Modeling Space System to provide global coherency from design to operations phases

E. Poupart[1] and M.C. Charmeau[2]
CNES, Toulouse, France, 31401

A. Cortier[3]
ATOS, Toulouse, France, 31024

There is a big semantic "gap" between textual information spread into the many documents (*space system manuals, etc.*) used in operations and what is really produced (software, hardware, procedures, spacecraft database, etc.) and used for validation (simulators, test beds, failure analysis tools, etc.). Operational user is "taken between" a huge amount of documentation and the very low level information spread into many different specialized formalisms and tools. It is very difficult to get quickly the information necessary to understand how the system works, which is a key point for many operation tasks like the design of operational procedures, alarm or error analysis, etc. A formalized model that captures the system knowledge would not only help designers and operators to catch it much more efficiently rather than in documents, but this may also be used by the computer to ease the many verification and validation tasks to be done, and may enable to keep a more efficient traceability with requirements during the whole system life-cycle. Some attempts to go towards this formalized, global and centralized system view have already been made, for instance with the "Space System Model" from the European ECSS-E-ST-70-31 standard. However, it needs to be much more enriched to take into account all the different views of the system and obtain all the expected benefits. In this paper, we present the results of a CNES R&T study called EGPO done with the help of ATOS Company to go further in this direction, initiating a richer view of the space system, and formalizing it with a customization of the SysML standard to the space domain.

I. Introduction

AT each different phase of the whole space system life-cycle, from creation to withdrawal, its formalization as a central system view may be useful.

Already in the preliminary design phases, when the space system becomes more concrete, hierarchical system views may be formalized by describing:

- Its logical architecture view with its main functions and sub-functions;
- Its physical architecture view with the allocation of functions on physical elements;
- Its dysfunctional view for failure mode analysis;

[1] Ground Segments Studies and Product Development Engineer, Ground Segments and Products Department, Erwann.Poupart@cnes.fr.
[2] Monitoring and Control Engineer, Ground to Space Department, Marie-Claire.Charmeau@cnes.fr.
[3] Software development Engineer, Innovation & Methods team, Alexandre.Cortier@atos.fr.

- Its operational view using time invariants from the highest level taking into account mission phases, to the lower operational levels with the operating modes occurring during mission phases, and finally operating states to coordinate activities inside operating modes.

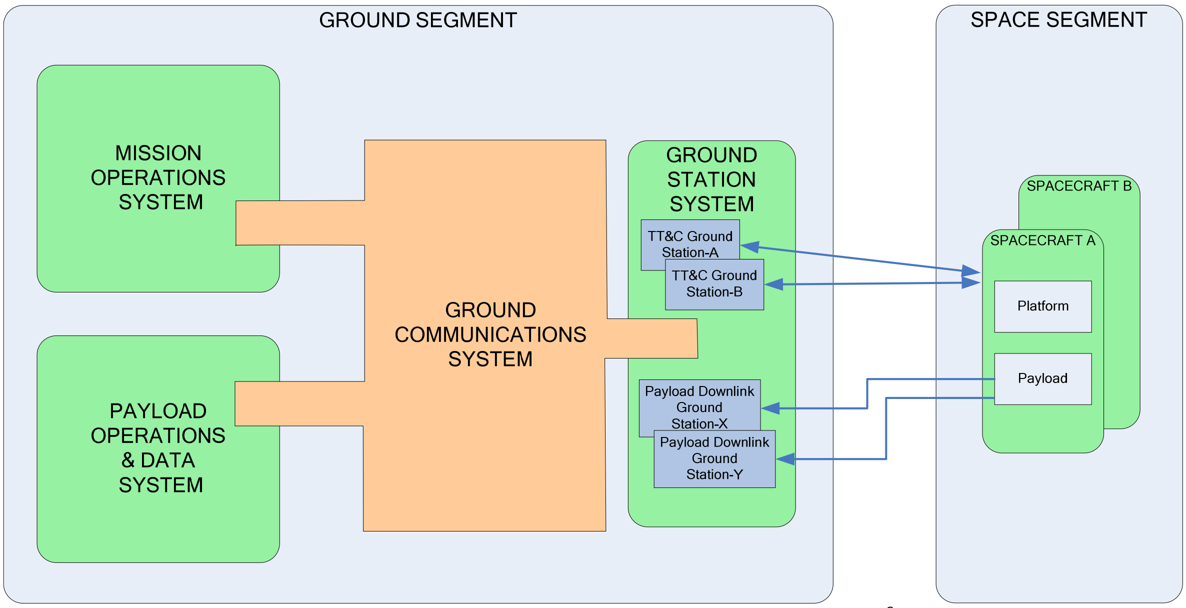

Figure 1: Space system from European ECSS-E-ST-70-31[6]

