# AN APPROACH TO INTEGRATE RISK MANAGEMENT IN CROSS-STRUCTURE SYSML-MODELS

**Kunnen, Steffen Georg; Adamenko, Dmytro; Pluhnau, Robin; Nagarajah, Arun**

University Duisburg-Essen

## ABSTRACT

Demands on developers are increasing due to the growing complexity of products in engineering. As many different disciplines are involved in planning the communication and data exchange becomes difficult. Systems engineering and especially the model-based development have proven themselves for this sector. However, the different languages for system modeling, such as SysML, offer considerable potential for optimization. A corresponding data model must be modelled so that data is available continuously and across all levels. Based on this data model, various engineering processes like risk management can be integrated into this model. New stereotypes are defined within SysML so that errors and risks can be implemented in the system model. This makes it possible to determine influences and effects that risks and errors have on other components of a product across all structures.

**Keywords**: Systems Engineering (SE), Model-based Systems Engineering (MBSE), Risk management, Ontologies, Product architecture

**Contact**:
Kunnen, Steffen Georg
University Duisburg-Essen
Institut for Product Engineering
Germany
steffen.kunnen@uni-due.de

**Cite this article:** Kunnen, S.G., Adamenko, D., Pluhnau, R., Nagarajah, A. (2019) 'An Approach to Integrate Risk Management in Cross-structure SysML-models', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.364



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

## 1 INTRODUCTION

In the design of complex products and plants, many different disciplines are involved. Communication takes place exclusively via written or telephone communication. This leads to small time delays in almost every process, which can quickly add up to a large delay. The flow of information must be uncomplicated, so that the developer always has access to all necessary information. The increasing complexity of interdisciplinary products and systems can, however, no longer be mastered with classical document-based SE methods. Model-centric approaches are, in contrast to document-centric methods, capable of formal definition and description of systems, which are necessary for enabling, inter alia, the systems thinking perspective and the automatic processing of interrelationships between system elements as well as the impact assessment of changes. Hence, Model-Based Systems Engineering (MBSE) approach gained growing attention specially during the last decade and is going to be one of the main pillars of the next industrial revolution named "Industry 4.0". (Hooshmand 2017, S. 101–110).

In general, the question arises how negative influences and connections can be identified at an early stage in order to eliminate them before a negative effect occurs. Risk management makes it possible to identify these critical points through parallel analyses and evaluations of components during the development phase. However, this leads to some problems, especially in plant construction. Risk management in plant engineering is usually done at component level and is therefore not Cross-system and cross-domain. Furthermore, communication between the different disciplines is very difficult. In addition, many modelling languages for systems do not offer the possibility of creating risks and errors, so that the impacts can be determined across structures. The goal is thus to answer the question of how the critical points of a project or product development can be determined based on a system model.

A continuous data model can accelerate and simplify communication between the disciplines. Processes like risk management can be performed at an early stage and identified risks can be communicated in real time with participating disciplines. This involves risks that can threaten the realisation process of the product and thus cause a time delay.

For the description of a system model a modelling language is necessary, which can represent not only the architecture but also other system-relevant elements. The Systems Modelling Language (SysML) was chosen for this application. SysML currently does not offer an opportunity to model risks and errors within a system model. For this reason, an adaptation to the modelling language must be done.

This publication presents a method to solve this problem using a continuous data model and a new stereotype to display risks and failures in SysML. For this purpose, the individual levels of a system architecture should be connected to each other. This method is suitable for several industries, but the validation is done using an example from power plant construction.

## 2 SYSTEMS ENGINEERING

Systems Engineering (SE) describes an interdisciplinary approach for the successful realisation of a system. It uses different approaches to ensure the functionality and safety of a product during the development phase, so that potential problems and malfunctions can be identified and corrected at an early stage. Systems Engineering has its origins in the aerospace industry, where early validation is necessary. The organization of processes, the increasing complexity of tasks and the increasing level of technology are just some further reasons for the development of systems engineering. (ISO/IEC/IEEE 15288 2015; Walden, Roedler, Forsberg 2017; INCOSE 2018)

In general, a system is the combination of technical and organizational units which are supposed to perform a specific task. Under certain boundary conditions, it should comply with the desired requirements. Figure 1 illustrates some disciplines that can be summarized under the generic term Systems Engineering. In addition to engineering processes such as risk management, this also includes points such as software and hardware.

Systems Engineering represents system design or system development but there are many different definitions of the term "systems engineering". Within the context of this paper the definition of Oliver Alt is used, who describes SE as the totality of all activities necessary for the development of a system (Alt 2012). The essential structure of the SE is limited to the blocks system architecture, system requirements and system behaviour (Alt 2012; Walden, Roedler, Forsberg 2017).



