INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

SUPPORTING THE MODELING OF TRACEABILITY INFORMATION

N. Koehler, T. Naumann and S. Vajna

Keywords: product development, sociotechnical systems, traceability, complexity, systems engineering, engineering objects

1. Introduction

Over the last decade, significant changes turned the automotive industry into a highly competitive engineering environment. Especially trends related to globalization, climate change and economic crises have contributed to the manufacturers' competition. As a result, automotive companies have been forced to launch new cooperations in order to reduce engineering costs. To keep the development process effective and efficient, firms have focused on front loading to enhance time-to-market. Product lifecycle management technologies have been constantly improved to increase product quality in early phases. Furthermore, individualization has led to a strong increase in vehicle variants. The automotive companies have tried to handle the increased complexity by communality and modularization. Accordingly, the technical and process-related complexity in the automotive industry's product development has risen significantly within the past decade. The existing tools and methods of product development, however, are not able to deal with this dimension of complex dependencies in vehicle development [Naumann et al. 2011].

Our *systems approach* is destined to overcome the boundaries of today's tools and methods in order to meet the needs of prospective engineering in the automotive industry. The objective of this paper is to demonstrate the need of a well-defined traceability approach in product development. Our broad approach of social systems engineering merges social and technical systems. The social systems engineering concept was refined and eventually resulted in a meta-model of sociotechnical systems. This meta-model describes the interaction of social systems with technical systems as technical genesis [Naumann et al. 2011]. With respect to the sociotechnical meta-model, a framework of product development supporting the effective and efficient development of complex systems was established. This framework was implemented to a software prototype which supports the modeling of EORs between different EOs in product development. The goal was to establish as much transparency as possible regarding the raising complexity [Königs et al. 2012].

However, the general challenge related to *traceability* to make implicit relations explicit is still present [Königs et al. 2012]. In particular, the high effort to manually model EORs is the main obstacle for using traceability tools in practice [Aizenbud-Reshef et al. 2006]. To date, none of the existing commercialized product lifecycle management approaches support the modeling EORs comprehensively [Storga et al. 2011a]. Further research is needed to enhance creation and maintenance of traceability information [Storga 2004]. Future traceability tools have to support engineers with automatic methods to model traceability information. Accordingly, the research question of this paper deals with the ability of today's product development approaches to model full traceability. Regarding this, the paper contains an in-depth analysis indicating that manual modeling of traceability is not an appropriate approach for complex technical systems. Based on the analysis, implications for new approaches supporting the modeling of traceability are discussed.

2. Traceability

Over the past years, the automotive industry has struggled to handle the increased complexity of its mechatronic products. Trends like a rising number of functionalities, modularization and reductions of hardware prototypes have led to a significant increase of engineering objects and their engineering object relations [Gausemeier et al. 2009], [Vajna et al. 2009], [Königs et al. 2012]. Engineering objects (EOs) are defined as all artifacts arising from product development represented by information frag- ments and its graphical representations and defined by an integrated product and process model, e.g. requirements, functions, and elements [Faisst and Dankwort 2007], [Bitzer et al. 2007]. Accordingly, engineering object relations (EORs) are all connections between two EOs with the objective to describe the semantic relations among all EOs, such as aggregation, composition and definition [Zimmermann et al. 2002], [Königs et al. 2012]. Companies tried to overcome these challenges due to the rising number of EOs and EORs with approaches like reusability, adaptability and partial design solutions. However, insufficient design documentation causes fundamental problems particularly in later phases of the product development process. These problems are intensified by the variety of software tools currently used in the automotive industry, which make an integrative product development process even more complicated [Storga 2004]. Therefore, analyzing the impacts of changes of product characteristics across different departments is hardly possible. This results in intransparency, inconsistency and redundancy of EOs. An approach addressing the above mentioned problems from the 1970s focusing on the EORs is traceability [Königs et al. 2012]. The approach is originally defined by the IEEE Standard Glossary of Software Engineering Terminology:

> "The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor - successor or master-subordinate relationship to one another" [IEEE Std 610.12 1990].

The given *definition* focuses on software engineering, however, the concept of traceability is transferable to any system development [Buur and Andreasen 1989], [Ramesh and Jarke 2001]. Generally, traceability represents "a quality factor of designing - a property that product development environment should possess" [Storga 2004]. Transferred from requirements engineering to the development of mechatronic products, traceability comprises the ability to follow every EO with the help of EORs from its origin to its use in the final product [Gotel and Finkelstein 1994]. In addition, traceability of EOs could play an important role concerning knowledge and information management [Ouertani et al. 2011]. The goal is to enable the understanding of semantic relationships within and across different engineering contexts [Storga et al. 2011b]. Due to its complex mechatronic systems the product development of the automotive industry is especially made for an integrative traceability approach [Königs et al. 2012]. However, research about traceability in product development is still immature due to a lack of common understanding [Ramesh and Jarke 2001]. Recently, researchers of the product development domain have emphasized the need for further research and improvements. Especially, more research supporting the modeling of traceability information is needed for the application of the concept in industry practice [Königs et al. 2012].

