SysML for the Analysis of Product-Service Systems Requirements

C. Durugbo, W. Hutabarat, A. Tiwari, J.R. Alcock,
School of Applied Sciences, Cranfield University, Bedfordshire, MK43 0AL, United Kingdom
{c.durugbo, w.hutabarat, a.tiwari, j.r.alcock} @cranfield.ac.uk

Abstract

In this paper, the Systems Modelling Language (SysML) is introduced as a technique for the design of a product-service system (PSS). In examples of the requirements diagram in SysML, this paper shows that: (1) a PSS can be analysed by the SysML technique, and (2) the SysML can model the requirements of a PSS in orientations according to product, use or result. Recommendations based on the example will then be used to suggest improvements in the design of a PSS.

Keywords

Product-Service Systems, SysML, Conceptual design, Requirements, Traceability

1 INTRODUCTION

The design of a product-service system (PSS) involves three main stages: analysis, concept and execution [1]. The analysis phase captures specific customer needs and *the requirements to fulfil them*. The concept phase identifies structural, physical and behavioural characteristics for the PSS. The execution phase involves planning, scheduling, prototyping, simulation and any other design process required for PSS realisation.

For systems design in general, the analysis and concept stages are areas of design that lacks efficient support [2]. The problem centres on how to efficiently develop methods to deal with difficulties in converting customer needs into functions/use cases and to realise a clear vision of the system to ease system integration and communication among team members.

Recently, a modelling technique known as the Systems Modelling Language (SysML) has been proposed to support the analysis and concept phase of systems design. SysML is based on the Unified Modelling Language (UML) and was developed by the Object Management Group (OMG) consortium and the International Council on Systems Engineering (INCOSE) [3]. The SysML approach has been applied in areas such as requirement modelling, describing physical systems connectivity and modelling the frameworks of organisations [4].

The focus of this paper is to show an example of the SysML requirement diagram for the conceptual design of a PSS. It seeks to contribute to PSS research by assessing the benefits of the technique for the analysis phase of PSS design.

This paper begins with an overview of product-service systems. The SysML approach will then be introduced as technique for the analysis phase of systems design. The focus of the technique is to support systems designers in identifying the requirements to fulfil customer needs. Recommendations based on worked examples of the SysML requirements diagram for a PSS will then be used to suggest improvements in the design of a PSS.

2 PRODUCT-SERVICE SYSTEMS

According to the United Nations, a Product-Service System (PSS) is 'a competitive system of products, services, supporting networks and infrastructure' [5]. It is an approach to production that shifts focus from 'product thinking' to 'systems thinking' [6, 7].

In principle, a PSS involves intermixing product and service components for added customer value [6]. Added value for customers is realised by delivery systems that offer value propositions that utilise PSS granularity i.e. the level to which a PSS is broken down into subsystems and components to aid flexibility and customisation.

2.1 Delivery systems for a PSS

Delivery systems for a PSS are realised in various orientations according to product, use or result [7, 8] as shown in Figure 1.

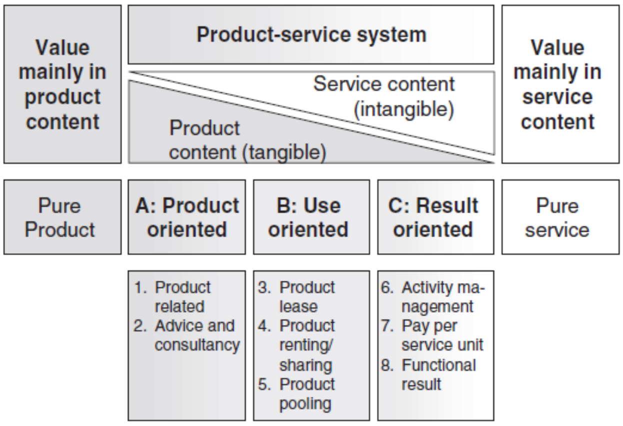

Figure 1: Product-service system configurations [8]

In the *product-oriented* approaches, the provider offers the sale of products and also opens service channels for additional product related services such as repairs, upgrades as well as professional advice / consultancy [8].

For the *use-oriented* approach, the provider maintains the rights to a product. This product is made available for use in a service environment via services such as product leasing, sharing, pooling and renting.