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

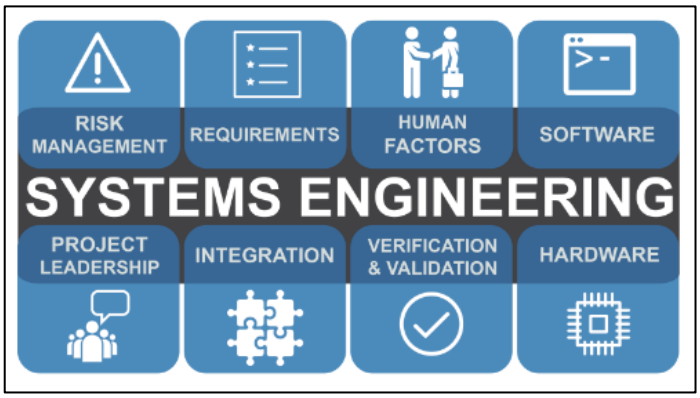


*Figure 1: Contents of Systems Engineering (INCOSE 2018)*

## 2.1 Model-based systems engineering (MBSE)

MBSE is a subordinate method of system engineering. It applicates visual modelling principles to systems engineering activities. The basic idea of MBSE is thus to formalize the system description as well as to connect the relevant information, needed for the creation of various artefacts during the system development, in a system model (Kaufmann, Schuler 2016, S. 343–352).
The objective of MBSE is to control the increasing complexity of products and processes, by improving communication between different disciplines and domains (SEBoK 2016). Risks and errors within the planning process can be better communicated and delays reduced. The trend in the industry is a transition from document-centred system development to model-based system development since one model can contain all the necessary information from multiple documents necessary for the development of a product (Kaufmann, Schuler 2016, S. 343–352). Figure 2 describes an MBSE-based data structure. From the system requirements a system model is developed, which is the centre of the data structure. Using specific services, software systems or methods can be connected to the model in order to integrate different engineering processes to the system design.

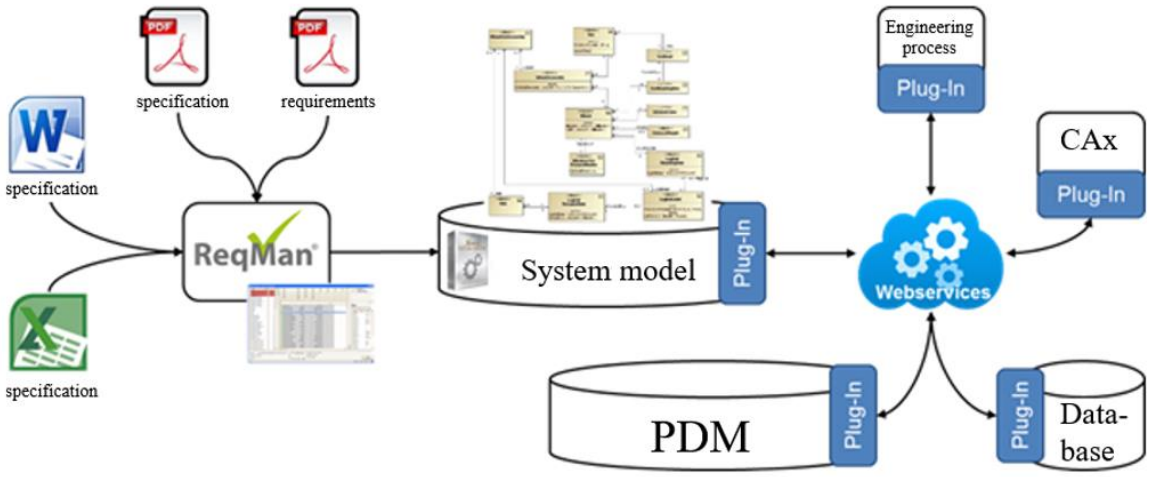


*Figure 2: Concept of an MBSE data structure with SysML (Hooshmand et. al. 2017)*

For MBSE, a uniform, cross-domain system modelling language is necessary to visualize the systems architecture (Friedenthal, Moore, Steiner 2015). The System Modelling Language (SysML) is a graphical language based on the Unified Modelling Language (UML) and was developed for modeling technical systems of all kinds. This is the difference to UML, which was developed especially for software development. SysML picks up a large amount of the UML, but leaves out the software-specific parts and replaces them with technical ones (Alt 2012). The SysML was first introduced in version 1.0 by the Object Management Group (OMG) in March 2007 and is regarded as the successor of UML (OMG SysML). Many UML functions have also been added to the SysML function library, with new concepts that have proven themselves in the course of system development. These were not considered in the UML and can be found as functional extensions in the SysML. SysML provides the ability to create a variety of chart types such as Activity diagrams, Requirement diagrams and Block



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

diagrams. The most common variants are block definition diagrams (BDD), which can be further specified by internal block diagrams (IBD). A distinction is always made between structure and behaviour diagrams. The Systems Modelling Language offers the possibility to assist in the development of complex systems. System requirements can be modelled to analyse and evaluate the system (Alt 2012) (Kaufmann, Schuler 2016, S. 343–352).
Although systembased approaches are now commonly used, many modeling languages do not offer the possibility to integrate risk or error models. For this reason, an extension of the SysML is necessary to create a stereotype for risk management with the existing diagram types. This topic will be addressed starting with chapter 4.

