Requirement diagramming in SysML

Chaïm van Toledo
4237471
Utrecht University

Introduction

Developing complex systems can go beyond budget, can cost more time than is scheduled or can fail in addressing the right requirements. To remedy these problems, good requirements engineering is needed. One of the solutions and that is the topic of this paper, is requirement diagramming in the Systems Modelling Language (SysML). This introduction starts with the description of the goals of SysML, followed by a description of the procedure with the used notation and the needed performing development activities. Subsequently, the creators of SysML are discussed. The second part of this paper will elaborate on an example of SysML in the context of requirement diagramming. The third consists of a literature review. The fourth and last part will show a process deliverable diagram, where the steps of requirement diagramming are modelled and elaborated.

Friedenthal, Moore and Steiner (2014, p. 31) describe SysML as "a general-purpose modelling language that supports the analysis, specification, design, verification, and validation of complex systems." These complex systems may consist of hardware, software, data, personnel, procedures facilities. SysML can be helpful when a system engineer tries to improve precision and efficiency of the communication with other co-workers and stakeholders (SysML Forum, 2015). The modelling language can be traced back from the unified modelling language (UML) and took elements from it, namely: sequence, state-machine, use case, and package diagrams. SysML also extended and created requirements and parametric diagrams (Santos Soares & Vrancken, 2008).

For diagramming requirements with SysML, a box can be used with the stereotype on top (<<Requirement>>) and the title below. It can be extended by adding a text, id or other information below, just like a UML class diagram. Requirements can have relationships among other requirements and can therefore be linked to each other or they can have a relationship with a part of the system. These relationships can be more specific with the type of the relationship, this can be hierarchy, derive, satisfy, verify, refine and trace (Santos Soares & Vrancken, 2007). Figure 1 shows a small example of this.

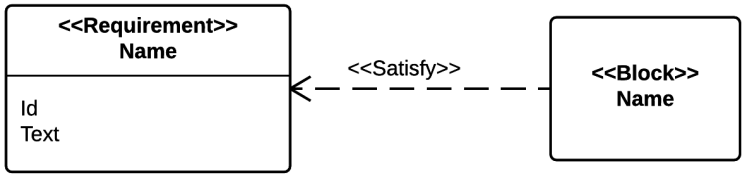

Figure 1: Requirement with relation to other part of the system.

Another important factor in system design is requirements traceability, it "helps in identifying the sources, destinations and links between requirements" (Santos Soares & Vrancken, 2008, p. 61). Theretofore a table is useful where the id, name, type, the relationship with other requirements and how it relates can be stored. The format of the table is not set and can be extended if necessary.

Use cases are not necessary with requirement diagramming, but can be helpful to represent the requirements to each stakeholder. Use case diagramming in SysML is the same as in UML and is not modified or extended. It can easily show which stakeholders are interacting with different parts of the system.

SysML was in 2003 created by the SysML Partners, which was founded by Cris Kobryn, a system engineer and entrepreneur. SysML Partners is an informal association of experts in system engineering and software modelling tool vendors (SysML Partners, 2015).

Example

In this part an example of requirement diagramming in SysML will be discussed. In total two deliverables will be shown and explained: The SysML requirement diagram and the requirements table. The case will be part of an online ticketing system. An online ticketing system has several concerns like handling the many customers when an event is popular. There is a maximum of how many visitors a system can handle per second and therefor customers shall have a waiting screen when the system has to much visitors. Customers also want to see how many tickets there are left. And at last, a customer must be able to buy a ticket. Figure 2 shows how these requirements can be drawn in SysML.