The result-oriented approach involves a provider delivering service content independent of product choice. This approach is based on pay-per-service-unit schemes that delivers functional results, and on activity management approaches.

2.2 Modelling for PSS delivery

Various models for product-service systems have been proposed, but as yet there is no final definition of a standard model. This is because a company seeking to deliver a PSS needs to consider business requirements such as: *life-cycle of a product* from its engineering through to its decommissioning (recycling and disposal), *close knitting* of products with services and services with products, and *establishing links* and networks (with customers and other manufacturers) to aid PSS delivery.

Function-oriented modelling, object-oriented modelling, conceptual modelling and service modelling are some approaches to modelling the business requirements for a PSS [9].

Given the variety of possible contexts, achieving a comprehensive standard model can be a very difficult goal [10]. It is likely that bespoke models need to be developed for each PSS context [11].

It then becomes worthwhile to consider contributions from the field of systems engineering as a starting point for any model being developed. Systems engineering has always had to deal with complex, dynamic, and multidisciplinary systems, so that a systems engineering tool such as SysML may be useful at some level in any PSS context, possibly as an initial PSS model. The SysML approach benefits system engineers because the approach identifies the minimal set of the popular UML technique required to specify, analyse, design, verify and validate a wide range of systems (such as PSSs) [4].

3 SYSML AND THE REQUIREMENTS DIAGRAM

3.1 Overview

SysML is a general-purpose graphical modelling language for representing systems that may include combinations of products, data, people, facilities, and natural objects [12]. In software engineering, the use of modelling is increasingly being adopted through the Unified Modelling Language (UML). SysML was developed by starting with a subset of UML, which was then extended with some new diagrams. SysML consists of nine diagrams (Figure 2) that are classified into:

1. Structure diagram, consisting of Block Definition, Internal Block, and Package diagrams, which represent structural elements and their organisation and interconnection; and parametric diagram, which represents real-world constraints on property values.
2. Behaviour diagram, consisting of Activity, Sequence, State Machine, and Use Case diagrams. These diagrams represent behaviour and functionality of entities within the model.
3. Requirement diagram, which represents individual requirements and their relationship with other requirements, design elements, and test cases to support requirements traceability [12].

Each diagram utilises a different viewpoint, with different abstraction and notation; each diagram thus represents partial views of the complete model. The complete set of diagrams provides a complete, coherent description of the model of the system being developed. This model can then be used to support analysis, specification, design, and verification of the system.

The system model is generally created using a modelling tool and stored in an on-line model repository. This facilitates collaboration and provision of useful capabilities such as automatic updating of other diagrams when details are changed in one diagram. For example, because requirement diagram dependencies to constructs in other diagrams are tracked automatically, the design team will be alerted when a design change compromises a requirement.

In SysML, the requirements diagram documents what the system must accomplish and provides many useful features for a PSS design team. The rest of this paper will focus on the requirements diagram and how it can be applied in a PSS context.

The reader is referred to reference [2, 3, 4, 12] for the detailed explanation and description of the SysML approach, diagrams and the arguments which underlie this technique.

3.2 The Requirements Diagram

By introducing the requirement diagram, SysML offers an opportunity to:

1. Trace initial requirements during the design and development of a system,
2. Capture design rationale in terms of how design decisions were made, and
3. Generate test cases during the verification stage of a design process [4].

It is for this reason that the requirement diagram is vital to the SysML technique.

