MBSE & SysML Applied to the Development of EGSE for Satellites Assembly, Integration and Testing (AIT) - A Practical Case

Marcelo Coicev¹, Geilson Loureiro²

¹National Institute for Space Research, INPE, Brazil
Email: marcelo.coicev@lit.inpe.br
²National Institute for Space Research, INPE, Brazil
Email: geilson@lit.inpe.br

Abstract — This paper targets to present a proposal for the use of MBSE and SysML applied to a case study of the analysis in a component of a typical Electrical Ground Support Equipment (EGSE) used in Satellite Assembly, Integration and Testing (AIT).

The approach aims to describe the process flow used in the analysis, providing a methodological background for the practical application of SysML notation for EGSE development.

Keywords — Systems Engineering; Model Based Systems Engineering; MBSE; SysML; Electrical ground Support Equipment; EGSE.

I. INTRODUCTION

Although SysML, in recent years, has become the de facto standard for MBSE, a supporting methodological basis is still needed, as SysML is just a graphical language and defines a set of diagrams, modeling elements, syntax and semantics. Like any language (formal or informal), it can be used in many different ways, including inappropriate ones. Most notably, it is possible to misuse the language for creating unrepresentative or even incorrect models.

The flow of the analysis processes used in this paper seeks to implement, as much as possible, the sequence presented in the GSE Integrated Development Guide proposed by Vintecinque (2017), respecting the limitations imposed by the SysML notation language and the modeling tool used. (Cameo Systems Modeller).

The added value of the methodology with the MBSE approach consists of:

- To select a suitable subset of SysML diagrams and artifacts to be generated conveniently and pragmatically;
- To define semantics to ensure meaningful diagrams and rules to check the model for consistency;
- To define an obvious sequence of diagrams that ensures modeling efficiency in relation to organizational processes, and is well understood by all stakeholders throughout the life cycle;

II. METHODOLOGY - MBSE APPROACH TO THE GSE INTEGRATED DEVELOPMENT GUIDE

The process flow that will be used seeks to follow the major process phases of the Total Vision Framework proposed by Loureiro (2010), as shown below, as well as the SysML diagrams that will be used:

Table 1 - Phases of Analysis Processes and Modeling Approach

| Phase       | Subphase                  | SysML Diagrams                  | Modeling Approach                                      |
|-------------|---------------------------|---------------------------------|--------------------------------------------------------|
| Mission     | 1.1 - Context / Concept of Operations | Block Definition Diagram (BDD)   | Type Definitions (Parts Catalog)                       |
|             | 1.2 - Life Cycle / Life Cycle Scenario | Internal Block Diagram (IBD)    | Description of Structures (blocks), internal components, relationships and interfaces between Structures. |
|             |                           | Activity Diagram                | Definition of the phasing of processes and subprocesses. |
### Phase Subphase SysML Diagrams Modeling Approach

#### 2. Stakeholder analysis

| Subphase | Modeling Approach |
|----------|-------------------|
| Stakeholder Identification | Actors (used in Use-Case Diagrams) Are created as “types” in a BDD |
| Stakeholder Concerns | Description of stakeholder concerns in Use-Case diagrams, System or Interest Organization in various scenarios |
| Stakeholder Requirements | Stakeholder requirements and their dependencies are represented in Requirements Diagrams. |

#### 3. Requirements Analysis

| Modeling Approach |
|-------------------|
| Stakeholder requirements and their dependencies are represented in Requirements Diagrams. |

#### 4. Functional Analysis

| Subphase | Modeling Approach |
|----------|-------------------|
| Functional Interface | N² chart generated from interfaces defined in functional scope analysis |
| Functional Behavior | Tables with list of identified functions, modes, and states |
| Functional Architecture | Refinement of states and modes identified for functions |
| Establishment of the functional architecture | Establishment of control, data and material flows between functions and functional elements of the system. (*1) |

#### 5. Implementation Analysis

| Subphase | Modeling Approach |
|----------|-------------------|
| Physical Architecture | Using both types of block diagrams for physical architecture proposals |
| Function Allocation | Through "Allocate" type relationships and automatic generation by modeling tool |

#### NOTES *

1. The Guide proposes the establishment of functional architecture via Data Flow Diagrams (DFD's), which are not part of SysML. Using the Use-Case Diagram instead implies implementation limitations that need to be further evaluated throughout the application of the guide.
III. CASE STUDY - UMB SCOE ANALYSIS

As proposed by Vintecinque (2017) UMB SCOE is an element of EGSE proposed for future PMM platform satellite missions, which allows use during both the AIT and the Satellite launch phases, and aims to reduce the amount and volume of equipment to be transported to the launch base. The mission of UMB SCOE can be stated as:

“To be the only satellite-connected element of EGSE that allows to power up, operate and monitor its vital signs during the AIT and launch phases” Vintecinque (2017).

