Transdisciplinary Engineering: Crossing Boundaries
M. Borsato et al. (Eds.)

doi:10.3233/978-1-61499-703-0-1071

Driving Product Design and Requirements Management with SysML

Edmar HINCKEL[a,1] , Milton BORSATO[a] , Juliana SCHMIDT[a]
Fabio MACCARI [b], Paulo STORRER [a] and Eduardo ONOFRE [a]
[a] *Federal University of Technology, Paraná, Brazil*
[b] *CNH Industrial Latin America, Brazil*

Abstract. The Product Development Process (PDP) multidisciplinary aspect, under Concurrent Engineering (CE) principles, leads to overwhelming complexity, where several systems, methods and tools are used in a process with intensive information flow. Nonetheless, the absence of a common language for describing information components, regarding product, process, design and business, give rise to multiple interpretations, hindering full understanding and therefore produces rework and quality issues. This scenario highlights the need for ways to make the availability of product requirement information more dynamic and scope-sensitive (i.e. different levels of abstraction) along the PDP stages. In this context, Model-based System Engineering (MbSE) and supporting system-modeling languages such as SysML propose a product representation structure, through a unique and timeless model, which potentially drives the whole product lifecycle, as the single and ubiquitous information source to stakeholders. In this sense, the goal of this work is to propose a system model that provides reliable product representation, able to support product requirement definition tasks and their use along the PDP, allowing significant gains in productivity and reduction of non-conformities. The methodology adopted in this work follows the principles of DSR (Design Science Research), considering a real scenario inserted in a multinational enterprise context, in the agriculture-applied machinery sector. The model proposed is expected to assist the generation and usage of product information at various abstraction levels, by all stakeholders during the PDP, therefore reducing rework and enhancing design quality.

Keywords. Model-based Systems Engineering, System Modeling Language, Requirements Management, Product Development Process.

Introduction

The Product Development Process (PDP) is characterized by complex information flow. Information are shared across different languages (e.g. 3D models; spreadsheets; texts; mathematical formulas, videos, images), without a common language, giving rise to multiple interpretations on the whole product [1]. This information discontinuity, mainly the difficulty in interpreting and manage requirements, generates the need for product changes, requiring extra efforts and costs [2]. Several computational tools have been developed to work with Product Lifecycle Management (PLM) for requirements management (e.g. Polarion, DOORS). However, in practice these tools are applied in a

[1] Corresponding Author, E-Mail: ehinckel@utfpr.edu.br

standalone way, therefore not ensuring integration along the product lifecycle and potentially causing misalignment between requirements and product features [3]. Thus, this scenario demands models that help designers of several different technical skills to access the required information easily, quickly and reliably, allowing traceability [4].

Model-based Systems Engineering approach (MbSE) has been proposed as an alternative to information flow representation to complex products. This approach aims to allow the maintenance of complex engineering systems [5]. In this context, System Modeling Language (SysML) has been highlighted. SysML has been applied in [6]various industries for the purpose of modeling requirements, describing physical connectivity and modeling the structure of organizations [7]. In this sense the research questions addressed in this work are:

Q1. Could a SysML model allow better correlations between different requirements and product components?
Q2. Could a SysML model allow unique understanding of the requirements among stakeholders?
Q3. Could a SysML model allow better requirements management and traceability?

The goal of this work is to propose a model that can support requirements management of complex products across the PDP, being the unique stakeholders' information source.

The present article is organized as follows: Section 1 of this paper presents a brief literature review on SysML, applied to requirement management in the PDP, as well as a theoretical overview on the types of information handled along the PDP, fundamentals of SysML and requirement management. Sections 2 and 3 describe the methodological aspects of the present research and the proposed model. Sections 4, 5 and 6 bring preliminary results of the study, a short discussion on the findings and the conclusions.

1. Conceptual Background

This section describes the basics concepts used for developing the proposed model and a brief literature review on the main applications of SysML in the context of PDP and Requirements Management (RM).