In the reality today, a space system view is sometimes partially formalized in the first design phases, but then it is usually lost and spread first into documents and finally into specialized and low level formalisms for dedicated activities like:

- Software models and code for on-board software sub-system design;
- functional and dysfunctional behaviors models etc. for failure modes and effect analysis;
- MATLAB/Simulink models for attitude and orbit control sub-system algorithms design;
- Spacecraft databases formalisms for monitoring and control design;
- Operation formalisms for sequence of operation design;
- etc.

A uniform space system view shared by all the different actors designing or operating it, does not exist today.

For instance, spacecraft database formalisms contain only monitoring parameters, checks and commands but not the mission phases, operating modes, activity transitions inside modes, and behavioral constraints which are defined separately in on-board software models, code, or operational formalisms and tools.

This uniform space system view could be useful to help procedures design, to analyze an alarm or a failure, or many other design or operational activities. This would also facilitate coherency at system level, for example to be sure that a possible future modification of operating modes will be correctly reflected everywhere and especially in the constraints of related procedures.

Using this uniform system view, the computer may provide warnings of a possible system inconsistency or mismatch.

The semantic information provided by this uniform space system view would not only help designers and operators to catch it much more efficiently in a system model rather than in documents, but it may also be used by the computer to ease the many verification and validation tasks to be done, and finally keep a more efficient traceability with requirements during all the system life-cycle. Those benefits are not only expected for the design phases but also for verification phases before the launch and during operation activities to keep the system operational.

Some attempts to go towards this uniform and global system view have already been made, for example with the "Space System Model" (SSM) from the European ECSS-E-ST-70-31 standard. However, SSM is mainly restricted to the physical architecture view with the allocation of functions on physical elements, and needs to be much more enriched to take into account all the different views of the system and obtain all the expected benefits.

II. Initiating a space system view

A. The approach

A CNES R&T study done in 2011, called EGPO (study for Generation of Operational Procedures), performed with the help of ATOS company, showed some very promising results to initiate this space system view in a standard way.

A possible choice would have been to define another dedicated language to describe space systems, but it would have required to redefine also all the basic concepts for both structure and behavior system views.
We decided to take another option which is to reuse and customize SysML[9] which is a general-purpose modeling language for systems engineering applications.

Space system concepts were taken from the following space standards:

- ECSS-E-ST-70-31Ground systems and operations monitoring data definition European standard for the ground to space link and its corresponding CNES ISIS tailored version, called ISIS System Data Model;
- ECSS-E-70-41[7] Ground systems and operations monitoring telemetry and telecommand packet utilization European standard for the ground to space link;
- ECSS-E-ST-70-01[5] Space Engineering – Spacecraft on-board control procedures standard or OBCP, and ECSS-E-ST-70-32[4] Space Engineering – test and operation procedure language standard, or FCP, and their corresponding CNES ISIS tailored versions, called "ISIS Exchange Format for OBCP and Elementary Procedures", and "ISIS Operations Procedure Language";
- Mission Operation services CCSDS draft standards[1,2,3];

The main difficulty identified was the semantic gap between SysML and space system concepts, so the main task was to build links between the two. To this end, we defined a mapping of space system concepts to SysML, first to measure this gap, and then, if not too big, use this mapping to customize SysML by specialization of its concepts.

Another potential difficulty was the maturity of existing tooling that can support SysML customization based on UML profiles and stereotypes. We choose to use open-source TOPCASED[8] toolkit for this study with the new Papyrus UML/SysML editor which seemed very powerful to ***specialize SysML by the definition of UML profiles and stereotypes*** and using them to customize editors.
Note: To ease the reading of this paper, we call this customization "*SysML profiles*" in the following even if it is a misuse of language.

B. Result of the mapping of space system concepts to SysML

One of the major result is that space domain concepts are relatively close to SysML. Three main space system views were mapped to SysML, this is described hereafter.

