Infotech@Aerospace 2012
19 - 21 June 2012, Garden Grove, California

AIAA 2012-2532

Multi-disciplinary MBSE Approach in Industrial Phases

Joachim Fuchs*
European Space Agency / ESTEC, 2200 AG, Noordwijk, The Netherlands

The European Space Agency has undertaken a number of R&D activities to advance the state of MBSE for the design and development of space systems. The main organizational context is presented, focusing on the main issues of the industrial phase of a project life-cycle. The underlying principles of data linking and interoperability between system and domain level will be described, and related emerging standards discussed. The prototype of a supporting engineering environment illustrates the potential application of the paradigm.

I. Background

The European Space Agency has a recognised track record in building, launching and operating space systems together with industrial partners. Common systems engineering practices have been standardized[1] in the frame of ECSS (European Cooperation for Space Standardization), which further helped to apply system engineering consistently across organizations, companies and countries.

One of the fundamental points of the “System Engineering” standard is the description of the System Engineering discipline, its relationship with production, operations, product assurance and management disciplines. Its internal partition defines the following functions:

- Requirement engineering, which consists of requirement analysis and validation, requirement allocation, and requirement maintenance;
- Analysis, which is performed for the purpose of resolving requirements conflicts, decomposing and allocating requirements during functional analysis, assessing system effectiveness (including analysing risk factors) and complementing testing evaluation and providing trade studies for assessing effectiveness, risk, cost and planning;
- Design and configuration which results in a physical architecture, and its complete system functional, physical and software characteristics;
- Verification, which objective is to demonstrate that the deliverables conform to the specified requirements, including qualification and acceptance;
- System engineering integration and control, which ensures the integration of the various engineering disciplines and participants throughout all the project phases.

This breakdown is rather similar to the one used by INCOSE, which defines it as focussing “on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem (Operations, Cost & Schedule, Performance, Manufacturing, Test, Training & Support, Disposal). Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation.”[3]

Although both definitions address technical as well as managerial aspects, the technical aspect and the associated engineering data is usually the focus during large parts of the project life-cycle, although it is obviously recognised that an efficient mastering of the technical process is intimately linked to the managerial and programmatic aspects. However, the main challenge of the integrative function of system engineering is the inter-disciplinary aspect between the different domains.

In the context of institutional projects (such as led by the European Space Agency) the different organisational levels and responsibilities in the engineering process need also to be properly represented. It is important to establish and respect the contractual boundaries, and to define and manage the corresponding technical interfaces. The boundaries are typically between a customer organisation (e.g. the Agency), the system engineering functions as implemented by a system engineering organisation within the supplier (industrial prime), who in turn is in charge of

* Head of System Modelling and Functional Verification Section, TEC-SWM, AIAA Senior member.

transforming the requirements of the customer into a system solution delivered by the supplier (industrial subcontractor). Relations to the supplier organisation needs to remain traceable at all times. In this process (as also reflected in the "traditional V cycle") it is clear that the early phases of the life-cycle are primarily focusing on the requirements and specification part, whereas the later phases are more concerned with the verification aspects of the system built. At all levels it is important to coordinate the necessary iteration between individual disciplines, covering analysis and design activities, and orchestrated by the system engineering function.