1.1. State of art

Some authors have proposed the use of SysML and UML to optimize development and requirements tracking throughout the PDP [8]. Very few studies focus on the standardization of SysML modeling inputs for the beginning of the PDP, and how this information can be viewed and tracked by stakeholders from different disciplines and different abstraction levels towards the PDP. [9] propose the use of a framework called RSL (Requirements Specification Language) based on SysML to formalize the requirements in embedded systems development applied to automotive systems. However, there is still a gap in the literature, focusing on how standardized information

are converted into SysML diagrams at early stages of the PDP, and its correlation with other system components.

1.2. Model-based Systems Engineering.

The MbSE approach allows to develop system and process models, through specifics languages, as well as the description and application of relationships between components of a system through transformation models [10]. In the PDP context, MbSE allows semantic consistency between various knowledge domains through formal languages, machine-readable statements, thus enabling interoperability between different tools (e.g. CAD, CAE, ECAD, discrete simulations) as shown in Figure 1 [11]. The scientific literature suggests two main languages related to MbSE applications, which are the United Modeling Language (UML) and SysML.

1.3. Unified Modeling Language

UML is widely used in industry as a standard for object modeling [12]. It allows graphical representations of products by means of several kinds of diagrams. Such diagrams, arranged in classes, define how product and process items are related. This representation is known as meta-model [13]. However, UML is limited to representing only high-level functional requirements.

1.4. System Modeling Language

SysML basically uses various components of UML, but suggests two additional diagrams, called requirements and parametric diagrams. These diagrams help assisting the search, analysis, validation and documentation of requirements in addition to quantitative analysis [7]. A SysML model allows system visualization from various perspectives, maintaining consistency between them.

1.5. Boilerplates

A boilerplate is a template with some fixed syntax elements and variable parts to be filled in by engineers, during the requirements definition. This template allows the requirements formalization through textual expressions. Figure 1 shows system function, description, object related and performance values and units, which allow requirements traceability [14].

The <bulkhead> shall be able to <carry><driver> less than <69> <dB>

The <system> shall be able to<function> <object> less than <performance> <unit>

Figure 1. Boilerplate structure, adapted from [28].

1.6. ReqIF

The Requirements Interchange Format (ReqIF) is a standard XML-based to exchange and storage requirements between different tools. Through this standard, requirements can be accessed and managed by different stakeholders, which use different RM tools [15]. Therefore, ReqIF can be used as a strategy to export requirements of a SysML model and import it to other RM tools. Figure 2 shows the basic structure of ReqIF. The standard carries the requirement identification, a textual description, status and revision.

Figure 2. ReqIF structure, adapted from [29].

2. Methodology

The present research has been conducted based on DSR methodology (Design Science Research). This method aims to develop ways of achieving goals through a set of artifacts [16]. DSR must start with the identification of the research problem and solution goals. The next stage is to develop the model, evaluate it and report the results. Currently, the present research is in the development phase. Figure 3 shows the steps taken for the execution of this work [17] .

Figure 3. Work procedure steps.

Initially an information was collected in a partner company through interviews with key stakeholders of an agricultural tractor project (e.g. Marketing, Product Engineering, Manufacturing) in addition to the analysis of available project documentation. It was possible to identify discontinuity and ambiguity of requirements across the PDP. To facilitate future evaluation of the model, a tractor subsystem was chosen, named Vehicle System (i.e. tractor cabin), as study scenario. The requirements list was generated through documents provided by the department that performs product approval, in addiction to discussions with stakeholders.

3. Proposed Solution

Figure 4 shows an overview of the proposed model. The inputs are standardized information (i.e. ReqIF with boilerplates) which allow the SysML diagram building

and establishing the relationships between then, represented by black connectors. The following sections further detail the steps taken to construct the model. The tool used for SysML modeling was Eclipse Papyrus due its open source characteristic.

