Systems Engineering – Classical vs. Model Based Approach and First Lessons Learned

Katarina Jesswein [1], Michael Brahm [1], Stephan Jahnke [1], Mohammad Chami [2]

[1] *OHB System AG, Universitätsallee 27-29, 28359 Bremen (Germany) Email: katarina.jesswein@ohb.de michael.brahm@ohb.de*
[2]*SysDICE GmbH, Franz-Volhard-Str. 5, 68167 Mannheim (Germany) Email: mohammad.chami@sysdice.com*

MBSE2022 Objectives: O-1, O-2, T-1

INTRODUCTION

This paper presents the Model Based Systems Engineering (MBSE) methodology, developed at OHB System and its first deployment as a pilot project in a phase A of a quantum key distribution (QKD) mission. In this pilot project the MBSE implementation is running in parallel to the classical, document-based systems engineering (DBSE) approach and offers comparison of both and the derivation of lessons learned.

CLASSICAL SPACE SYSTEMS ENGINEERING

Within the last years a major part of our systems engineering activities included definition and verification of hand-written, textual requirements. These requirements were defined for different decomposition levels of the system to be developed, connected via traces to justify low-level requirements and to show completeness and to be able to impact analyses. Theses requirement specifications were frequently accompanied by manually created figures or reference documents giving background information or, eventually, refining requirements further.
Even when choosing a common requirements management tool, like IBM Rational® DOORS®, and agreeing on a set of common quality criteria, like testability, traceability, justifications, definition of “atomic” requirements, “shall” statements, etc. the following major problems can be observed:

- **missing or ambiguous relation between requirements**
- **missing or unclear semantics**
- **missing digital continuity**

The **missing or ambiguous relations between requirements** partially arise from the use of a table-based creation of requirements eventually resulting in a big list of requirements which are initially not connected to each other. This could be circumvented by manual measures, like rules for headings or using traces within the same specification, but, in the end, this would be manual work with no immediate support by the requirements engineering tool.

The **missing or unclear semantics** become apparent when looking at figures which are added to the specifications, sometimes to compensate missing connection between atomic requirements. These figures are created by additional tools and integrated pictures into the requirements. Being pictures, the content of these pictures has no connection to the hand-written requirements, except for, manually maintained, identical names which could be found in the figures and the textual requirements. Furthermore, these figures are often created by different domain experts and no common semantic is defined for the picture elements, e.g.: What is the meaning of a rectangle? What is the meaning of a rectangle with rounded edges? What is the meaning of a line? What is the meaning of a dotted line? These semantics could be standardised for a given project or the whole company, but this would be significant work and the rules would still have to be checked and applied manually.

The **missing digital continuity** is caused by specifying and designing the system mostly based of pure text and within many, only loosely, connected documents. The text has a meaning within its immediate (e.g. inside the requirement) but also extended context (e.g. the entirety of documents within a project), which need to be considered to understand its exact meaning. In space projects, many engineers work together, often focussed on a particular aspect of the system, but with interfaces to other domains. While working with text-based, distributed specifications it is very difficult to judge whether the same work used in different documents really has exactly the same meaning, because there is no additional sematic defined. Even if using consistent naming was targeted, for example by respective wording rules, renaming of elements easily leads to outdated documents, because adapting the names usually is manual work.

In summary, many of the previously mentioned aspects can be improved by defining detailed rules for the specification and the design of a system and checking these rules by additional tools which interpret the text. Nevertheless, semantic consistency and digital continuity can only be reached if the system design language itself allows to define semantics of system description elements and if the used engineering tools support applying and checking these semantics during the system engineering process.

MODEL-BASED SYSTEMS ENGINEERING @ OHB

The term MBSE comprises multiple modelling concepts: modelling language, modelling method, and modelling tool in order to produce one systems model or more. A systems model contains model elements (e.g., requirements, functions, logical components…) and relationships in between (e.g., refine, allocate, derive...). Indeed, MBSE does not necessary change the “what to do” by systems engineers, instead changes the “how to do it”. Particularly, MBSE goes beyond the DBSE approach by considering the use of systems models instead of documents as the primary artifacts produced during the life cycle activities. Moreover, such models are specified, reviewed, and released using a systems modelling tool (following a modelling language such as SysML) and not just a drawing or documentation tool as Visio, PowerPoint or Excel.