The utilization of traceability offers various *advantages* for product development. First, traceability makes it possible to understand existing information in its context by the help of semantic EORs. Thus, traceability supports reusing of design knowledge during the product development. Second, if any change occurs during the product development process, traceability facilitates the identification and verification of its impacts [Storga et al. 2011b]. Traceability is also a suitable approach for the efficient propagation of changes enabled by EORs [Königs et al. 2012]. Third, traceability enhances the credibility of engineering information by documenting the history of all EOs and their EORs [Storga et al. 2009]. Finally, traceability supports the product development process ensuring that EORs are "complete, correct, consistent and error free" [Storga 2004].

Despite the numerous advantages, many *difficulties* connected to applying the traceability approach have been identified. This is illustrated by Storga's [2004] question that arose a decade ago and is still up-to-date: "Why is the achievement of engineering information traceability in modern highly-automated product development environments, still so difficult?" [Storga et al. 2011b]. There are several reasons why traceability is not yet widely implemented in today's product development environments. On the one hand, human factors as well as established development processes play an important role. On the other hand, the heterogeneous tools and methods of the automotive industry complicate the introduction of traceability approaches [Königs et al. 2012]. Some tools already support traceability in product lifecycle environments. However, this support is mostly limited to specific partial models. Moreover, present product data management systems lack management of semantic EORs between EOs. A basic reason for this limitation of product data management systems is the focus on hierarchical structures particularly for later phases of the product lifecycle [Königs et al. 2012]. In addition to the mentioned limitations, Storga [2004] has identified four main difficulties concerning the application of traceability in product development:

- Diverse traceability needs in development: needs differ in organizations, projects and users
- Complex and extensive knowledge base: wide range of content and multidimensional relations
- Formal and informal sources: informal information is easy to capture but difficult to process
- Interference with tools: besides main functionalities, most tools support traceability partially

As a *result*, there is no existing approach compensating for the above mentioned obstacles while at the same time achieving full traceability [Storga et al. 2011b]. A lack of knowledge about EOs and their EORs then leads to an increased risk of future product defects. Making use of the advantages and overcoming the obstacles, researchers and tool suppliers developed new approaches, including knowledge integration [Hicks et al. 2002], [Mohan and Ramesh 2007], [Ouertani et al. 2011], requirements management [Ramesh and Jarke 2001], [Sutinen et al. 2002], [Winkler and Von Pilgrim 2010], design structure matrices [Browning 2001], [Danilovic and Browning 2007], [Lindemann et al. 2009] and systems engineering [Stark and Figge 2011], [Königs et al. 2012]. Based on these approaches, implementations of traceability in software prototypes [Diehl et al. 2008], [Wynn and Clarkson 2008], [Stark and Figge 2011], [Königs et al. 2012] have been developed. Besides the prototypical implementations, software producers developed commercial tools using partial traceability, e.g. requirements management tools such as RationalDOORS or Reqtify and tools for systems engineering such as Loomeo or METUS. Product lifecycle management systems integrated traceability approaches as well, e.g. Teamcenter by Siemens and Catia V6 by Dassault System [Königs et al. 2012]. To summarize, there is a wide range of approaches dealing with traceability, however, most approaches focus on partial models of traceability. The main problem of the described partial approaches is the missing integration of fragmented process and product information on common platforms [Storga et al. 2011b].

Our *focus of research* is on the development of an integrated product and process approach supporting the modeling of traceability in order to handle today's rising complexity. The overall objective of our approach is to draw more attention to traceability in a multidisciplinary product development environment. To make implicit relations between EOs explicit and thus enhance transparency is the general challenge. Not all implicit relations can be modeled efficiently, though. Further research is hence needed to focus on detecting beneficial EORs [Königs et al. 2012]. Especially, the absence of automatic techniques supporting the modeling of EOs and their EORs leads to redundancies, inconsistencies and limited reuse of EOs. Therefore, our research points to one of the major drawbacks of traceability: the effort of labour intensive modeling of traceability information [Aizenbud-Reshef et al. 2006], [Pavkovic et al. 2011], [Stark and Figge 2011], [Storga et al. 2011a,b], [Königs et al. 2012]. This is emphasized by Aizenbud-Reshef's et al. statement that "a well-defined, automated method of accomplishing traceability would be of value in any domain, on any project, with any methodology" [Aizenbud-Reshef et al. 2006]. Thus, a well-defined traceability approach allows for gaining a competitive advantage in the automotive industry. Our research question for future projects about traceability in product development at Daimler is the following:

> How can new product development approaches support engineers with the efficient handling of today's rising complexity regarding interdependent relations of engineering objects?