## 2.2 System development on different system levels

There are many different approaches to develop a system. One possibility is to follow the V-model shown in Figure 3, which was developed based on VDI Guideline 2221 (Eigner, Roubanov, Zafirov 2014). This was originally developed for the software industry, but later found great recognition in engineering (Informationstechnikzentrum Bund 2015). It basically consists of two sides; the descending side describes system development and the ascending side describes system integration. The V-model aims to establish a logical sequence of the essential sub-steps for developing a system. It is intended to ensure that the development is done properly. This approach enables the identification of critical subsystems and risks at an early stage (Informationstechnikzentrum Bund 2015). Within the development using the V-Model different levels such as Requirements, Functional, Logical and Physical (RFLP) must be designed. The left side of the V-model is required for an approach at RFLP level (Eigner, Roubanov, Zafirov 2014).

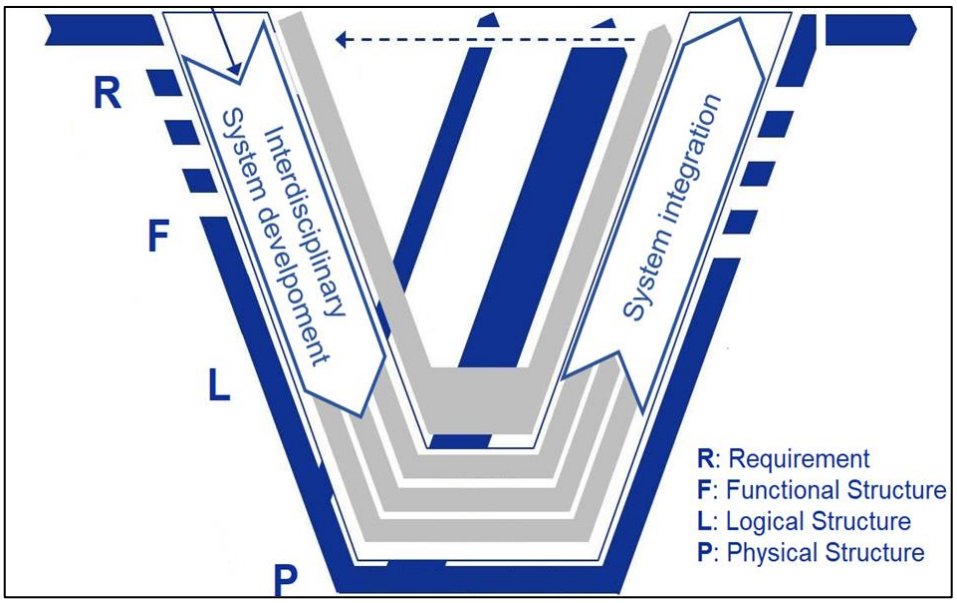


*Figure 3: V-Model to support multidisciplinary system development (adapted from (VDI-Richtlinie 2206); (Eigner et al., 2014)*

A system development/design usually starts with a requirement list. This results from a specific development order and is later considered to be a verification unit on which the product can be evaluated. The system design is used to define possible solution concepts and describes the physical or logical (PAS 1059 2006) structure. From the information in the requirements list, the system's overall function is determined in a further step, which describes the superordinate functions the system must fulfil. The main function consists of several sub-functions. A first cumulated collection of all the sub-functions contained in the system is called a functional structure. It is structured in a similar way to a tree diagram, whereby the main function can be further detailed on different sublevels. A split into sub-functions only makes sense until, in the course of time, a solution element has been assigned to each function. On that basis, the next step will be to connect the individual functions by the general flow parameters mass, energy and information flow, so that the correlation between input and output variables is made clear. These connected functions are called function net and describe the behaviour of the system. The previously mentioned allocation of solution elements illustrates the logical level of


https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

system development. Solution elements can be machines, plant components or in general any kind of components. In this case, a component is defined as a solution element that fulfils a function. It is important to note that a solution element can perform several functions at the same time. At the logical level, all the information is present that is theoretically necessary to meet the requirements of the system. The theoretical solution is put into concrete terms until there are in principle possible solutions for the product. A final level of system development is the physical level (Eigner, Roubanov, Zafirov 2014). The general opinion differs, whether physical models like CAD models or concrete assignment of components to the operating principles of the logical level represent the physical level. In this article, the logical level is regarded as a theoretical arrangement of components and the physical level as computer-aided models such as CAD models, so that a delimitation is possible. (Alt 2012; Eigner, Roubanov, Zafirov 2014)