The OHB MBSE Method is organized by two main dimensions, which are the **Engineering Level** and the **Engineering Perspective**. The **Engineering Level** distinguishes different decomposition levels of a System of Interest (i.e. a system to be developed), typically Overall System Level, System Level, Subsystem Level, Equipment Level. For a Spacecraft, these levels could be: Mission Level, Spacecraft Level, Subsystem Level, Equipment Level.
Depending on the responsibility of the system design authority, additional levels can be added as the engineering approach is designed to be fully recursive. The **Engineering Perspectives** define which aspects of a system shall be designed in the frame of a model-based systems engineering process. These Engineering Perspectives are used for each Engineering Level, unless explicitly excluded for a given level.

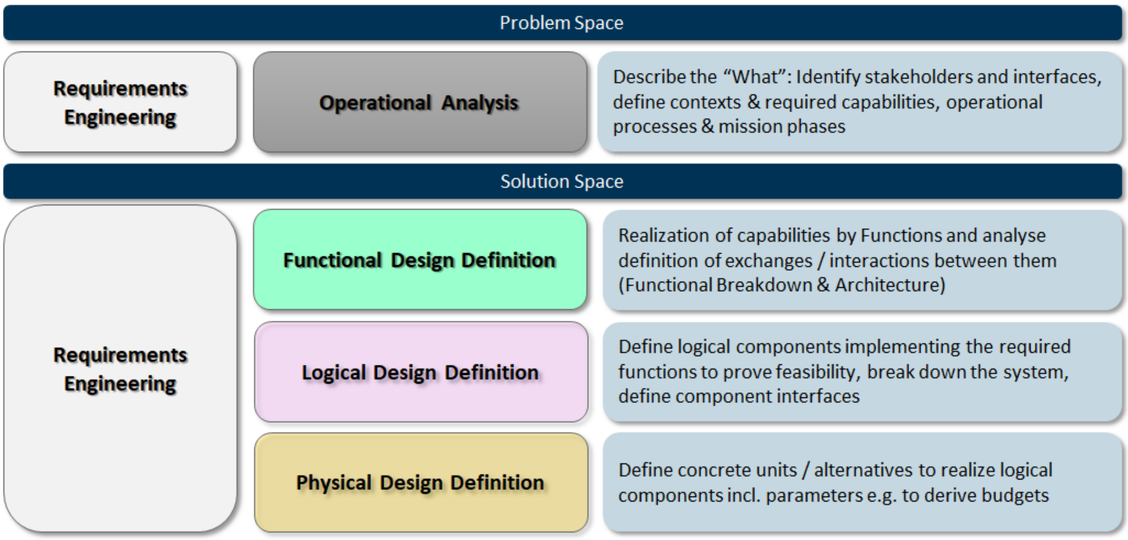

Fig. 1. OHB MBSE Method Overview: Main engineering perspectives structured in Problem and Solution Space

The Engineering perspectives, as summarized in *Fig. 2*, are divided into a Problem Space which focusses on defining the "problem" to be solved by the system to be developed and the Solution Space, which defines how this problem will be solved.

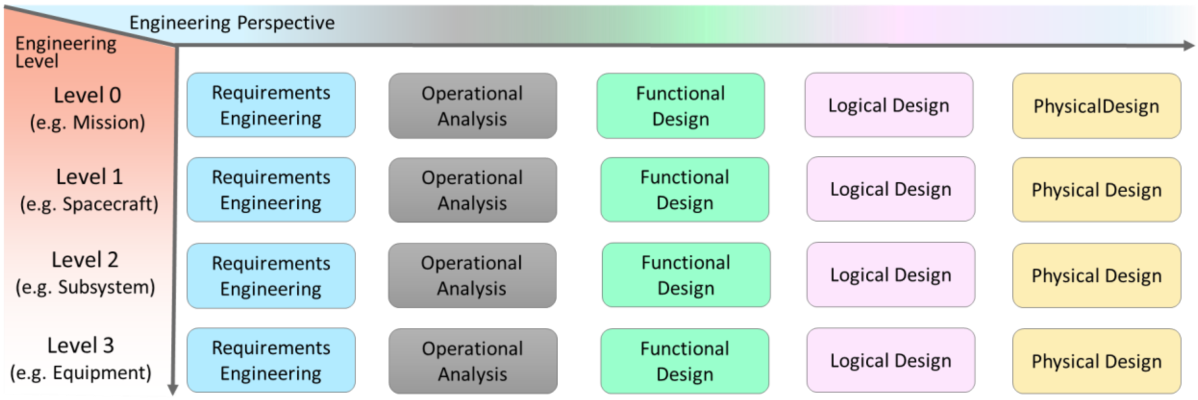

Fig. 2. OHB MBSE Method Overview: Recursive & adaptable application of engineering perspectives for the typical engineering levels

For each engineering perspective the OHB MBSE method defines all *method steps* to be executed to ensure that all necessary inputs are available when working of subsequent Engineering Perspectives. Transitions between two Engineering Perspectives (as per **Fig.2**) are usually performed from left to right, on the same Engineering Level, or vertically, crossing the Engineering Levels. The OHB MBSE Method defines for each of the *method steps*: the questions to be answered, the used inputs, the generated outputs, the involved model elements & diagrams and a justification for having this *method step*.

