An Investigation of Functionalities of Future Tool-chain for Aerospace Industry

Jinzhi Lu, De-Jiu Chen, Didem Gürdür, Martin Törngren
School of Industrial Engineering and Management, KTH Royal Institute of Technology, Stockholm 10044
E-mail: jinzhi@kth.se

Abstract. Model-based systems engineering (MBSE) is advocated as one important means to deal with increasing product complexity. In this paper, we investigate the increasing interest and use of MBSE in the aerospace domain. The growth of MBSE tools causes new challenges in the corresponding tool-chain development including data and information management, tool’s applicability identification, tool-integration, etc. In this paper, a literature survey into trends and usages of large-scale MBSE tool-chains is presented. It is found that the key functionalities include functional roles, elements, abstraction levels and tool integration techniques. Additionally, we propose a methodology to provide guidance in optimizing future MBSE tool-chain functionalities for the aero-industry. In closing, we present a case study in order to illustrate the decision making process required to develop expected tool-chains by the proposed methodology.

Introduction

Various factors contribute to the increasing complexity of the high-technology products which make up today’s aero-systems. The first factor is the trend of an increasing number of functionalities supported by software, hardware, control and other domain specific techniques. This leads to challenges in engaging a much broader set of developers during the development of such functionalities across multiple domains (Bayer, 2011). The second factor is that new functionalities are interactive and multi-layered requiring additional stakeholders, actors, and system interests, and therefore posing challenges for traditional organizational structures to satisfy the demand of future aero-system development (Young & Fellow, 2015). The third factor concerns the growing numbers of views and system concerns that require new design tools and methodologies in order to implement more complicated design tasks. Finally, the increasing product complexity has resulted in development processes, which are no longer flexible, efficient and agile (Verma & Fabrycky, 1997).

All these challenges motivate a more structured and formalized approach than pure document based systems engineering. Industries are therefore investigating the use of Model-Based Systems Engineering (MBSE) approaches where formalized system descriptions (i.e. computer manipulatable) are used to analyse different aspects of a complex system (Ramin, 2015). Each model and formalism may be used at different abstraction levels for the given purposes, captured by a specified language, which is defined by its meta-model. MBSE results in a renovation of design activities and workflows and additional needs for new tools and methodologies.

However, such new tools and methodologies supporting multi-domain, inter-disciplinary and increasingly complex design tasks are challenging to adopt and sometimes even ignored by current stakeholders. This is partly due to an incomplete understanding and a lack of a system perspective about these new techniques, and partly due to product-centric and design task-first approaches (Selic, 2012). On the other hand, weak communication between tool users and tool developers, poor
understanding of tool functionalities and insufficient motivations also hinder the adoption of new technical practices. This combination of problems results in a barrier to the application of new tools or methodologies for aero-systems.

In this paper, our contribution is threefold. Firstly, we investigate the trends and usage of tool-chain evolution in the aerospace industry. From literature reviews in the aerospace domain, covering contributions from industry and academia, we analyse current functionalities and application domains to assess MBSE tool-chain stakeholders and viewpoints. Secondly, we propose a methodology supported by a systems engineering approach to define the process of setting up an MBSE tool-chain. Relying on the definitions put forward in IEEE Standard 1471-2000, we construct an architecture description which clarifies the relationships between stakeholders, views and viewpoints for development of MBSE tool-chains. Thirdly, in our methodology, an initial conceptual model-based approach is proposed to represent the expected MBSE tool-chain architecture with the purpose of understanding the tool-chain’s complexity and interoperability. The intention with this model-based approach is to improve communication between different stakeholders and reinforce the developers’ understanding of MBSE tool-chain functionality. Finally, the conceptual model-based approach is introduced to represent the process of setting up an MBSE tool-chain in a case study.

The method used in this paper follows steps: (1) literature surveys and interviews with stakeholders are launched to define mission of MBSE tool-chain and select stakeholders and viewpoints as the input of our methodology. (2) based on our proposed methodology, a model-based tool-chain representation is generated. (3) a case study is used to illustrate the methodology to design a tool-chain.

The remainder of this paper is structured as follows: Definitions of important keywords used throughout the paper are introduced first. Then the conducted literature review is explained. Based on the literature review findings and IEEE Standard 1471-2000, stakeholders are selected for collecting viewpoints. Then a model-based representation of the expected MBSE tool-chain architecture is constructed by our proposed SPIT framework (Jinzi 2016). In the case study, a propulsion system is demonstrated in order to validate this approach. And finally, our conclusions and suggestions for future work are presented for our plan to implement this methodology to develop an MBSE tool-chain.

**Definitions**

This section defines important terms that we use throughout the paper. The dependability terms are inspired by (Algirdas Avižienis 2004) and system architecture terms from IEEE Standard 1471-2000 (Anon 2011). We slightly adapted some terms to fit an MBSE context.