## 3 RISK MANAGEMENT IN PLANT ENGINEERING

Securing the future of the organization and ensuring the security of employees and the environment are topics of risk management (Wolke 2016). Risks arise in every company. They must be resolved in a conscious and controlled manner at an early stage. The Function of risk management is to examine strategic and operational activities to identify risks and errors. These uncertainties have to be identified and eliminated at an earliest possible stage (Schneck 2010). Among other things, risk management also serves as a tool for the financial control of a company. Only the disclosure of activities and the effectiveness of risk management enable the company to use these methods effectively. It must be defined beforehand what constitutes a risk. A risk is a hazard with negative or positive target deviation and is represented as a combination of the probability of occurrence and the effect of an event. Although a risk is generally considered more of a negative effect, it should be noted that any risk also brings with it the chance of improvement. Various methods of risk assessment, analysis and evaluation can be used to assess the tendency of the risk to the system. A meaningful and unambiguous description of the risk is important in the context of risk management. It must be formulated in a way that is comprehensible to all participants and must include not only the description but also the impact. The analysis and evaluation of risks is referred to as risk assessment and includes all steps required to identify and reduce a risk (ISO 31000 2009).

Another point is the security, which is defined as the ability of a system to prevent hazards. Security is divided into plant and occupational security in plant engineering. Plant security includes hazards and risks that may arise during operation as a result of plant process technology. In this case, the source of the hazard is the plant itself. Occupational security is not considered in the further course of the study. (Voigt 2010; Vanini 2012)

There are numerous methods to analyse and evaluate risks, for example the Failure Mode and Effects Analysis (FMEA) or the Hazard and Operability Method (HAZOP). The FMEA is considered as the most effective and common method for evaluating product- or process risks (Voigt 2010). Although the FMEA is generally valid, it cannot assess the project risks within the developed approach to integrate it into the system landscape. Compared to the HAZOP method, the FMEA is significantly less complex, since a rough estimation of the risk is already possible on the basis of three criteria.In the case of classic FMEA, a risk priority number (RPN) is determined based on the probability of occurrence, the severity of the event and the detectability of the error. Each of these three factors has a theoretical value of 1-10, so that the RPN can be between 1 and 1000. Company-specific scales define from which RPN value on a risk must be handled and counter measures initiated. In general, the higher the RPN, the greater the risk. In practice, this classification from 1-10 for each factor is often changed because it allows too much room for manoeuvre and therefore risks are not clearly defined. (Eberhardt 2013)

Since, as compared to the HAZOP method, no expert team is necessary for the evaluation, the results can be quickly and easily determined and transferred. The representation in SysML is simple, because in SysML inner elements can be assigned to a block or diagram. In the case of the FMEA, these would be the three evaluation criteria of the RPN. This would allow an initial assessment of whether the component is critical or not.

Especially in plant engineering, risks and faults are usually only considered at the component level of the system. The various methods are used, but the effects of the errors are not considered across domains.



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

The following approach is necessary in order to identify errors and risks of such a large product as a plant at an early stage and to identify possible influences.

# 4 APPROACH TO INTEGRATE RISKMANAGEMENT INTO SYSML

In order to model a system properly, it is necessary to have an ontology that defines certain areas and can therefore manage all knowledge. To successfully integrate engineering processes into a system model, it is essential that these are already noted in the ontology. Figure 4 describes the required ontology. A system can thus be described by three different View. The model contains the requirements for the product to be developed the risks which arise during the development and the architecture on different system levels. The requirements within this ontology are differentiated between functional and non-functional requirements. Functional requirements determine which function the new product should fulfil. The non-functional requirements define boundary conditions for the system that are not mandatory for the function. The system architecture is modelled according to the RFLP approach shown in chapter 2.2. In this example, the requirements are modelled separately using the requirement diagramm type from SysML but can be referenced to the architecture. The architecture is displayed on three levels in the form of configuration items (CI). The modelled levels are the functions, the logical and physical structure.
Different Engineering processes like risk management can be integrated within this model. To perform risk management, errors and risks must be created within the model and linked to components, requirements or other system elements. This enables cross-structural traceability, which can be used to determine influences in the event of an error occurring.

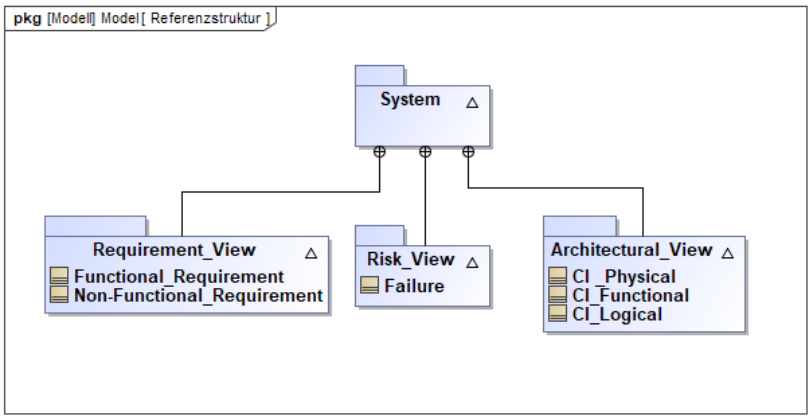


*Figure 4: Referencestructure of a SysML-modell to integrate engineering-processes (Braune 2018)*

The mentioned approach offers the possibility to attach errors and risks within a system model described using SysML, to certain components, without extending an extra diagram type. All necessary information is stored in the metadata of the components or the system. By linking risks with requirements and the system architecture, possible influences and links can be identified in the occurrence of an error. In addition, chains of effects and influences can be determined and visualized.

