMS analysis process using RAMSCheck.

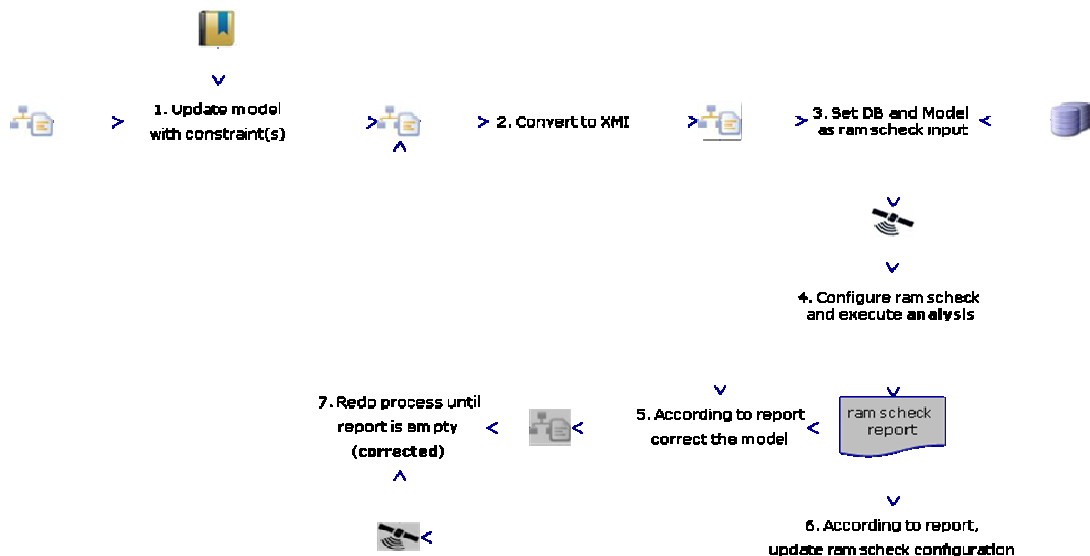


Figure 16 – RAMS Analysis Process overview

The analysis process is an iterative task, the report leads to new model corrections that must be analyzed again. The process normally finishes when the report is empty, meaning that no RAMS threats were detected by the tool.

4 Case Studies

4.1 FDF

The FDF is an element of Galileo Ground Segment of the Galileo System.

The RAMS analysis focuses on the critical elements of FDF that are components from its external interface also known as FDF EXIF.

The FDF-EXIF software is classified as critical level C (MAJOR).

The FDF EXIF features the following functional features:

- Command line and GUI;
- Data processing, managing data related with a Galileo system/element (CMCF, ESCC, GMC, OPF, SCCF GCS , SCPF GCS, TTCF);
- Communication process, that manages the communication with the Galileo system/element using protocols like CORBA servicing, SNMP, GFTS, FTP and HTTPS.

4.2 SENTINEL 3 OLCI-ICMSW

Sentinel-3 is an Earth Observation Mission in the frame of GMES that will provide the following mission products and services:

- Ocean colour measurements;
- Sea and land surface temperature;
- Sea surface topography.

The Unit Flight SW in processor or ICM-SW is the software embedded in the Instrument Control Module (ICM) on board the OLCI Electronic Unit (OEU), the intelligent core of the Ocean & Land Colour Instrument (OLCI).

ICM-SW is in charge of:

- Processing SMU TC packets;
- Managing the Events Log Table;
- Handling the PPS and EQSOL signals;
- Producing TM packets to the SMU;
- Monitoring/controlling instrument items & OEU;
- Managing the instrument modes;
- Performing the active thermal control instrument items & OEU;
- Delivering Science Clock (SCLK) to DPF;
- Managing the OBT.

ICM-SW project develops ICM-SW software which is classified as critical level B (CRITICAL).