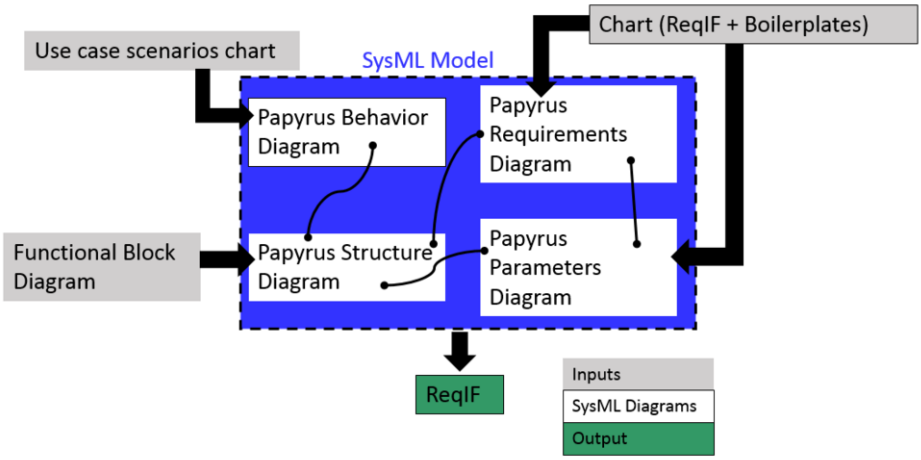

Figure 4. Proposed model concept.

3.1. Behavior modeling

The chart shown in Figure 5 presents use cases about the scenario, identified and described textually in order to guide the modeling of the Use Case Diagrams (UCD), which are one of the environmental diagrams types used in SysML for the purpose of describing the scenario of study. Each use case action is described in sequence, identifying the driver as an actor, and the subsequent actions (e.g. 'start tractor engine is a consequence of "turn ignition" action').

Based on the chart, the UCD was modeled, as shown in Figure 6. Lines link the actor to his actions, and the arrows indicate dependencies between actions (e.g. 'tractor only turns if the ignition is turned').

ID: UC1 NAME: Drive tractor Actor: Driver			
Main Scenario:			
1.	Open the door;	5.	Handle levers;
2.	Close the door;	6.	Press command buttons;
3.	Sit on the seat	7.	Turn ignition;
4.	Adjust seat;	8.	Start tractor engine

Figure 5. Use case scenario chart.

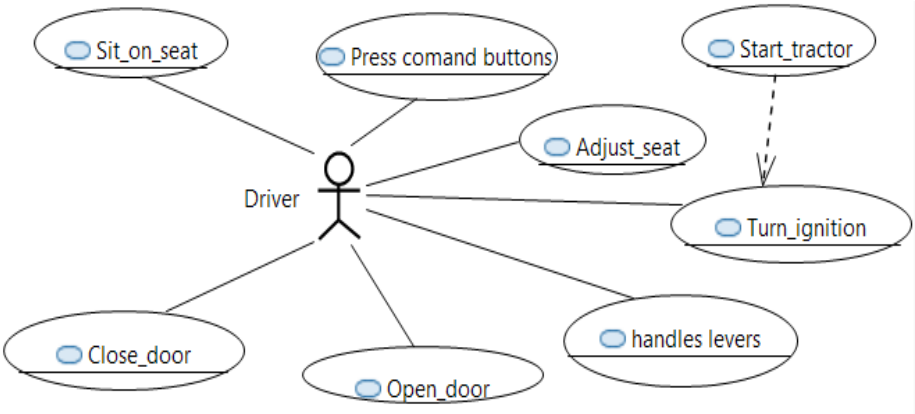

Figure 6. Use Case Diagram (developed in Papyrus).

3.2. Requirements Modeling

The information collected from requirements was in textual form, in several different documents. Therefore, for standardizing requirements, another chart was used, following the ReqIF structure, as shown in Figure 7. The field description was defined using boilerplates. The second column identifies the requirement constrain (i.e. brake pedal load). The chart allows one to figure out that the pedal has the function "support the user foot", and the maximum stress value is 180 MPa. Furthermore, it identifies the requirement category (i.e. functional) and the last two columns indicate early stages of the PDP (i.e revision 1.0 and 'starting' maturity)

Category	Constraint	ID	Text	revision	maturity
Functional	Brake_pedal_load	3.3	The **<brake_pedal>** shall be able to **<Support><user foot load>** less than **<180> <Mpa>**	1.0	Starting

Figure 7. Information input standard chart.

Based on the chart previously described, a Requirements Diagram (RD) was created in Papyrus, namely "Break pedal load", as shown in Figure 8. The RD shows that the part of the product that satisfies this requirement is the brake pedal, connected to the requirement by a dotted arrow labeled 'Satisfy1', which must be verified by a Finite Element Analysis (FEA) simulation, also connected to the requirements by a dotted arrow and labeled as 'Verify1'.