# 5 IMPLEMENTATION OF RISK MANAGEMENT IN A SYSTEM MODEL

## 5.1 Development of a cross-structural data model

Within the scope of the research project, the three levels for the cooling circuit of a power plant were modelled. Figure 5 shows an exemplary section of the logical level. The Functional and physical structure are modelled in the same way.
The main element of the logical level is the CI_Logical Power Station. This consists of further elements, such as the condenser and the cooling circuit (CC). Already on the 2nd sublevel, the logical elements are linked to functions, e.g for example the cooling circuit, which is connected to the "Cooling Turbine Steam" function. On the 3rd sublevel, the individual components of the cooling circuit are shown as "CC Component". On this level, functions as well as physical elements are



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

assigned to the logical elements. In order to represent the entire architecture, all three system levels must be modelled consecutively so that a fully-linked and continuous data model is created.

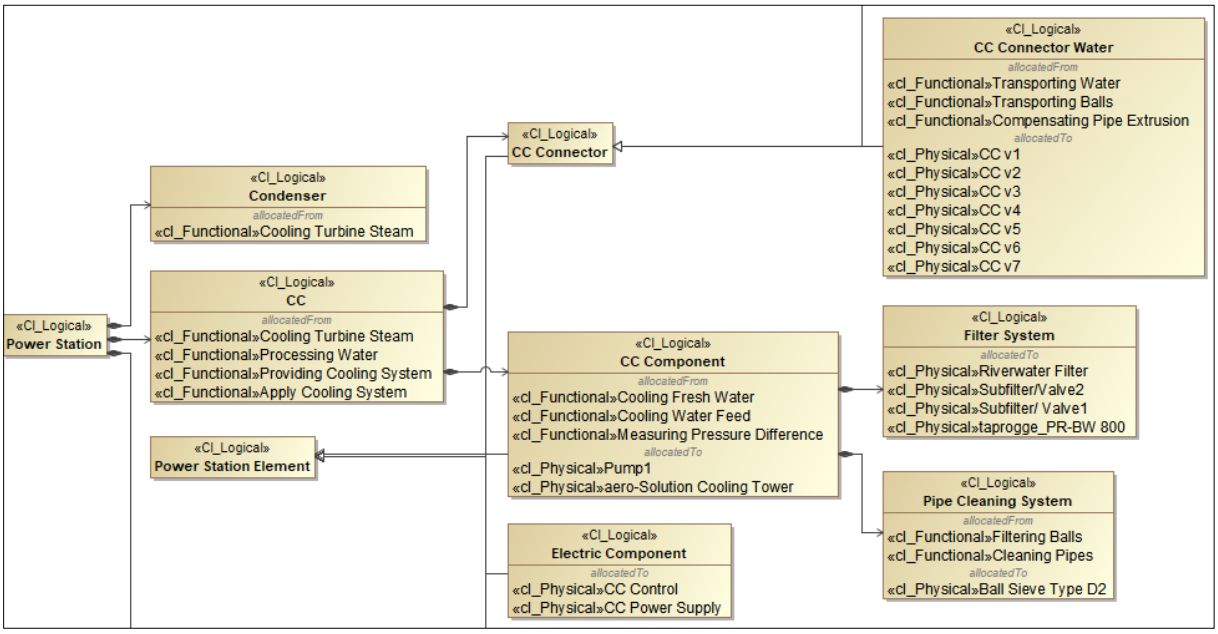


*Figure 5: Section of the logical level in the system model (Braune 2018)*

The functional structure and the concept of the cooling circuit lead to CIs, which describe the implementation of the functions with suitable active elements, without naming specific physical components. The user is thus able to quickly and easily exchange data between two CI. The creation of information items and the correct allocation of the flow direction, as usually done in the internal block diagram (IBD), is not necessary. If the structure is changed, the property values can simply be deleted or supplemented with new ones. A graphical modelling is not necessary in this case.
Figure 6 shows a visual representation of the links and connections of the system model. The logical structure of the system is in an interaction relationship with the PDM system, which manages the plant data. It is noticeable that two physical objects or models are linked via the functional network and the logical structure. One engineering process that was implemented in the systemarchitecure is risk management. The component supplier can save errors and risks in the PDM system. Influences from other components can be determined more quickly via the data model and taken into consideration in the case of a change.