The OMG Systems Modeling Language (OMG SysML™) is used as the systems modelling language and the OHB MBSE method is supported by Cameo System Modeler (CSM) profile which defines stereotypes for model elements to be used when performing the *method steps*, icons, colour codes, toolbar elements, standard attributes of model elements, etc. These stereotypes enforce the desired semantics on top of the SysML semantics and lead to improved readability and reuse of models among different projects. While performing MBSE, a common model library is built-up, which can be used as a template and starting point for upcoming projects.

QKD PILOT PROJECT IMPLEMENTATION

The MBSE activities on the pilot project have been started mid-way through the phase A of the QKD mission and are kept as a parallel activity to the classical DBSE approach of this project. This offers the unique opportunity to compare both approaches and identify potential benefits and drawbacks. In addition, the team gained experience on working with the OHB methodology in a project and identified lessons learned, gaps or areas of improvement. The Systems Model includes firstly the data transformed and enhanced from the document-based practices of the project and this includes **(c.f. Fig. 3)**:

- The customer and system requirements imported from the requirements management tool IBM DOORS into CSM.
- The operational analysis elements including the operational contexts under scope, operational entities, exchanged items, and the capabilities with their activities
- The functional analysis elements including the functional blocks and their interfaces and exchanged items
- The logical analysis elements including the logical blocks and their interfaces and exchanged items
- The security risk analysis elements aiming at only translating the document-based information and tracing it to the logical analysis

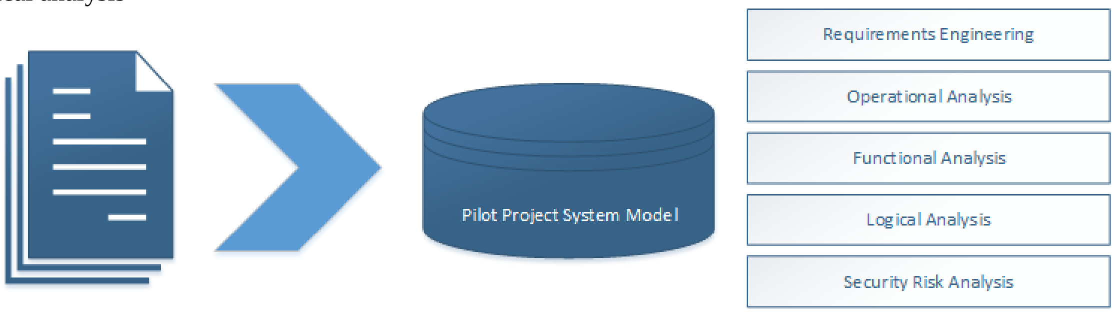

Fig. 3. Data Transformation into the Pilot Project Systems Model

LESSONS LEARNED

The lessons learned from the implementation in the MBSE pilot project can be structured into three areas: the data factors, the project factors and the human factors. The **data factors** include the model elements and their structure as well as the model relationships. Lessons learned on data factors are:

- **Data Structure:** For complexity management and knowledge interoperability the systems models elements need to be structured in an optimized manner through the applied method: evaluate and decide for the suitable and common data structure to be used before starting with modelling.
- **Data Traceability:** The usage of the suitable model relationships between the different types of data enables representation of the data linking and enhances change management. It is essential to define unique and minimum traceability relations with their own benefits and usage, avoid duplication of traceability links to ensure a minimum of traceability relations.
- **Data consistency and reusability:** It is difficult to keep the modelled data consistent and reusable when multiple personnel are involved simultaneously in the modelling activities. There are solutions to automate the validation and guarantee the data consistency and reusability, such as the usage of validation rules.
- **Identifying the need for Variability Management:** The need of defining and managing commonality and variability of the system under consideration has been identified to enable the trade-off analysis of different architectural solutions.
- **Integration between DOORS and CSM:** The integration between requirements management tools and systems modelling tools is a key for an optimized model-based requirements engineering activities.