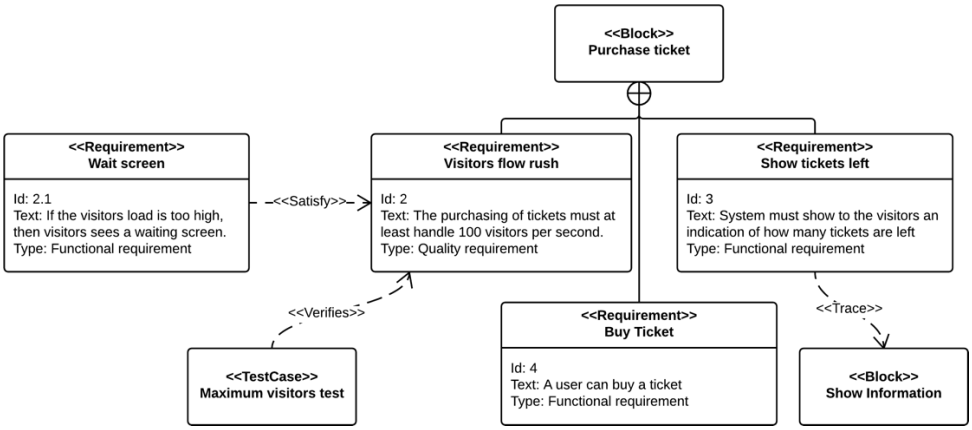

Figure 2: Deliverable 1, the requirements diagram of the ticketing system

The Purchase Ticket block is a part of the system. It can be seen as a black box, how the system looks is not further detailed. The block has some requirements though, the first is Visitors flow rush, which is a quality requirement and has the wish for the system to handle at least 100 visitors a second. The test case of Maximum visitors test should verify if this is possible. The Wait screen satisfies the Visitors flow rush when there are too many visitors. Show tickets left has a traceability to the system that always shows information. Another representation of the requirements diagram is provided at the table, table 1 shows the requirements with other elements of the ticketing system.

Requirements table ticketing system

Id	Name	Relates To	Relates How	Type
1	Purchase Ticket			
2	Visitors flow rush	1		Quality requirement
2.1	Wait screen	2	Satisfy	Functional requirement
3	Show tickets left	{1, 6}	{6, Trace}	Functional requirement
4	Buy Ticket	1		Functional requirement
5	Maximum visitors test	2	Verifies	
6	Show information			

Table 1: Deliverable 2, the requirements table of the ticketing system

Related Literature

This literature review provides critique, research and developments in requirement diagramming in SysML. Schneider, Naughton and Berenbach (2012, p. 202) are not satisfied with requirement diagramming in SysML and state that the language "[focusses] on the technical description of the envisioned system, and not on the upstream rationale for the requirements." Another point of critique came from Ringert, Rumpe and Wortmann (2012). Although they elaborate that SysML can be used for requirements in general, SysML is not suitable for modelling the behaviour requirements of cyber physical roboting systems. What the solution will be for this problem remains unknown and need to be investigated further.

On the other hand, there are also positive findings. Scanniello, Staron, Burden and Heldal (2014) compared requirement diagramming in SysML with UML use case diagrams. The experiment, held on students from Italy and Sweden, showed that the understanding on specification documents of complex systems improves, without affecting time. Although the results are in favour for SysML, the experiment is only conducted in universities and not in the field. Sung and Kim (2012, p. 534) state that "SysML requirements diagram can facilitate the transformation of user requirements into system requirements and improve the requirements' traceability throughout the design life cycle."

The modelling language itself is also under development and improvement. One of these developments are extending the language with grouping requirements and custom stereotypes (Santos Soares & Vrancken, 2008). With grouping requirements, sub requirements can easily be attached to a requirement and an extra box is grouping the requirements. Custom stereotype can be extended to describe directly what kind of requirement it concerns. The stereotype can be changed from <<Requirement>>, to <<QualityRequirement>> or other types such as functional requirements.

Process Deliverable Diagram

The following part contains the Process Deliverable Diagram (PDD) with the corresponding tables. The PDD, notated in figure 3, shows a process, modelled on the left, based on an UML activity diagram. On the right of the PDD the deliverables are modelled in on an UML class diagram based model (Weerd & Brinkkemper, 2007). The activities and concepts of the requirements diagramming technique are derived from the article of Santos Soares and Vrancken (2008).

The first stap of the PDD begins with classifying the atomic requirements. Atomic requirements are “primarily, `well-formed requirements'”, which means that they “are abstract, unambiguous, traceable and validatable” (Salzer & Levin, 2004, p. 46). It is important to have such requirements to begin with modelling the requirements into the system, for the sake of understanding wat different stakeholders want. With classifying the requirements, the requirements engineer knows what kind of types of requirements he/she has.

The second step is requirements modelling. Some requirements can be combined, and that will eventually upgrade the readability of the diagram. Then the requirements can be modelled in into the SysML Document and after this done, the relationships can be drawn between the requirements and between the requirements and the system. Important is to have a requirements table where there requirements are further elaborted. Recommended, but not neccesary are linking the requirements to the use cases. The last part is to evaluate the diagrammed requirements with the stakeholders. Thereby the requirements can be break down in to smaller requirements to give a better understanding of the requirements.

To go into detail about the PDD, the tables will provide the necessary documentation. The activities are elaborated in table 2. In table 3, the delivarables are detailed elaborated.