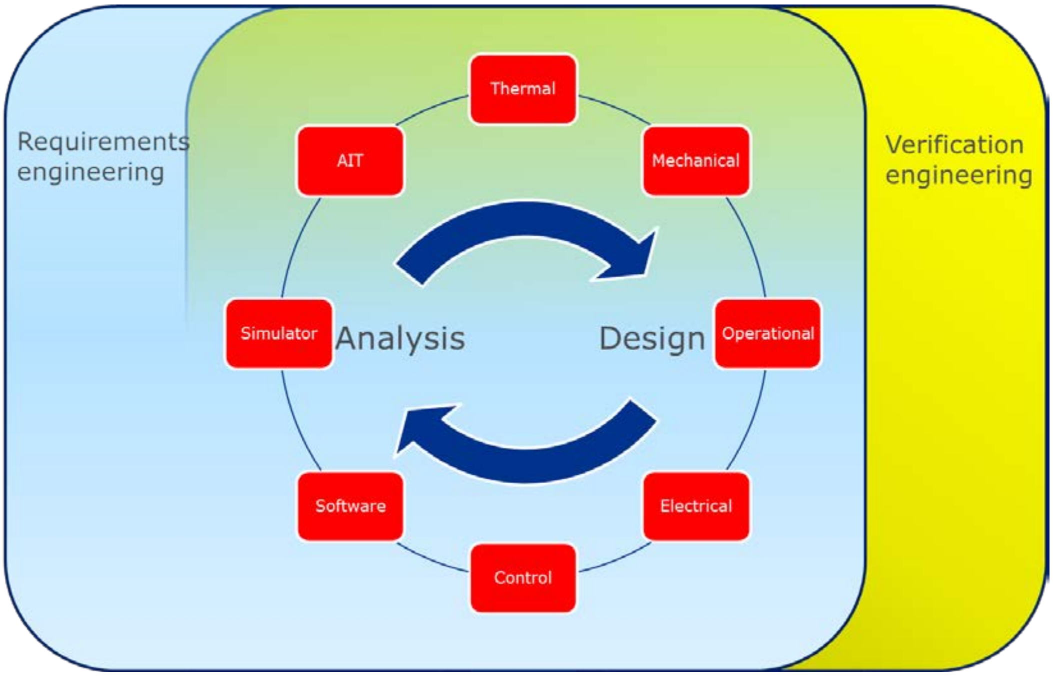

Figure 1: Multi-disciplinary iteration of SE process

Although this process is well understood, future challenges include the increased complexity of systems and systems of systems to be built. This leads to an increased burden of the corresponding data and process management, in particular in a context of decreasing resources. It is therefore mandatory to improve the efficiency and to provide support to manage this system complexity.

II. Model-Based System Engineering

In a very generic way, any representation of an engineering artifact is a model. Models have been used throughout the history of science and engineering for the most diverse objectives. They have in common that they try to represent specific aspects of a system in order to support a particular purpose. In the recent decades, computer (virtual) models have gained an increased importance as they are generally cheaper to build, replicate and exchange / share.

These models tend have two main purposes

- The appropriate *description of an element*: In this descriptive nature they are either the direct result of a design process, or in some areas the mere translation of a design expressed in other ways into a more structured form by a dedicated "modeling" activity. These models are instrumental to the communication of (engineering) information between different stakeholders. The descriptive nature is normally static.
- The *use in exploitation tools*: This usually is associated with specific analysis tools, where models (in the former sense) are used (exploited) to make complex calculations to e.g. predict performance or behavior of the element. The use of models therefore is often focusing on the dynamic aspect of an element. However, this notion can also be applied to formal checks of a model, or any transformation applied to a model which is generating another description, possibly more suitable in another domain.

Model-based approaches – i.e. the use of models in the engineering process – are already used in many disciplines today. Some well-known examples include CAD models where the models are the result of the physical design process and are used as the basis for use in specific analysis tools (such as e.g. Finite Element Analyses). In the functional domain there is the practice of formalizing the requirements and the design of functional chains, and to apply model transformation to these models to ultimately generate embedded software for the system management and operation.