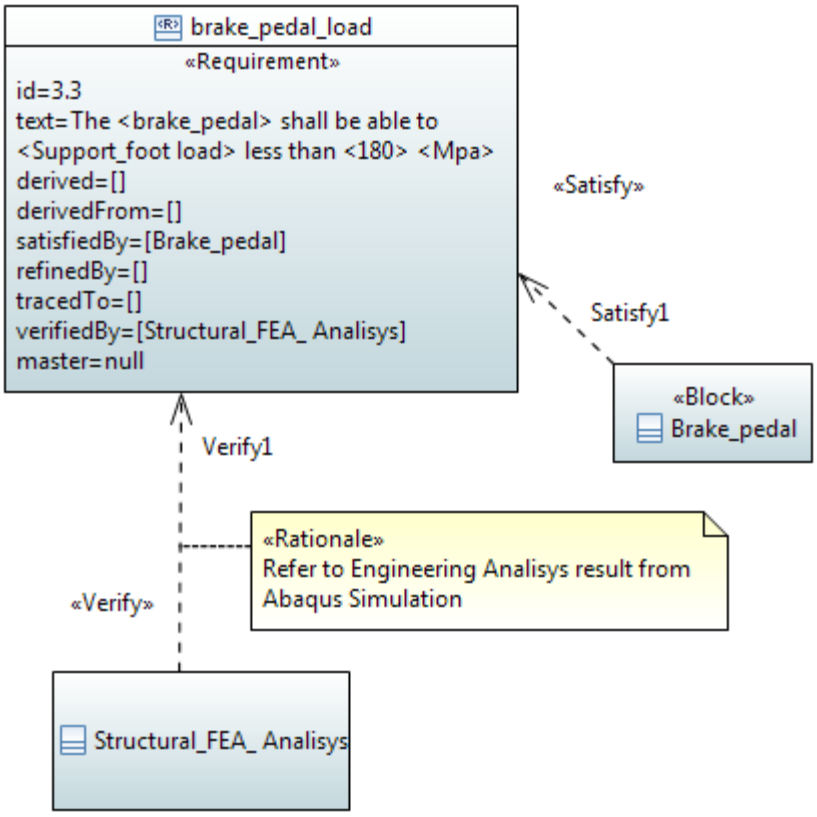

Figure 8. Requirements Diagram (developed in Papyrus).

3.3. Structure Modeling

Figure 9 shows the decomposition of the tractor structure, based on a product Functional Block Diagram (FBD), provided by the partner company. Each block represents a subsystem of interest (e.g. Vehicle System), and the lines represent its decomposition (e.g. "Structure System" is a "Tractor" subsystem, and "Vehicle System" is a "System Structure" subsystem). Besides the diagram in Figure 10, another diagram was created to represent the hierarchy of "Vehicle System" components. However due to its large size, it will not be illustrated in this paper.

3.4. Parametric modeling

In order to allow the verification of requirements constraints a Parametric Diagram (PD), as shown in Figure 10, was created. The "Brake pedal load" constraint was decomposed into two other constraints, representing the force and pressure equations. Thus the PD represents the system of equations to verify the requirement.