**MBSE tool-chain:** more than one modeling tool that, when combined, can support and construct a system engineering workflow, which has the following features:

- The workflow can support system requirements, design, analysis, verification and validation activities through the whole life cycle of product development.
- The workflow supports the stakeholders’ view of system functionalities and other extra-functional aspects, like project management, dependability and so on.
- The workflow includes data, knowledge, and information exchange and model transformation.
- The ontology of this workflow can support system engineering.
- The workflow could integrate existing engineering tools and system development platform.
- The workflow could support heterogeneous model integration.

**Dependability:** The ability to deliver a system function or software service that can justifiably be trusted.
**Availability:** readiness for correct system function or software service.

**Reliability:** continuity of correct system function or software service.

**Safety:** absence of catastrophic consequences on the user(s) and the environment.

**Integrity:** absence of improper system function or software service alterations.

**Maintainability:** the ability to undergo modifications and repairs.

**Security:** absence of unauthorized system function or software service alterations.

**Stakeholders:** An individual, team, or organization (or classes thereof) with interests in, or concerns related to, a system.

**View:** A representation of a whole system from the perspective of a related set of concerns.

**Viewpoint:** A specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

**Social factor:** The surrounding environments are major influences on an individual's behaviour, thought processes and culture.

**Aero systems:** The equipment and cyber physical system (CPS) used in aerospace industries or aerospace products.

**Cyber-physical system (CPS):** We used the definition provided by Edward A. Lee (Lee 2010), “CPS are integrations of computation with physical processes”. In CPS, cyber system includes embedded computers and network systems. Physical system includes all the physical actors in the real world.

### Tool-chain Literature Review

During the literature review, we have surveyed papers from 28 journals and 26 conference of AIAA, from aerospace and aircraft domains. Google scholar digital database has been used for searching the keyword “tool-chain” and “MBSE” in the scope of the journal and conferences. Figure 1 illustrates the frequency of occurrence of these two terms in response to the searches. It is found that the research papers about tool-chain and MBSE have rapidly increased since 2010. Moreover, as shown in Figure 2, most of the publications concern tool-chains for developing integrated systems. Most developers expect to integrate different viewpoints to capture system performance during the aero system development by a tool-chain. Therefore most of such tool-chains should be able to support design activities for multiple purposes in the fields of system design, subsystem design, hardware design, and software design.
We summarize the functional roles, elements, and abstraction levels of application and tool integration techniques from the partial literature review shown in Table 1. These data could help us to define the system boundary of the expected MBSE tool-chain in the next section. Functionalities are investigated to capture the purposes of tool-chain. The elements are models used in such tool-chains. Abstraction levels represent views we need to consider while developing tool-chains. Tool integration techniques are considered as important system concerns during MBSE tool-chain development.

The functional roles of the tools in an MBSE tool-chain are system modelling, verification and validation, simulation, geographic modelling, HIL simulation, data modelling and management, ontology, knowledge management and information management. Furthermore, the functional roles of the tool-chain are integration of two or more items from each tool’s functionalities. The elements (models) used for each tool-chain include data model, CAD model, FEM model, dynamic simulation model, requirement model, system model (software and hardware), information model and process model. The abstraction levels are process level, requirement level, function level, behaviour level, implementation level and verification& validation level. The tool integration techniques are linking data, model-driven and code generation, standard based model and data exchange (for example using FMI (Anon 2010) and STEP (Eckert et al. 1990)), ‘meta-model’ based transformation and ‘ontology’ based transformation.

Based on the literature reviews, we identified MBSE tool-chains being used in a wide range of settings and from a variety of disciplinary perspectives. The views of MBSE tool-chain development are in multi-abstraction level and inter-disciplinary fields. At the boundary of the tool-chain, it overlaps with more disciplines, such as system design, software development, simulation, data management, integrating with PLM and so on. Thus it seems that all these disciplines need to be considered when constructing an MBSE tool-chain.

Currently, an MBSE tool-chain is developed by a problem solving strategy. Stakeholders from different domains often develop the tool-chain for their own purposes. The target product supported by the tool-chain often decides its functionalities. This results in a large number of specific tool-chains and design platforms in organizations which creates challenges for tool integration and data integration.

The literature review shows that a tool-chain’s functionality for specific integration is a key concept driving tool-chain development. Researchers develop standards and interfaces to support tool integration. However, it takes time and cost to develop such techniques and a considerable amount of repetitive tasks to realize this in practice.
Developers expect to develop tool-chains in a trend integrating viewpoints across the whole abstraction level and whole life cycle. Design activities are expected to be implemented in a unified platform integrated with other design tools and development platforms. This needs a systematic way to design the tool integration strategy and develop the tool-chain which supports all domains and needs of the stakeholders.