Requirement diagrams contain three main diagrammatic

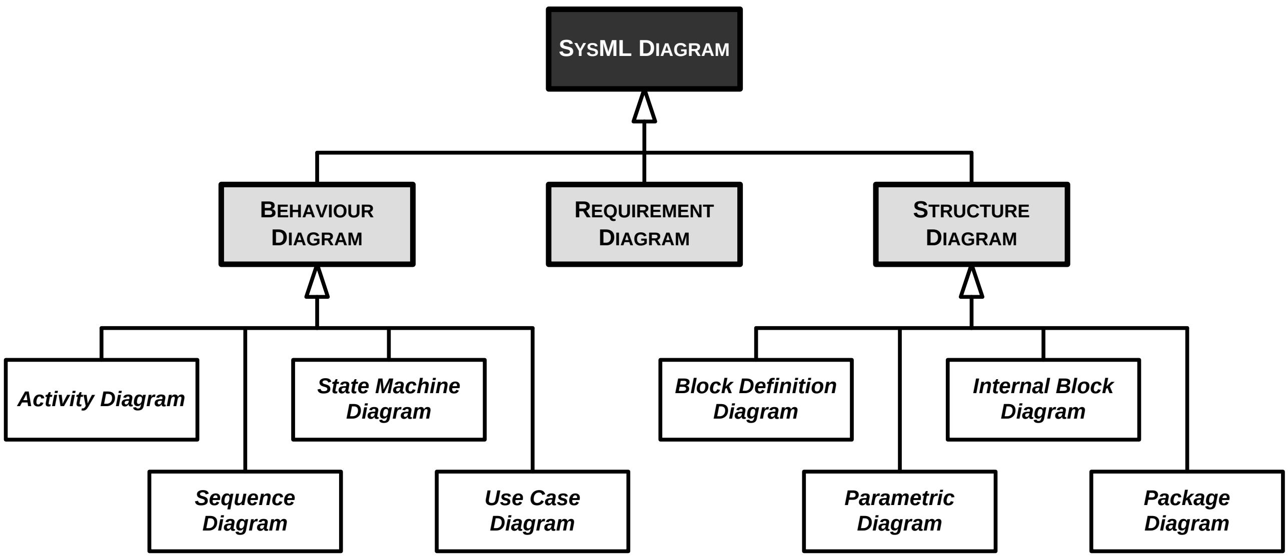

Figure 2: Systems modelling language (SysML) diagram types

modelling components: *boxes* to represent requirements, *rationale* to represent design rationale and to track changes to requirements and *arrows* to represent relationships between requirements [3, 13] as shown in Figure 3.

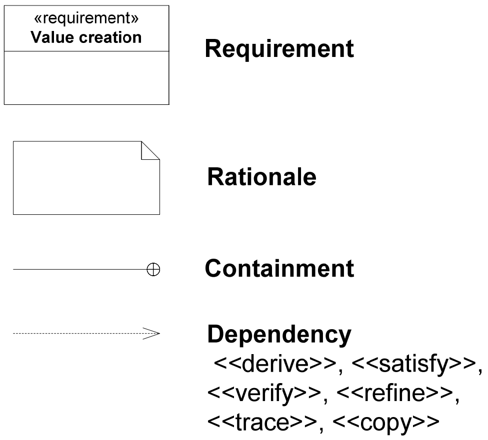

Figure 3: Requirement diagram symbols

The term ‘requirement’ is used to specify ‘a capability or condition that must (or should) be satisfied’ [3]. In real life systems, a standard requirement is also defined by a unique identifier, textual description and extra explanations such as verification status and requirements consistency checks.

The rationale construct is a general-purpose construct that can be used by a systems designer to provide explanations for decisions made during system analysis and conceptualisation. For example, a PSS delivery system requirement to ‘reduce waste generation’ could be supported by a design rationale containing textual descriptions such as ‘new environmental regulations’ or ‘motivated by a customer survey conducted on …’.

Within the context of SysML, a requirement may be related to several sub-requirements as defined by relationships of containment or dependencies [3].

The containment relationship is used to show how a complex requirement can be decomposed into a set of child requirements.

SysML also recognises five main dependencies between requirements and between other requirement diagram elements: derive, satisfy, verify, refine, trace and copy. The <<derive>> relationship defines the link between derived requirements and source requirements whereas the <<satisfy>> relationship shows links for satisfying requirements. The <<verify>> relationship shows how test cases are used to verify a requirement. The <<refine>> relationship is applied to explain how one element refines another element whereas the <<trace>> relationship can be used to describe a general-purpose link between requirement diagram elements. The <<copy>> relationship creates a copy of a requirement.

4 PRODUCT-SERVICE SYSTEM REQUIREMENTS

William [13] captured and discussed cases of business-to-customer (B2C) type product-service systems in the automotive industry. A B2C automotive PSS sells (or delivers) cars, vans, trucks, buses and other forms of commercial vehicles to end consumers. The B2C type business lies at the spectrum of a supply chain where a business delivers products and services to end users or consumers. A barbershop and a laundrette are common examples of B2C for end consumers.