1. Mapping of space system physical architecture view to SysML

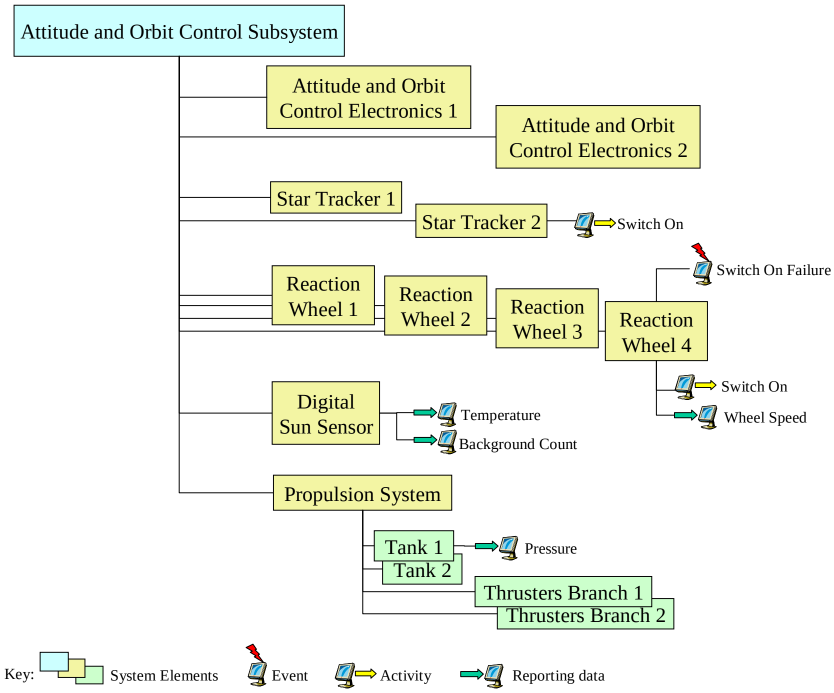

Figure 2: SSM physical architecture view with M&C function allocation (ECSS-E-ST-70-31)

This view, showed in figure 2, is the most complete in the ECSS-E-ST-70-31 standard, and its hierarchical aspect can be easily represented using SysML Block Definition Diagrams and their *parts* also typed as SysML primitive types or as a SysML Block Definition. The root elements are the first parts stereotyped as "System". Then each lower space system element (Sub-System, Equipment, Assembly, Part, Sensor, etc.) has been represented with a dedicated stereotype extending the SysML *part* concept. Finally, for the tree leaves, for monitoring parameters for example, we had also to use the UML instance specification concept because this concept was absent at class level. This was one of the main difficulty, the main reason is that we need a model including also instance level.

Then, function definition has been mapped to SysML *interface* definition concept and function allocation to SysML *port* concept. As an example showed in the figure 3 below, we have defined several equipment instances belonging to the AOCS (Attitude and Orbit Control Sub-system) each providing service independent of instances.