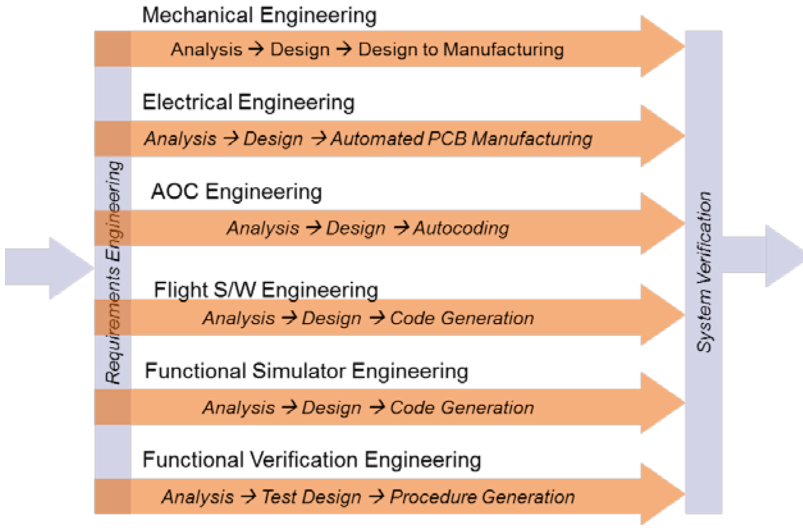

Figure 2: Model-based engineering practices

In the examples highlighted above models are used throughout the full life-cycle, supporting the different phases from requirements engineering to the verification process. Unfortunately, these lines still tend to remain within specific disciplines, so that the consistency of the domain specific models underlying these processes cannot be easily guaranteed. In frequently used cases there exist solutions to map model data from one domain to another and therefore ensure the coherence of these models at system level. However, these solutions tend to be point-to point, implying that any change in one domain will impact the mapping, and that multi-disciplinary mappings cannot be easily achieved. The challenge of model-based system engineering is the formalized exchange of models / model-data *at system level*: The Data Challenge of System Engineering.

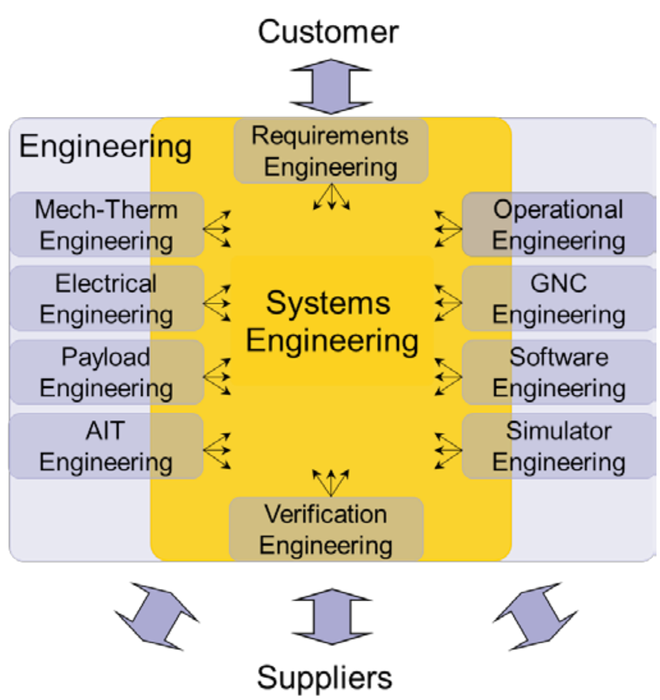

Figure 3: Required model coordination at system level for MBSE

III. A proposed solution

A. Conceptual Data Model

As outlined above, the backbone of an efficient MBSE process is a coherent and consistent model and data exchange at system level. This can be achieved through an integrated “system level model”. One of the major challenges for this problem is the fact that internal data structures (and the associated data model) of different engineering tools are incompatible with each other. These models are usually based on long heritage of domain specific engineering, and they represent the formalized knowledge of that domain. These Domain Specific Languages often lead not only to a syntactic, but also a semantic incompatibility with other domains.

To combine the different domains into a compatible “system description” proves very difficult. When analyzing the different models, it is important to ensure the interoperability of the data. This requires to find a description of a “system level model” at a conceptual level. Such a conceptualization can then be used to map to different logical and physical models for particular implementations. This implies therefore that it is necessary to abstract existing data descriptions and models (which are usually at physical or at best at logical level) up to the conceptual level, to ensure the interoperability.