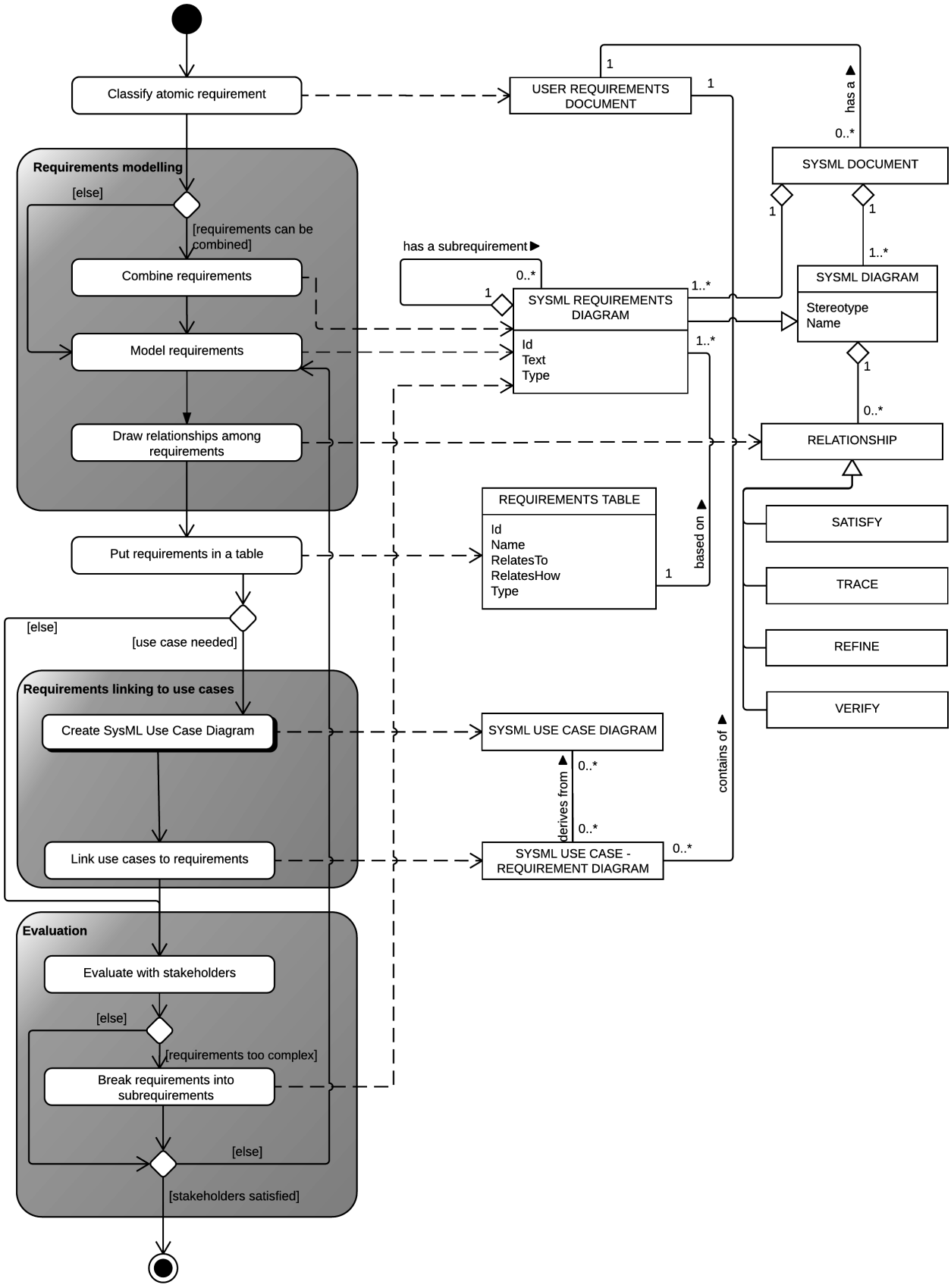

Figure 3: PDD of the Requirements diagramming in SysML technique.

Activity	Sub activity	Description
Classify atomic requirement		The classification for each atomic requirement will avoid confusion of which type of requirement is written. Put this classification in the USER REQUIREMENTS DOCUMENT.
Requirements modelling	Combine requirements	Some requirements can maybe combine. This is depended on their semantics.
	Model each requirement	Each model comes in the SYSML REQUIREMENTS DIAGRAM.
	Draw relationships among requirements	Requirements can have relationships among each other or to parts in the system, these relationships can be drawn with a RELATIONSHIP (Holt & Perry, 2013).
Put requirements in a table		Besides the modelled requirements, requirements must also be stored in tables, therefore the REQUIREMENTS TABLE comes into place. Hereby the id, the name, the relationship with other requirements or system parts, how it is related and the type needs to be stored. If necessary, the table can also be extended with extra information (Santos Soares & Vrancken, 2008).
Requirements linking to use cases (not required)	Create SysML Use Case Diagram	SysML Use case diagrams can be helpful, it is not mandatory in the SysML Requirements diagramming technique, but it can helpful to trace the requirements to the use case (Santos Soares & Vrancken, 2008).
	Link use cases to requirements	As mentioned before, the use cases can be linked in an extra document, namely the SYSML USE CASE - REQUIREMENT DIAGRAM
Evaluation	Evaluate with stakeholders	Evaluation is a great part of SysML. Stakeholders can give feedback on the requirements in the SYSML DOCUMENT (Friedenthal, Moore, & Steiner, 2014).
	Break requirements into sub-requirements	If the stakeholders do not understand some requirements or found them too complex, requirements can be break down into sub-requirements (Friedenthal, Moore, & Steiner, 2014, p. 175).