This section makes use of the PSS orientations identified in [13] to demonstrate the possible use of the requirements diagram for a PSS. The first sub-section presents the requirements for a PSS in the automobile industry based on product-orientation whereas the second and third sub-sections present the requirements for a PSS in the automobile industry based on use-orientation and result-orientation respectively.

4.1 Product-oriented requirements

In Figure 4, an example of a requirement diagram has been produced for an automotive company that delivers a product-orientated PSS for product related services and professional advice and consultancy.

In this configuration for a PSS, a company sells and transfers ownership of a product to a customer. When ownership of a product is transferred from the company to the customer, the responsibility of dealing with the risks and costs associated with after sales is also transferred. These risks and costs include product failure, maintenance costs and availability of spares. In the product-orientated PSS the business offers product-related services as well as professional advice and consultancy to alleviate some of these risk and costs. These services are also provided to create more business.

Delivery system requirement

The clear vision or main requirement of the delivery system in this example is customer mobility. This main requirement is satisfied in requirements for affordable customer solutions (customer requirement) and profitable business operations (manufacturer requirement). It is also determined (i.e. refined) by the choice of automobile that delivers the product-orientated PSS (provision requirement). Consequently, the automotive company will need to deliver a selection of automobiles to meet the main requirement of customer mobility.

For this example, automotive (i.e. the product) functions are contained in the sub-requirements for reliability and performance.

Services offered in this orientation as shown in Figure 4, are required to satisfy customer and manufacturer requirements of affordability and profitability. To satisfy the requirement for affordable customer solutions, the manufacturer offers cheap repairs and maintenance for automobiles whereas for profitable operations, the manufacturer provides automobile data and advice to customers. In addition, the manufacturer collects services data about automobile use and life to refine the services offered to customers.

Value proposition

For the example of a product-orientated PSS in Figure 4, the main requirements of the value propositions for the automotive company include: initial warranty for automobile, zero interest financing and take back scheme at automobile end-of-life.

The initial warranty is a scheme that the manufacturer offers to the customer to guarantee repairs, maintenance and replacements service for a fixed duration. In other words, the automotive company closely links automotive services to an offered automotive to deliver affordable customer solutions. The terms and provisions of the warranty are derived from customer contracts for service provision.

Zero percent interest is an initiative that the automotive company offers to customers to promote sales. It offers customers a fixed period during which the manufacturer expects no payment for the purchased automobile (and the closely linked services).