No globally accepted standards have yet been agreed for the description of conceptual data models. It is therefore still difficult to formalize such a system model in a globally accepted manner. Within the ECSS a first attempt has been undertaken to describe such a model to represent system engineering data[2]. This first conceptual data model for space SE specifies the data structures required for the integrated “system level model” and have provided the basis for a semantic integration platform.

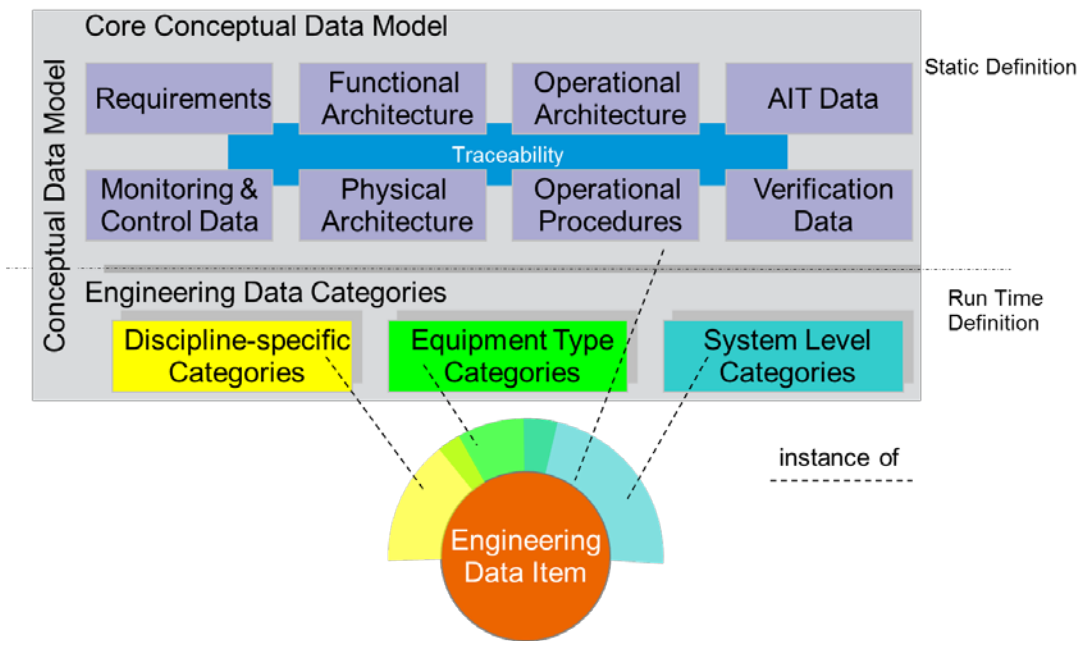

Figure 4: Conceptual Data Model for SE, as implemented in a prototype

The static definitions of the concepts for an Engineering Data Item (or System Element) are extended by Engineering Data Categories, which represent groups of parameters specific for disciplines or equipment types. These categories can be extended more easily if required, without having to change the core data model underlying the system description.

B. Model management

The next problem to be faced by MBSE is the management of different models, at system level as well as the domain specific models. Configuration control of the models itself is probably the smaller problem posed in the system context. The traceability of engineering data from one model to the next including the semantic mapping and ownership management for data items used in models is required to achieve the ultimate goal of model and data coherence at system level. Model management needs to be broken down to data element level.

A further difficulty is the need to allow different stakeholders to work with "partial" models, and to work in parallel (concurrent engineering). For that purpose it is necessary to be able to branch model versions, to be able to split parts off with well-defined interfaces, to compare and eventually to merge those parallel and partial models back into a complete and consistent system model. This merging process needs to be supported by adequate checks, so that inconsistencies can be highlighted, and corrective actions at system level be initiated.

