

REQUIREMENTS ANALYSIS FOR DECISION-SUPPORT SYSTEM DESIGN: EVIDENCE FROM THE AUTOMOTIVE INDUSTRY

Nikolaos Madenas
Manufacturing and Materials Department
Cranfield University
Cranfield
Bedford, MK43 0AL, UK
n.madenas@cranfield.ac.uk

Ashutosh Tiwari
Manufacturing and Materials Department
Cranfield University
Cranfield
Bedford, MK43 0AL, UK
a.tiwari@cranfield.ac.uk

Christopher J. Turner
Manufacturing and Materials Department
Cranfield University
Cranfield
Bedford, MK43 0AL
c.j.turner@cranfield.ac.uk

Sophie Peachey
Axillium Research
Daventry
Daventry, NN11 0QB
sophie.peachey@axillium.com

ABSTRACT

The purpose of this paper is to outline the requirements analysis that was carried out to support the development of a system that allows engineers to view real-time data integrated from multiple silos such as Product Lifecycle Management (PLM) and Warranty systems, in a single and visual environment. The outcome of this study provides a clear understanding of how engineers working in different phases of the product-lifecycle could utilise such information to improve the decision making process and as a result design better products. This study uses data collected via in-depth semi-structured interviews and workshops that includes people working in various roles within the automotive sector. In order to demonstrate the applicability this approach, SysML diagrams are also provided.

Keywords: requirement analysis, decision support system, automotive.

1 INTRODUCTION

Decision-making within automotive manufacturers has become more complex due to the number of participants involved, the volume and the structure of information required, and the high number of interrelated functions that are impacted both upstream and downstream in the value chain. Such organisations aim for “right first time” decisions by allowing engineers to make better decisions through the utilisation of information. Throughout the product lifecycle, from concept definition and design to the final assembly and the in-service support, Original Equipment Manufacturers (OEM) work closely with the suppliers to bring products to markets earlier and meet customers’ expectations while reducing costs. During the last decade the evolution of information technology has improved the collaboration and integration of companies. Enterprise applications such as ERP, PDM and PLM are evolving in order to support the decision-making process within the supply chain. However the large amount of information exchanged within the supply chain throughout the whole product lifecycle has created the landscape of “Isolated Islands of Information” where information is locked in different repositories making it difficult to share (Mahdjoub et al. 2010). Although some of these systems allow the information exchange in a dynamic and direct way, organisations still need to work closely with suppliers to improve the decision making process and the entire supply chain performance (Fiala, 2005; Rungtusanatham et al., 2003). Companies are also restricted by their internal legacy systems, usually developed to support proprietary methods of working. While information exists inside organisations, the integration of those disparate sources is currently missing.

Rupnik et al. (2006) defines decision support systems (DSS) as interactive computer-based systems intended to help decision makers utilise data and models in order to identify problems, solve problems and make decisions. DSS are designed to support decision makers by providing the right information at the right level in order to minimise the decision efforts while increasing the decision efficiency. In order to do so, engineers need to be able to reduce the amount of time spent searching for information and access the right information from the multiple silos required in order to make better decisions.

The purpose of this work is to outline the requirements analysis that was carried out to support the development of a system that allows engineers both internally and in the supply chain to visualise data integrated from multiple silos. The methodology presented demonstrates the approach that was followed to ensure that user requirements have been fully specified and transformed into system functionality in order to meet customers' expectations. It ensures that the requirements can relate the business objectives with the use cases while ensuring that the system developed is capable of meeting the user requirements. SysML diagrams were also developed to document, validate and communicate user requirements with the stakeholders while developing the system. In this paper, a SysML use case and a SysML requirement example is presented to show the methodology applied in relation with the business scope of the system.

2 RELATED RESEARCH

Davis et al. (2007) defines requirements engineering (RE) as the discipline of determining, analysing, pruning, documenting, and validating the desires, needs and requirements of stakeholders for a system. RE is a critical activity for the success of every software system. It allows engineers to capture user requirements and transform them into system functionality. Moreover RE can be used as the basis for planning, validating and testing the system (Konrad and Gall, 2008). There are several techniques for capturing and analysing requirements. Jiang et al. (2008) evaluated the most common techniques used in each phase of the RE process and proposed a methodology called Methodology for Requirements Engineering Techniques Selection (MRETS) to support the selection process by enabling engineers to link the attributes of the project with the attributes of each RE technique. Konrad and Gall (2008) discussed the challenges associated with requirements engineering in large-scale projects. The lessons learnt presented in Konrad and Gall (2008) were instructive in supporting the development of the proposed methodology.