3. Sociotechnical systems

The *meta-model of sociotechnical systems* describes the development and utilization of technical systems by social systems. The objective of the meta-model is to strengthen the ability of sociotechnical systems' selforganization. In addition, the meta-model of sociotechnical systems provides the basis for developing new approaches to improve the interaction of a social system with a new technical system. The approach bases on two main concepts of Ropohl's [2009] and Willke's [1991] system theory. First, the formal structure of the meta-model applies the three concepts of Ropohl's [1999] system theory: the functional, the structural and the hierarchical concept. Secondly, the meta-model of sociotechnical systems regards as well Willke's [1991] functional genetic system theory of social systems. Based on Ropohl's [2009] and Willke's [1991] understanding of system theory, the meta-model of sociotechnical systems forms a sociotechnical system consisting of more than one social system, and one technical system. The essential part of the meta-model of sociotechnical systems is the coupling of social and technical systems via the function of technical genesis [Ropohl 2009].

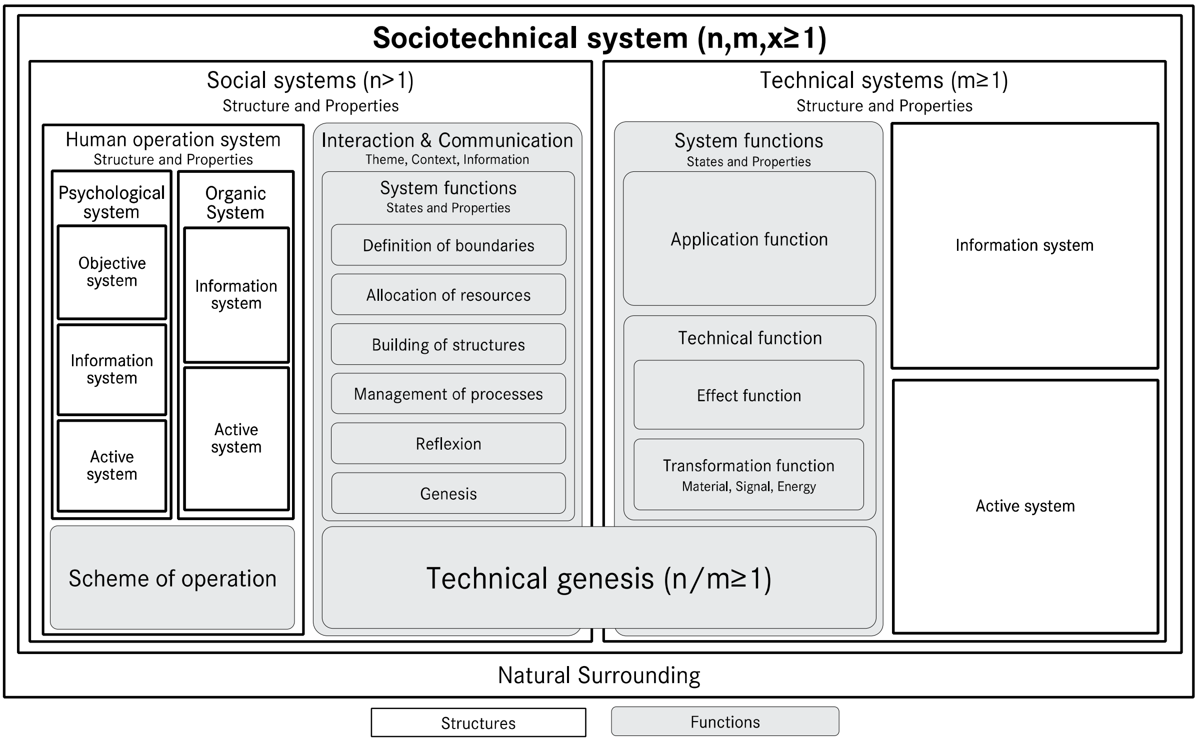

Figure 1. Meta-model of sociotechnical systems [Naumann et al. 2011]

The *social system* considered as a system is represented by its structures of subsystems and its functions. The human operation system as a part of the social system consists of a psychological and an organic system. Both systems are characterized as an operation system according to Ropohl [2009]. The psychological system then comprises an objective, an information and an active system. Contrary to the psychological system, the organic system is only composed of an information and an active system. Therefore, the psychological system is the only system defining objectives for the human operation system. These objectives are the basis for the functional scheme of operation represented and simplified as TOTE scheme [Ehrlenspiel 2007], [Pahl et al. 2007], [Ropohl 2009]. In order to achieve the objectives a social system arises, which is composed of at least two individuals. To ensure the cooperation of the human operating systems, interaction and communication are the essential functions. This provides the basis for the social system's functions: definition of boundaries, allocation of resources, building of structures, management of processes, reflection and genesis [Willke 1991]. To summarize, the social system is represented by its psychological and organic structure and tries to achieve the objectives of the scheme of operation by interaction and communication.
The *technical system* also consists of structures and functions. Compared to the social system, the structures and functions of a technical system are individually shaped. The structure of the technical system is described by Ropohl's [2009] operation system. However, the structure contains only an information and an active system. To date, there is no technical system that is able to process