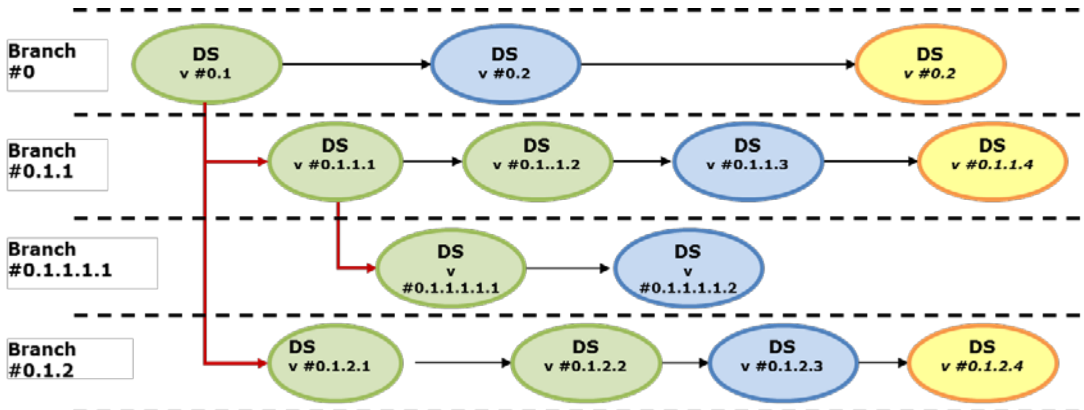

Figure 5: Branching of datasets, as implemented in prototype (datasets can be merged again at the end)

C. Integration of COTS tools

The underlying principle in the proposed approach is the respect of the domain specific languages and data models. As explained above, specific representations are traditionally linked to specialized design and analysis tools. It is important to maintain the expertise captured in these representations and tools, and not to impose an overall "system" language and representation which might only be suboptimal to capture the relevant concepts. The approach is therefore a hybrid one, analyzing which data needs to be exchanged between different domains, and then providing a mapping towards a common space model able to capture the required concepts.

This philosophy allows on the one hand to maintain the domain specific models and processes, whereas it allows a neutral exchange of relevant data between different disciplines. A transformation towards this neutral system level model representation allows also to maintain model transformations / mappings independent of each other. The advantage of this solution over individual point-to-point solutions is the fact that an independent evolution of specialist models is only requiring the adaptation of one interface at a time. It also enables to exploit system data at system level directly (e.g. maintenance of budgets at system level).

The cost of this approach is the fact that it is necessary to keep the domain models as a whole model under configuration control, in order to maintain the proper traceability and consistency between engineering information and related design and analysis information. It is obviously also necessary to maintain the system level data model in case new domains are included. As outlined above, this requires also an analysis of the respective concepts to ensure that they can be represented (and exchanged) correctly.

The demonstrator has implemented corresponding adaptors for DOORS (requirement models) and CATIA (CAD models). Further links are considered, including the mapping to other modeling tools.