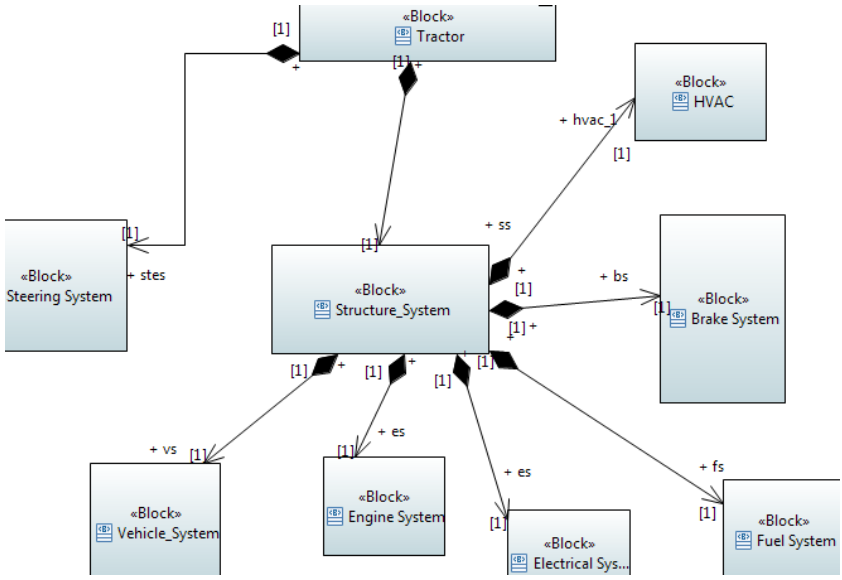

Figure 9. Structure Diagram (developed in Papyrus).

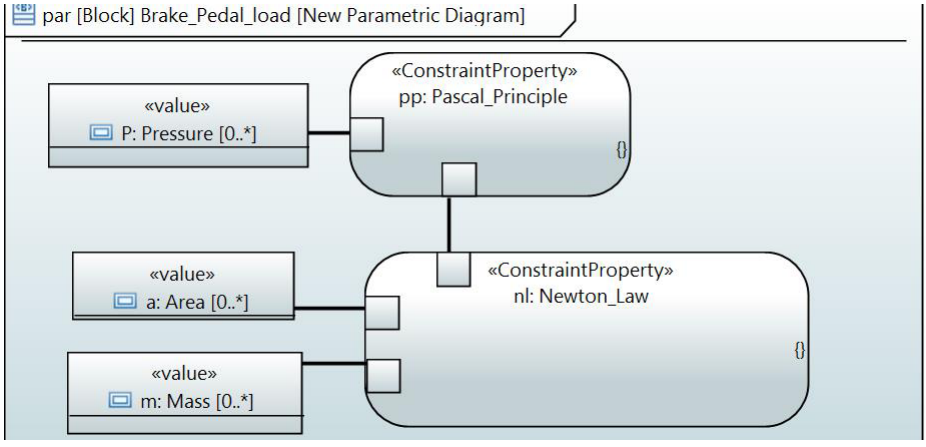

Figure 10. Parametric Diagram (developed in Papyrus).

3.5. Requirements traceability and changes control

The requirement control and traceability is possible through the link between the SysML model and a RM tool. As shown in Figure 11, the requirements specified in the model should be exported from Papyrus, through ReqIF standard and imported in RM tool. However, to allow the link between ReqIF and other model artifacts SysML (e.g. diagrams relations) an application is required to perform this function, which will not be addressed in this paper. In this way through the RM tool can perform the traceability

requirements from SysML model. Changes in requirements along the PDP are controlled by ReqIF revision number, which should always be synchronized between the model and the RM tool. Creating new requirements will be carried out by importing new information ReqIF in Papyrus. The system engineer does the detailed specifications of new requirements and new correlations manually.

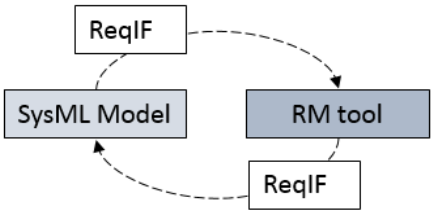

Figure 11. SysML control cycle.

4. Preliminary Results

The proposed model allows representing the correlation of the product with its subsystems' "vehicle system" and component "brake pedal" requirements (Q1). In addition, the ReqIF standard and boilerplates made it possible to formalize requirements, and consequently provided unique interpretation of requirements along the PDP (Q2). Thus, the proposed model allows the management and traceability of requirements, along the PDP, through correlations represented in SysML diagrams and the link with a RM tool (Q3).

5. Discussion

Research questions Q1, Q2 and Q3 of this study have been answered. However, to validate the proposed model, a more detailed system is necessary in order to verify if the concept allows consistence and continuity requirements in all PDP stages in the entire product. With this detailing, it will also be possible verify the validity of the proposed model for multiple related requirements within multiples subsystems, components and behaviors.

The critical task in this work has been the link of standardized information into ReqIF to SysML diagrams in Papyrus. In order to solve this issue, the next research step will be to apply a Papyrus plugin (e.g. ReqCycle) that supposedly allows this link. In addition, another important future research step is to demonstrate how ReqIF could be exported from SysML, allowing its use with other PLM tools (e.g. CAD, CAE).