One of the key parts of any RE technique is the modelling technique used for the analysis, documentation and validation of the requirements captured. During the last decade UML and SysML diagrams have received significant attention as they provide a common language in order to better understand the developed systems. Chauldron et al. (2012) provided empirical evidence to demonstrate the effectiveness of UML modelling. Soarez et al. (2011) proposed a structured approach for capturing and communicating requirements specifications using standardised models such as SysML requirements diagrams, SysML tables and SysML use case diagrams. UML and SysML are widely used and contribute in transforming user requirements into system behaviour while at the same time offering a shared understanding and enabling more effective communication through a standardised language. However, Schulz (2001) stated that most UML-based methodologies are focus on the solution and define the system functions using use-case diagrams instead of addressing the business-oriented application requirements, proposing a requirements-based UML methodology to support the development method.

3 METHODOLOGY

Figure 1 shows the holistic approach that was followed to gather, analyse, manage and document requirements. Due to space limitations only the highlighted area of this methodology will be presented in this paper.

Requirements elicitation was conducted through semi-structured interviews and workshops with various members of the automotive industry such as OEMs and other actors in the supply chain. All the interviews arranged were face-to-face and included people from various roles such as Quality Managers, Quality Data Managers, Warranty Engineers etc. In several cases, interviews were conducted with a group of people in order to include technical people and engineers who are the actual users of the systems involved. Through the requirements elicitation approach a generic

requirements catalogue was created to ensure that the solution could be applied in different areas of the business. A low-level requirements catalogue was also created to capture requirements for specific use cases and form the basis for the development of a proof-of-concept system. The low-level requirements captured highlight the business requirements defined from the users without taking in consideration what the solution will look like. As Schulz, 2001 defines this catalogue answers the what? question rather than the how?. The requirements catalogues were then prioritised according to the MoSCoW ratings (Tudor and Walter, 2006 ; Hatton, 2007). Hatton, 2007 demonstrates different prioritisation methods and states that significant benefits can be achieved through early prioritisation. The MoSCoW method is probably the best choice as it is simple to perform and defines very accurate priorities.

- (M)ust have: Requirements that represent the core elements of the proof-of-concept system. Failure to meet these requirements will have an impact on the success of the project.
- (S)hould have: Represents high-priority requirements that would be an advantage to have in the early prototypes. Selection of these requirements depends on the project resources.
- (C)ould have: Features that are usually considered as “under development”. These requirements are desirable to have but they can be omitted if necessary.
- (W)on’t/(W)ant to have: These requirements are not unimportant but they are recognised as requirements that will not be part of the proof-of-concept system. These requirements are an important element in an incremental approach as they can be used during the next phases of the development.

A preliminary business-case analysis was conducted to support the development of a system. SysML diagrams such as requirements diagram, use case diagram, activity diagram and sequence diagrams were created based on the low-level requirements catalogue to transform the user requirements into system functionality. The SysML diagrams created were also used as a communication method between stakeholders and system developers to share a common understanding of the system; in addition SysML models were used as a validation point throughout the development of the system. A traceability matrix was created to ensure that test cases can be linked back to the use cases to ensure that the outcome of the system developed will always meet customers’ expectations. This tool helped in tracing the source requirements to their low/test level and from their low/test level back to source (Soonsongtanee and Limpiyakorn, 2010). Cost-benefit analysis will be carried out at the end of this project to measure the outcome and justify further investments required. Throughout the whole process, requirement management activities were carried out to manage and validate the outcome of this methodology.

