

# Towards a Collaborative Working Environment to Support Model-Based Systems Engineering

Matthias Merk<sup>1</sup>, Gabriela Tullius<sup>2</sup> and Peter Hertkorn<sup>3</sup>

**Abstract:** This paper presents an approach to develop a collaborative working environment for engineering support in the field of model-based systems engineering. The need for such a system is motivated and a concept for an adaptive CSCW environment is shown.

**Keywords:** CSCW, Model Based Engineering, Engineering Support Tools, Collaboration, Working Environments

## 1 Introduction

According to Eigner et al. a modern product design process in engineering requires the following skills from an engineer: First the willingness to optimize the complete design process not only for product functionality, but also to support all subsequent product lifecycle phases like maintainability or recyclability (Design for X). Second, Engineers need to collaborate in early design phases and work interdisciplinary during all phases of the product lifecycle. Third the willingness to work with co-located engineers (distributed teams) as well as the willingness to use new methods, processes and software tools to support product development in all aspects of the engineering process [ERZ14].

This paper is focused on the research in the field of software tools for systems engineering, especially for the support of collaborative work (CSCW). According to Eigner et al. the usage of software tools in systems engineering means to use software systems not only to generate the product itself, but also to generate documentation of the work result [ERZ14]. Over the last years the focus of the development of such tools was on the support of product management issues and not on the functionality of the tools to support the design of the product [BR11].

According to Eigner et al. a decade ago the engineering community focused on CAD-Systems as software tools in the design process. Today their focus shifted to Product-Data-Management-Systems (PDM) that allow an integration of product related data throughout all the steps in the product lifecycle. This change happened because engineering activities in design and development switched from creative activities towards administrative, communicative, informative and team-oriented decision making where knowledge and

---

<sup>1</sup> Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, matthias.merk@reutlingen-university.de

<sup>2</sup> Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, gabriela.tullius@reutlingen-university.de

<sup>3</sup> Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, peter.hertkorn@reutlingen-university.de

data (digital product and process model) is the central element [ES01]. The digital representation of the product itself is described through different sequencing levels according to Bordegoni et al. The product is represented through a model with increasing complexity. To deal with this increasing complexity, engineers need to add knowledge to the model at each level to ensure that the engineers are able to reach the design goals [BR11]. According to Eigner et al. the usage of models instead of documents is an advantage especially in interdisciplinary projects [ERZ14].

New trends in engineering like model-based systems engineering (MBSE) are demanding new software systems to meet the changed requirements in the engineering process. MBSE is defined as a formalized usage of modeling to support the definition of requirements, development, verification and validation of a system beginning with the concept- and development phase as well as all the subsequent phases in the product lifecycle [In07]. Albers et al. are listing the main advantages of MBSE: fewer inconsistencies, less redundancies, clear communication and sustainable documentation [AZ13].

PDM-Systems are focusing only on the management and provisioning of product related data, not creation of the artefacts. Software tools to support model-based engineering need to provide a user-interface for the generation of the models PDM-Systems and MBSE-Support-Systems should work together. The PDM-System is delivering data in terms of knowledge and MBSE-Support-Systems like tools for modeling are used to design the model. You could see the PDM-System as a database for other tools.

Eigner et al. define software tools in a model-based engineering context as tools that use a formal language to describe the product in each phase of the product lifecycle. The modeling of the product is often supported by an easy to learn form of graphical input that is built upon digital models. The model is then provided to other product lifecycle phases through model transformations. The basis for a MBSE centred design process is formed by software and modeling tools that support the specification of requirements, functions and behavior of the product e.g. SysML, CAD and product life cycle management (PLM) systems [ERZ14].

According to Bordegoni et al. there should be one unique, consistent, and software independent model, that is used through all of the product phases. Such a model would improve both productivity in the engineering process and the resulting quality of the model itself [BR11]. Albers et al. show, that the problems of such a model is in the interdisciplinarity of engineering design processes. Term understanding for example can become a major communication problem. As an example Albers et al. show the different understanding of the word “function” in interdisciplinary engineering teams that is described differently as an appearing phenomenon, an effect, a behavior, a description what a system should do or a piece of software code processing data [AZ13]. We need to encounter the problem of differences in term understanding to ensure an effortless documentation, collaboration and communication throughout the system and its users.

Model-based virtual product development (MVPE) is an extension of MBSE. It extends the practices of MBSE with the aim of using as many virtual prototypes as possible [ERZ14]. Bordegoni et al. describe a paradigm shift in product development from a geometrical to a functional view [BR11]. According to Bordegoni et al. the shape of an object is not static. It is a “function of the environment, of time, of the history of the phenomena affecting the object” [BR11].

