Experience from Introducing SysML into a Large Project Organisation

Erik Herzog, Henric Andersson, Jessica Hallonquist
SAAB Aerosystems
SWEDEN
{erik.herzog, henric.andersson, jessica,hallonquist}@saabgroup.com

Abstract. Introducing a model centric way of working in large organisations is difficult. At SAAB Aerosystems it has been tried over multiple years in many projects. Success has been limited.

In our latest attempt to introducing SysML as a tool for system design it was decided to change focus. Instead of trying to promote models as the new way of documentation it was decided to setup the tools with the objective to create consistent and readable document centric deliverables. Instead of giving the modellers the freedom to use the language at its extreme, it was decided to limit language use to an absolute minimum.

As a result there have been some quite drastic initial complaints from our some experienced modellers, but surprisingly strong acceptance of the new approach to modelling from the stakeholders responsible for quality control, but not directly involved in the core development team.

Introduction

Communication between individual engineers, between IPTs and between projects - instantly as well as over time is problematic. In large organisations a lot of time is spent on ensuring that knowledge is transferred properly among stakeholders. Still, as a rule, a lot of problems surface late in projects due to omissions and misinterpretations. For complex systems this is not surprising, but must be minimised as it is driving cost.

Model Based System Engineering has been promoted as a way to improve specification quality and improve communication. (Oliver et al, 1997; Alford et al, 1992; Friedenthal et al, 2000; Wymore, 2002 and Long and Baker, 1996) have all promoted the use of models. The advent of SysML and the gradually maturing SysML implementation is COTS tools have further increased the interest in the Model Based Systems Engineering - at SAAB Aerosystems as well as internationally. An overview on methodologies for use with SysML is presented by Estefan (Estefan, 2007).

Still there are few papers reporting on experience made and problems encountered when introducing SysML. Piggott et al (Piggott et al, 2008) has reported on the experience from introducing MBSE in a comparably small project in a small organization. Piggott report that the core development team - the modelers - was generally content with the use of MBSE, but problems were encountered when trying to use models to communicate design throughout the organization. Similar results can be read out from SAAB Aerosystems (Andersson et al, 2009) report on

experience from introducing UML/SysML in the development of an autonomous helicopter. In this case the emphasis was placed on training of the modelers, but problems was encountered when the models was distributed for review and implementation to external stakeholders.

Both of these examples, no matter how successful they were, are comparatively small projects. We have not found any firm guidance on how to introduce MBSE in really large project organizations. But we have a growing suspicion that many problems encountered when introducing MBSE are due to the way models are presented to stakeholders outside the core development organization, i.e., the people not directly involved in the systems modeling activity. In our internal modeling projects we have seen a tendency from modelers to embrace many, if not all, new possibilities offered by the modeling tool and paradigm, while less emphasis is place on traditional textual documentation. Thus leaving the rest of the organization uneasy with the results. What do the models really convey as compared to traditional documentation?

Based on this a decision was taken to ensure that the next MBSE methodology developed should focus on how to integrate the models as seamlessly as possible with traditional system engineering deliverables. Also focus should not be embrace all as aspects of the modeling languages, but as little as required to ensure that design is properly communicated to stakeholders. Naturally this implies some restrictions on the freedom of the modelers, but those are acceptable as long as the acceptance of the results of modeling among all stakeholders improve.

The rest of this paper is outlined as follows: The rationale for introducing SysML in the Gripen NG project is presented followed by an analysis of inhibitors to the acceptance of MBSE identified. The model usage and selected model structure is presented prior to an analysis of the introduction activities in development teams and the experiences made. A summary and conclusion concludes the paper.

Figure 1: SAAB-39 Gripen

Rationale for Introducing SysML

SAAB Aerosystems is developing airborne systems and their support systems. The primary product is the SAAB 39 Gripen lightweight fighter aircraft. Gripen has been in development for about 25 years and in operational service since 1997, and is still being updated. Currently

development work on a substantial upgrade – the Gripen NG – is underway. Modifications include:

- Modifications to airframe including larger fuel tanks and repositioning of the landing gear
- Introduction of a active electronically scanned radar
- Introduction of an Arinc-653 based avionics system
- Installation of a new more powerful jet engine

The upgrade is far reaching and will result in modifications or replacement of many subsystems in the aircraft. Consequently, there is a need to communicate the new architecture and design to large number of people who need to review, understand, influence and react to the information provided.