The approach in the analysis and modeling to obtain UMB SCOE will be presented next.

a. Mission Analysis

Starting from the mission statement, the possible concepts of operation, as exemplified in Fig. 1, are analyzed for one of the situations.

b. UMB SCOE Life-cycle

This analysis only states and establishes the sequence of expected life-cycle processes, as exemplified in Fig. 2.

c. UMB SCOE life-cycle scenarios

The UMB SCO life cycle scenarios are shown in Fig. 3. The activity diagram was used, with the inputs or controls of the old IDEF0 being replaced by SysML “Accept Event Action” or “Time Event” type elements.

Fig. 1 - Example of UMB SCOE operation concept for telemetry and telecommand data link in the launch tower (internal block diagram).

Elements and their interactions are described textually through the appropriate fields of the model. Flows of energy, material or information from an early point of view are represented as “Information Item” stereotyped elements, denoting the directions of the flows. It does not matter at the moment the physical concept of interfaces.

d. Stakeholder Analysis

i. Identification of UMB SCOE Stakeholders

For identification of product and process stakeholders, block definition diagrams (BDD) will be used with SysML stereotyped actors, as illustrated in Fig. 4.

Fig. 2 - UMB SCOE (activity diagram) life-cycle example.

Fig. 3 - UMB SCOE life-cycle scenarios example (activity diagram).

Relevant scenarios within the development effort will be reviewed by the UMB SCOE developer organization.

ii. UMB SCOE stakeholders concerns

Product and process stakeholder concerns raised previously are analyzed together with the System or Organization of Interest in various scenarios. The result is described through Use Case diagrams stereotyped as “Concern”. Fig. 5 illustrates an example of how this will be done.
iii. UMB SCOE stakeholder requirements

Stakeholder requirements derived from the concerns raised before are analyzed and the outcome is described through requirements diagrams or requirements table as exemplified in Fig. 6. Dependency or trace-ability relationships between requirements and stakeholder concerns can be made explicit in this kind of diagram. Additional attributes can be added to the requirements by the use of Tagged Values provided by the SysML notation.

iv. UMB SCOE measures (MoE / MoP / TpM)

Measures of Effectiveness (MoE) are operational measures of success closely related to the achievement of the mission objective being evaluated and they are derived from the mission objectives related to the stakeholders concerns.

Measures of Performance (MoP) are measures that characterize physical or functional attributes relating to system operation, measured or estimated under specified testing and/or operational environment conditions.

Technical Performance Measures (TPM) measure attributes of a system element to determine how well that element is satisfying, or expected to satisfy, a technical requirement.

The SysML parametric diagrams will be used for the MoEs, MoPs and TpMs. It is desired that the measures could be quantitatively assessed, in a form that could be estimated or even simulated in supporting tools, integrated with the modeling tool. In order to allow this, the default SysML meta-model needs to be extended in order to allow then to be expressed in Value Types, but also in a textual description, which can be traced back to the Stakeholders Concerns and Requirements. This can be achieved by a Custom Profile which extends the SysML meta-model, with MoEs, MoPs and TpMs Specification elements, as shown in Fig. 7.

Fig. 5 - UMB SCOE Stakeholder Concerns Example

In the UMB SCOE scenario under development (Use Case Diagram).

Fig. 6 - UMB SCOE Product stakeholder requirements example (Requirements Diagram).

Fig. 7 - MoEs, MoPs and TpMs Specification Meta-Model.

The MoE, MoP and TpM specifications can be traced back to its source as the example shown in Fig. 8.

Fig. 8 - Deriving Specification for MoEs Example.

Finally, the MoE, MoP or TpM quantitative metrics can be expressed by Value Types (Properties), as shown in Fig. 9.
Fig. 9 - Parametric Diagram for MoEs of interest.

e. UMB SCOE Requirements Analysis

From the stakeholder requirements and MoEs defined before, as well as the assumptions that emerged during their analysis, the technical requirements (both for product and organization) are derived, but now from the UMB SCOE point of view. Similarly, requirements analysis is described through requirements diagrams or requirements table, as exemplified in Fig. 10.