information and perform actions with respect to an inherent objective. The system functions of a technical system are classified into two different subfunctions: an application and a technical function. The application function represents the connection between a technical and a social system. The technical function enables the technical system to execute its required behavior. Further, the technical function is classified into the effect function describing a change of a system's state and the transformation function relating to a transformation of material, signal and energy. Concluding, the described structures and functions contribute to the general objective of a technical system to take over human action by performing physical work or processing information [Hubka and Eder 1988].
The *technical genesis* represents the application of a technical system by a social system. According to Ropohl [2009], this application is a coupling of technical and social systems comprising the utilization and development of technical systems. The coupling of the two systems follows the above described social system's scheme of operation. This means that during the technical genesis the functions of the scheme of operation are applied. Focusing on the development of technical systems the technical genesis contains the steps of setting objectives, planning and realization. The scheme of operation is passed iteratively and completed by achieving the objectives. As the technical genesis couples the technical and the social system the equivalence principle is applied, meaning that the social systems' complexity and the complexity of the developed product are equivalent. As a result, the social system has to adapt its structures and functions in order to cope with those of the technical system. This adaption of the social system is achieved by self-organization and self-governance. To efficiently develop technical systems, the social system has to adapt to environmental influences as efficiently as possible applying these two principles [Naumann et al. 2011].

4. Framework of product development

The *framework of product development* emerged from the meta-model of sociotechnical systems. Especially, the rising complexity of the technical genesis, that is, the interaction between a social and a technical system, led to the development of the framework. It represents a systematic approach to handle all EOs with their EORs that arise in the product development. The framework has a product model representing the technical system as well as a process model representing interaction and communication as activities of the social system.

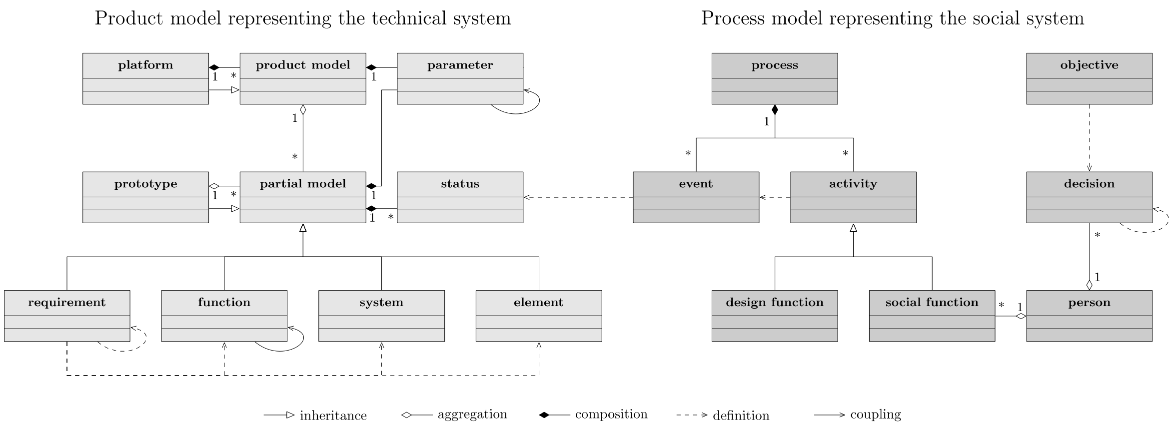

Figure 2. Framework of product development

As a result, the framework of product development represents a comprehensive integrated product and process model. The product model is characterized by different partial models, e.g. requirements, functions and elements. The process model focuses on the communication of social systems to describe the processes with e.g. objectives, decisions, activities and events. The framework of product development is developed with the help of the Unified Modeling Language in order to implement the framework in a software prototype. Overall, the framework aims at supporting the effective and efficient modeling of technical systems by defining a distinct syntax and semantic for EOs and their EORs [Naumann et al. 2011].

Figure 2 shows a *simplified extract of the framework*. To describe the semantics between the EOs, five types of EORs are used: inheritance, aggregation, composition, definition and coupling. As mentioned above, the framework is partitioned into the product model and the process model. The product model, representing the technical system, is formed by its partial models: the requirements model, the functional model, the systems model and the elements model. Besides the partial model, parameters, platforms and prototypes are relevant EOs for a comprehensive product model. The process model, representing the social system, is defined by a design and a social function. The design function is characterized by the part design, the validation and the documentation, among others. Engineers who make decisions aligned to organizational objectives constitute the social function. The social and design functions cause activities and events. In contrast to many other process models, our framework defines a process as an aggregation of activities and events in the social system. Both, the product and the process model are connected by the status of the partial model. In summary, the described framework defines the relevant EOs and their EORs for its implementation in a software prototype.