Model based systems engineering languages has been in use for a long period of time at SAAB Aerosystems. Tool usage has, however, been limited to isolated parts of the organization. There has been no modeling methodology common to the whole organization and no common modeling language. Also, throughout the organization there is a tradition of expressing design in terms of text formulated in requirements syntax. Design documentation is frequently filled with paragraphs expressing how the systems is to be realised using the “shall” form with little graphical design content. The described approach is feasible, but requires substantial effort for readers to decode and synthesize the overall design from individual text statements.

For Gripen NG development the objective is to capture system design in visual format to improve communication and specification quality. There is also the objective to introduce modeling tools that support domain modeling, e.g., control systems modeling, software modeling, and code generation.

SysML has been selected as the base notation for capturing system design within the Gripen NG project. The rationale for selecting SysML is that:

- It is a standard, and the standard is supported by multiple vendors. This is important as it is expected that the Gripen product will be refined and maintained up to 2040. The fact that SysML is a standard does, of course, not guarantee that it will be around longer than any alternative language, but the risk is perceived to be lower.
- It is UML - but simpler. SysML is still a large language but is less excessive than UML. The smaller footprint of SysML allow for lower cost in training and fewer alternative usages of the language.
- The fact that SysML and UML are siblings allow for minimizing the transition cost from systems to software development.
- There is a substantial positive experience pool from using a UML subset very close to SysML on an existing product – the Skeldar UAV helicopter.

MBSE inhibitors

As stated above, there have been multiple attempts to increase the use of models in system design at SAAB Aerosystems. We also have a paradox in the sense that use of models has yielded positive experiences in all projects where it has been applied. This is especially true, as mentioned above, in the Skeldar project. So how could something that is generally perceived as being positive be so

difficult to introduce? There are clearly some strong inhibitors to the wider acceptance of model based systems engineering within SAAB Aerosystems. Our internal analysis point at the following MBSE inhibitors:

1. Training of third parties: While the modelers themselves are keen to take on a new language it is very difficult for other stakeholders to keep up with the pace. This problem is not only related to interpreting the models, but also in navigating the model structure. Unlike a textual document there is no readily available strategy for a traversing a large model database and ensure that all parts of the model have been visited.
2. The relevance of the models related to other deliverables: When a model has been created it is not clear how it relates to traditional document deliverables. Should these be completely replaced by the model or should they be complimentary. In the second case the question is how a minimal amount of redundancy can be ensured? The question is also which model elements should be maintained over time. All or just a subset? The items described above are in a sense trivial from a technical point of view but does cause a lot of waste unless there is a clear strategy formulated.
3. Configuration management: Our organization has a well established infrastructure for document CM. Establishing proper change and version management in SysML tools is feasible, but require some careful thinking to prior to implementation.
4. Acceptance of model deliverables: This may also sound trivial, but in fact we have a good understanding on how to manage and approve document based deliverables, but no comparative experience or support for doing this on a model database. Even if there are means to state that a model database is approved there is the problem whether this implies that all information in the database is so, or just a subset?
5. Too ambitious expectations on MBSE: The modeling tools open many news opportunities in terms of system specification - trying to attain them all result in a poor overall performance. The purpose with the model must be clearly stated and understood by all stakeholders.

Selected approach

When defining the methodology the following areas were primarily considered:

- The importance of documentation generation
- Modeling language usage

In line with the identified inhibitors the first priority was to ensure that proper document based design description deliverables can be generated. The document is the primary artifact for communicating design – not the model. A consequence of this is that the model may contain information that will not end up in the official documentation. May it be transient information used for communication during design, or alternative solutions investigated but not selected.

The objective was not to create the ultimate deliverable in terms of depth and content – just something marked superior to the current local state of practice. The objective is to encourage models to capture overall system design, i.e.,

- Identification of system components and component interfaces
- Identification of overarching system functionality

- Definition of sub-functions, functional interaction and the allocation of functionality to components

Here it is important to note that all parts of the deliverable need not be SysML formatted. Information that is more conveniently represented in free text or graphics shall remain in this format and integrated in the deliverable with the document is generated. This does not imply manual integration; most SysML tools provide the facility to associate Microsoft Word files with model elements and include these files in the generated documentation. Using this approach model and document manipulation is managed from the SysML tool, thus creating an integrated environment.