Fig. 10 - UMB SCOE Product Technical Requirements Example (Requirements Diagram).

f. UMB SCOE Functional Analysis

i. UMB SCOE boundaries/interfaces identification and environment modeling

To identify system boundaries, relevant scenarios are chosen within the product and organization life cycle of interest (Vintecinque, 2018).

Scenarios and circumstances for the product and organization of interest will then be described as blocks and their interfaces with the environment by using internal block diagram (IBD), as illustrated in Fig. 11.

From the analysis of circumstances, it is possible to identify the events and expected responses of the system, as well as the associated measures of effectiveness.

Fig. 11 - UMB SCOE Environment Modeling Example during scenario U33: Launch Operation and its Circumstances (Requirements Diagram).

ii. UMB SCOE functions definition

For each scenario and circumstance identified, all flows of energy, material and information are gathered and, from them, the system's external functions are identified, which can then be listed in a generic SysML table, as exemplified below:

Table 2 - Example of definition list of functions identified for UMB SCOE (generic SysML table)

| # | Name | Documentation |
|---|------|--------------|
| 1 | F1: Provide Satellite Umbilical Interface and EGSE | UMB SCOE must be EGSE interface with satellite umbilical connector during AIT and Launching |
| 2 | F2: Protect the Satellite | Minimize fault propagation and reduce the severity of failure effects on satellite and other EGSE elements |
| 3 | F3: Signals Monitoring | Monitor vital satellite signals present in the umbilical connector and important signals to Status including EGSE signals. |
| 4 | F4: Command | Generate on/off command pulse for satellites Through Umbilical Cable, or Through simulation devices (eg, separation simulation |

iii. Scope analysis of UMB SCOE functions

Scope analysis of each function helps to identify inputs and outputs and scope limits for previously identified functions, as shown in Fig. 12.

Fig. 12 - F1 Function Scope Analysis Example for UMB SCOE (Use Case Diagram).

iv. UMB SCOE functional interfaces identification

From the analysis of the inputs and outputs of each function it is possible to establish the interfaces between the functions.

Normally, this is represented through an N² chart (a.k.a N² diagram) which are not part of the SysML standard. But the main idea can be achieved through the use of internal block diagrams (IBD), in a matrix form, with the same rules of an N2 chart:

1. All Functions (or sub-functions) are on diagonal;
2. All Inputs are vertical (to down or to up)
3. All Outputs are Horizontal (to left or to right);
4. Inputs and outputs are items, not functions;

Fig. 13 Gives an example of this diagram used for some of the UMB SCOE functions.
v. UMB SCOE states and modes definition

From the functions identification, it is possible to identify states and modes of each function external to UMB SCOE. The definition is made by State Machine diagrams, for the purposes of tracing the states and modes to the functions defined before. The final representation is made by using a generic SysML table, as the example shown in Table 3.

Table 3 - UMB SCOE Function State Analysis Example (Generic SysML Table)

| # | Functions / Modes | Name | Documentation |
|---|-----------------|------|---------------|
| 1 | F1: Provide Satellite | SIMULATED MODE | Connected | Cable connected to Satellite |
| 2 | F2: Protect the Satellite | REAL MODE |_Alert | Alarm Cable |
| 3 | F3: Signals Monitoring | REAL MODE | Monitoring Disabled | For each protection circuit (Initial Mode) |
| 4 | F4: Command | REAL MODE | Command-Active | |
| 5 | F5: Monitor | REAL MODE | Monitoring Running | |
| 6 | F6: Configure | REAL MODE | Monitoring Enabled | |
| 7 | F7: Control | REAL MODE | Monitoring Disabled | |
| 8 | F8: Test | REAL MODE | Monitoring Enabled | |
| 9 | F9: Test | REAL MODE | Monitoring Disabled | |
| 10 | F10: Test | REAL MODE | Monitoring Enabled | |
| 11 | F11: Test | REAL MODE | Monitoring Disabled | |
| 12 | F12: Test | REAL MODE | Monitoring Enabled | |
| 13 | F13: Test | REAL MODE | Monitoring Disabled | |
| 14 | F14: Test | REAL MODE | Monitoring Enabled | |
| 15 | F15: Test | REAL MODE | Monitoring Disabled | |
| 16 | F16: Test | REAL MODE | Monitoring Enabled | |
| 17 | F17: Test | REAL MODE | Monitoring Disabled | |
| 18 | F18: Test | REAL MODE | Monitoring Enabled | |
| 19 | F19: Test | REAL MODE | Monitoring Disabled | |

vi. UMB SCOE functional behavior analysis

After identifying the states and modes of operation, they are then refined using SysML state machine diagrams for each function external to the UMB SCOE. The Fig. 14 shows an example of that.