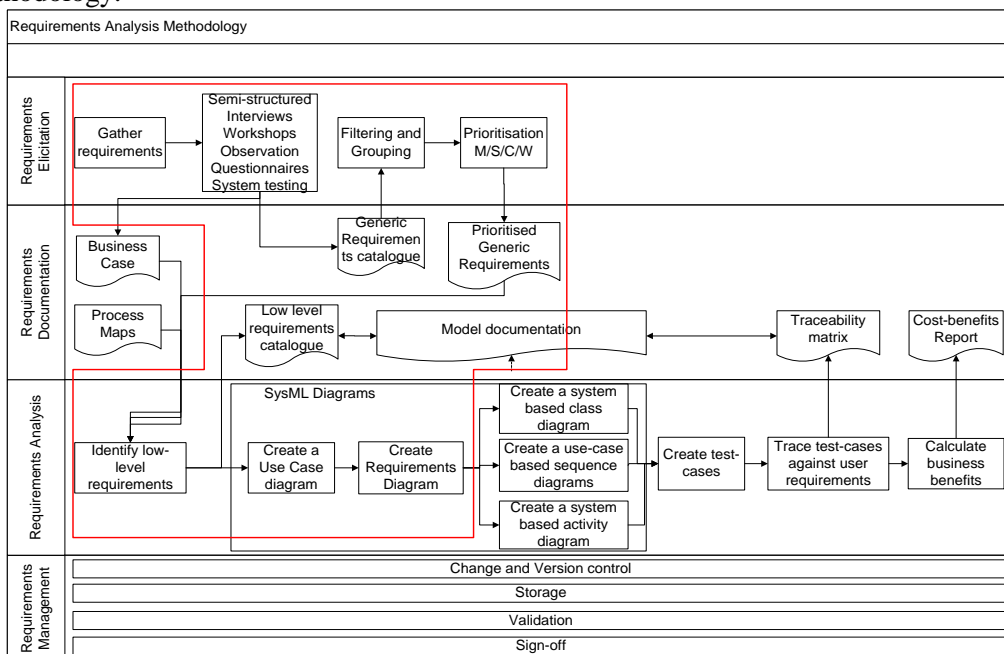


Figure 1: Requirements Analysis Methodology

The proposed methodology ensures that the system developed will satisfy the business requirements and as a result bring benefits to the business and its supply chain. The use case presented in the next section demonstrates one part of the system developed and shows the applicability of the above methodology followed.

4 RESULTS

One of the objectives of this integrated solution is to support the Quality department of the case study company to improve the time to detect and resolve an issue. As such this department is responsible for reducing the need for repairs and therefore cost of warranty claims, responding to customer feedback, and ensuring that the dealers are equipped sufficiently to repair the vehicles in the appropriate manner. In order to meet these objectives, this department has a close working relationship with suppliers by providing all the information required whilst monitoring their performance throughout the analysis of faults during manufacturing and in-service. There is a common requirement to reduce the time engineers, within the OEMs and the supply chain, spend on searching for information as opposed to resolving customer concerns. The application targets of this system are to support the decision-making process by allowing engineers to investigate data, more deeply and to further enrich the level of knowledge within less time. Moreover by gathering relevant data, engineers will be able to understand the customer issues and the dealer diagnosis and repair methods used in relation with other important supporting information. This process will allow them to improve the dealer diagnostic, repair, or warranty returns analysis procedures, close the information gaps and reduce the amount spent in warranty returns.

4.1 Establish Use Case Model

SysML use case diagrams provide a good graphical representation of the functional requirements of a system as they illustrate the different use cases, the actors and their interactions with the system. Actors may include users, systems or any other entity that has a direct or indirect communication with the system (OMG, 2002). A SysML use case diagram is shown in Figure 2. Within the boundaries of the diagram each use case is associated with a unique ID and as a result low level requirements and test cases documented later in the process can be traced back to the source use case. Outside the boundaries the diagram demonstrates two types of actors. Firstly it shows the actual users of the system such as: 1) Quality managers who are responsible for monitoring the performance of a supplier. The system will enable these actors to overview the quality process by accessing all relevant information in a less time-consuming way. 2) Warranty reduction engineers who are responsible for delivering the quality projects raised by analysing different types of data such as faults recoded by customers. They are responsible for assigning liabilities on the faults found and ensuring that the dealers are equipped sufficiently to repair the vehicles in the appropriate manner. 3) Supply Chain engineers have similar tasks within their own organisations and they require the capability to integrate data coming from dealers as well as other types of data provided by the OEMs in relation with their own internal systems. Secondly SysML use case diagrams show the different types of system that need to be accessed: 1) PLM systems that hold the product data such as CAD data and 2) Data Silos, usually developed within organisations to hold all the other meta-data required to support the decision process.

