Transdisciplinary Lifecycle Analysis of Systems
R. Curran et al. (Eds.)

doi:10.3233/978-1-61499-544-9-100

Using the "Model-based Systems Engineering" Technique for Multidisciplinary System Development

Carolin ECKL[a,1], Dr. Markus BRANDSTÄTTER[ab] and Dr. Josip STJEPANDIĆ[b]
[a] *Technische Universität München, Institute of Astronautics, Garching (Germany)*
[b] *PROSTEP AG, Darmstadt (Germany)*

Abstract. "Model-based Systems Engineering" is currently a hot topic at INCOSE (International Council on Systems Engineering). It involves multidisciplinary development based on the usage of models as main artifact. The frequent use of models during the development of the pico-satellite MOVE (Munich Orbital Verification Experiment) was attributed to the long history of the chair for astronautics at the TU München with Systems Engineering. The development of MOVE displayed many of the characteristics of a real-world multidisciplinary engineering project and resulted in a successful space flight of the engineered satellite. Within the satellite, communication was lead through a central bus between the different components and required expertise and coordination from all of the involved disciplines. An equivalent task of distributing information and energy can be found in automotive engineering: in the wire-harness. In contrast to the satellite bus, it does not distribute centrally created coordination commands, but supports the orchestration between distributed systems. Even though these two systems and their development processes are inherently different, they exhibit similar difficulties during their design phase (e.g. with compatibility) and can be modeled similarly. This paper uses the design of satellite bus systems and automotive wire-harnesses as examples, describes their common pitfalls, explains "Model-based Systems Engineering" and demonstrates how the development of communication systems in both satellite and automotive engineering can benefit from relying on it in early design and concept phases.

Keywords. Systems Engineering, Model, Model-based Systems Engineering.

Introduction

The development of most technical products involves specialists from different engineering domains. A modern communication interface, for example, requires both electrical knowledge to transmit signals as well as an abstract understanding of the protocol and the hardware required to send/receive the signals. Additionally, flexibility, connectivity and performance gains demand this separation into the realms of different domains to be realizable at all.

The number of domains involved in the development of a product influences the number of different components, because a typical breakdown of the tasks of a system regards the boundaries of domain knowledge. The increase in the number of different

[1] Corresponding Author, E-Mail: c.eckl@tum.de

and new components as well as the flexibility required from each of the components increases the complexity of the system [1].

Automotive development is at one of the extremes of complex product development as it requires a lot of very flexible components, which interact in various configurations and variations [2]. At the other end of the spectrum, satellite development produces complex systems, where many components have to be newly developed and are specifically engineered to interact with each other. Most other engineering domains exhibit some characteristics of both types of domain presented in this paper [3]. The most important common feature of both domains is the involvement of engineers from various domains.

As a result, methods for fostering multidisciplinary cooperation and alleviating the risks introduced by these challenges have been on the agenda of both engineering branches for some time, e.g. by Model-based Systems Engineering (MBSE) at INCOSE (International Council on Systems Engineering) [4,5].

For example, MBSE for multidisciplinary teams has been prototyped by the German chapter of INCOSE (GfSE [6]), an organization with origins in the space industry in the "Telescope systems modelling by SEˆ2" and "Space Systems Modelling" [7] projects.

1. Engineering (Bus) Systems Differently

Differences in engineering between automotive and space derive mainly from the differences in the contexts and are detailed in the following subsections for the example of their bus systems.

1.1. Satellite Engineering and Bus Systems

Almost all satellite development is initiated by a customer order. The customer's use cases provide the basis for the requirements analysis. Resulting requirements reflect the wishes of only one customer for a certain purpose. Typically, this customer-driven development leads to the engineering of a single (or a few similar) satellites without the need for variation. Reliability is typically one of the highest aims due to high system costs and impracticality of repair in orbit.

Satellites are built with a similar general structure [8], which contains a bus system [9] that comprises all components, which contribute to the life support of the satellite (e.g. power unit or the attitude control system). Equally important is the payload of the satellite, which is defined by the satellite mission. For example, if the mission was to take pictures of the earth, the payload would contain a camera to take them. Additionally, the satellite typically contains a mechanical connection of all components of the satellite (the structure), a power supply such as solar cells and battery, an attitude determination and control system (ADCS), which orientates the satellite in space, a communication unit for communicating with the ground station, a central steering unit (the on-board computer) and a thermal system, which regulates the temperature budget of the satellite. The full system "satellite" also comprises the launcher and the ground station, which largely contribute to the success of the satellite's mission.