5. Prototypical implementation

Based on the framework of product development, a *prototypical implementation* was developed by Königs et al. [2012] at Daimler's research department. The prototype is a systems modeling and management tool (SysMT) applying model-based systems engineering. The objective of SysMT was to develop a system template approach reducing the effort for the creation of system models. Due to its high extent of mechatronic systems' reuse, the automotive industry is suitable for such a system template approach. In particular, a multitude of mechatronic systems are used for different vehicle generations with only minimal adaptations. Besides the system template approach, SysMT enables traceability of EOs. To specify, with its graph based representation, SysMT has the ability to model systems as lattice structures. The framework of product development forms the data model of SysMT and, therefore, describes all possible EOs and EORs [Königs et al. 2012].

Table 1. Qualitative analysis of traceability approaches [Königs et al. 2012]

	management				*concept*				*application*			*properties*			
	central storage	acquisition of objects	manipulation of objects	original objects	traceability framework	lattice data structures	parameter granularity	modeling support	graphical representation	traceability purpose	change handling	quantitative relations	typed semantic	across processes	across contexts
Rational DOORS by IBM	●	○	●	●	○	○	○	◑	●	◑	◑	○	○	○	●
Reqtify by Dassault Systems	○	●	○	●	●	○	○	○	●	○	◑	○	●	●	●
METUS by ID-Consult	●	○	○	○	●	○	○	○	●	○	○	●	○	●	●
Loomeo by TESEON	●	○	○	○	○	○	○	○	●	○	◑	●	○	●	●
CAM by Cambridge EDC	●	○	○	○	○	●	○	○	●	◑	◑	●	●	○	○
Teamcenter by Siemens	●	●	●	●	○	●	●	◑	◑	●	●	○	○	●	●
Catia V6 by Dassault Systems	●	●	●	●	●	●	●	◑	◑	●	●	○	●	●	●
ToolNet by Daimler	●	●	○	●	○	●	○	○	○	○	◑	○	○	●	●
Atego Workbench by Atego	●	●	●	●	○	●	○	◑	◑	○	◑	○	●	●	●
SysMT by Daimler	●	●	●	○	●	●	●	◑	●	●	●	●	●	●	●

● full implementation, ◑ partial implementation, ○ no implementation

SysMT's detailed *functionalities* are presented by means of Königs' et al. [2012] proposed traceability criteria. These standardized criteria were developed to enable the analysis of existing traceability methods and tools on the same basis. Königs et al. [2012] used their criteria to conduct an analysis of existing traceability approaches. Table 1 shows a summary of the results extended by the performance

of SysMT. The traceability criteria are classified into four groups: data management, traceability concept, application of the traceability concept and properties of the EORs. Concerning data management, SysMT has a client-server architecture with a central storage of data. SysMT also represents an integrative tool that allows the acquisition and manipulation of EOs based on copied data. Regarding the traceability concept, SysMT has a comprehensive traceability framework, supports lattice data structures and uses parameters as finest granularity. Moreover, SysMT offers partial support of traceability modeling by reusing mechatronic systems in different contexts. With respect to the application of traceability, SysMT uses a graph based SysML visualization. In addition, the prototype has a wide range of traceability purposes in product development including verification, analysis, synthesis, documentation, process monitoring and synchronization. SysMT is able to handle changes by notification and propagation of changes. Considering the properties of the EORs, SysMT supports qualitative as well as quantitative EORs in order to propagate parameters. Further, SysMT's EORs have a typed semantic based on the framework of product development. SysMT's EOs are used across processes and contexts. As shown in table 1, SysMT satisfies nearly all trace ability criteria to the full extent and, therefore, represents an appropriate traceability approach.
The *evaluation* of SysMT was performed by modeling various mechatronic systems in early phases of Daimler's product development process. Within the last two years, the prototype has been used for at least three substantial projects by approximately 15 users. Figure 3 displays the EOs and their EORs created in the first 18 months of SysMT's application. In this period, over 65,000 objects have been modeled including about 20,000 EOs and 45,000 EORs. The number of EOs can be subdivided into the partial models, containing about 4,000 requirements, 140 functions, 219 systems, 9,169 elements and 2,400 parameters. An analysis of SysMT with regard to the implementation of traceability is shown in Figure 3.