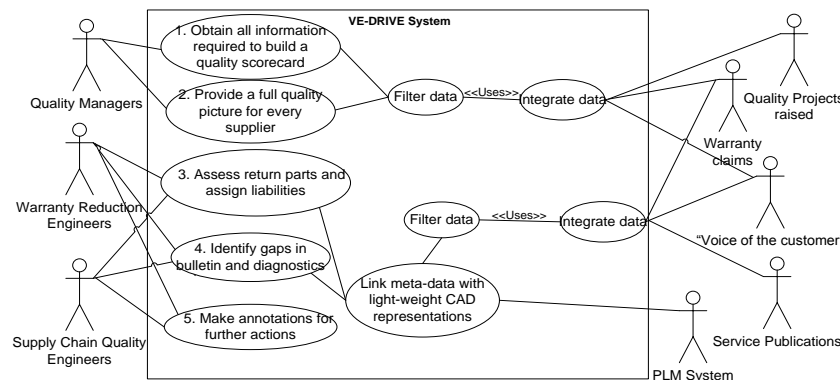


Figure 2: SysML Use Case Diagram

4.2 Establish the Requirements Model

The SysML requirements diagram is a standardised way of decomposing high-level functional requirements from the use case diagram into low level requirements while showing the relationships between them. It also represents the relationships drawn from text-based requirements with additional modelling elements (OMG, 2002). The basic template for a SysML requirement diagram includes the ID, the name and the text. The diagram shown in Figure 3 uses the approach proposed by Soares et al. (2011) and includes the MoSCoW ratings prioritisation for each one of the low-level requirements. Developing a system that is capable of fulfilling all the user requirements captured is not always feasible. Prioritisation will support the selection of the “Must” and “Should” requirements that will be used in developing a proof of concept system. Having the prioritisation within the SysML requirement diagram will allow the analyst to calculate the risk of a requirement that has not been met as well as determine its impact on the whole system based on the relationships defined. The diagram shown expands on two of the use cases presented in the previous use case diagram. For simplification reasons only a sample of priorities are included in the diagram below.

During the requirements elicitation process it was highlighted that engineers spend a significant amount of time searching for information and manually accessing disparate silos. As warranty engineers work in a later phase of the product lifecycle accessing information held in PDM and PLM systems could be a challenge. In order to improve the decision making process it was highlighted that it is critical to support the engineers on gathering relevant information in a single visual environment. The requirements analysis showed that it is critical for engineers to view different clusters of data in a single environment in order to build a full quality picture for a part, sub-system or whole system. Data such as the “voice of the customer” (surveys, feedback from forums, breakdown and recovery reports); service and repair publications (manuals, bulletins and diagnostics); product data (CAD data, specifications and tolerances) and warranty claims and faults recorded from the dealers are critical in order to build a full quality picture. Although every use case in Figure 2 is linked with the SysML requirements diagram for the sake of simplification only elements of use case number three and four will be decomposed in Figure 3.