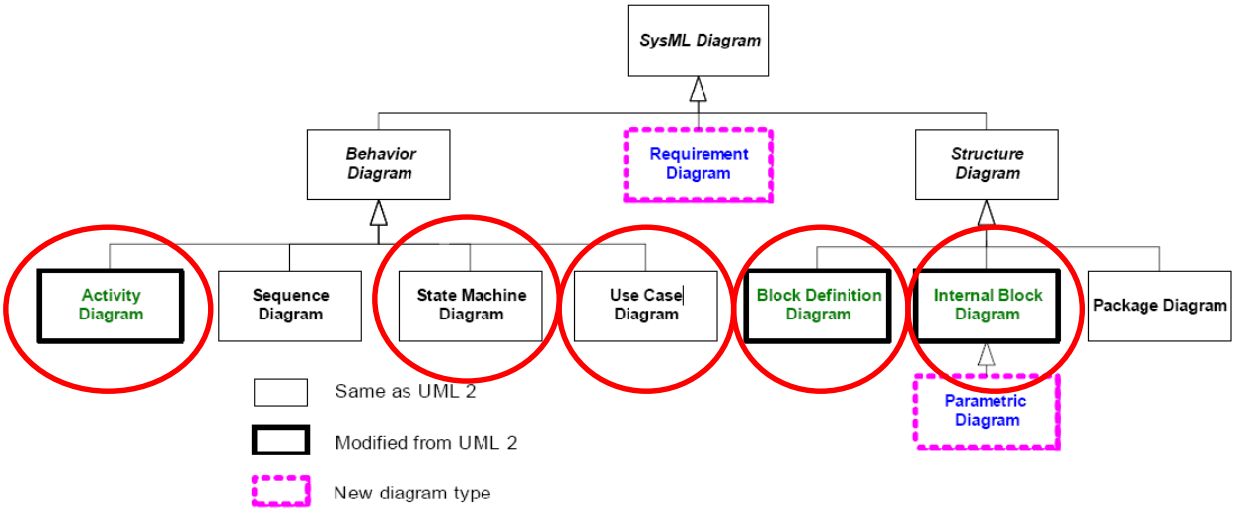

Figure 2: Used Subset of SysML

Figure 2 captures the scope of SysML in terms of diagrams. The encircled diagrams are those selected for use each with the following purpose:

- *Use Case Diagrams* are used to capture the top-level functions provided by a system. As clear separation as possible is sought for Use Cases. Multiple Use Case Diagrams may be used to capture the functionality of a complex system.
- The functionality of individual Use Cases are defined in one or more *Activity Diagrams* or *Statechart diagrams*. Sub-functions identified in the Activity diagram are represented using Action objects and allocated to system components using Swimlanes. The Swimlanes serves as the link between the functional and the physical view on the system.
- System components are identified in *Block Definition Diagrams* and component interfaces captured in *Internal Block Diagrams*. Depending on system complexity the capture of a physical architecture may be sufficient; alternatively it may be required to capture a logical architecture and the allocation of logical entities onto physical components.

Textual documentation is primarily captured on diagram level as opposed to documenting individual objects. The rationale for this is that a heading is created in the generated documentation for each object that information is taken from. If there are many objects and each documented with a short text then the generated documentation will be too fragmented for interpretation.

Requirements are not managed within the SysML model at all. All requirements are maintained in

IBM DOORS, using the same methodology as previously applied within the organization.

Model structure

The complexity of the Gripen NG system motivates maintenance of system models at different abstraction levels. The system architecture model is used to identify the overarching functionality and how this is allocated to the main system components. At this level component interfaces are identified, but not captured in detail. The system architecture model defines the baseline for further detailed development. Complex system functions are is refined in separate detailed design models. These capture the detailed functional design and functional allocation to system components for implementation.

The use of separate models for system architecture and detailed design is important as it allow separate baselining of the two models types. A baseline of the System architecture model may be created for development of a function model development increment. Once the baseline is taken work may proceed at System architecture level towards the architecture for the following increment.