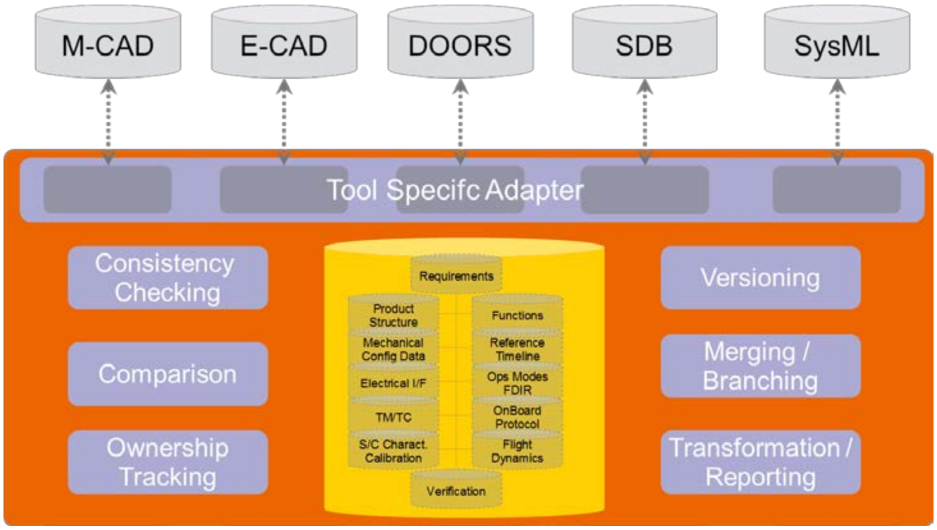

Figure 6: Principle of adaptors for domain specific models and data

The overall implementation of the prototype to support the presented principles turned out to be a challenge for the limited resources available. Due to the fact that the system data model might evolve (extend) in the future, it was necessary to choose an efficient / pragmatic implementation technology. The Eclipse framework with its associated plug-ins has been found to be a practical solution to the rapid prototyping of the Virtual Spacecraft Engineering Environment. The conceptual datamodel was formalized in an Ecore implementation and formed the common core for the development of associated modeling tools, data editors and underlying database infrastructure in a model-driven approach. In that way it was possible to quickly iterate the tools, even after a change / adaptation in the datamodel.

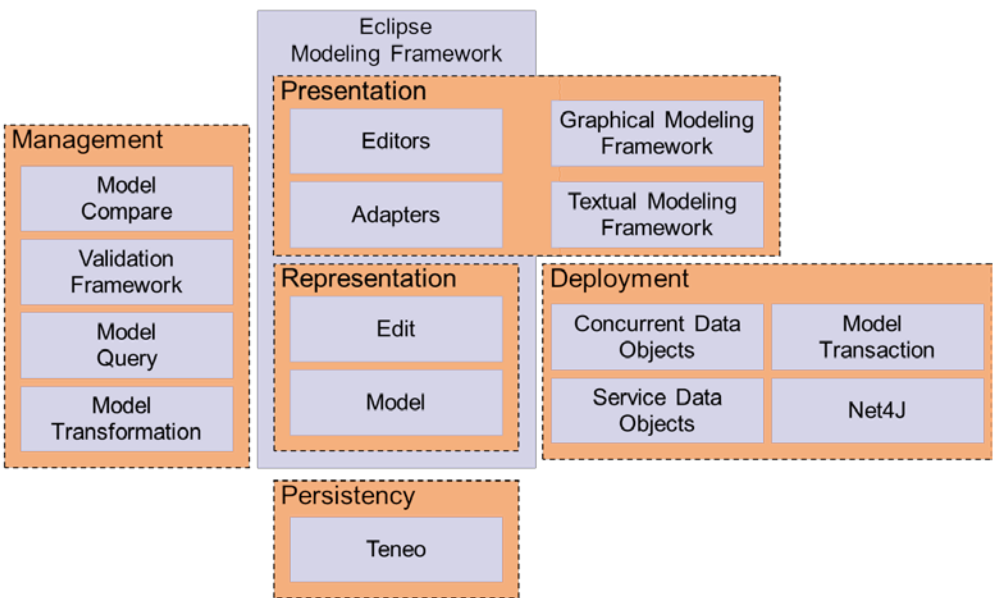

Figure 7: Implementation technology for the prototype

IV. Main achievements

The main result of the prototype and demonstrator development in the frame of the R&D activity " Virtual Spacecraft Design" is the availability of an open and modular framework as a backbone for efficient and effective multi-disciplinary data sharing (Virtual Spacecraft Engineering Environment [VSEE]). Although it has been driven by concepts of space projects, the principles are generic, and it should be possible to apply them in different application areas. By applying the underlying principle it is possible to achieve increased consistency and traceability by means of commonly shared data, nevertheless respecting domain specific notations. It needs to be mentioned that the notation used in the VSEE is in some areas highly inspired by UML and SysML, but has been extended to include more representations such as tables, trees, and dynamic model element trace diagrams. Domain specific notations have been introduced where possible. The prototype provides also the possibility of visualization and navigation through all elements by means of the physical (3D) model. The underlying data exchange mechanisms (split, branch and merge) support a concurrent engineering process adequately.