Few suppliers are involved in the development of one of these spacecraft. For the sake of certification of the whole system, each of them has to provide full documentation of the delivered components. Especially the payload of the satellite is

usually created by one highly specialized supplier (in case of scientific satellites it is typically the customer), who develops independently and delivers the built and tested payload. In parallel, the satellite structure is created by the developer of the satellite specifically for this satellite system. It includes a specific bus system, if no commercial bus system fits the purpose. Finally, the payload and all the other satellite components are integrated and tested as a whole.

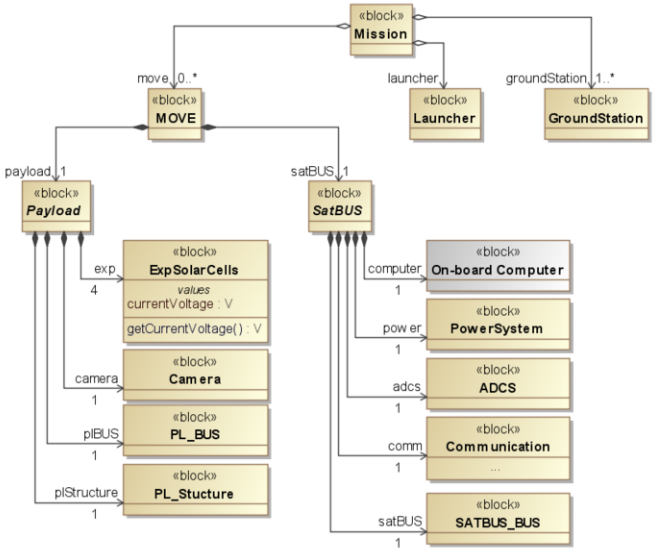

Figure 1. Simplified structure of the satellite system "MOVE".

All parts of a satellite are steered centrally by the "on-board computer" (which is part of the satellite bus). Its signals are distributed and routed to the components of the satellite through the satellite harness. Because of the dedicated master represented by the central "on-board computer", no special coordination of components for signal transmission and bus arbitration on the harness is necessary. This architectural feature contributes to the determinism and testability (and therefore reliability) of the bus system – mutual interferences and unwanted communication via the satellite harness are improbable.

Each of the signals transmitted by the bus system has two facets: an observable electrical manifestation on the satellite harness and the contained information. The information can be viewed as software signal, which is virtually transmitted between the encoder (converting the information to electrical signals) and the decoder (converting the signal back to information). Even though there are commercial-off-the-shelf alternatives for (partial) bus systems in satellites, the higher communication layers of the harness conveying more abstract information rather than signals have to be created specifically to allow communication with the specialized payload. In order to send and receive correct information on all of these communication layers, a close communication between the supplier of the satellite bus and the developers of the components is required.

Another mechanism to increase the determinism is the state-based behavior of the satellite. This means that the satellite has at least the states "initialization mode" (the initialization phase), "nominal mode" (normal operation) and "fail-safe mode" (for error handling)[8,10]. The "on-board computer" knows the required actions for all of these states and distributes the knowledge about the current mode to the components. A transition from one state into another requires a certain trigger and/or condition to hold.

Almost all of the standard satellite components are also visible in the student project to engineer the very small satellite MOVE [11], which has been developed at the chair for astronautics of the TU München. The development exhibited many of the characteristics of a real-world satellite development in its multidisciplinary approach and resulted in a successful space flight. The models, which were created after the development, show the structure and behavior of this concrete satellite.

In general, the development of technical systems such as satellites in the space industry is steered by a Systems Engineering group [12], which is responsible for the coordination and distribution of design information. It collects design information and generates an abstract model of the whole system. The objective of the model is to provide an overview of the system for involved engineers: the general context, behavior and structure of connected components [13].