Internally within each model a common package structure is used. A package in SysML (and UML) corresponds loosely to a folder in a file system. Packages may contain other packages just as folders in the file system may contain other folders. The following top-level packages are mandated:

1. Introduction, capturing the scope, introduction and reference sections of the deliverable generated from the model. All information kept in the introduction section are formatted in Microsoft Word and associated with the model (The term controlled file is used within the tool to represent such information).
2. Overarching design decisions, capturing the design principles applied as well as the design decisions taken. Information in this section is expected to be in formatted in Microsoft Word with some odd model elements.
3. System components, identifying the physical and logical components a system comprises of. The information in this section normally comes Block elements captured in Block Definition Diagrams and the textual descriptions associated with the diagram and the diagram elements. External files may be associated with any diagram or element.
4. Component interfaces, identification of component interfaces is captured in Internal Block Diagrams, and textual definitions are captured per diagram. External files may be associated with any diagram or element as appropriate for documentation.
5. Concept of operation, all behavioural diagrams are captured in the Concept of operation package. Use case diagrams serve as the entry point where each system function is identified. Any number of Activity diagrams or Statechart diagrams may be associated with a use case. Textual descriptions are captured for each use case and each associated diagram.
6. Requirements allocation, this package captures the allocation of requirements to components/behaviour oriented diagram. The information is captured in a Microsoft Word file generated from DOORS where the source information is captured.
7. Notes, the notes package captures any additional information relevant for system design.

The information in this package is expected to be captured in Microsoft Word format.

The set of packages listed above corresponds to the structure of a traditional SSDD type document. This structure is traversed by the document generator macro defined in such a way that information is extracted such that a properly formatted SSDD document is created. The benefit of this approach is that the system design description deliverable is in a format familiar to all internal stakeholders in the organization. The content of the document generated is not the same as it has traditionally been within the organisation, but model interpretation is simplified as each diagram from the model is presented in the context defined by the document structure. Also the extensive usage of information in Microsoft Word formatted files allow the modelers to present information using the best suited format.