4.3 Results

The case studies analysis can be summarized with following results:

- RAMSCheck was capable to detect a large amount of RAMS threats; a total of more than 900 corrections/ improvements in the model.
- RAMSCheck increased in 178% in FDF and 141% in OLCI-ICMSW model the amount of new RAMS threat prevention features (e.g. safety barriers).
- The case studies model quality was quantified being detected a gain with RAMSCheck of 597% in FDF and 258% in OLCI-ICMSW
- When compared to the results stated in the case studies RAMS Reports derived from the RAMS Techniques (HA, HAZOP, FMECA etc...), RAMSCheck increases confidence level since additional RAMS threats are identified and more RAMS solutions were implemented in the model.

5 Results and Conclusions

Further analysis of the results led to the following final conclusions on MODSARE concept for RAMS Analysis.

- It reduces time and effort in the verification and validation of critical software by detecting and implementing RAMS solutions at early stage of the engineering;
- Improves the quality of the model;
- Provide alternative resolutions for an existing RAMS threat;
- Provide pre-built test suits that support RAMS requirements;
- Bring RAMS-related issues to the engineering team attention;
- In many cases it is capable to recommend more than one solution for the protection of a detected RAMS threat;
- The RAMSCheck report is also an input for the execution of Product Assurance documents and base for the generation of standard software RAMS Techniques;
- It does not require the knowledge of new modeling languages and formalisms;
- It presents a cost-effective standalone analysis method when compared to manual methods used in RAMS Techniques;
- It is capable to cover all kind of functional RAMS requirements and function model objects.

RAMSCheck meets the project objectives and still presents a huge potential for improvements and future work such as:

- Become applicable to RAMS software of other business areas a part from space;
- Become applicable to non-RAMS software;
- Have an editable and flexible RAMS Database;
- Become a full functional software tool;
- Being integrated in a software development platform;
- Being portable towards many platforms and operating systems;
- Increase in the tool performance.

6 REFERENCES

- [1] "Software Dependability Technique". The ESA SME Initiative Training Courses. ECSS-Q-80B-SW Dependability Techniques.
- [2] J. M. Faria, S. Mahomad, N. Silva ,” Practical Results from the application of Model Checking and Test Generation from UML/SysML Model of On-Board Space Applications”, Paper for ‘DASIA 2009 Conference, 26 – 29 May 2009 (ESA SP-669, August 2009)
- [3] Kenneth Kvinnesland, Øystein Torget. “Use of Models to Verify and Validate Requirements for Critical Software”, Paper for ‘DASIA 2009 Conference, 26 – 29 May 2009 (ESA SP-669, August 2009)
- [4] <http://www.conformiq.com/>, date of visit: 29-01-2009.
- [5] <http://www.leirios.com/>, date of visit: 29-01-2009.
- [6] <http://www.pst.informatik.unimuenchen.de/projekte/hugo/>, date of visit: 29-01-2009.
- [7] Flight Dynamic Facility (FDF) , Ground Control Segment (GCS) , Galileo System, European Space Agency (ESA)
- [8] Instrument control unit OLCI Electronic Unit (OEU) of Sentinel-3, is an Earth Observation Mission in the frame of Global Monitoring for Environment and Security (GMES), European Commission (EC) and European Space Agency (ESA)
- [9] ECSS-Q-30B “Dependability”. Space Product Assurance. European Cooperation For Space Standardization, Date: 08-03-2002
- [10] ECSS-Q-40B “Safety”. Space Product Assurance. European Cooperation For Space Standardization.,Date:17-05-2002

- [11] ECSS-Q-80C "Software Product Assurance". Space Product Assurance. European Cooperation For Space Standardization., Date: 06-03-2009
- [12] "RAMS Modeling Guide", MODSARE-V, contract ESTEC 21404/08/NL/EM,
- [13] T Schäfer, A Knapp, and S Merz. "Model Checking UML State Machines and Collaborations." Electronic Notes in Theoretical Computer Science, Vol. 55, No. 3. (2001), pp. 357-369. [<http://www.loria.fr/~merz/papers/sw-mc01.pdf>]
- [14] B W Boehm, "Verifying and validating software requirements and design specifications", IEEE Software. Vol. 1, no. 1, pp. 75-88. 1984