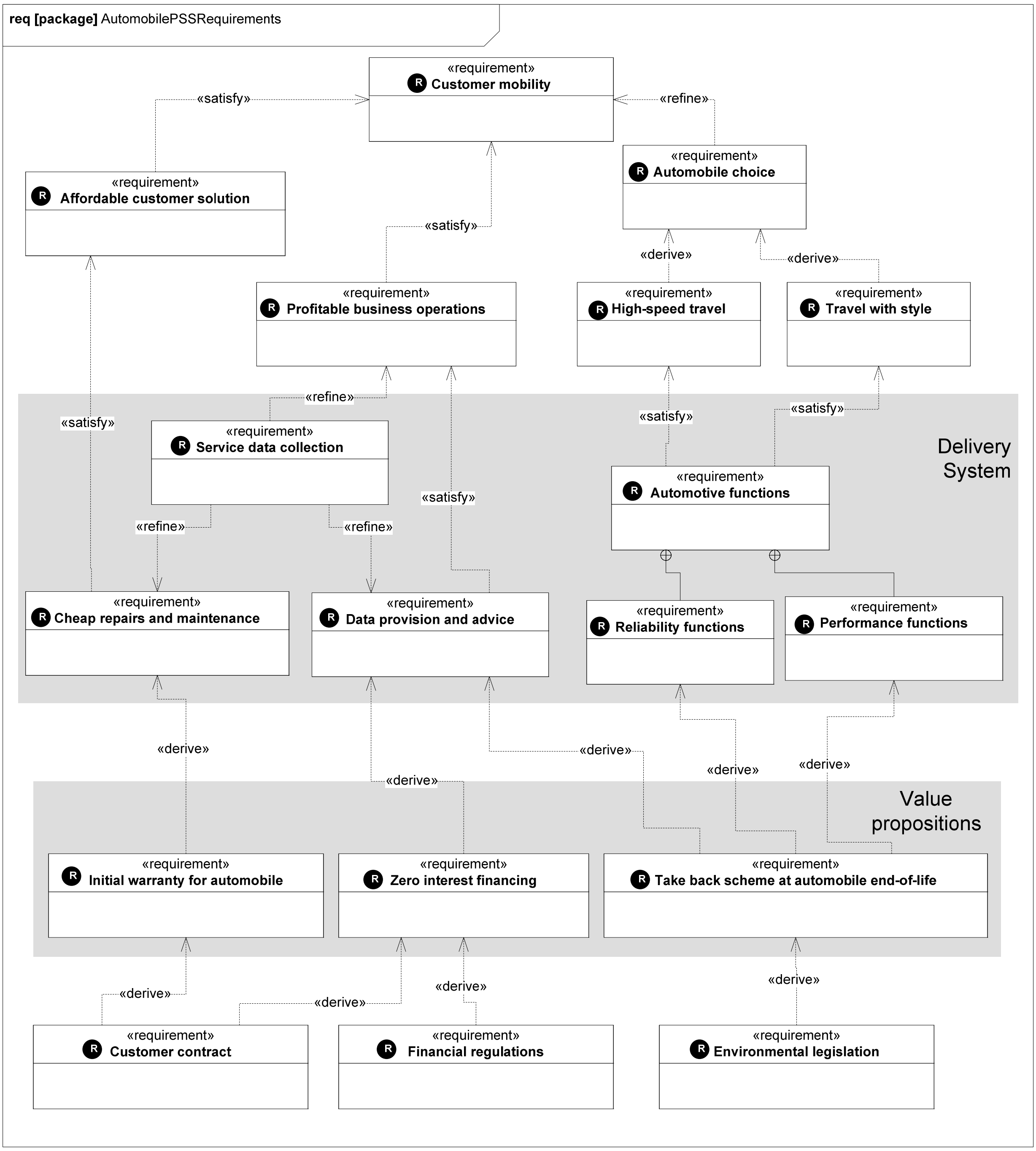

Figure 4: Requirements diagram for a product -oriented product-service system in the automotive industry

Following the completion of the period, the manufacturer may then request full payment, instalment payments or extend the period for zero interest financing. The stipulations of the zero interest financing are determined by the customer contract and financial regulations defined by institutions such as the Financial Services Authority (FSA) in the United Kingdom, the Securities and Exchange Commission (SEC) in the United states and Finansinspektionen in Sweden.

Take back schemes are motivated by environmental legislation within the automotive industry that encourage companies to improve the environmental impact of actors associated with automotive lifecycles [13].

These legislations require companies to be more active and responsible for taking back automotives at the end-of –life stage (i.e. when automotives are no longer functional or required by the customer). The take back scheme may include assuring customers that end-of-life automotives will be taken back and recycled. It may also involve recovering or salvaging materials and components from end-of-life vehicles for use as spare parts.

To derive the take back schemes, the company must consult appropriate environmental legislation to capture requirements relating to the laws that govern aspects such as: material recovery from vehicles, automotive disassembly, recycling and garages.