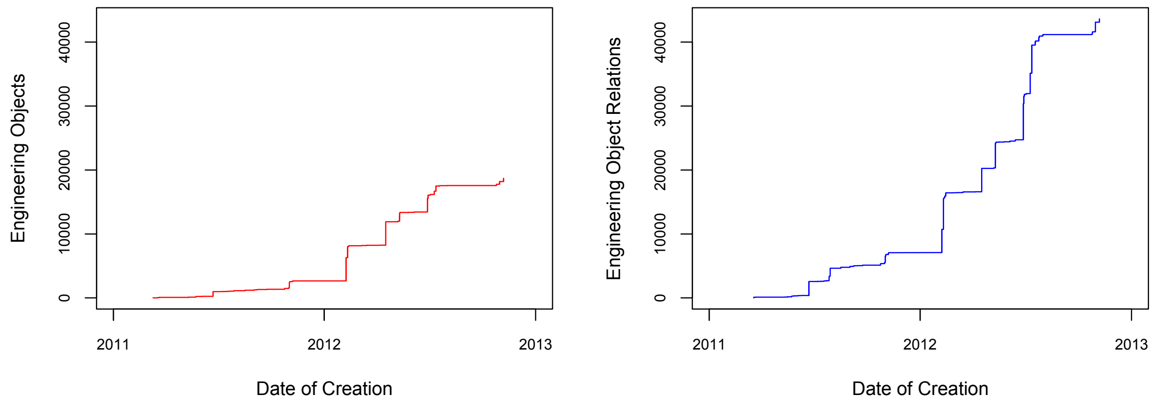

Figure 3. Quantitative analysis related to the number of EOs and EORs

The *modeling* of the first mechatronic systems started in the second quarter of 2011. In the middle of that year, the first project on the technical validation of design models was initiated. SysMT acquired relevant requirements for the project by its interface with a requirement management tool. This is shown by the first notable increase in the number of EOs. In parallel, a mechatronic rear view system was modeled by the help of SysMT's systems engineering approach, which is reflected by a growth in EOs and EORs [Königs et al. 2012]. At the end of 2011, a Daimler specific product structure was partly imported to embed the modeled systems. However, a neutral product structure was needed in order to link elements to requirements, functions and systems on a common basis. Thus, at the beginning of 2012, it was imported, which is shown by a jump in the number of EOs and EORs. Additionally, the parameters of the Global Cars Manufacturers Information Exchange were integrated in order to link these parameters to the requirements of the design validation model. The following jumps in the amount of EOs and their EORs represent the application of Königs' et al. [2012] template approach. At the beginning of the second quarter of 2012, the neutral product structure as well as the requirements of the design validation project were instantiated to other car projects, which can be seen by further increases in EOs and EORs. Besides this, the development of a side window guidance system was modeled in SysMT, displayed by a smaller increase of EORs in the middle of 2012. The final strong rise in EORs can be explained by the instantiation of the product structure including all modeled systems.

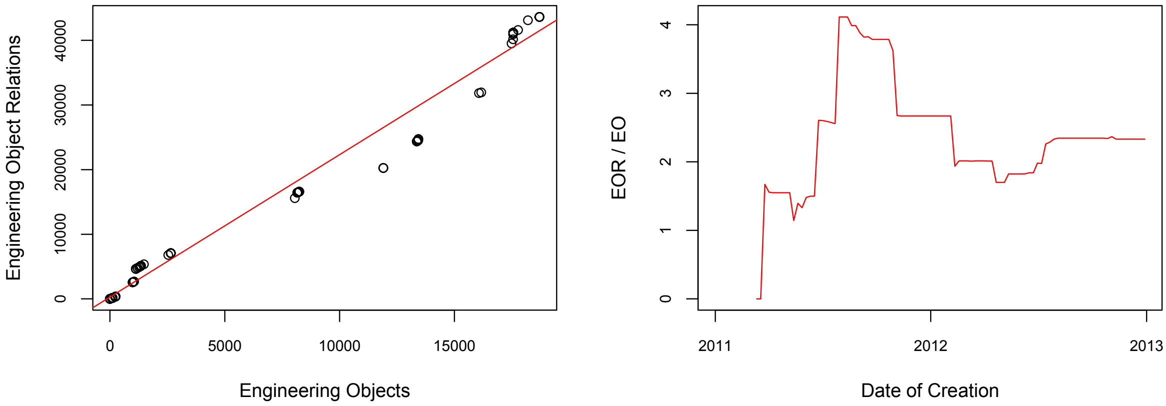

Figure 4. Quantitative analysis related to the ratio of EORs and EOs

The *analysis* reviewed the projects that were modeled with the prototype. Regarding SysMT's ability to model traceability information, Figure 3 gives a first impression. The number of EORs rose nearly twice as fast as the number of EOs, meaning that every EO has approximately two EORs. Figure 4 presents an in-depth analysis of the correlation between EOs and EORs on a weekly basis. Figure 4 shows the number of EOs and EORs at a specific point of time. The linear regression confirms the relation between EOs and EORs with a correlation coefficient of r = 0.98:

$$EOR(EO) = 294 + 2.2 \cdot EO \tag{1}$$

Thus, every EO has about 2.2 EORs in the sample of 65,000 objects. Considering that every EO has at least one hierarchical EOR, every EO has only about one additional semantic relationship. However, a network system is not characterized by a linear relationship between EOs and EORs. Assuming that EORs are undirected, the relationship between EOs and EORs should be of an polynomial nature:

$$EOR(EO) = EO \cdot \frac{(EO - 1)}{2} \tag{2}$$

Besides this quantitative finding, Figure 4 illustrates the development of the quotient of EORs and EOs over time. The higher the ratio, which is achieved by an increasing number of EORs, the more suitable it is for modeling the multitude of interdependencies of mechatronic systems. At the beginning of the evaluation, the ratio equaled 1.5, which is comparatively low. With respect to the modeling of the rear view system, the quotient rose significantly to 4.0. This indicates that the lattice of EOs in manually modeled mechatronic systems is very dense. However, the ratio declined when requirements and parameters were acquired at the end of 2011. In the middle of 2012, the quotient increased again with rising EORs due to the modeling of the window guidance system. This emphasizes the strength of SysMT's ability to manually model mechatronic systems. Thus, SysMT is an appropriate approach for the modeling of traceability information in small to medium-sized mechatronic systems. However, for modeling an entire vehicle consisting of numerous systems an overall average of 2.2 EORs per EO is too low to ensure full traceability. Furthermore, the modeling of the three systems took a tremendous amount of time. To summarize, the quantitative analysis verified that a well-defined traceability approach with automatic methods is essential for the modeling of complex mechatronic systems.

6. Conclusion

The aim of our paper was to evaluate the ability of today's product development approaches to model full traceability. Our *findings* related to the analysis of the prototype confirm the need for further research on the modeling of traceability information. The development of the prototype on the basis of the framework of product development was a first step in this direction. Comparing SysMT with other traceability approaches shows that it outperforms competing approaches in terms of an effective and efficient manual modeling. However, the quantitative analysis of the mechatronic systems modeled in SysMT points to several difficulties concerning the manual modeling of complex systems such as an entire vehicle. Thus, new approaches focusing on automatic methods are essential to provide a sufficient amount of EORs to apply the traceability approach. In total, the analysis forms a solid basis for further research on improvements of existing traceability approaches.

The present paper's main *limitation* is its analytical character. The paper merely analyzes SysMT's application and states that new approaches are necessary to achieve traceability. However, it does not present new methods to overcome these obstacles. Besides this, the quantitative analysis of the prototype considers primarily the product model and its partial models. The meta-model of sociotechnical systems, however, highlights the social system of the product development. The main reason for this limitation is the partial implementation in SysMT. Especially, some parts of the process model, such as objectives and decisions, representing the social system were not considered during the development. To summarize, the exact findings of this paper are limited to the conditions of the prototype's implementation. Nevertheless, the analysis identifies general challenges regarding the application of traceability that can be generalized to industrial practice.

The above discussed limitations give rise to *future research* opportunities. As shown, the major drawback for the application of traceability in industry practice is the effort for the manual modeling of traceability information. This implies that future research should focus on new approaches supporting the effective and efficient modeling of EOs and their EORs. To support the creation and maintenance of traceability information, new automatic methods are needed to ensure the utilization of EOs and their EORs. In addition, future research might focus on finding the beneficial EORs due to the fact that not all EORs can be modeled efficiently. Semantic web technologies present a possible research area supporting the finding of EORs between EOs. Particularly, the application of ontologies, rule based approaches and taxonomies might be valuable. To conclude, future research could contribute to a beneficial application of traceability in a complex and competitive industry environment.

References

Aizenbud-Reshef, N., Nolan, B., Rubin, J., Shaham-Gafni, Y., "Model Traceability", IBM Systems Journal, Vol. 45, No. 3, 2006, pp. 515-526.

Bitzer, M., Eigner, M., Langlotz, M., "Management Decision Support by Plm Solutions", Proceedings of the 16th International Conference on Engineering Design, Paris, 2007.

Browning, T. R., "Applying the Design Structure Matrix to System Decomposition and Integration Problems - Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions", Transactions on Engineering Management, Vol. 48, No. 3, 2001, pp. 292-306.

Buur, J., Andreasen, M. M., "Design Models in Mechatronic Product Development", Journal of Design Studies, Vol. 10, No. 3, 1989, pp. 155-162.

Danilovic, M., Browning, T., "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices", Journal of Project Management, Vol. 25, No. 3, 2007, pp. 300-314.

Diehl, H., Hellenbrand, D., Lindemann, U., "Transparent 3d Visualization of Mechatronic Systemstructures", Procedings of the 10th International Design Conference, Dubrovnik, 2008.

Ehrlenspiel, K., "Integrierte Produktentwicklung", Carl Hanser, München, 2007.

Faisst, K. G., Dankwort, C. W., "New Extended Concept for the Usage of Engineering Objects and Product Properties in the Virtual Product Generation Process", Proceedings of the 16th International Conference on Engineering Design, Paris, 2007.

Gausemeier, J., Pöschl, M., Deyter, S., Kaiser, L., "Modeling and Analyzing Fault-tolerant Mechatronic Systems", Proceedings of the 17th International Conference on Engineering Design, Standford, 2009.