Especially the use of an overview model during the early development phases has been well tested in the so-called concurrent design facilities [14]. During the course of the development, the Systems Engineering group continues to enhance the model. If connections to development models (such as models from CAD (Computer-aided Design), FEM (Finite Element Method) or Software-descriptions) are required, they are handled by links (e.g. via the OSLC (Open Services for Lifecycle Cooperation) protocol [15]). These development models detail the components, which have an abstract representation in the system model.

This system model is best used throughout all development phases and especially during the early system conception and for the central component. In the case of MOVE, it is engineered in SysML (Systems Modeling Language) [16] (as displayed in Figure 1), which is used for all models in this paper.

Figure 2 displays the detailed internal connections of the satellite including the satellite bus system and its connection to the payload.

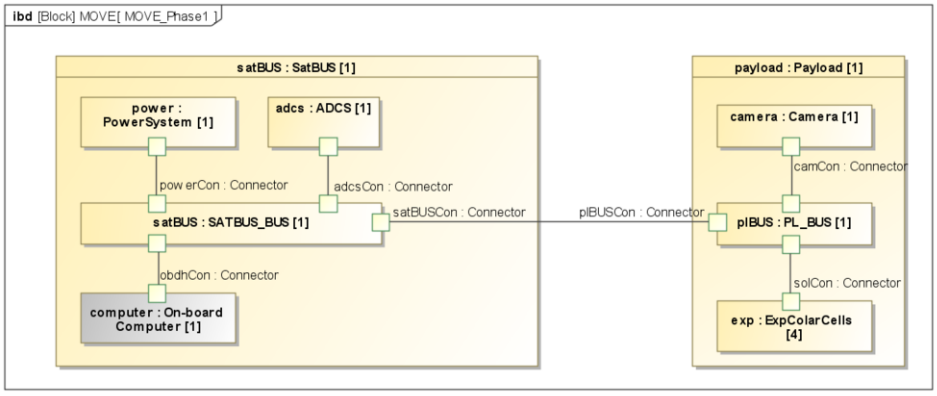

Figure 2. Details of the connections within the satellite. Usages of ports - these connections are realized in the system.

1.2. Automotive Engineering and its Bus Systems

The trigger of a development project in the automotive industry is not a customer order, but comes from the organization itself based on market analysis and studies. Abstract use cases for the product have to be anticipated.

The development is based upon a "master plan", which contains all components of the car and is detailed during the course of development. The master plan lays out the component development on a time schedule, but does not track connections between the developed components. Geometrical aspects of the components are also captured in a common model that provides a sketch of the completed car, but does not hold any invisible, intangible information such as behavior or software. There is no concrete central model of the system, which could provide an overview of non-geometrical connections between the components of a car.

A lot of specialized design models (i.e. mechanical & electrical CAD-models) are created during the early phases. The documents containing these engineering models are coordinated by Product Data Management (PDM - see e.g. [17]) systems, which contain all of the required information and may be exported to other systems. These systems contain the references to the separate models and provide possibilities to create links between them, but do not make the contents accessible for adding connections to parts of other models. A detail of the mechanical CAD model of the ignition switch of the car, for example, cannot be connected to its electrical signal, which is specified within a document containing the whole communication across a wire harness.

Every automotive development project leads to a large variety of vehicles: the customer determines the exact configuration of the car from a wide range of variation possibilities. This leads to the fact that almost all individual cars are built differently. The car is built after the order, but does not undergo a complete test anymore. Due to the large variability, not all of the cars that can be configured can be built for testing. Therefore, each configuration of closely connected components (which are much less than actual car configurations) has to be detected and tested before production.

Since the exact configuration is not known at design time, the organization of components/control units has to be flexible. Flexibility is introduced by using bus systems, which do not require a receiver/sender at every port and by a cooperative communication, which is not steered by a central unit, but is rather orchestrated between the control units. In some cases, smaller components are directly steered by a composite component.

The internal state of a car is defined by the current state of each component. This leads to a myriad of global states, because all combinations of component states have to be regarded. A transition cannot be defined clearly, because any of the contributing components may trigger the transition.