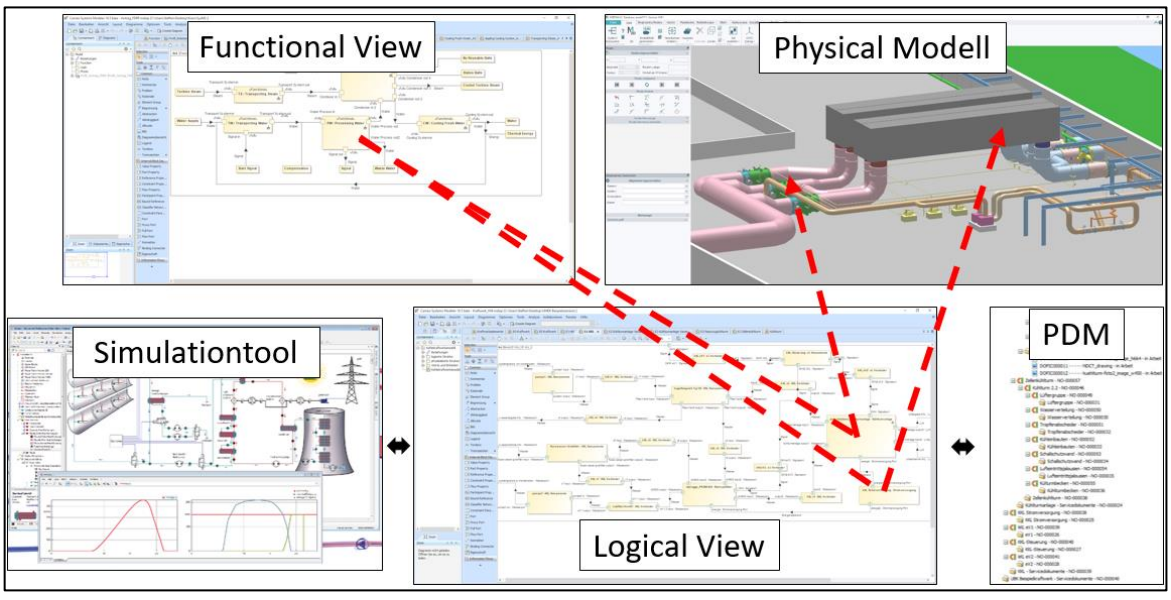


*Figure 6: Visualization of the connections within the system structure*



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

## 5.2 Integration of risk management in SysML

For risks and errors to be integrated into the SysML model, a necessary stereotype must first be defined. Since all system elements on the different levels are represented and linked as block diagrams, the stereotype “Failure” is also created as a child of the stereotype “Block”. SysML does not offer the possibility to link block diagrams across all structures. Therefore, an ontology was developed which is necessary for the implementation within the research project (Kunnen 2018).

The Failure block contains attributes that are derived partly from analysis methodologies of risk management and partly from the documentation of the research project. In addition to the ID the Risk, Action and Scenario properties are used to describe the failure. The “Risk Type” again uses an enumeration literal as data type for characterization. In order to establish a uniform terminology, the user has the option of choosing exclusively between technical, political, economic and ecological risk types. With the sum of the property values, each risk or error can be described in terms of key points, but in enough detail.

The most important property value refers to the effective range of the error. The stereotype is modelled using the association relationship in the profile. It exists in the profile between “Failure” and “CI_Functional”, since according to the requirements of the system model, the error is supposed to act on the function level. With the association ends “HasFailure” and “IsFailureOf” the relationship is concretized and shows which error affects which “CI_Functional”. Similarly, associations to “Functional_Requirement” and “Non-Functional_Requirement” were established. An example of an applied risk is shown in the top left of Figure 7, which shows the risk that the pipes within an installation can no longer be cleaned. This example comes from the research project. Several attributes were assigned to the stereotype block. If the error occurs that the pipes of the plant can no longer be cleaned, a countermeasure is first proposed. In this case, a component-specific countermeasure has been defined. The “IsFailureOf” relationship names the affected function. Furthermore, the error is assigned to a possible risk. Since this is a technical error, it is also assigned to the corresponding risk category. The scenario describes that this risk has no influence on the development because, as already mentioned, it is a component-specific problem. The data is entered using the PDM system and then transferred to and adjusted to the stereotypes in Figure 7, to the system model.