Fig. 14 - UMB SCOE Function State Analysis Example (State Machine Diagram).

vii. UMB SCOE functional architecture establishment

During functional behavior analysis, it is possible to identify the possible failures in each of the flows and thereby identify preventive and protective functions for the failures. (Vintecinque, 2017). After that, it will be possible to map how functions interact, and perform functional partitioning that will provide an "allocable" generic functional architecture, allowing architectural decisions to be taken, for the product and organization of interest. In the guide proposed by Vintecinque (2017) DFD diagram is used. With SysML, the use case diagram will be used, with some restrictions, use of stereotypes and minor implementation differences, as illustrated in Fig. 15.

Fig. 15 - UMB SCOE functional architecture example - Command and Monitoring (Use Case Diagram).

g. Implementation Analysis

i. UMB SCOE physical architecture establishment

At this stage, from the functional architecture, through the two types of block diagrams from SysML (BDD and IBD), it is possible to propose a viable generic architecture that seeks to satisfy all that has been raised before, and Fig. 16 represents this step. In the Guide proposed by Vintecinque (2017), the physical architecture
for UMB SCOE was not proposed, which will then be done from now on.

Fig. 16 - UMB SCOE Generic Physical Architecture Example (Use Case Diagram).

ii. UMB SCOE physical architecture’s functions allocation

Function allocation can be done through "allocate" SysML connectors, in block definition diagrams, which do not necessarily need to be documented (they can be temporary). From the relationships made between the functions and blocks of the physical architecture, the allocation matrix can be generated directly from the modeling tool, as illustrated by Table 4.

Table 4 - UMB SCOE Function Allocation Example - (Allocation Matrix)

iii. Trade-off analysis

This type of analysis is still open in the ongoing study. At first, the intention is to make the trade-off analyzes for “product” by using several block definition diagrams with different practical solutions (Product BreakDown Structures) for the same physical architecture, with estimates and simulations for different vendors / data acquisition solutions, protection systems, etc.

The approach most appropriate in this case seems to be taking advantage of the characteristics of SysML parametric diagrams, as well as external tools integration and simulation, in order to simulate various scenarios of cost, ease of implementation, material availability, technical performance, etc., besides of morphological charts with adequate criteria and weights for each component in order to achieve a balanced Product Specific Physical Architecture, but this issue is beyond the scope of this study effort.

IV. CONCLUSION

The use of MBSE through the notation language SysML, supported by the use of appropriate modeling tools, allows to cover virtually the entire product and organization systems engineering life-cycle, obviously while respecting the limitations of the notation itself and maturity of its utilization, as well as the different methodological implementations that make use of it.

There are still difficulties in applying the notation fluidly with respect to the non MBSE approaches used in previous working frameworks, but future versions of the notation itself, such as SysML V2, as well as its future adoption by vendor modeling tools can simplify and better tailor their use more widely.

REFERENCES

[1] Kaslow, David & Ayres, Bradley & Cahill, Philip & Hart, Laura. (2018). A model-based systems engineering approach for technical measurement with application to a CubeSat. 1-10. 10.1109/AERO.2018.8396443.
[2] Cameo Systems Modeler 19.0 LTR SP2 User Manual, 2019. 573 p. No Magic, Inc., a Dassault Systèmes company.
[3] Loureiro, G. A system engineering and concurrent engineering framework for the integrated development of complex products, 1999. 530 p. Thesis (PhD in Systems Engineering) - Loughborough University.
[4] Venticinque, Guilherme. Engenharia de sistemas aplicada ao desenvolvimento do equipamento de suporte em terra - GSE, 2017. 400 p. Dissertação de Mestrado em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais Instituto Nacional de Pesquisas Espaciais.
[5] Crochet, Brad. A case study in the application of model-based systems engineering to laboratory research science, 2017. 72p. Dissertation for the Degree of Doctor of Philosophy, Te University of Southern Mississippi - Te Aquila Digital Community