## 2 State of the Art

There are a number of computer-aided innovation (CAI) systems that are used to support the actual design of the product. These systems are often based on TRIZ, the theory of inventive problem solving developed by Genrich Alschuller [Al84]. The focus of TRIZ is the support of early phases in product development. Cugini et al. describe the problem of current CAI systems in the limited interoperability with other CAx tools [Cu09]. The PROSIT project tries to solve this problem through bridging CAI tools with PLM systems. According to Bordegoni et al. a modeling approach capable of representing a product at different detail levels is needed to further address this problem [BR11].

Web 2.0 technologies are used to extend CAD, PDM or PLM tools with social components to allow collaboration by discussing and sharing CAD models between co-located team members (Autodesk social share plugin<sup>4</sup>, GrabCAD Workbench<sup>5</sup>). According to Bordegoni et al. PLM systems claim to support any stage of product development but are still far away from a systematic support in the early design phases [BR11]. The work of Ubik et al. highlights that low latency remote access to 3D models can enable effective distributed collaboration over large distances [UT13]. Kim et al. proposes a method for using the X3D visualisation standard to include part geometry, product structures, and manufacturing-related properties in a web browser to have a CAD independent distributed form of visualisation [Ki15]. This technology could be used to extend web based collaboration platforms.

There is also a number of commercially available tools for the collaborative creation of different diagrams that could be used for model-based engineering methods. A small overview of three tools is presented. The focus of these early observations in commercially available tools is on the availability of data exchange and thus the possibility of integration into existing workflows. This listing is far from being complete. Further research and analyses needs to be done, a suitable method for classifying and evaluating has to be found.

---

<sup>4</sup> <https://apps.autodesk.com/ACD/de/Detail/Index?id=837454399538248119>

<sup>5</sup> <https://grabcad.com>

Cacoo<sup>6</sup> is a web-based tool for collaborative creation of diagrams including UML class diagrams, mind maps or flowcharts. It offers features like role and resource management and the ability to share and export the diagrams. Export is limited to PNG, SVG, PDF, PS or PPT. What is missing is a universal interchange format that would allow a tight integration in the model-based engineering domain.

Creately<sup>7</sup> is another web-based tool like Cacoo. It is able to show changes in real time to all collaborating users and supports commenting and discussing on models as well as a revision history of the changes made to the model. Export is available in PDF and SVG.

VisualParadigm<sup>8</sup> offers the ability to design diagrams like UML or SysML and is using a central repository allowing stakeholders to comment directly on the created diagrams which can be shared online. It offers collaborative modeling functionality where you can checkout, commit and update the diagrams stored in the repository. It is possible to integrate custom plug-ins written in Java, so integration in an existing workflow is possible. The export of the diagrams in a universal interchange format (XML) is also supported.

### 3 Concept

In this part the concept of the system is described. The concept is divided in five parts: (1) a description of our user group, (2) a basic methodology for the development of domain-independent collaborative modelling tools, (3) the requirements of the system and (5) its basic system architecture. An example scenario at the end of this section puts the parts in relation.

#### 3.1 User-Group

Our System is designed for engineers using model-based engineering methods. We follow the approach of the human-centred design for interactive systems [ISO10], therefore it is essential to develop a good understanding of the typical users and their activities. To gather the requirements of the main user-group, an initial persona [Co99] based on statistics e.g. provided by the “Verein Deutscher Ingenieure” (VDI - German Engineering Society) and interviews with engineers has been created. We interviewed research associates with engineering background and experience in model-based engineering to identify the requirements of our user-group. In a semi-structured open expert interview, we asked the discussion partners about collaboration tools they use for work and private tasks. We also asked them to describe their typical engineering workflow and the differences to the model-based approach. Further questions were asked to find problems they encountered

---

<sup>6</sup> <https://cacoo.com>

<sup>7</sup> <https://creately.com>

<sup>8</sup> <https://www.visual-paradigm.com/features/collaborative-modeling>



while working with model-based engineering software in general. The interviews were finished with an open discussion about future trends in engineering support and needs and requirements the discussion partners expect to see in the future. However, we will need further interviews to gain a much deeper insight in the working practices of this user-group.

### 3.2 Methodology

In [GBR12] a model-driven method for the development of domain-independent collaborative modeling tools is proposed. Gallardo et al. focus their work on domain-independent modeling tools, supporting the work of co-located designers working in shared workspaces using a whiteboard metaphor. The method describes four types of users. First, the user who is developing the collaborative modeling tool, second the domain expert with knowledge about the domain the tool will be used in, third a software engineer who may participate in phases where software development tools need to be manipulated, and fourth the user who will use the developed tool to build models in collaborative design sessions.