In order to develop an MBSE tool-chain for a large scale system in a systematic way and decrease the R&D cost for tool integration and data integration, a new systems engineering approach to investigate functionalities of future MBSE tool-chain for aero-systems is proposed in the next sections.

**A system thinking way to develop MBSE tool-chain**

The methodology we propose rests on the following pillars:

- Application of systems thinking and a systems engineering approach to design tool-chain architecture (Anon 2011)(Cloutier et al. 2015)
- Use of a SPIT based framework for structuring viewpoints(Jinzhi 2016) (Sillitto 2010)
- A model-based approach to support tool-chain development

The systems engineering approach is used to design the target MBSE tool-chain architecture. We focus in particular on systems engineering practice applied to MBSE tool-chain development. This represents a more narrow scope compared with previous researcher, such as Hallqvist (Hallqvist & Larsson 2016) using a system engineering principle to introduce MBSE in Saab or Bayer (Bayer T J, Cooney L A 2010) adopting IEEE 1471 to capture relationships between views, viewpoints and architecture at the Jet Propulsion Laboratory (JPL).
### Table 1: Investigation of tool-chain functionalities, abstraction levels and tool integration techniques

| References | Functional Role | Element | Abstraction Level | Tool integration Technique |
|------------|-----------------|---------|------------------|---------------------------|
| (Graignic et al. 2013) | Simulation management | Data model, CAD/CAE model | Function/behaviour/structure/CAD&CAE | Data linking |
| (Brusa et al. 2015) | Integration of Heterogeneous simulation and MBSE | FEM model, dynamic simulation model, TBD model, requirement model | Requirement/Function/physical models | Functional Mock-up interface (FMI) |
| (Votiveeva et al. 2012) (Scheeren & Pereira 2014) | System modelling and simulation verification | System model, Simulation model | Requirement/Function/System Structure/Verification (simulation) | Model-driven (Code generation) |
| (Schumann et al. 2011) | Data exchange between generic tools and simulation | Product model, Geometry model, multi-body dynamic model | System requirements/CATIA model/Refine Geometry/Multi-body dynamic model/Simulation | STEP |
| (Hein et al. 2010) | Integration of views of requirement, design, implementation and verification | System model, software model and hardware models | Requirement views/Design views/Implementation View/Verification View | Common modeling theory based views integration |
| (Kalawsky et al. 2013) | MBSE tool-chain support HIL simulation | System model, XML, HIL simulation model and c code | Requirement/Platform independent model/Platform specific models | XML-processor |
| (Shani U 2013), (Broodney et al. 2013) | Ontology mediation support linking data | Ontology, MBSE tools and models | Architecture/Preliminary design/subsystems design/integration | Ontology-based data fusion and model transformation, OSLC |
| (Johnson et al. 1999) | Data model in MBSE | Standards-based Data model, tools | System architecture/Requirements/functional design/breakdown design/configuration management/physical design&Arch/Properties/Graphics/Data Type | STEP AP233 |
| (Young & Fellow 2015) | Web deployment support MBSE | Web deployment, MBSE data, tool and model | Social level/Technical level | ----- |
| (Pavalkis 2016) | Integration of MBSE tool-chain and PLM | MBSE tool, model, and PLM | Requirement Arch/Functional Arch/Logical Arch/Physical Arch/Testing Arch | Meta-model called RFLP Model |
| (Do et al. 2012) | Knowledge sharing between project stakeholders | Shared model environment, project environment, and system model | Project process level | ----- |
| (Do et al. 2014) | Knowledge sharing across the contractual boundary | WSAF meta model, script, project process | Requirement analysis/functional analysis/synthesis | ----- |
| (Paredis et al. 2013) | Subsystem models can be integrated into a system model before proceeding to subsystem build and integration. | System model, behavior model | System models of subsystems, integrated model of subsystems/Subsystem integration and Qualification(I&Q)/System I&Q/System acceptance | ----- |
| (Petnga & Austin 2013) | Ontology for information model of MBSE | Information model, Ontology | Information level/ontology | Ontology |

IEEE-Std-1471-2000 is described as “IEEE Recommended Practice for Architectural Description for Software-Intensive Systems”. The MBSE tool-chain is a system consisting of software tools, physical systems and human activities which satisfy the definition of a software-intensive system. Therefore we make use of IEEE 1471 in order to develop the descriptions of the MBSE tool-chain architecture and analyse the architecture description.