Gotel, O. C. Z., Finkelstein, A. C. W., "An Analysis of the Requirements Traceability Problem", Proceedings of the International Conference on Requirement Engineering, Colorado Springs, 1994, pp. 94-102.

Hicks, B. J., Culley, S. J., Allen, R. D., Mullineux, G., "A Framework for the Requirements of Capturing, Storing and Reusing Information and Knowledge in Engineering Design", International Journal of Information Management, Vol. 22, No. 4, 2002, pp. 263-280.

Hubka, V., Eder, W., "Theory of Technical Systems", Springer-Verlag, Berlin, 1988.

IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology", 1990.

Königs, S. F., "Konzeption und Realisierung einer Methode zur templategestützten Systementwicklung", PhD Thesis, Technische Universität Berlin, 2012.

Königs, S. F., Beier, G., Figge, A., Stark, R., "Traceability in Systems Engineering", Journal of Advanced Engineering Informatics, Vol. 26, No. 4, 2012, pp. 924-940.

Lindemann, U., Mauerer, M., Braun, T., "Structural Complexity Management", Springer-Verlag, Berlin, 2009.

Mohan, K., Ramesh, B., "Traceability-Based Knowledge Integration in Group Decision and Negotiation Activities", Journal of Decision Support Systems, Vol. 43, No. 3, 2007, pp. 968-989.

Naumann, T., Königs, S., Kallenborn, O., Tuttass, I., "Social Systems Engineering", Proceedings of the 18th International Conference on Engineering Design, Copenhagen, 2011, pp. 357-368.

Ouertani, M. Z., Baina, S., Gzara, L., Morel, G., "Traceability and Management of Dispersed Product Knowledge during Design and Manufacturing", Computer-Aided Design, Vol. 43, No. 5, 2011, pp. 546-562.

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., "Konstruktionslehre", Springer-Verlag, Berlin, 2007.

Pavkovic, N., Bojcetic, N., Franic, L., Marjanovic, D., "Case Studies to Explore Indexing Issues in Product Design Traceability Framwork", Proceedings of the 18th International Conference on Engineering Design, Copenhagen, 2011.

Ramesh, B., Jarke, M., "Toward Reference Models for Requirements Traceability", Jounral of Transactions on Software Engineering, Vol. 27, No. 1, 2001, pp. 58-93.

Ropohl, G., "Allgemeine Technologie: eine Systemtheorie der Technik", Universitätsverlag, Karlsruhe, 2009.

Ropohl, G., "Philosophy of Socio-Technical Systems", Society for Philosophy and Technology, Vol. 4, No. 3, 1999, pp. 59-71.

Stark, R., Figge, A., "Eco Tracing - A Systems Engineering Method for Efficient Tracelink Modelling", Proceedings of the 18th International Conference on Engineering Design, Copenhagen, 2011.

Storga, M., "Traceability in Product Development", Proceedings of the 8th International Design Conference, Dubrovnik, 2004.

Storga, M., Bojcetic, N., Pavkovic, N., Stankovic, T., "Traceability of Engineering Information Eevelopment in Plm Framework", Proceedings of the 8th International Conference on PLM, Eindhoven, 2011.

Storga, M., Darlington, M. J. C. S., Marjanovic, D., "Toward a Process and Method for Tracing the Development of Information Objects used in Engineering Design", Proceedings of the 17th International Conference on Engineering Design, Standford, 2009.

Storga, M., Marjanovic, D., Savsek, T., "Reference Model for Traceability Records Implementation in Engineering Dsign Environment", Proceedings of the 18th International Conference on Engineering Design, Copenhagen, 2011.

Sutinen, K., Almefelt, L., Malmqvist, J., "Supporting Concept Development Using Quantitative Requirements Traceability", Proceedings of the 12th Annual International Symposium INCOSE, Las Vegas, 2002.

Vajna, S., Weber, C., Bley, H., Zeman, K., Hehenberger, P., "CAx für Ingenieure", Springer-Verlag Berlin, 2009.

Willke, H., "Systemtheorie", Gustav Fischer Verlag, Stuttgart, 1991.

Winkler, S., Von Pilgrim, J., "A Survey of Traceability in Requirements Engineering and Model-Driven Development", Software and Systems Modeling, Vol. 9, No. 4, 2010, pp. 529–565.

Wynn, D. C., Clarkson, P. J., "Linkage Meta-Modelling to Support the Development of Design Process Improvement Tools", Proceedings of Tools and Methods of Competitive Engineering, Izmir, 2008.

Zimmermann, J. U., Haasis, S., Van Houten, F. J. A. M., "ULEO - Universal Linking of Engineering Objects", CIRP Annals- Manufacturing Technology, Vol. 51, No. 1, 2002, pp. 99-102.

Nico Koehler, Daimler AG, Digital Product Modeling
Otto-von-Guericke University Magdeburg
Wilhelm-Runge-Str. 11, 89081 Ulm, Germany
Telephone: +49 (0) 731 / 505 2832
Email: nico.koehler@daimler.com