The method of Gallardo et al. is based on the following three frameworks:

**A methodological framework:** a series of phases that must be followed by the non-expert user who wishes to develop a collaborative modeling tool. These phases are: the identification of the domain, the modeling of the domain and the workspaces, the production of the collaborative modeling tool, which includes the model transformations and the generation of the tool itself and the usage of the generated tool.

**A conceptual framework:** made up of the models that are used in the meta-modeling process. These models are mainly the domain and workspace meta-models

**A technological framework:** consists of a series of plug-ins for the Eclipse platform that have been modified and extended to generate collaborative applications

For now, we will focus our conceptual work on the methodological framework. The methodological framework will form the base to all future work.

### 3.3 Requirements to the System

First, there are requirements which are special to the user group of our system or are defined in literature written by the user-group itself: Bordegoni et al. outlines that “there is the necessity of developing new, very friendly, very interactive interfaces, which will attract new users and support faster design iterations” [BR11]. User experience and usability research will be a major part in the design and development of the system. We focus on creating a system for the user group that adapts to the user. We don't want the user to adapt to the system. Lemberg et al. outlines that early artefacts in the first design

phases need to have a preliminary character, so that making changes in these artefacts feels easier for the designers [LF09]). To our system this requirement means, that we need to provide data structures that allow fast, easy and unrestricted editing, manipulation and augmentation of the project data or its visualisation. The data and its visualisation needs to be divided in a representation and a data layer. Because of the adaptive design of the system, different representations and visualisations of the data can coexist in different versions. Eigner et al. describe essential things a software support needs to do in order to improve the engineering process which can be divided into two main parts: Collaboration support and the provision of the right information in the right time [ES01]. Communication support forms the basis of our system. This basis is then extended through other modules / visualisations allowing the representation of data suitable for the current task needed by a specific user group.

Second, there are requirements originating from general CSCW and innovation research: According to Herstatt et al. the stimulation of creative thinking is a prerequisite for the generation of breakthrough ideas [HK05]. According to Kyng, the most severe problems of CSCW systems are originating from a lack of end-user involvement in the development of the CSCW system [Ky91]. To overcome this problem, we will use a user-centred-design approach throughout the development of our system.

Third we have the wishes and needs of our user group. One insight we gained from these interviews is, that the interviewed engineers would like to have classical creativity techniques that should be supported by a CSCW system. As an example method 365 by Bernd Rohrbach [Ro69] was named. There is a huge number of creativity techniques from different domains (see [HM12] for an overview) that have to be evaluated if they can be integrated into a CSCW system. Another insight is the need for adaptive user interfaces. Engineering projects are often interdisciplinary and every domain uses its own working methods and tools. Therefore we need to develop a highly adaptive and customizable system to support the interdisciplinary field of engineering. With this high level of adaptivity, we are able to implement the ideas and needs of our user-group. Because of the interdisciplinary nature of engineering projects, we will need to implement our system using modular approaches to support the different needs of the domains involved in an engineering project.

### 3.4 Basic System Architecture

Figure 1 gives an overview of the basic concept of our adaptive CSCW environment to support collaboration in engineering. The technological basis of this concept is a central model, describing the whole project the engineers are working on. The central model is shared by all domains involved in the engineering process. Based on this model we can identify the current project phase, the data and artefacts necessary for working in the current phase as well as information about the environment the engineers are working in.

This includes time, location and devices which can be used by the engineer. Other project related knowledge and data needed for the design process is available through an interface

to a PLM-system. We will have a look at engineering design grammars like the design graph as a possible implementation for our model [AR03]. Using a graph database like Neo4j, we are able to store the model structure in a performant and persistent way. Using the transaction management of such a database, synchronization and merging problems can be solved. This is an ongoing research project with some open questions that needs to be answered in future research. We will examine on the methods used in software engineering as well as in database systems and how we can adapt them to our system to define a suitable architecture for our system.