The target of this approach is twofold. Firstly, this approach addresses the functionalities of MBSE tool-chains for future aero-systems from a systems engineering perspective. Secondly, based on IEEE 1471, an initial conception of domain specific models is proposed which could replace documents capturing requirements for MBSE tool-chain development. With this domain specific model, a more user-friendly and powerful approach with multiple analysis functions would be provided in order to evaluate the functionalities of MBSE tool-chains in the future (Biehl 2013).
Using the architectural description of IEEE 1471, we define the mission to be that of developing an MBSE tool-chain for a particular aero-system. The system is a target specific MBSE tool-chain, and the organization developing this MBSE may have certain policies (environmental factors) which influence the development process, for example, restricting the type of data base. This MBSE tool-chain has one architecture and multiple stakeholders. Different stakeholders have their own system concerns, for example, mechanical engineers have system concerns about assembling features of their product. The system architecture is described by architectural descriptions identified by stakeholders and their system concerns. Stakeholders provide viewpoints which cover system concerns. Often, viewpoints could derive from a viewpoint library. In the architectural description, viewpoints are confirmed by the views organizing architectural description. Viewpoints establish techniques of an abstract method for modelling, data interaction and other design activities. Different technologies which meet the requirements as posed by the viewpoints could confirm tools or tool interactions used in the MBSE tool-chain. Different tools and tool interactions make up a tool-chain. The analysis process and reasons adapting this new MBSE tool-chain are represented and shared as a rationale.

**Stakeholder survey**

The stakeholders involved about MBSE tool-chain development and usage are selected first. Following the IEEE 1471, the stakeholders are classified into tool-chain users, acquirers, developers and maintainers. As presented in Table 2, stakeholders are selected from different domains based on literature reviews. Stakeholders who have interests in the tool-chain or tools are tool-chain developers. Those with interests in the IT system are system maintainers. Here, we assume that stakeholders of the IT system have the responsibilities of tool-chain, cloud, database and operation system maintenance. Stakeholders who have an interest in the aero-system are tool-chain users and acquirers.

System interests are further classified into functional and extra-functional aspects. For the functional aspect, the system architecture represents an architecture of the whole aero-system. Domain engineering is a system interest focused on the subsystems of the aero-system. In this paper, system interests of domain engineering are classified by the definitions of CPS. For the extra-functional aspects, project management, risk management and dependability are taken into account (Hein et al. 2010). For dependability aspects, definitions in the Glossary are used to classify the system interests. From the list of system interests, stakeholders shown in Table 2 are considered to summarize the viewpoints used to capture the functionalities of MBSE tool-chains.

**SPIT Framework**

The viewpoints come from interviews with stakeholders, literature studies and industrial standards. However, since the collected viewpoints are unstructured and wide in scope, it is difficult to organize and analyze them in a systematic way. Therefore as illustrated in Table 3, we make use of a slightly revised SPIT framework oriented from (Sillitto 2010) to arrange and classify these viewpoints into four views: Social, Process, Information and Technique (Jinzi 2016). From the business case for systems engineering study (Elm & Goldenson 2012), we found that a project plan and project monitoring and control are important factors influencing project performance when the research team adapts a system engineering approach. We therefore take Process as an important aspect in our framework in order to satisfy the demand of the MBSE tool-chain.
### Table 2: Stakeholders from system interests

| System Interest | Stakeholders |
|-----------------|--------------|
| Design Tool     | System Engineer |
| Cyber system    | Software Engineer |
| Software (Embedded System) | Hardware engineer |
| Hardware (Embedded System) | Hardware engineer |
| Control         | Control engineer |
| Mechanical system | Mechanical engineer |
| Hydraulic system | Hydraulic engineer |
| Electronic system | Electronic engineer |
| Power system    | Power System engineer |
| ...             | ...           |

#### Stakeholders of MBSE Tool-chain Development for Aero systems

| Stakeholders | System Interest |
|--------------|-----------------|
| Design Tool  | System Engineer |
| Cyber system | Software Engineer |
| Software (Embedded System) | Hardware engineer |
| Hardware (Embedded System) | Hardware engineer |
| Control       | Control engineer |
| Mechanical system | Mechanical engineer |
| Hydraulic system | Hydraulic engineer |
| Electronic system | Electronic engineer |
| Power system  | Power System engineer |
| ...           | ...             |

### Table 3: Viewpoints of each layer

| Layer Name    | Viewpoints of each layer |
|---------------|--------------------------|
| Social Layer  | Functionality about MBSE toolchain’s human machine interface and the network of its users: Policies, Standards & Specification, Culture, human machine interface... |
| Process Layer | Functionality about target product development process: Design Process Management and Control... |
| Information Layer | Functionality about target product systems and its IT system requirement: Requirement, architecture, verification and validation... |
| Technical Layer | Functionality about MBSE tool-chain Techniques: Simulation, configuration... |

The SPIT framework is a system architecture description template organized by these four layers which are treated as specific views. The Social Layer relates to social factors of the organization for which the MBSE tool-chain is implemented. The Process Layer regards design process management and control including agreement process, enterprise process, project process, technical process and special process (Estefan 2008). The Information Layer is a collection of viewpoints covering aero-system concerns and IT system managers’ concerns. The Technical Layer addresses the tool-chain and tool developer’s concerns and is constructed by viewpoints of technical aspects.