The environment allows the proper management of the system engineering process. It enables the linking of scattered repositories, and the mechanisms for consistency checks for all relevant data. With that it is possible to detect “broken ends” and ease an impact analysis in the case of distinct iterations of parts of the model. Review of the model is possible through the concept of “annotations”, which allow to attach comments, discrepancy reports, problems, actions… to individual engineering items.

Tracing from requirements to verification is enabled through the availability of all required traceability links defined in the conceptual data model: Between different requirements levels, from requirements to design artifacts and finally to test and verification activities (incl. analyses). These traces play an important role in ensuring consistency of the overall system engineering information and an easier verification process at the end of the project.

System level design forms essential set of data to be shared between disciplines for analysis and therefore allows early V&V activities. AIT procedures can be defined and evaluated at an early stage, and functional system simulation can be included as a system-level analysis tool. Budget analysis can be achieved through an automated tie-in into Excel for the corresponding system parameters.

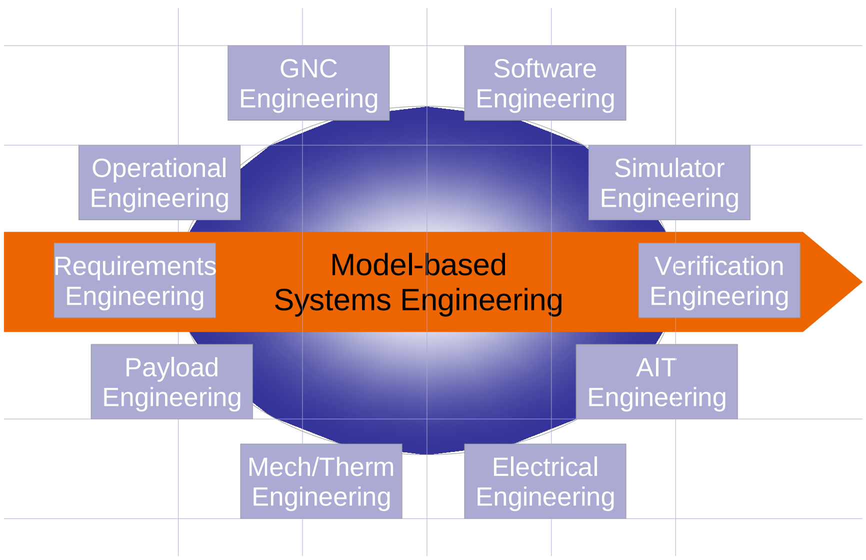

Figure 8: Overall MBSE process

V. Outlook

The principle of a “Virtual Spacecraft Design” has been applied and its applicability and possible tool support demonstrated by (re-)modeling a real mission. However, major challenges remain to be addressed:

- The link between the semantic data integration and the configuration control capabilities of current PLM systems needs to be formalized. The complementarity of these systems needs to be further explored and improved.
- It is necessary to increase the number of number of links to disciplines. Automatic generation / instantiation / parametrization of domain specific models (e.g. functional simulation models, Modelica models…).
- The formal management of status and progress of verification activities will improve the usefulness in the overall V&V activities.
- It is necessary to achieve a general acceptance of the conceptual data model underlying the used tools in order to increase the possibility to share (part of) models and consistency between contractual partners.

It is envisaged to support a real project with the developed prototype tools as a pilot analysis in order to consolidate the tool requirements and the underlying conceptual data model.

Acknowledgments

The work has been mainly funded through the Technology Research Programme (TRP) of the European Space Agency. Special thanks to Harald Eisenmann from Astrium, industrial project manager of the VSEE development and to my colleague Serge Valera, trying to educate me in the art of conceptual data modeling.

References

[1] "System Engineering" ECSS standard, E-ST-10C, accessible through www.ecss.nl

[2] "System Engineering Data Repository", ECSS Technical Memorandum E-TM-10-23, accessible through www.ecss.nl

[3] "What is System Engineering?", INCOSE definition, URL: http://www.incose.org/practice/whatissystemseng.aspx [cited 27/5/2012]