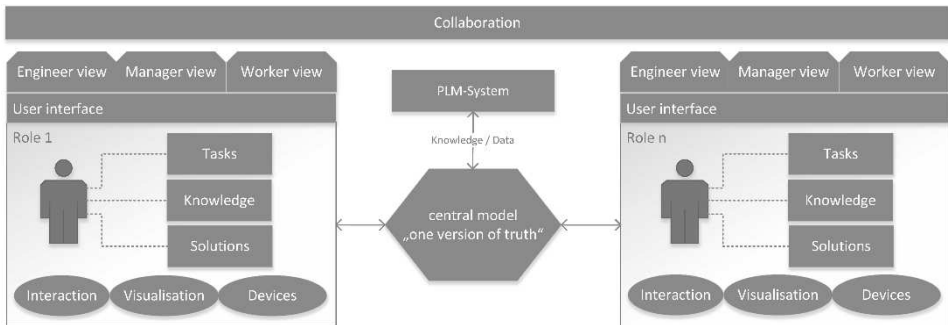


Fig. 1: Basic concept of our adaptive CSCW-Environment for engineering collaboration

To support the different views each project member can have regarding the different roles of the other members in the project, the system needs to be adaptive. There will be different forms of adaption. First technical adaption in the form that the user interface must change appropriately depending on the device the engineer uses and second the system needs to adapt its support capabilities on the needs based on the current project phase or the demands of the engineers. We will have to investigate how the different views can affect each other in such a scenario. In the future we will support different CSCW use cases like distributed collaborative design, virtual team rooms that will be defined later on based on the requirements gathered from further interviews. For now, we will focus on the support of communication in early brainstorming phases in the product lifecycle using a modular online collaboration platform as a basis for further work.

### 3.5 Example Scenario

The following example scenario will help to understand our concept and the problem space: A co-located team of engineers is working on a structural element for a car. That element is used in every car of the company, so the product engineering team in Germany decided to use a model-based engineering process which allows them to model the part once and generate different variants of that part according to the specification for a number of future cars. To do that they need to model that part in their model-based design tool. To ensure the manufacturability of the part, a production planner sitting in the production site in China is involved in the design process. Because it is a visible part of the body of the

car, some Italian car exterior designers are also involved in the design process. In a monthly status meeting the project management is informed about the current project state. Let's have a look at possible visualizations and interactions with the model for the roles product engineer, exterior designer and project management.

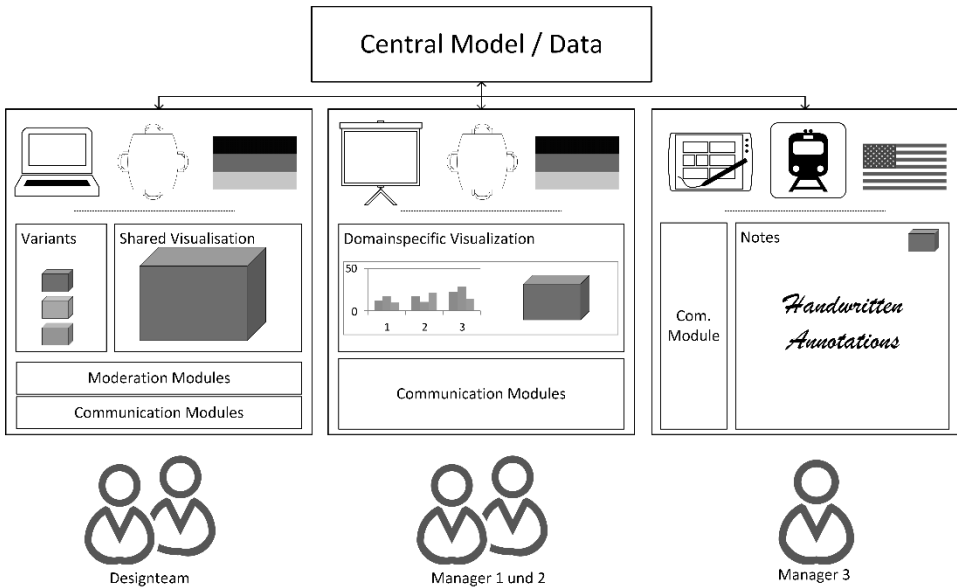


Fig. 2: Illustration of the adaption of the user interface for different roles and environments

The methods and tools these different roles are used to work with are likely to be different. Product engineers may work with mind maps and hand drawn technical sketches to communicate the technical or structural properties of the product, while the product designers are used to work with tools like mood boards to visualize the product styling. The managers are used to look at different kinds of diagrams to visualize the projects key figures needed for managing the project. Not only the methods or tools could be different. Also the devices used to interact with the project data or the workspace needs to be appropriate to the user and his needs.

There is also a need for collaboration between the different roles in a project. The designers could come up with designs that suits their requirements but are impossible to produce or are simply too expensive for the designated use-case and thereby could violate the requirements the managers are working with. So there have to be some kind of communication between these roles. With the concept of one model of truth all information needed to make collaborative decisions are in one place, ready to be visualized to all of the roles helping them to make a decision.