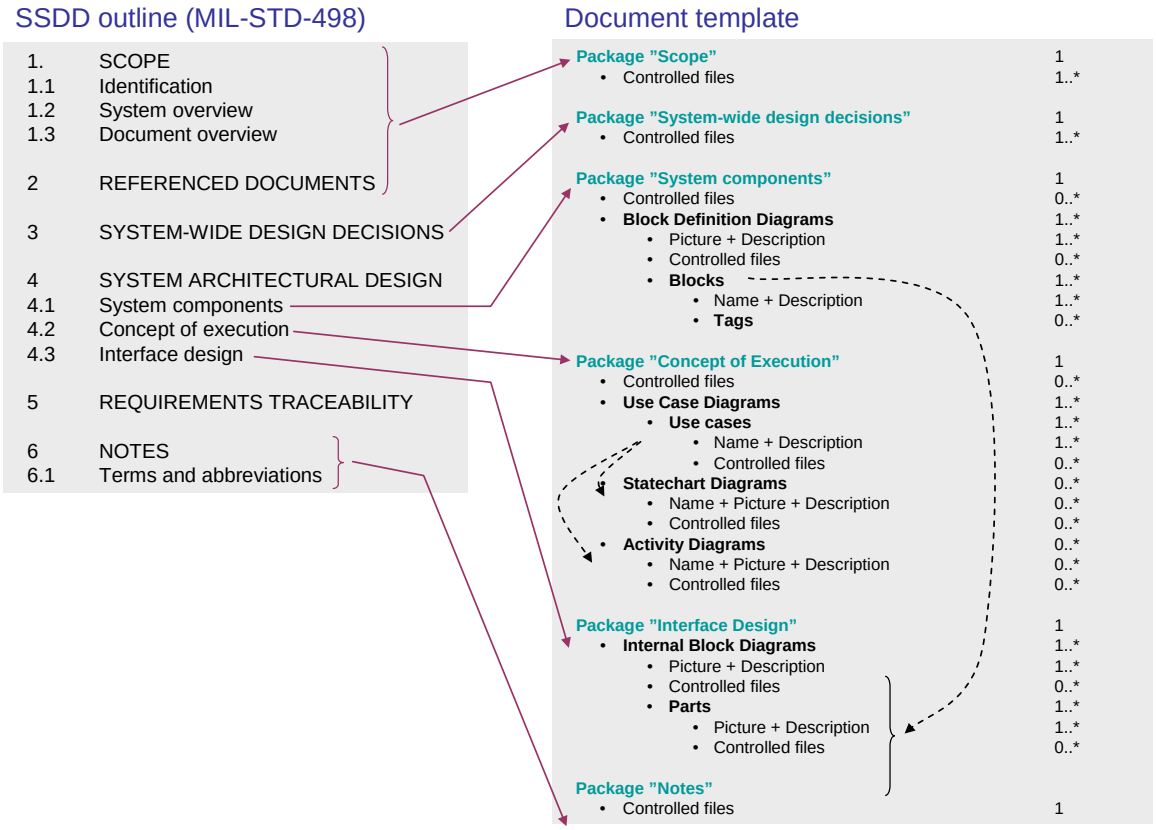

Figure 3: Information - Documentation mapping

The system modelers may elect to create any number of top-level packages beyond the seven listed above. The information kept in any such package will be part of the model database, but not included in the deliverable generated from the model. Using this approach it is possible to keep track of the information for inclusion in deliverables and support information used as part of the design activities.

Methodology introduction in development teams

The modeling approach described herein has been communicated to development teams and other internal stakeholders through a 2-day integrated SysML, modeling methodology and tool familiarization course. The course captures the basic method, the rationale for the methodology

selected and an overview over diagrams, diagram elements to use. More than half of the course time is used for tool exercises – in this case IBM Rhapsody. In addition the methodology is documented in the wiki on the local intranet. In addition support from methodology experts are offered when new subprojects are started.

Experiences made

The basic premise in the Gripen NG project has been to use modeling more. This premise has been well communicated internally in the organization, but how to perform the modeling took time to define. After introducing the methodology in teams the reactions have been diverse.

- A common negative reaction stems from the exclusion of certain SysML diagrams in the methodology. This is in particular true for Sequence diagrams that have been in extensive use in previous modeling projects.

Most negative comments in this area have been managed by underlining that focus is on identifying functionality, functional interaction in terms of flow rather than the message passing paradigm inherent in the Sequence diagrams. When modeling in practice modelers have also been positive about the support for including more complex behaviour in a single Activity diagrams as compared to what is possible in a Sequence diagrams.

- Likewise there have been negative reactions to the strong emphasis one the generated documentation and the constraints that result from this emphasis.

Such negative comments primary came from people with previous modeling experience and interestingly from people attached to the system architecture modeling team. Their main argumentation was along the lines that generally the methodology is fine, but for their particular task another modeling methodology is required.

This turned out to be a very difficult conflict to resolve. Multiple meetings was performed to resolve the differences with little progress to show of as results. In fact it turned out that the conflict was more about the freedom of the engineers to decide their working methodology themselves as opposed to having methodology specialists dictating how the tools should be used. In the end the system architecture group decided to use its own methodology. The biggest problem encountered was that the information inserted was very difficult to export to a readable document.

We believe that such conflicts between the actual development engineers and methodology experts are not uncommon. The lesson learned is very clear: Introduction of new methodology need to be performed in due time prior to project initiation.

On the other hand all function development teams readily accept the methodology and the scope it gives to design. In this group these modeling aspects are appreciated

- The clear scope on what shall be captured in the models
- The freedom to use non-SysML formatted information in deliverables. In particular the importance of text is underlined.
- The focus on the deliverables, and the ease with which the deliverables can be communicated. In particular, the design, and open issues related to design, is easier to communicate to management and other external stakeholders.
- Dependencies between system elements are easy to capture and communicate. Moreover,

the complexity of the system under development is clearly communicated by the model.

- No problems have been experienced among third party stakeholders to accept the document paradigm and the limited scope of SysML language artifacts used as it simplifies interpretation. The favourite anecdote is that one skeptical chief engineer exclaimed that the models as manifested in the deliverables are no different from the block diagrams he used to construct in the late 1980'ies.
- The common model structure is appreciated as is it clear which information goes where. This simplifies work for the individual team members when working in their own models, but also for external parties browsing the model of another team.