All components, which are connected to the bus system, communicate with each other through this channel. Standard frameworks (such as the CAN (Controller area network) bus protocol [18], [19] or the MOST (Media oriented systems transport) protocol [20] as described in [21]) and drivers for accessing the bus infrastructure are available – especially for extracting information from the communication. Additionally, frameworks for supporting the development (e.g. AUTOSAR [22]) are widely available.

Even though this infrastructure is readily available, the bus including the attached components has to be thoroughly tested to rule out unwanted effects of one component on another.

Each of the components that have to work together may come from different suppliers as many are involved in the development of automotive components. Since each supplier develops its components independently and there is no immediate need for certification, the documentation remains with the suppliers.

The application of SysML for model-based systems engineering (MBSE) has not been adopted in an automotive context and there is no model of the whole system. Systems Engineering is usually applied on a smaller level to steer the development within one department. Therefore, the following models depict the rough structure of a fictional car and its bus system. In contrast to the satellite model, in which the "on-board computer" is responsible for coordinating the whole system, the car contains many control units, which organize themselves by listening on a bus system to receive a free communication slot. Communication starts when a free time slot is detected.

Similar to the satellite model, the bus system connects all control units structurally. Modern cars contain several bus systems for specialized tasks within the vehicle. Basic functions are, for example, steered through the CAN bus [18], [19] whereas entertainment functions are handled by the MOST bus [20], [21].

Because of the orchestration of components without central control unit, all communication paths (including the paths of "virtual" software signals) as well as possible interferences with other signals have to be known to understand the communication between the components.

2. Differences between Automotive and Space Engineering in Model-based Systems Engineering

As has been outlined in the previous section, both the engineered product (including the use of its bus systems) and model-based systems engineering experience vary within the extremely different contexts in the automotive and space industry.

2.1. Stakeholder Analysis, Use Case Creation and Requirements Elicitation

In the development of spacecraft, the customer is known before the development is started. The customer issues the order. In contrast, the automotive engineer does not know the concrete customer, but a scheme developed from customer analysis and studies. Both types of customers lead to a similar stakeholder analysis, with more concrete or abstract definition of the stakeholder "customer".

The creation of use cases and the derivation of requirements from the use cases can be modeled equally in both contexts.

2.2. Structural Modeling

The model of the structure of a satellite and a car differs only by small parts. A satellite model contains one representation for each component of the satellite. This one variant of the component is used within the model of the concrete satellite. This instance model is exactly equivalent to the customer order. Figure 1 displays the structure of a satellite on an abstract level.

The structural model of a car contains – in general – many different components that could theoretically be built into a car. In contrast to the single-variant instance model (which would be a "100% model"), this is termed a "150% system model" (see

e.g. [23]). Because of the variety of components, one instance model of a car cannot contain all possible variants. Figure 3 displays the "150% system model" of a car on an abstract level, where the customer can order at most one navigation system in variant "Standard" or "Exclusive".