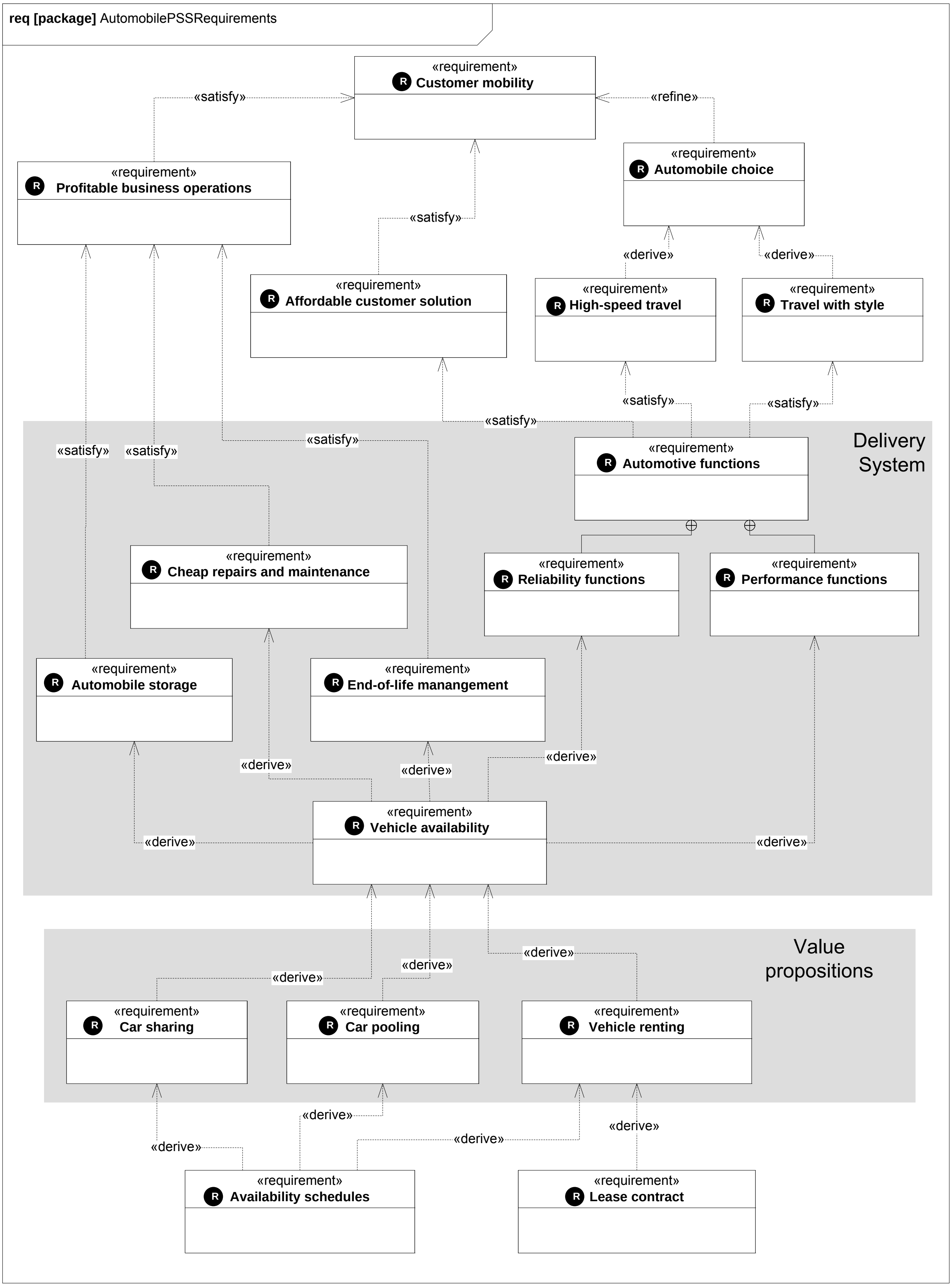

Figure 5: Requirements diagram for a use-oriented product-service system in the automotive industry

4.2 Use-oriented requirements

A second example of the requirements diagram is highlighted in Figure 5. The figure shows how the requirements for a use-orientated PSS can be captured by SysML.

For this configuration of a PSS, the automotive company in the B2C automotive business retains ownership of vehicles. These vehicles are then made available to customers for leasing, sharing or pooling. As a result, the automotive company assumes the risk and cost of vehicle maintenance, repairs, upgrades and replacement.

Delivery system requirement

As in the product-oriented PSS, the clear vision or main requirement of the delivery system in this example is customer mobility. Since the customer is given the option to choose the type of vehicle for mobility, the company must also deliver a selection of automobiles. The customer requirement for affordable customer solutions and the company requirement for profitable business operations are maintained.

The main service requirement offered by the automotive company is vehicle availability. This requirement is dependent on how the automobile is stored to enable access by customers, repairs and maintenance for conditioning the vehicle and end-of-life management for recovering spares from vehicles.

Value propositions

The main value propositions for the example of a use-orientated PSS (Figure 5) offered by the automotive company include: car sharing, car pooling and vehicle renting.