Figure 3 represents the workflow analysing the MBSE tool-chain development. First, the system stakeholders are selected (developers, users, maintainers and acquirers) and investigated to provide viewpoints to develop the MBSE tool-chain. Then, these viewpoints are classified into the four layers of SPIT. Viewpoints in the social, process and information layers are requirements and constraints which need and limit viewpoints in the technologies level. The third step is based on constraints and technical viewpoints to involve selection of techniques. In the next step, considering environmental policy constraints, tools and tool interactions are selected. For example, in company A, there is a policy that only MySQL is the only database. Therefore, some softwares developed based on another database cannot be selected. Finally, a proposed tool-chain prototype is provided.
Model-based approach for MBSE tool-chain analysis

In our methodology, we also propose using a model-based approach to support the decision making flow from the stakeholders through to the MBSE tool-chains. The meta model is designed based on the system architecture description in IEEE 1471. The nodes represent (Social, Process, Information, Technical) viewpoints, techniques (Tech), environmental constraints (Envi), tools (Tool), and interactions of tools (Tool_Int) as shown in Figure 4. Edges represent the conditional dependencies between these nodes. The input for creating such a model approach is the information collected from the stakeholder survey and the literature reviews. The output is the expected MBSE tool-chain after instantiation from Tool and Tool_Int.

![Figure 3. Workflow of analyzing tool-chain analysis](image)

Figure 3. Workflow of analyzing tool-chain analysis

![Figure 4. Towards a model-based approach to support tool-chain development](image)

Figure 4. Towards a model-based approach to support tool-chain development

Figure 4 represents one view as part of the development of a tool-chain. An example case is provided to demonstrate this model-based approach. The stakeholders who develop this MBSE tool-chain have a system concern in that they need a method to deal with system integrated simulation. Currently, Modelica (Association 2005), co-simulation (Gomes et al. 2017) and specific tool-based integrated simulation are candidate techniques to deal with this. In order to compare and make a choice between the alternative candidates, we use Modelica first as an example.

The viewpoint for technical layer, T, is provided: Use a Modelica technique to deal with this problem. There are three constraints. The first factor is S is a viewpoint for which model-based analysis is a requirement from the social level. The second factor is P is a viewpoint for which this technique could be used in the system integrated phase. The third factor I is a viewpoint from information layer: if this
technique could verify system performance. If \( T \) could satisfy the demand of \( S, P \) and \( I \), the technique, Tech would be selected. If not, another technique would be used to repeat the analysis again. Environmental rules, Envi, are the policies used in the organization. Tech and Envi are two factors to select a tool, Tool constructing the MBSE tool-chain. In this case, if there is a tool that would support Modelica and it could satisfy the demand of the policy of security, this tool would be selected. If not, alternative candidate tools would be analysed. Tool interactions, Tool_Int are selected in the same way as tools. After Tool and Tool_Int selection, Tool and Tool_Int are instantiated as tools and tool interactions in the tool-chain’s architecture represented by a model-based approach.

Case Study

Using the methodology described in the previous section, a case study of the propulsion system for a liquid fuel rocket, shown in Figure 5, is used to illustrate the corresponding process for its tool-chain development. The propulsion system consists of PFS (Propellant Feed System), conduit head and engine. The PFS enables the fuel and oxidizer to be pushed from the conduit head to the engine in a specified ratio and the engine produces thrust. The propulsion system is the head of the launch vehicle whose carrying capacity is decided by the propulsion’s specific impulse and thrust.

In order to develop the MBSE tool-chain for this system, we take viewpoints, as examples, from partial stakeholders in Table 2, analyse this tool-chain by our proposed approach and provide a conceptual model-based representation of this tool-chain. Table 4 summarizes the ten stakeholders, four design processes, one social factor and several information viewpoints. In this case, the organization wants to use a model-based approach for propulsion system design, therefore “model-based” approach is the exclusive social factor. Technical viewpoints could be proposed as input to specify each technique, e.g. Modelica could deal with multi domain simulation. In each cell of Table 4, a technique is needed to satisfy the corresponding viewpoints. We have a hypothesis that partial techniques could be integrated into one tool. For example, a system modelling tool could support the function of system modelling and static modelling and calculation. Some tools could support multiple function required by stakeholders. Therefore, in the end, eight tools are needed in this case study as shown in Figure 6.