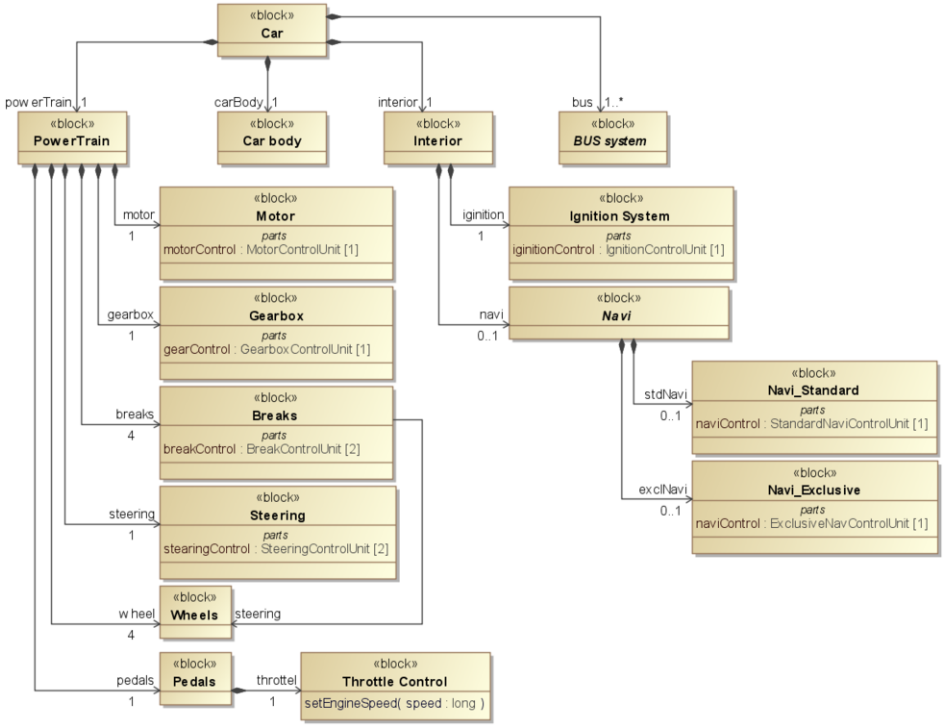

Figure 3. Structural model of a 150% car.

This type of structural model displays all possible connections between components, but does not show in detail, which alternatives can be composed. Figure 3 displays that none or one type of navigation system is used – the semantical relationship between these connections is not explained in detail (the model would allow for choosing both navigation systems in parallel). This relationship is fairly simple and can be annotated, but when more variants are introduced, the full combinatorial view of the relationships is not possible.

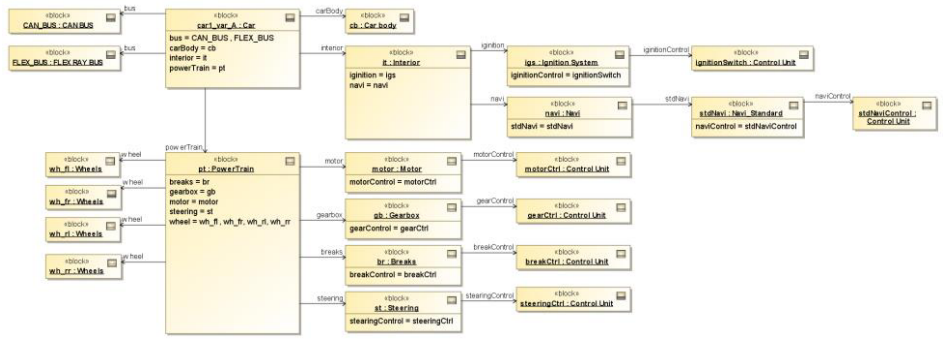

Figure 4. Model of an instance of a car with a standard navigation system.

Instance models describing one concrete car can be derived, which serve as witnesses for correctly composed choices. One of the belonging instances or "100% models" is displayed in Figure 4 – the car, where the customer chooses the standard navigation system.

2.3. Behavioral Modeling

The biggest differences in the model lie in the type and complexity of the behavioral model. Since the satellite components are centrally controlled by the master control unit "on-board computer", most communication and component activities occur sequentially (Figure 5).

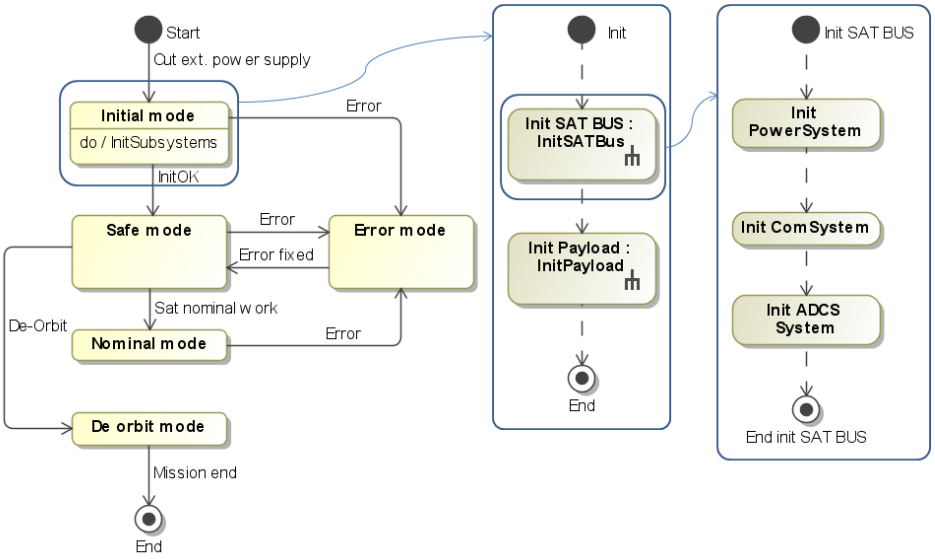

Figure 5. SysML state chart of a satellite including the sequential progress during its initialization.