Based on interviews made with project members it is clear that, overall, the development teams are happy with the methodology. No one has expressed a desire to revert to document based methodology previously in use. The following items are fed back in the interviews:

- The defined methodology allows the teams to focus on their work. There is no need to discuss the advantages and disadvantages of different modeling approaches.
- Likewise, it is important that the objectives with the models is understood by the modeling team and external stakeholders.

Still there are items for improvement. It is noted that:

- SysML implementations in tools are not yet stable. During the course of time the modeling tool has been updated and the new version included a case where two end user items replaced what was previously a single one. As a result the engineers had to go through and update all models in order to maintain model consistency in the new tool version.
- The document generation software delivered with IBM Rhapsody is very powerful, but not always consistent in finding model elements.

Conclusions

The novel aspect in the presented approach to introducing SysML into a large organization is not the extent of model artifacts used, but the minimalist approach taken and the focus on the ability to create traditional documentation deliverables. In fact the approach is probably not novel at all, we are convinced that it has been applied in multiple projects, but not, to our knowledge, published. Still, we believe, the principles presented herein are important for the acceptance of SysML and other modeling languages in large organizations. The focus must be one of finding usage for models within the bounds given by the knowledge base and infrastructure of the organization rather than using modeling to its outmost. Once modeling is introduced and accepted the modeling methodology may of course be extended to capture more aspects of the system.

Successful introduction of SysML need not require focus on all capabilities provided by the language, but to find the key capabilities required to improve the performance of the organization.

Acknowledgements

The authors would like to thank the Gripen NG project and in particular the NG Radar project for the support offered when preparing this paper.

References

M. Alford et al. Improving the Practise in Computer-Based Systems Engineering, Proceedings of the 2nd Annual Symposium of the National Council on Systems Engineering, INCOSE, 1992.

H. Andersson, E. Herzog, G. Johansson, J. Johansson, Experience from Introducing UML/SysML at SAAB Aerosystems. In H. Andersson, Aircraft Systems Modeling. Licentiate Thesis No.1394 Department of Management and Engineering. Linköpings Universitet, 2009.

J. Estefan, Survey of Model-Based Systems Engineering (MBSE) Methodologies, INCOSE MBSE Focus Group, INCOSE 2007. http://syseng.omg.org/MBSE_Methodology_Survey_RevA.pdf, accessed 2009-11-07

S. Friedenthal et al, Adapting UML for an Object Oriented Systems Engineering Method, Proceedings of the 10th annual International Symposium of the International Council on Systems Engineering, INCOSE, 2000.

J. Long and L. Jr. Baker Specifying A System Using ERA Information Models. Proceedings of the 6th annual International Symposium of the International Council on Systems Engineering, INCOSE, 1996.

D. Oliver et al, Engineering Complex Systems–with models and objects. McGraw-Hill, 1997.

Piggott, Stephen; Melanson, Philip; Hartman, Leo; Adourian, Chahe, Advancing Model-Based Systems Engineering at a Small Space Agency, In Proceedings of the 18th annual International Symposium of the International Council on Systems Engineering, INCOSE, 2008.

W. Wymore, A System Theoretical Framework for V & V. Proceedings of the 12th annual International Symposium of the International Council on Systems Engineering, INCOSE, 2002.

Dr. Erik Herzog is a Systems Engineering specialist at Saab Aerosystems AB. Dr. Herzog received his Ph.D. at the Department of Computer and Information Sciences at Linköping University, Sweden. His professional interests include development and introduction of Systems engineering processes, specification methods, information modelling and tool integration techniques. Dr. Herzog is active in multiple INCOSE working groups and has implemented the INCOSE i-pub system.

Henric Andersson is a control engineer at Saab Aerosystems in Linköping. He works in the Systems Engineering Division responsible for recommending methods and tools for use on Saab aerospace development programmes. The key interest is modeling techniques, requirements management and tool integration with special focus on systems design and simulation. He is also an industry PhD student in model based systems engineering at Linköping University, Sweden.

Jessica Hallonquist is a Systems Engineer at Saab Aerosystems. She works in the Tactical Systems Division integrating the radar onto the Gripen fighter, using the new methods. She has a Master of Science degree, Wireless systems, from the Royal Institute of Technology, Stockholm, Sweden.