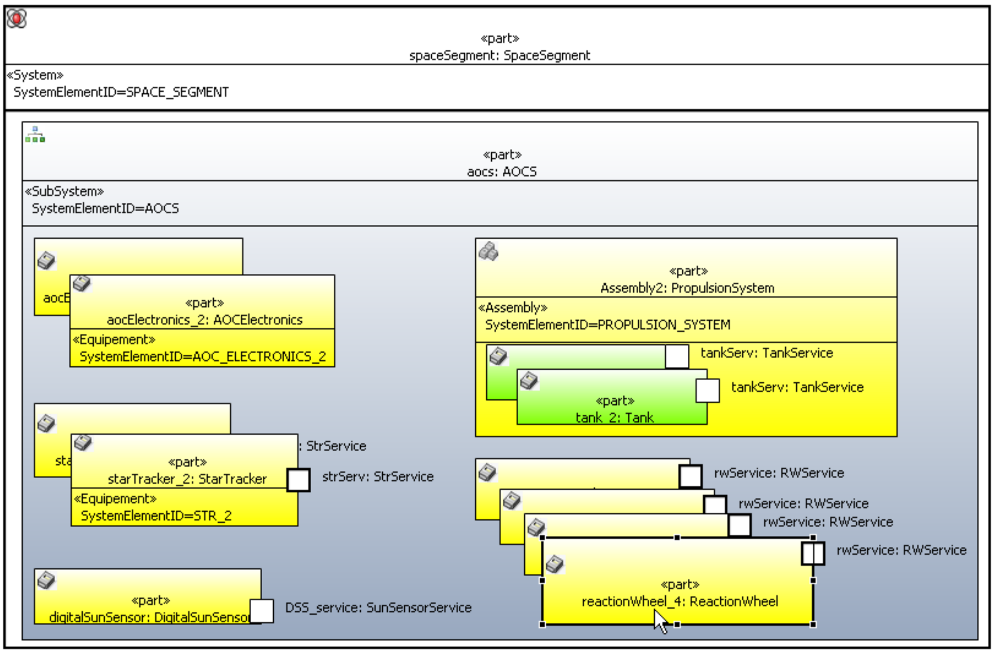

Figure 3: SSM physical architecture view mapped to SysML

Then, advanced editing capabilities provided by the new Papyrus UML/SysML editor, from open source TOPCASED toolkit, were experimented to provide a dedicated view to build or edit a Space System Model. Once SSM SysML profiles built, a diagram palette can be customized to reflect it as illustrated in figure 4 below.
The user will then mainly see SSM concepts instead of SysML or UML ones.

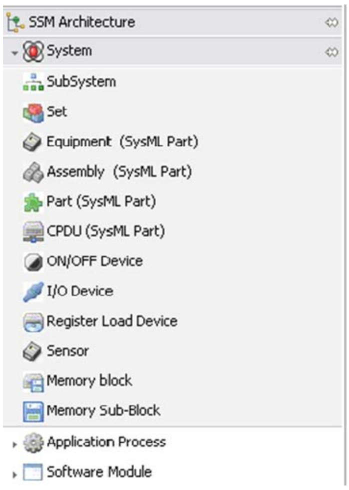

Figure 4: Diagram palette corresponding to the "SSM" SysML profile

2. Mapping of space system monitoring and control logical architecture view to SysML

We have seen in the previous paragraph that functions have been mapped to SysML interface concepts. However, we had to complete this logical view with ECSS-E-ST-70-31 Monitoring and Control concepts like *Activity*, *Reporting Data* and *Events and* CCSDS MO Activity, Status and Event concepts. This resulted in a dedicated Monitoring and Control SysML profile. Those space domain concepts can be mapped to SysML as described below:

- ECSS-E-ST-70-31 and CCSDS MO Event concepts are equivalent and can be mapped to SysML *ChangeEvent* concept;
- ECSS-E-ST-70-31 "Reporting Data" and CCSDS MO Status concepts are similar and can be mapped to SysML *SignalEvent* concept;
- ECSS-E-ST-70-31 and CCSDS MO "Activity" concepts are similar and can be mapped to SysML *Activity* concept;

It was necessary to enrich SysML concepts with some additional information added in the corresponding ECSS-E-ST-70-41 PUS and CCSDS MO SysML profiles.

Then, as for the physical architecture, a diagram palette was customized to reflect it as close as possible to the space domain view as illustrated in the figure 5 below.

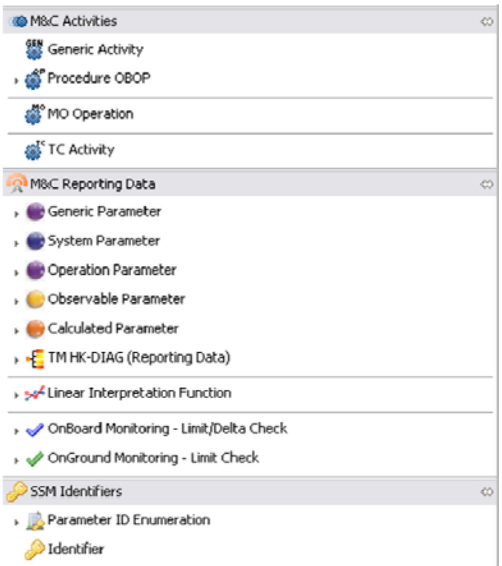

Figure 5: Diagram palette corresponding to the Monitoring and Control SysML profile

3. Mapping of space system operational view to SysML

We have seen in the previous paragraph that ECSS-E-ST-70-31 and CCSDS MO "Activity" could easily be mapped to SysML *activity* concept. To be more precise, this was applied to the "Space Engineering – Spacecraft on-board control procedures" ECSS-E-ST-70-01 standard, or OBCP, and the "Space Engineering – test and operation procedure language" ECSS-E-ST-70-32 standard, or FCP, and their corresponding CNES ISIS tailored versions, called respectively "ISIS Exchange Format for OBCP and Elementary Procedures", and "ISIS Operations Procedure Language". The experiment showed that it was possible to import/export SysML Activity Diagram using its corresponding profile from/to the ISIS Exchange Format for OBCP and Elementary Procedures.