6. Conclusions

This paper showed the great potential of SysML in the development of requirements, which key to all steps of a PDP. In addition, the importance of having a single model

that centralizes all product gift information was highlighted, therefore not allowing ambiguities and discontinuities of information. Furthermore, this work shows the need to standardize requirements in the partner company, due to the large number and complexity of requirements in the tractor product, with several systems and subsystems, in addition to critical consequences that may occur to the product, due to missing requirements or mismodeling. Addition the requirements, there is the opportunity to apply SysML to control and manage other information along PDP (e.g. the Figure 10 structure diagram can provide an early product structure and be used as input to a bill of materials.

References

[1] M. Chami and J.-M. Bruel, Towards an Integrated Conceptual Design Evaluation of Mechatronic Systems: The SysDICE Approach, *Procedia Computer Science,* Vol. 51, 2015, pp. 650-659.

[2] W. Brace and K. Ekman, CORAMOD: a checklist-oriented model-based requirements analysis approach, *Requirements Engineering,* Vol. 19, 2012, pp. 1-26.

[3] M. G. Violante, E. Vezzetti and M. Alemanni, An integrated approach to support the Requirement Management (RM) tool customization for a collaborative scenario, *International Journal on Interactive Design and Manufacturing (IJIDeM),* 2015, doi:10.1007/s12008-015-0266-3.

[4] H. Zhang, X. Han, R. Li, S. Qin, G. Ding and K. Yan, A new conceptual design method to support rapid and effective mapping from product design specification to concept design, *The Int. Journal of Advanced Manufacturing Technology,* 2016, doi:10.1007/s00170-016-8576-6.

[5] K. Bhanumathi and B. Haridas, Communication through Digital Engineering Processes in an Aircraft Program, *International Journal of Advanced Information Technology,* Vol. 3, 2013, p. 19.

[6] F. Mhenni, J.-Y. Choley, O. Penas, R. Plateaux and M. Hammadi, A SysML-based methodology for mechatronic systems architectural design, *Advanced Engineering Informatics,* Vol. 28, 2014, pp. 218-231.

[7] C. Durugbo, Integrated product-service analysis using SysML requirement diagrams, *Systems Engineering,* Vol. 16, 2013, pp. 111-123.

[8] A. Noyer, P. Iyenghar, E. Pulvermueller, F. Pramme and G. Bikker, Traceability and interfacing between requirements engineering and UML domains using the standardized ReqIF format, in: S.Hammoudi et al. (eds.) *3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2015*, 2015, pp. 1-6.

[9] M. Adedjouma, M. Wojciech, H. Dubois and F. Terrier, Efficient Methodology from Requirements to Design Models for an Automotive Application, in *Proceedings of 6th European congress on Embedded Real Time Software and Systems (ERTS 2012)*, 2012, pp. 5-7.

[10] M. Törngren, A. Qamar, M. Biehl, F. Loiret, and J. El-Khoury, Integrating viewpoints in the development of mechatronic products, *Mechatronics,* Vol. 24, 2014, pp. 745-762.

[11] N. Ivezic, B. Kulvatunyou and V. Srinivasan, On architecting and composing through-life engineering information services to enable smart manufacturing, *Procedia CIRP,* Vol. 22, 2014, pp. 45-52.

[12] J. Lasalle, F. Peureux and F. Fondement, Development of an automated MBT toolchain from UML/SysML models, *Innovations in Systems and Software Engineering,* Vol. 7, 2011, pp. 247-256.

[13] M. Fowler, *UML distilled: a brief guide to the standard object modeling language*, Addison-Wesley Professional, Boston, 2004.

[14] E. Hull, K. Jackson and J. Dick, *Requirements engineering*, Springer-Verlag, London, 2010.

[15] OMG, *Requirements Interchange Format Specification 1.1.* [Online]. Available: http://www.omg.org/spec/ReqIF.

[16] S. T. March and G. F. Smith, Design and natural science research on information technology, *Decision support systems,* Vol. 15, 1995, pp. 251-266.

[17] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, A design science research methodology for information systems research, *Journal of management information systems,* Vol. 24, 2007, pp. 45-77.