When we look at a web-based conference tool for example, supporting communication between the different roles, we could augment a presentation with visualizations customized for the different roles for example. Let's say that the designers initiated an online design review. They moderate the review in the online conference tool and provide 3D mockups of the different designs they came up with. This view is then shared to all of the participants. In addition to this view, the manager for example can see the price of the materials used for the design as well as the estimated quantity the factory would be able to produce on a daily basis. All the numbers are calculated in the background based on the design variant the designers have currently selected on the 3D Model viewer.

## 4 Conclusion and Further Work

We introduced the idea of a collaborative working environment to support model-based systems engineering. The next steps developing such a working environment are the following: We need to work on a catalog of support tools (for example creativity techniques) including a mapping to different phases in the engineering workflow. To do this we need deeper insight into the working practices of engineers. A model of the engineering process needs to be designed. This model will include project data allowing a reasoning engine to detect the current project state. A first prototype of the system supporting communication of distributed teams is developed based on modular software engineering methods. This prototype forms the basis of all following prototypes. Therefore modular expandability is a major requirement in the development process.

## References

- [Al84] Alshuller, G. S.: Creativity as an exact science: The theory of the solution of inventive problems, volume 5 of Studies in cybernetics. Gordon and Breach, New York, NY, 1984.
- [AR03] Alber, R.; Rudolph, S., eds. 43 - A Generic Approach for Engineering Design Grammars, Proc. of the AAAI Spring Symposium 'Computational Synthesis', Stanford, CA, 2003.
- [AZ13] Albers, A.; Zingel, C.: Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML. In: Proceedings of the 23rd CIRP Design Conference, Bochum, Germany, March 11th - 13th, 2013. pp. 83–92, 2013.
- [BR11] Bordegoni, M.; Rizzi, C.: Innovation in Product Design. Springer London, London, 2011.
- [Co99] Cooper, A.: The Inmates Are Running the Asylum. Macmillan Publishing Co., Inc. Indianapolis, IN, USA. 1999.

- [Cu09] Cugini, U.; Cascini, G.; Muzzupappa, M.; Nigrelli, V.: Integrated Computer-Aided Innovation: The PROSIT approach. *Computers in Industry*, 60(8):629–641, 2009.
- [ERZ14] Eigner, M.; Roubanov, D.; Zafirov, R.: *Modellbasierte virtuelle Produktentwicklung*. Springer Vieweg, Berlin, 2014.
- [ES01] Eigner, M.; Stelzer, R.: *Produktdatenmanagement-Systeme: Ein Leitfaden für Product Development und Life Cycle Management*. Springer, Berlin, 2001.
- [GBR12] Gallardo, J.; Bravo, C.; Redondo, M. A.: A model-driven development method for collaborative modeling tools. *Journal of Network and Computer Applications*, 35(3):1086–1105, 2012.
- [Gr88] Grudin, J.: Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. In: *Proceedings of the 1988 ACM Conference on Computer-supported Cooperative Work. CSCW '88*, ACM, New York, NY, USA, pp. 85–93, 1988.
- [HK05] Herstatt, C.; Kalogerakis, K.: How to use analogies for breakthrough innovations. *International Journal of Innovation and Technology Management*, 02(03):331 - 347, 2005.
- [HM12] Hanington, B.; Martin, B.: *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers, 2012.
- [In07] International Council on Systems Engineering: *SYSTEMS ENGINEERING VISION 2020*, 2007. [ISO10] ISO 9241-210:2010: *Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems*. 2010.
- [Ki15] Kim, E.; Hwang, J.; Hahm, G.-J.; Lee, J. H.: 3D CAD model visualization on a website using the X3D standard. *Computers in Industry*, 70:116–126, 2015.
- [Ky91] Kyng, M.: Designing for cooperation: cooperating in design. *Communications of the ACM*, 34(12):65–73, 1991.
- [LF09] Lemburg, J. P.; Feldhusen, J.: *Methodik der schrittweisen Gestaltsynthese: Lehrstuhl und Institut für Allgemeine Konstruktionstechnik des Maschinenbaus*. PhD thesis, Shaker and Zugl.: Aachen, Techn. Hochsch., Diss., 2009, 2009.
- [Ro69] Rohrbach, B.: *Kreativ nach Regeln—Methode 635, eine neue Technik zum Lösen von Problemen*. *Absatzwirtschaft*, 12(19):73–75, 1969.
- [UT13] Ubik, S.; Trávníček, Z.: Distributed Collaboration in Engineering by Low-Latency 3D Model Sharing. In: *Cooperative Design, Visualization, and Engineering*, volume 8091 of *Lecture Notes in Computer Science*, pp. 43–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.