![Figure 5. System Architecture of Propulsion System](source-url)
Table 4: Viewpoint examples in each phase from each stakeholder

| Stakeholders | Viewpoint example in each phase |
|---------------|---------------------------------|
| **Process**   | **Requirements Definition and Analysis Process** | **Architectural Design** | **Implementation Process and Integration Process** | **Integration and Verification Process** |
| **Tool or tool-chain developers** | Lightweight software, fast calculation, reliability. **Techniques:** Readiness level for software. Concurrent and collaborative design of system architecture, traceability, database and relative technical viewpoints. **Techniques:** Data and model sharing and management, integrated simulation, multi-domain and multi physics simulation. Interface design between system model and mechanical model. **Techniques:** General Interface standards. |
| **System Engineer (propulsion)** | Thrust and specific impulse. **Techniques:** static modelling and calculation, requirement modelling. Interface and contract of propulsion and POGO analysis. **Techniques:** static modelling and calculation, system modelling, and multi physics simulation. Function and performance of propulsion. **Techniques:** static modelling and calculation, system modelling, multi-domain simulation. Matching degree of engine and PFS. **Techniques:** Integrated simulation, system modelling. |
| **Software engineer** | Not exist. Sequential control algorithm analysis. **Techniques:** Embedded system modeling and analysis. Function design of sequential control algorithm. **Techniques:** State Machine modeling and code generation. Verification and validation of sequence control software. **Techniques:** HIL simulation. |
| **Hardware engineer** | Not exist. Controller hardware interface design. **Techniques:** Embedded system modeling and analysis. Design of controller hardware. **Techniques:** Embedded system modelling and analysis and static modelling and calculation. Verification and validation of sequence control hardware. **Techniques:** HIL simulation. |
| **Control engineer** | Control interval of engine operation. **Techniques:** Requirement modelling. Sequence of propulsion system and controller interface. **Techniques:** State Machine modelling and system modelling for control system. Integration of software and hardware of sequential controller. **Techniques:** State Machine modelling and system modelling for control system and code generation. Integrated verification and validation of sequence control system. **Techniques:** HIL simulation. |
| **Mechanical engineer** | Rough estimation of mass (Fuel, engine, conduit head). **Techniques:** static modelling and calculation and Requirement modelling. Load calculation and feature of mass calculation. **Techniques:** static modelling and calculation. Strength calculation, mass feature calculation and digital assembling verification. **Techniques:** static modelling and calculation and 3D assembling. Verification and validation of mass feature. **Techniques:** Requirement modelling, system modelling and static modelling and calculation. |
| **Hydraulic engineer** | No exist. **Techniques:** Requirement modelling. Parameter interval of engine with hydraulic system. **Techniques:** multi-domain modeling. Function of PFS. **Techniques:** multi-domain modeling. Verification and validation of PFS. **Techniques:** system modelling and multi-domain modelling. |
| **Project manager** | R & D cycle and cost estimation. **Techniques:** Business model and static modelling and calculation. Integrity of each subsystem design strategy. **Techniques:** Project management and process management. Integrity of each subsystem, interface coordination and risk management of project schedule. **Techniques:** Information sharing and process management. Project schedule of system verification validation, system production and experiment. **Techniques:** Project management and process management. |
| **Test Engineer or Simulation Engineer** | Verification and validation of engine experiment data. **Techniques:** Requirement modeling, Simulation verification and validation of mechanical and conduit head. **Techniques:** system modeling, Multi-domain Simulation verification and validation of propulsion system. **Techniques:** system modeling and integrated Experiment verification validation of propulsion system. **Techniques:** |
Tool interactions 1 and 2 in Figure 6 are decided by viewpoint, ‘Data and model backups and management in cloud’. IT managers hope to back-up all the data in these two platforms; therefore these two data flows are needed. Tool interactions 3,4,5,6 and 7 in Figure 6 are decided by viewpoint, ‘Concurrent and collaborative design of system architecture, engine, PFS, traceability, database and relative technical viewpoints’. The specific technical viewpoint is not provided by this case study resulting here in only an abstract mentioning of techniques and tools. These technical viewpoints will be considered during MBSE tool-chain development and a modelling environment would support this approach in the future.

**Conclusion and Future Work**

The paper has proposed a methodology to analyse the functionalities for MBSE tool-chain development. The methodology aims to design an MBSE tool-chain by a systems engineering approach and provide a concept of model-based approach to represent the expected MBSE tool-chain to share its ideas with stakeholders in order to enhance their understanding of its functionalities. The case study illustrated that before developing an MBSE tool-chain, investigating and understanding the tool-chain’s functionalities are vital to its developers. In our methodology, we used system engineering principles to capture the stakeholders, viewpoints and views of the tool-chain to design its architecture. The conceptual model-based approach proposed in this paper allows stakeholders to share the development ideas with each other. In future work, we will address a user-friendly modelling environment to extend the proposed methodology in order to support assessment of different MBSE tool-chains and assist the decision making of MBSE tool-chain developers.