Moreover, the operational procedures designer can be guided to use only the concepts he is used to, instead of learning SysML activity concepts (see figures 6 and 7 below). As soon as there is an UML profile defined for that, the tools can adapt and show the profile concepts rather than native SysML or UML concepts.

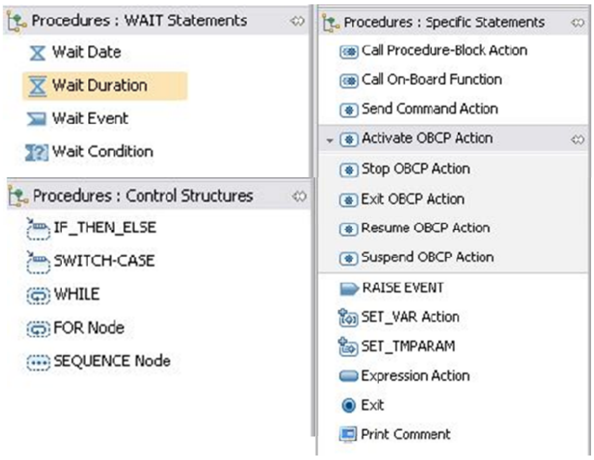

Figure 6: Diagram palette corresponding to the activity SysML profile

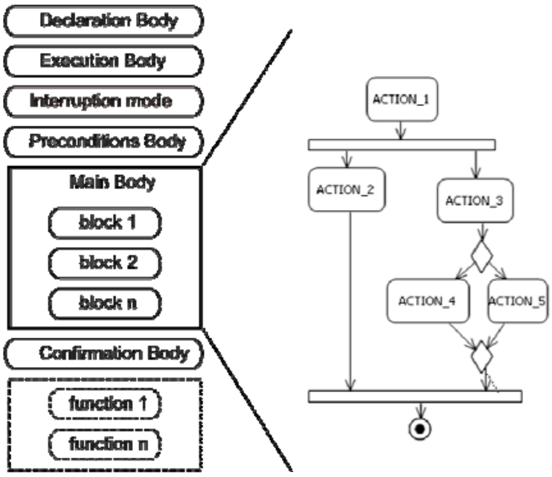

Figure 7: Resulting operational procedure described with a dedicated SysML profile

However, to be more complete for this space system operational view, we need also to map mission phases, operational modes, and equipment states to SysML concept and in particular the SysML *StateChart* concept.

Due to resources constraints, we have only carried out the mapping of the operational mode concept. Some additional work remains to complete this mapping.

Then, we have successfully experimented an assistance (wizard) and control (warning notification) provided by the tool to help the end user to also check editing rules in real time (see figure 8 below).

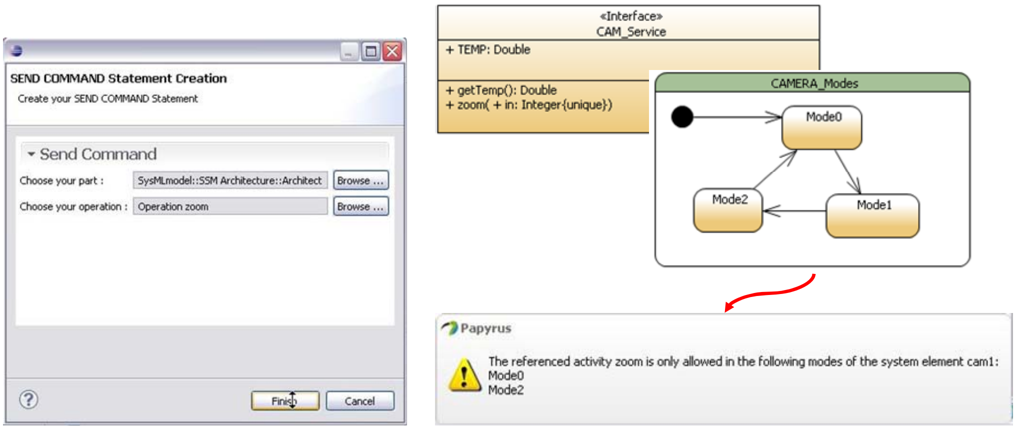

Figure 8: Wizard associated to the operational mode SysML profile

.