Table 2: Activities of the Process Deliverable Diagram of figure 1

Concept	Description
USER REQUIREMENTS DOCUMENT	The USER REQUIREMENTS DOCUMENT stores all the requirements. This can be a database or a simple word file (Santos Soares & Vrancken, 2008).
SYSML DOCUMENT	All the diagrams are drawn in the SYSML DOCUMENT. This can be requirements but also other system parts (Holt & Perry, 2013).
SYSML DIAGRAM	A SYSML DIAGRAM is diagram of a system. It can contain blocks, requirements, graphic paths, interface blocks, ports and other system parts (Friedenthal, Moore, & Steiner, 2014, p. 133). It needs a stereoptype (such as requirement or block) and a name. In this PDD only the requirement stereotype is notated in the form of SYSML REQUIREMENT DIAGRAM.
SYSML REQUIREMENTS DIAGRAM	SYSML REQUIREMENTS DIAGRAM is an aggregation of a SYSML DIAGRAM. It contains all the requirement diagrams. Besides the properties stereotype and name, there are also an unique Id and a text description needed.
RELATIONSHIP	The different diagrams of SysML can have relationships with each other. This can be SATISFY, TRACE, REFINE and VERIFY (Holt & Perry, 2013).
REQUIREMENTS TABLE	The REQUIREMENTS TABLE provide further specifications of the requirements (Holt & Perry, 2013).
SYSML USE CASE DIAGRAM	SYSML USE CASE DIAGRAM shows the use cases (Santos Soares & Vrancken, 2008).
SYSML USE CASE - REQUIREMENT DIAGRAM	In the SYSML USE CASE - REQUIREMENT DIAGRAM the relationships between use cases and requirements can be drawn (Holt & Perry, 2013).

Table 3: Concepts of the Process Deliverable Diagram of figure 1

References

Friedenthal, S., Moore, A., & Steiner, R. (2014). *A practical guide to SysML: the systems modeling language.* Waltham, USA: Morgan Kaufmann.

Holt, J., & Perry, S. (2013). *SysML for systems engineering. 2nd Edition: A model-based approach.* Stevenage, UK: The Institution of Engineering and Technology.

Ringert, O., Rumpe, B., & Wortmann, A. (2012). A requirements modeling language for the component behavior of cyber physical robotics systems. In N. Seyff, & A. Koziolek, *Modelling and quality in requirements engineering* (pp. 133–146). Münster, Germany: Monsenstein und Vannerdat.

Salzer, H., & Levin, I. (2004). Atomic requirements in teaching logic control implementation. *International Journal of Engineering Education, 20*(1), 46-51.

Santos Soares, M. d., & Vrancken, J. (2007). Requirements specification and modeling through SysML. *ISIC. IEEE International Conference on Systems, Man and Cybernetics.*, 1735-1740.

Santos Soares, M. d., & Vrancken, J. (2008). Model-driven user requirements specification using SysML. *Journal of Software, 3*, 57-68.

Scanniello, G., Staron, M., Burden, H., & Heldal, R. (2014). On the effect of using SysML requirement diagrams to comprehend requirements: results from two controlled experiments. *18th International Conference on Evaluation and Assessment in Software Engineering*(1), 433-442.

Schneider, F., Naughton, H., & Berenbach, B. (2012). A modeling language to support early lifecycle requirements modeling for systems engineering. *Procedia Computer Science.*(8), 201-206.

Sung, C., & Kim, T. G. (2012). Collaborative modeling process for development of domain-specific discrete event simulation systems. *IEEE Transactions on Systems, Man, and Cybernetics.*(42), 532-546.

SysML Forum. (2015). *SysML FAQ: Frequently Asked Questions about SysML*. Opgehaald van SysML Forum: http://sysmlforum.com/sysml-faq

SysML Partners. (2015). *SysML Partners: Creators of SysML*. Opgehaald van SysML.org: http://sysml.org/sysml-partners/

Weerd, I. v., & Brinkkemper, S. (2007). Meta-modeling for situational analysis and design methods. In J. Ralyte, S. Brinkkemper, & B. Henderson-Sellers, *Handbook of research on modern systems analysis and design technologies and applications* (pp. 313-327). Berlin Heidelberg: Springer-Verlag.