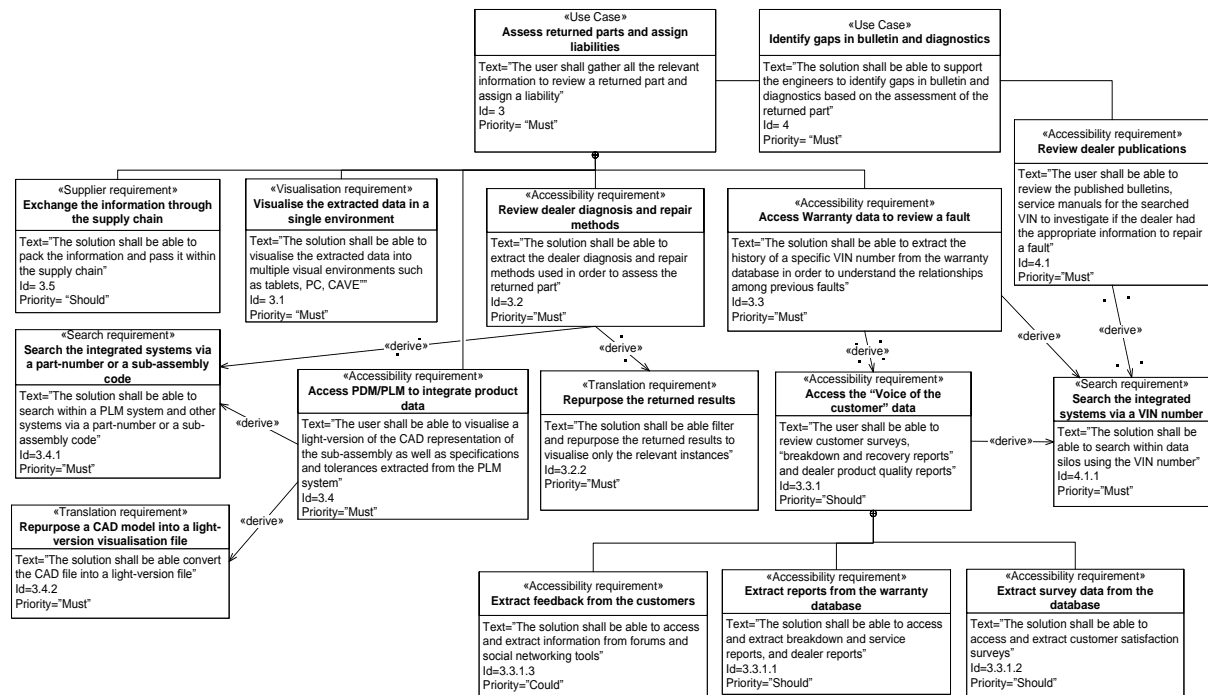


Figure 3: SysML Requirements Diagram for the Quality Use Case

5 CONCLUSIONS

The requirements gathered and described through SysML modelling diagrams will drive the development of a novel system that allows engineers to integrate and visualise, in a single environment product data from PLM systems in relation with other meta-data such as warranty data leading to an improvement in the decision making process. Due to space limitations, the transformation of business scope requirements into system functionality through SysML diagrams has not been presented in this paper. The methodology presented shows the holistic approach that needs to be followed in order to develop a proof-of-concept system that validates the requirements captured. Though this paper concentrates on the quality department use case requirements the wider purpose of the system is to support multiple engineers working in different phases of the product lifecycle.

ACKNOWLEDGMENTS

This research was conducted as part of 'VE-DRIVE'; a two year collaborative R&D project which started in December 2011. The authors would like to thank the project sponsors. This research is co-funded by the Technology Strategy Board, the UK's innovation agency and the project partners.

REFERENCES

- Chaudron, M.R.V., W. Heijstek and A. Nugroho. 2012. How Effective is UML Modeling? - An Empirical Perspective on Costs and Benefits. *Journal of Software and Systems Modeling* 11:571–580.
- Davis, A. M., A. M. Hickey, Ó. D. Tubio, N. J. Juzgado and A. M. Moreno. 2007. A Quantitative Assessment of Requirements Engineering Publications - 1963-2006. *Requirements Engineering: Foundation for Software Quality Lecture Notes in Computer Science* 4542:129-143.
- Fiala, P. 2005. Information sharing in supply chains. *Omega - The International Journal of Management Science* 33:419-423.
- Hatton S. 2007. Early prioritisation of goals. In *Proceedings of the 2007 conference on Advances in conceptual modeling: foundations and applications*, 235-244, Auckland, 5-9 November.
- Jiang, L., A. Eberlein, B. H. Far and M. Mousavi. 2008. A methodology for the selection of requirements engineering techniques. *Software and Systems Modeling* 7: 303-328.
- Konrad S. and M. Gall. 2008. Requirements Engineering in the Development of Large-Scale Systems. In *16th IEEE International Requirements Engineering Conference*, 217-222, Catalunya, 8-12 September.
- Mahdjoub, M., D. Monticolo, S. Gomes and J. Sagot. 2010. A collaborative Design for Usability approach supported by Virtual Reality and a Multi-Agent System embedded in a PLM environment. *Computer Aided Design* 42:402-413.
- OMG. 2012. OMG Systems Modeling Language v.1.3. Available via <http://www.omg.org/spec/SysML/1.3> [accessed December 5,2012]
- Rungtusanatham, M., F. Salvador, C. Forza and T.Y. Choi. 2003. Supply-chain linkages and operational performance: A resource-based-view perspective. *International Journal of Operations & Production Management* 23:1084-1099.
- Rupnik, R., M. Kukar, M. Bajec, M. Kripser. 2006. DMDSS: Data Mining Based Decision Support System to Integrate Data Mining and Decision Support. In *Int. Conf. Information Technology Interfaces*, Cavtat, 19-22 June.
- Schulz J.D. 2001. Requirements-Based UML. In *Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS39)*, 307-316. Washington: IEEE Computer Society, 29 July – 3 August .
- Soares M.D.S., J. Vrancken and A. Verbraeck. 2011. User requirements modeling and analysis of software-intensive systems. *Journal of Systems and Software*. 84:328-339.
- Soonsongtanee S. and Y. Limpiyakorn. 2010. Enhancement of Requirements Traceability with State Diagrams. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, Chengdu, 16-18 April.
- Tudor, D. and G.A. Walter. 2006. Using an agile approach in a large, traditional organization. In *Agile Conference*, 373-380. Minneapolis, 23-28 July.