The **project factors** concern the project work of the pilot project. These span over project management aspects and tool related topics:

- **Identifying the MBSE Adoption Scope:** One of the key challenges in MBSE adoption is to define a clear purpose and scope (i.e., the why and what). Ideally it must be precisely described before beginning with the MBSE deployment on projects
- **Collaboration between MBSE Development and Deployment:** The inputs provided from the team working on the MBSE Solution Development were helpful to accelerate the initial phase of MBSE deployment for the pilot project. The feedback collected from the MBSE deployment in the pilot project has been considered with the developments of the MBSE Generic solution at OHB.
- **Customer support related to MBSE Adoption:** Provides a leading initiative and support change management within an organization transforming from document-based to model-based systems engineering.

- **MBSE Model Views Presentation:** It is a key aspect to identify which stakeholders are involved in the projects and collect their expectations from MBSE adoption. It is important to identify and define which view is suitable for which stakeholder concern.
- **Adding Non-model Content to the Model**: It needs to be decided which model elements (e.g., documentation for exchange items) will the MBSE model be the single source of truth including their documentation and other attributes and which model elements will be documented outside the MBSE model.
- **Advanced MBSE Tools Adoption:** Consider the implementation of the MBSE Wizard before deploying MBSE on a specific project. Motivate the investment of the wizard implementation especially in how it can reduce the onboarding of new personal into the modelling activities and how to automate several manual activities.
- **MBSE Adoption Strategy on Projects:** Decide which MBSE adoption strategy will be chosen: On-Cycle or Off-Cycle. Avoid the combination of both strategies. In case On-Cycle is chosen, MBSE should be consider as a primary activity for the dedicated systems engineering activities.
- **Translation of Information from Documents into Models:** One of the key MBSE activities performed during pilot project was to transform the information from existing documents or personal experience and knowledge into the systems model. As information in the system models is defined in a unique and consistent manner, lack of consistency in the information sources can be identified.

MBSE triggered a constructive "system thinking" to analyse the problem space in order to describe the possible solution space. It enforces indirectly a common way of working among the team members, especially model developers and model readers. The lessons learned on the **human factors** are:

- **System Thinking and Mindset Change:** In classical approach the problem space level is normally shortened due to project timelines. MBSE instead ensures the required effort spent on the problem space phase to achieve the understanding of the problem before matching available solutions for unknown problems.
- **Thinking Specific and Achieving Completeness:** The usage of the specific MBSE method activities supports the involved personal in focusing on the specific aspects in order to describe the required specification.
- **Engineering Culture Change Management:** MBSE can trigger a huge human-factor change across the involved team. It is essential to manage this change and reinforce its states' transition.
- **MBSE Learning Curve:** MBSE similar to any new competence requires time and effort to learn. Consider the effort and time required for the learning curve of MBSE and that the learning curve differs among modelers with different backgrounds and experience.

IDENTIFIED AREAS FOR IMPROVEMENT

The QKD pilot project was able to benefit from the early OHB MBSE method developments and deploy the first releases of the method and profile implementation, collect feedback and proof the benefits of MBSE.

Identified areas of improvement are the adoption of **MBSE for the Security** related topics, especially Security Risk Analysis. Hereby, the approach is to use the traditional Security Risk Analysis and transform it into the suitable model-based approach in order to integrate it with the systems model.

Further improvements have been especially identified on the methodology and tool side. Several modelling tasks might be automated using the generation of templates by extending the actual profile with the required CSM **Wizards**. This will reduce the modelling effort and automate regular tasks.

CSM also supports the mechanism of creating own **validation rules** in the model and using them to check the model based on predefined constraints, e.g., check if a specific modelling element is documented. The usage of validation rules gives the modeler for confidence to ensure that the defined models are valid.

Although automatic document generation from the system model is not part of the scope of the pilot project, it is suggested to consider it as part of the areas of improvements due to its return of investment by reducing the high effort of data documentation beginning replaced with the automatic generation from the model with specific viewpoints. The document generation could also include the data validation rules to ensure that the data generated is valid with respect to pre-defined validation rules.

CONCLUSION AND NEXT STEPS

The adoption of MBSE for the pilot project was a complementary approach to the classical DBSE approach and allowed the test of a first implementation of the OHB methodology into a project. On the other hand, the direct comparison of both work approaches was possible in order to understand which parts of the methods from the classical approach can be replaced with MBSE methods or even extended. Clear benefits have been identified for the pilot project leading to the consideration of continuing with the MBSE model in the next project phase not as a parallel activity but on-cycle as part of the systems engineering. The lessons learned during the QKD project will be considered for the further OHB MBSE method development.