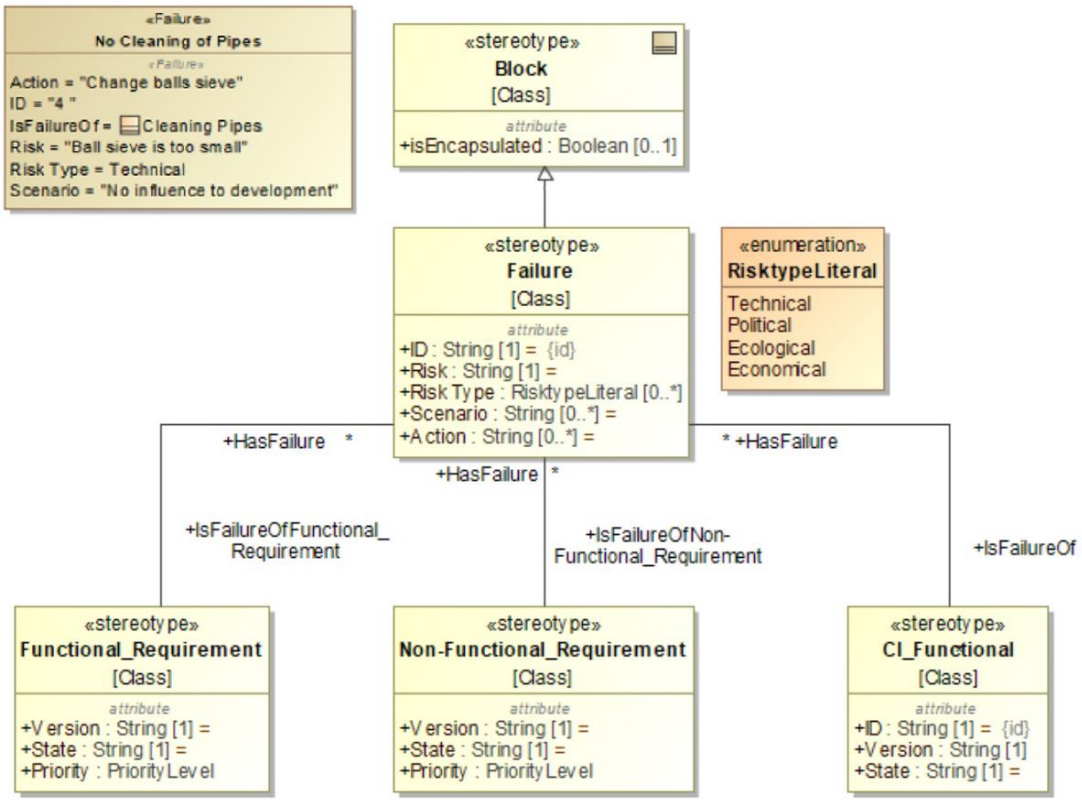


*Figure 7: Stereotyp failure and example for a specific component (Braune 2018)*



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

The Risk View of the system model contains all errors and risks of the CIs and requirements entered in the system for specific components. In this way, a countermeasure can be defined for the event of an occurrence and an internal ID. Under "IsFailureOf" in the specification, "CI_Functional", "Functional_Requirement" and "Non-Functional_Requirement" can be selected as property values. Due to the inverse relationship, the error appears as value under "HasFailure" in the selected element. In addition, the specific risk category is displayed.
In order to integrate a system model into the planning process of a product it is necessary to connect the system model with the PDM-System of the developer. Due to the overlapping of PDM system and system model, the manufacturers of components in the system model only work indirectly. Within the model the data can be managed for inhouse development and provided to the system engineer. The basis to integrate the RM-process into the system model is the representation and transmission into a PDM-system. Data can be attached to certain items in a familiar environment and retrieved at any time which is the central advantage. By constantly exchanging information from the system model with the PDM system, all disciplines involved are provided with the required information. Risks are no longer considered at the component level, but throughout the entire domain, so that the product developer can identify all connections and influences of risks and errors in a system.

## 6 SUMMARY AND OUTLOOK

Within the research project, the focus lay on multidisciplinary development and the corresponding support of all engineering processes necessary for successful product development. Many discussions with project partners showed that the multidisciplinarity is precisely what leads to problems repeatedly. One reason for this is that many disciplines use different tools for similar or identical development processes. A standardization of the processes for a development is therefore advisable. A data model that allows all participants to implement and store the data of their components can support this process.
Based on the V-model, a structure and level comprehensive data model was developed in SysML. For the products to be developed, a wide variety of engineering processes must be applied. To ensure that the data is consistently and permanently available, it is attached to the system model. Risk management was regarded as an example. Risk management in plant construction is not applied across domains, but only considers the individual components of the corresponding suppliers. A new stereotype was created in SysML so that risks can be attached to system elements. This contains the "Failure" block, which is filled with risk-specific information. This block can be linked in the system structure with all the components created. This enables the effects and influences of risks on connected elements to be determined quickly so that the product developer can determine a corresponding link chain in the event of a risk occurrence.
In this paper, a method was presented that allows risks and errors to be implemented in a cross-structural SysML structure. This simplifies the work of the product developer, as he can determine the influence of risks or the effects on linked components at any time.
However, the developed model offers great potential for optimization. At present, only the product data is implemented in the system. In the future, process data such as operating parameters etc. will also be transferred to the system model. This enables not only risk management but also the application of further engineering processes. Furthermore, necessary changes due to errors and risks can be processed more quickly. Simulations with critical process parameters can be carried out and thus more precise statements about the risk impact can be made.
The development of an own diagram set for the extension of the SysML would be a further constructive step. Thus, the currently necessary development of company-specific stereotypes for the development of risk models could be avoided and a universal solution could be developed.