III. Conclusion and perspectives

A. The main results

One of the major result is that space domain concepts are relatively close to SysML.

A second one is the confirmation of the good maturity of the tools like the new Papyrus UML/SysML editor from open source TOPCASED toolkit to customize SysML using UML profile extension mechanism.

Building a uniform space system view is possible with the addition of UML profiles as a Domain Specific Modeling Language (DSML).

It is possible to hide the SysML and UML languages from the space domain designers and operators by refining palettes, editors, menu sticking to the space standards (CCSDS, ECSS) terminology and customizing it with additional dedicated graphical "business views".

B. Medium term perspectives

But what to do next with those first results?

Do we intend to replace existing specialized dedicated tools and editors by this new generic DSML editor?

Not at all.

The intend is to provide a *complementary* tool to manage the space system knowledge, keep it coherent with all the specialized formalisms or even documents, partially automate some of their production, and keep a continuous traceability with requirements during all the system life. This would help not only space system designers but also operational users.

Another perspective is that this system view may enhance all the existing specialized tools. Information in each specialized tool may integrate a system view for its data. It mainly consists in organizing information with a standard system view taking into account the different levels instead of having a flat organization. One

typical and simple example is the space sub-system hierarchical information that can be added to telemetry parameters. This approach may be generalized for all the space system information.

C. Short term perspectives

A second following study has been decided at CNES that will start this year to focus on two main activities:

- Complete the semantic links between the SysML and space system concepts
- Build advanced structural checks related to physical architecture view using OCL, etc. and behavioral checks related to operational mode constraints, delays, etc.

Appendix A
Acronym List

CCSDS	The Consultative Committee for Space Data Systems
DSML	Domain Specific Modeling Language
ECSS	European Cooperation for Space Standardization
FCP	Flight Control Procedure
ISIS	CNES project called Initiative for Space Innovative Standards
M&C	Monitoring & Control for both ECSS and CCSDS contexts
MATLAB	Numerical computing environment and fourth-generation programming language developed by MathWorks
MO	CCSDS Mission Operation services draft standards
OBCP	On-Board Control Procedure
OCL	Object Constraint Language: Declarative language for describing rules that applies to UML
OMG	Object Management Group: consortium aimed at setting model-based standards
PUS	ECSS-E-70-41 Telemetry and Telecommand Packet Utilization Standard
SSM	ECSS-E-ST-70-31 Space System Model
SysML	OMG System engineering Modeling Language. General-purpose modeling language for systems engineering applications
UML	OMG Unified Modeling Language. General-purpose modeling language in the field of object-oriented software engineering

Acknowledgments

E.P. author thanks Marie-Claire Charmeau from CNES, Raphael Faudou and Alexandre Cortier from ATOS for their active contributions to this work and this paper.

References

Books

[1]CCSDS, CCSDS_521.0-B-1 “SM&C Message Abstraction Layer”, CCSDS, ESA-ESOC Germany, 2010.

[2]CCSDS, CCSDS_522.0-R-2 draft “SM&C Monitoring & Control Service”, CCSDS, ESA-ESOC Germany, 2008.

[3]CCSDS, CCSDS_521.1-R-1 draft “Mission Operations - Common Object Model”, CCSDS, ESA-ESOC Germany, 2011.

[4]ECSS, ECSS-E-ST-70-32C “Space Engineering – Test and operations procedure language”, ECSS, ESA-ESTEC The Netherlands, 2008.

[5]ECSS, ECSS-E-ST-70-01C “Space Engineering – Spacecraft on-board control procedures”, ECSS, ESA-ESTEC The Netherlands, 2010.

[6]ECSS, ECSS-E-ST-70-31C "Space Engineering – Ground systems and operations monitoring data definition", ECSS, ESA-ESTEC The Netherlands, 2008.
[7]ECSS, ECSS-E-70-41-A "Space Engineering – Ground Systems and operations - Telemetry and Telecommand packet Utilization", ECSS, ESA-ESTEC The Netherlands, 2003.

Electronic Publications
[8]TOPCASED platform: http://www.topcased.org [cited 09 May 2012].
[9]OMG System Modeling Language (SysML), The official OMG SysML site, http://www.omgsysml.org/ [cited 09 May 2012].