![Figure 6. Model-based representation of tool-chain in case study](image)

The following, eight stages (Kotter 1996), detailed below, will be implemented as part of future work:

1. A case study of a system will be selected and viewpoints from its stakeholders will be investigated.
2. A special team including specialists from domain and tool-chain developers will be formed to prepare to develop such an MBSE tool-chain.
3. Based on the investigation in step 1, the virtual MBSE tool-chain virtual prototype will be designed.
4. The MBSE tool-chain prototype will be shared with its stakeholders to discuss validity and value provided by this MBSE tool-chain prototype.
The first four steps are used for stakeholders to understand MBSE tool-chain and its importance from a system thinking way.

5. Following the virtual MBSE tool-chain prototype, the team will develop the MBSE tool-chain. The use of industrial standards and high maturity level techniques will be encouraged to be used for the new tool-chain.

6. Achievement of each technique, tool, and standard will be planned as a milestone called “short-term win”. Short-term wins will be created after the milestones which techniques, tools, and standards are validated by case studies,

7. With increasing credibility from each short-term win, the whole MBSE tool-chain will be developed. Traditional policies, organization structures and design process which do not fit this new tool-chain will be adapted accordingly. Training and education for this new tool-chain will be introduced, and new case studies will be used to improve the MBSE tool-chain iteratively.

Steps five to step seven are used for developing and implementing the new MBSE tool-chain.

8. Anchor this new MBSE tool-chain in the culture and articulate the connections between MBSE tool-chains and other new achievements of applications in practice.

References

Algirdas Avižienis, Laprie Jean-Claude, Randell Brian, L.C., 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 1(1), pp.11–33. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1335465&tag=1.

Anon, 2010. Functional Mock-up Interface for Co-Simulation MODELISAR. Available at: http://creativecommons.org/licenses/by-sa/3.0/legalcode [Accessed February 27, 2017].

Anon, 2011. IEEE Std 1471-2000, Systems and software engineering —Recommended practice for architectural description of software-intensive systems.

Association, M., 2005. Modelica specification, version 2.2. Available at: https://scholar.google.com/scholar?cluster=14771390706047104516&hl=zh-CN&as_sdt=2005&sciodt=0,5 [Accessed March 14, 2017].

Bayer T J, Cooney L A, D.C.L., 2010. An operations concept for integrated model-centric engineering at JPL. Aerospace Conference, pp.1–14.

Biehl, M. (KTH S., 2013. A Modeling Language for the Description and Development of Tool Chains for Embedded Systems.

Broodney, H., Shani, U. & Sela, A., 2013. Model Integration - Extracting Value from MBSE. INCOSE International Symposium, 23(1), pp.1174–1186. Available at: https://www.researchgate.net/publication/255720194_Model_Integration_Extracting_Value_from_MBSE.

Brusa, E., Ferretto, D. & Calà, A., 2015. Integration of heterogeneous functional-vs-physical simulation within the industrial system design activity. Systems Engineering (ISSE), 2015 IEEE International Symposium on, pp.303–310.

Cloutier, R. et al., 2015. Transitioning Systems Thinking to Model-Based Systems Engineering: Systemigrams to SysML Models. Ieee Transactions on Systems, Man, and Cybernetics: Systems, 45(4), pp.662–674.

Do, Q. et al., 2012. Requirements for a metamodel to facilitate knowledge sharing between project stakeholders. Procedia Computer Science, 8, pp.285–292. Available at: http://dx.doi.org/10.1016/j.procs.2012.01.059.

Do, Q., Cook, S. & Lay, M., 2014. An investigation of MBSE practices across the contractual boundary. Procedia Computer Science, 28(Cser), pp.692–701. Available at: http://dx.doi.org/10.1016/j.procs.2014.03.083.

Eckert, R., Mansel, W. & Specht, G., 1990. STEP AP233 + Standard PDM = Systems Engineering. PDM? Systems Engineering, (June), pp.405–412.

Elm, J.P. & Goldenson, D.R., 2012. The Business Case for Systems Engineering Study: Results of
the Systems Engineering Effectiveness Survey. (November), p.173.

Estefan, J.A., 2008. Survey of Model-Based Systems Engineering (MBSE) Methodologies. Environment.

Gomes, C. et al., 2017. Co-simulation: State of the art. Available at: http://arxiv.org/abs/1702.00686.

Graignic, P. et al., 2013. Complex System Simulation: Proposition of a MBSE Framework for Design-Analysis Integration. Procedia Computer Science, 16, pp.59–68. Available at: http://dx.doi.org/10.1016/j.procs.2013.01.007.

Hallqvist, J. & Larsson, J., 2016. Introducing MBSE By using Systems Engineering Principles. (Is).