## REFERENCES


Alt, O. (2012), *Modellbasierte Systementwicklung mit SysML*, Carl Hanser Fachbuchverlag, München.

Braune, S. (2018), *Datenaustausch in der modelbasierten Systementwicklung zur Visualisierung von SysML-Strukturen*, Universität Duisburg-Essen, Institut für Produkt Engineering, Masterarbeit.

Eberhardt, O.: Risikobeurteilung mit FMEA : Die Fehler-Möglichkeits- und Einfluss-Analyse gemäß VDA-Richtlinie 4.2 ; die Risikobeurteilung von Maschinen gemäß EU-Richtlinie 2006/42/EG. 3., überarb. Aufl. Renningen : expert-Verl., 2013 (Edition expertsoft 63)




https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press

Eigner, M., Roubanov, D. and Zafirov, R. (2014), *Modellbasierte virtuelle Produktentwicklung*, Springer Vieweg, Berlin.
VDI-Richtlinie 2206. (Juni 2004), *Entwicklungsmethodik für mechatronische Systeme*.
Friedenthal, S., Moore, A. and Steiner, R. (2015), *A practical guide to SysML : The systems modeling language. Third edition*, Morgan Kaufman, Waltham, MA.
Hooshmand, Y. (2017), *DS 87-3 Proceedings of the 21st International Conference on Engineering Design (ICED 17) : An approach for holistic model-based engineering of industrial plants*, The Design Society, Glasgow.
Hooshmand, Y., Köhler, P. and Manoharan, T. (2017), *Unterstützung von multidisziplinären Engineering-Prozessen im Kraftwerksbau (UMEK)*, Magdeburg.
INCOSE. Systems Engineering. URL https://www.incose.org/systems-engineering – Überprüfungsdatum 2018-12-13.
Informationstechnikzentrum Bund (Hrsg.): V-Modell XT Bund : Das Referenzmodell für Systementwicklungsprojekte in der Bundesverwaltung Version: 2.1, 2015
ISO 31000 (2009 ), *Risk management*, International Organisation for Standardization, Geneva .
ISO/IEC/IEEE 15288 (2015), *Systems and Software Engineering -- System Life Cycle Processes*, International Organisation for Standardization, Geneva .
Kaufmann, U. and Schuler, R. (2016), *Systems Re-Engineering - ein Beitrag zur Integration von MBSE und PLM : Tag des Systems Engineerings*, Hanser Verlag, München.
Kunnen, S. (2018), "Entwicklung und Anwendungsmöglichkeiten eines strukturübergreifenden Datenmodells", In: Klaus Brökel, Burkhard Corves, Karl-Heinrich Grote, Armin Lohrengel, Norbert Müller, Arun Nagarajah, Frank Rieg, Gerhard Scharr and Ralph Stelzer, (Hrsg.), *Digitalisierung und Produktentwicklung - vernetzte Entwicklungsumgebungen : 16. Gemeinsames Kolloquium Konstruktionstechnik : am 11. und 12. Oktober 2018 in Bayreuth : Tagungsband*, Universität Bayreuth Lehrstuhl für Konstruktionslehre und CAD, Bayreuth, pp. 200–211.
OMG SYSML: OMG SysML Home | OMG Systems Modeling Language. URL http://www.omgsysml.org/ – Überprüfungsdatum 2017-12-06
PAS 1059 (2006), *Planung einer verfahrenstechnischen Anlage — Vorgehensmodell und Terminologie* , Beuth Verlag GmbH.
Schneck, O. *Risikomanagement : Grundlagen, Instrumente, Fallbeispiele. 1. Aufl*, Wiley-VCH, 2010 (WILEY Klartext), Weinheim .
SEBoK (2016), *Transitioning Systems Engineering to a Model-based Discipline : Systems Engineering Body of Knowledge (SEBoK)*.
Vanini, U. (2012), *Risikomanagement : Grundlagen ; Instrumente ; Unternehmenspraxis. 1. Aufl. s.l.*, Schäffer-Poeschel Verlag.
Voigt, K.-I. (2010), *Risikomanagement im industriellen Anlagenbau : Konzepte und Fallstudien aus der Praxis*, Schmidt, Berlin.
Walden, D.D., Roedler, G.J. and Forsberg, K. (2017), *INCOSE systems engineering handbuch : Ein Leitfaden für Systemlebenszyklus-Prozesse und -Aktivitäten. [1. Auflage]*, GfSE, München.
Wolke, T. (2016), *Risikomanagement. 3., vollständig überarbeitete, erweiterte und aktualisierte Auflage*, De Gruyter Oldenbourg, Berlin, Boston.



https://doi.org/10.1017/dsi.2019.364 Published online by Cambridge University Press