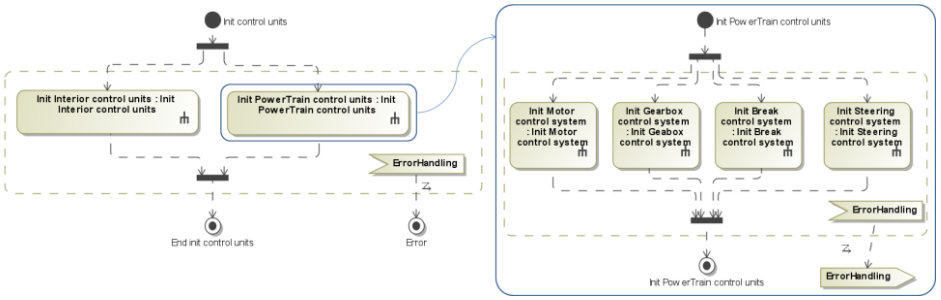

Figure 6. SysML activity diagram of parallel activities during initialization of an automobile.

The model of the system behavior discerns between models of the states, activities and sequences of collaboration. States are modeled similarly within the satellite and the car. The level at which this occurs is different – the satellite has defined global states, whereas the car requires the components to be defined first and then assigns a state machine to each of the components. The difference in activity models is the sequential description for the satellite versus a description of highly parallel and concurrent activities in the car. The same holds for models of the communication and

collaboration sequences. Even though the satellite exhibits a sequential structure at the abstract level, modeling more details leads to an increased concurrency in more concrete levels. This is contrary to the model of the automobile, which exhibits concurrency in each level, but displays more determinism in the details of the components (Figure 6).

2.4. Usage of the models

Not only the behavioral model, but also the priorities for their usage differ greatly, since the engineering of satellites is basically a one-time development and automotive engineering focuses on varying, multiple realization of a certain model.

In the context of satellite development, a common, coarse model of the system supports in the synchronization of views on an abstract level. The layout and behavior of the satellite system determines the necessary interfaces between the different components, which have to be realized and possibly defined. The definition of the interfaces and connections between the components provides a mild support for finding simple compatibility issues early on. Also, using the abstract common model in the early phases of development allows for a less costly exploration of implications posed by special requirements. Several alternatives can be modeled, communicated, discussed and evaluated without actually building flight hardware. And finally, the model documents the decisions made during the development of the satellite. These decisions can be the basis for knowledge transfer to subsequent satellite development projects.

Automotive engineering also benefits from a common model for the synchronization of involved disciplines. This synchronization is especially important for the interface definitions. As each car has to be defined in a variety of variants, one interface often has to be used by several components with similar functionality (such as the navigation systems in the previous example). The common ”150% model” helps in finding components connected to the interface, which are affected by changes (both of the structural interface and of the behavior supplying it). Since the components connected by one interface are define by the common model, it can also be used as basis for selecting groups of closely connected components for systematic testing. Additionally, all possible ”100% models” of the car can be created combinatorial from the model to be used as witnesses for extreme configurations. Finally, the model can be used to document the development for reuse purposes and to satisfy process requirements (such as imposed by [24], [25]).

3. Conclusion

Satellites and automobiles are inherently different in the main objectives that underlie their development: satellites are made to order while automobiles are constructed with variants that can be composed in a way that suits the customer.

The satellite and automotive domain are similar in some ways to construct products. Satellites are one-of-a-kind development, which requires a certain amount of manual design for each product and necessitates high reliability of all its components. Cars also require manual design of each group of similar choices in separate, descriptive instances, but do this on the basis of a catalog of different applicable variants.