Hein, A., Lopez, R. & Herzig, S., 2010. Object-oriented system models in spacecraft design: First steps towards an application in Phase B. Workshop on System & Environment, (1). Available at: http://www.congrex.nl/10C08/docs/10C08_SECESA_2010/s6_1145_hein.pdf.

Jinzh, L., 2016. A Model-driven and Tool-integration Framework for Whole Vehicle Co-simulation Environments. In 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016).

Johnson, J. et al., 1999. The future systems engineering data exchange standard STEP AP-233: Sharing the results of the SEDRES project. Proceedings of the 9th Annual INCOSE Symposium. Available at: http://www.dasi.unisa.edu.au/materials/99papers/johnson-futureSEDRES.pdf.

Kalawsky, R.S. et al., 2013. Bridging the gaps in a model-based system engineering workflow by encompassing hardware-in-the-loop simulation. IEEE Systems Journal, 7(4), pp.593–605.

Kotter, J.P., 1996. Leading change.

Lee, E., 2010. CPS Foundations. Proc. of the 47th Design Automation Conference (DAC), ACM, pp.737–742.

Paredis, C.J.J. et al., 2013. Model-Based System Integration (MBSI) – Key Attributes of MBSE from the System Integrator’s Perspective. Procedia Computer Science, 16, pp.313–322. Available at: http://www.sciencedirect.com/science/article/pii/S1877050913000343.

Pavalkis, S., 2016. Towards Industrial Integration of MBSE into PLM for Mission-Critical Systems. (Is).

Petnga, L. & Austin, M., 2013. Ontologies of time and time-based reasoning for MBSE of cyber-physical systems. Procedia Computer Science, 16, pp.403–412. Available at: http://dx.doi.org/10.1016/j.procs.2013.01.042.

Ramin, 2015. Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL. International Journal of Conceptual Structures and Smart Applications (IJCSSA), 3(2), pp.25–70.

Scheeren, I. & Pereira, C.E., 2014. Combining Model-Based Systems Engineering, Simulation and Domain Engineering in the Development of Industrial Automation Systems: Industrial Case Study. 2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, pp.40–47. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6899129.

Schumann, H., Zamov, P. & Escher, S., 2011. Model-based design and tool data exchange in aerospace: a case study. CEAS Aeronautical Journal, 2(1–4), pp.295–303.

Selic, B., 2012. What will it take? A view on adoption of model-based methods in practice. Software and Systems Modeling, 11(4), pp.513–526.

Shani U, L.A., 2013. Tools Interoperability Platform for Model-Based Systems-Engineering. 8th Workshop on Model Based Software, Data, Process and Tool Integration (together with the PMDE Workshop).

Sillitto, H.G., 2010. 1.3.1 Design principles for Ultra-Large-Scale (ULS) Systems. INCOSE International Symposium, 20(1), pp.63–82. Available at: http://doi.wiley.com/10.1002/j.2334-5837.2010.tb01057.x.

Verma, D. & Fabrycky, W.J., 1997. Systematically identifying system engineering practices and methods. IEEE Transactions on Aerospace and Electronic Systems, 33(2 PART 1), pp.587–595.

Votintseva, A. et al., 2012. Comparative study of model-based and multi-domain system engineering approaches for industrial settings. Lecture Notes in Computer Science (including subseries
**Biography**

Jinzhi Lu, Ph.D. student at KTH Royal Institute of Technology, Mechatronics Division. Meanwhile he works in Suzhou Tongyuan Software and Control Technology Company as a project manager. His research interest is MBSE tool-chain design and MBSE enterprise transitioning.

DeJiu Chen received his PhD degree in Mechanical Engineering with a research on embedded computer control systems from KTH in 2004. His main research areas include 1. Systems- and software-architecture design, 2. Model-based engineering, and 3. Advanced situation-awareness and self-adaptation, for safety- and time-critical cyber-physical systems. From 2007 to 2009, DeJiu Chen has worked for Enea Data AB, Sweden, as a senior technical instructor. He is currently an associate professor at KTH.

Didem Gürdür received her B.Sc. degree in Computer Engineering and M.Sc. degree from Sustainable Environment and Energy Systems program. Currently, she is a Ph.D. student at KTH Royal Institute of Technology, Mechatronics Division, in the research area of Integrated Engineering Environments. Her current research interest includes interoperability assessment, data visualizations, visual analytics, knowledge discovery, tool integration and interoperability in Cyber-physical Systems.

Martin Törngren has been a Professor in Embedded Control Systems at the Mechatronics division of the KTH Department of Machine Design since 2002. He has particular interest in Cyber-Physical Systems, model based engineering, architectural design, systems integration, and co-design of control applications and embedded systems. He has authored/co-authored more than 100 peer reviewed publications, and also been in charge of developing and leading graduate and continued education courses. He spent time as a post-doc at the EU-JRC, and did a 10 month sabbatical 2011/12 at UC Berkeley.