Both domains can benefit from model-based systems development using a central model, which describes the development object in detail – but in different ways: the satellite developer mainly from synchronizing global views, the possible exploration of design alternatives and knowledge transfer to subsequent projects and the automobile developer from finding components affected by interface changes, validating concrete combinations of components and documentation of the development.

References

[1] J.W.S. Pringle, On the Parallel between Learning and Evolution. Behaviour. 3, 3 (Jan. 1951), 174215.

[2] A. Katzenbach, Automotive, in: J. Stjepandić et al. (eds.): *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, Springer International Publishing, Cham, 2015, pp. 607–638.

[3] R.M. Kolonay, A physics-based distributed collaborative design process for military aerospace vehicle development and technology assessment, *Int. J. Agile Systems and Management*, Vol. 7, 2014, Nos 3/4, pp 242 - 260.

[4] *International Council on Systems Engineering*, 2015. Accessed: 04.04.2015. Available: http://www.incose.org/

[5] S. Friedenthal., R. Griego, & M. Sampson, INCOSE model based systems engineering (MBSE) initiative. In: INCOSE 2007 Symposium.

[6] Gesellschaft für Systems Engineering e.V.: http://www.gfse.de/. Accessed: 2015-04-04.

[7] Model Based Systems Engineering: http://mbse.gfse.de/. Accessed: 2015-04-04.

[8] W.J. Larson and J.R. Wertz, eds. *Space Mission Analysis and Design*, 3rd edition. Microcosm, 1999.

[9] *Spacecraft bus subsystems,* 2015. Accessed: 04.04.2015. Available: http://www.lr.tudelft.nl/en/organisation/departments/space-engineering/space-systemsengineering/expertise-areas/spacecraft-engineering/design-and-analysis/configuration-design/subsystems/subsystems/.

[10] A. Peukert, *Spacecraft Architectures Using Commercial Off-The-Shelf Components.* Technische Universität München - Lehrstuhl für Raumfahrttechnik, 2008.

[11] *MOVE — Munich Orbital Verification Experiment*, 2015. Accessed: 04.04.2015. Available: http://move2space.de/

[12] *NASA Systems Engineering Handbook,* 2007. Accessed: 04.04.2015. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080008301.pdf.

[13] R.C. Beckett, Functional system maps as boundary objects in complex system development, *Int. J. Agile Systems and Management*, Vol. 8, 2015, No. 1, pp. 53–69.

[14] M. Bandecchi, B. Melton, B. Gardini, and F. Ongaro, 2000. *The ESA/ESTEC Concurrent Design Facility. Systems Engineering.*

[15] *Open Services for Lifecycle Collaboration*, 2015. Accessed: 04.04.2015. Available: http://open-services.net/.

[16] *OMG Systems Modeling Language Version 1.3,* 2012. Technical Report #formal/2012-06-01. Object Management Group.

[17] J. Stark, *Product Lifecycle Management – Volume 1: 21st Century Paradigm for Product Realisation*, 3rd ed, Springer, Cham, 2015.

[18] Road vehicles - Controller area network (CAN) - Part 1: *Data link layer and physical signalling.* Technical Report #ISO 11898-1:2003. International Standards Organization (ISO), 2013.

[19] CAN Specification Version 2.0. *Robert Bosch GmbH,* 1991.

[20] MOST Specification Rev.3.0 Errata 2. *MOST Cooperation*, 2010.

[21] W. Zimmermann and R. Schmidgall, *Bussysteme in der Fahrzeugtechnik.* Springer Fachmedien Wiesbaden, 2014.

[22] *AUTOSAR*, 2015 Accessed: 04.04.2015. Available: http://www.autosar.org/.

[23] A. Seiberts, M. Brandstätter, and K. Schreiber, *Kompositionales Variantenmanagement - Ganzheitlicher Ansatz zur Komplexitätsbeherrschung im Systems Engineering Umfeld*, Tag des Systems Engineerings, 2012.

[24] *Road vehicles - Functional safety - Part 5: Product development at the hardware level.* Technical Report #ISO 26262-5:2011. International Standards Organization (ISO), 2011.

[25] *Road vehicles - Functional safety - Part 6: Product development at the software level.* Technical Report #ISO 26262-6:2011. International Standards Organization (ISO), 2011.