For car sharing, the customer is required to make a regular payment for the use of a vehicle that is shared by multiple customers at different times. The same condition applies for car pooling but customers are required to access the vehicle concurrently.

For vehicle renting, multiple vehicles are made available by the automotive company for customers to lease based on a regular subscription. In car sharing and car pooling schemes, a lease contract and availability schedule is required for defining how customers can use the vehicle availability service requirement. For the vehicle renting, the main requirement is the lease contract that defines the terms of vehicle use.

4.3 Result-oriented requirements

Figure 6 shows a third example of the requirements diagram for a result-orientated PSS. In this configuration for a PSS, the automotive company in the B2C automotive business also retains ownership of vehicles. Similarly, the automotive company assumes the risk and cost of vehicle maintenance, repairs, upgrades and replacement. Unlike the use-oriented PSS, the result-oriented delivers functionality independent of vehicle type, pay-per-travel schemes and outsources elements of business operations such as maintenance and service data collection.

Delivery system requirement

As in the previous examples, the customer, manufacturer and provision requirements are reused. For the result-oriented PSS in this example, the clear vision or main requirement of the delivery system for customer mobility is maintained. Similarly, customer requirement for affordable customer solutions and the company requirement for profitable business operations are maintained. A range of automobiles as suggested in [12] is also required to offer options for customer mobility. For instance, automotive companies may offer human-powered/electric vehicle, taxis or environmentally friendly car-ferry designs for variety in customer mobility.

Value propositions

In the example of the result-oriented PSS offered by the automotive company, three main value propositions are captured as requirements: pay per time, pay per output and mobility system.

In the pay per time and pay per travel schemes, a customer pays a flat fee for travel to destinations covered by the automotive company. For the pay per time scheme, the flat fee depends on travel duration (such as hourly, daily or weekly) whereas for pay per travel the customer pays for each trip or functionality provided by the vehicle. The mobility system is an initiative that allows customers such as locals and visitors to select travel means best suited to their needs.

5 DISCUSSION

As shown by the examples in this paper, for a PSS, with a clear vision of a delivery system that offers customer value, a requirements diagram can be generated. Engineers, designers, managers and other key company personnel may collaborate for the realisation of this requirements diagram.

For the analysis and conceptualisation of a PSS, a company captures customer, manufacturer and provision (product and service) requirements. These are then implemented to offer value propositions based on enhancing a product, making a product available or making the function of the product available to the customer.

5.1 SysML applications for product-service systems

Applying a systems modelling tool such as SysML for the design of a PSS can be beneficial in terms developing models to aid system collaborators in transforming customer requirements into functions and components. This is done to support systems development and to promote a common and clear vision of a system among system collaborators. The requirements diagram, an integral part of SysML can be used to models the requirements of a PSS as demonstrated by the examples in Section 4.

Furthermore, a major consideration for collaborators during PSS design involves designing the links and networks (with customers and other manufacturers) to aid PSS delivery, and the close knitting of services with products [11]. As a result, companies seeking to apply product-service systems will need to carefully analyse a PSS to capture key requirements to aid communication among PSS collaborators during development and delivery of a PSS. The requirements diagram offers a standardised notation that could be used to standardise communication of a PSS across departments or companies that collaborate to deliver a PSS.

In addition, the constructs of the requirements diagram are measures that SysML introduces to maintain traceability of requirements during systems design and development, to understand decisions made during design and to guide systems testing by creating test cases.

The requirements diagram is also a diagrammatic tool for maintaining traceability during the reconfiguration of a PSS. This is because it offers constructs that could enable PSS designers to explain the rationale for their decisions during the design process. Individuals are accountable for actions and can be consulted or contacted for further clarification on decisions.

5.2 Limitations of SysML

Developed recently, SysML is a new approach to systems modelling that is currently supported by relatively few tools and has not been accepted widely within the research community [14]. The approach also requires modifications and updates to enhance the use of the contained diagrams in industry. SysML like UML is developed as a notation for systems – not a methodology. Consequently, the use of SysML requires the study of the system to be modelled with the intention of: (1) understanding the characteristics of the system to be modelled, and (2) deciding on the level at which to model the system.

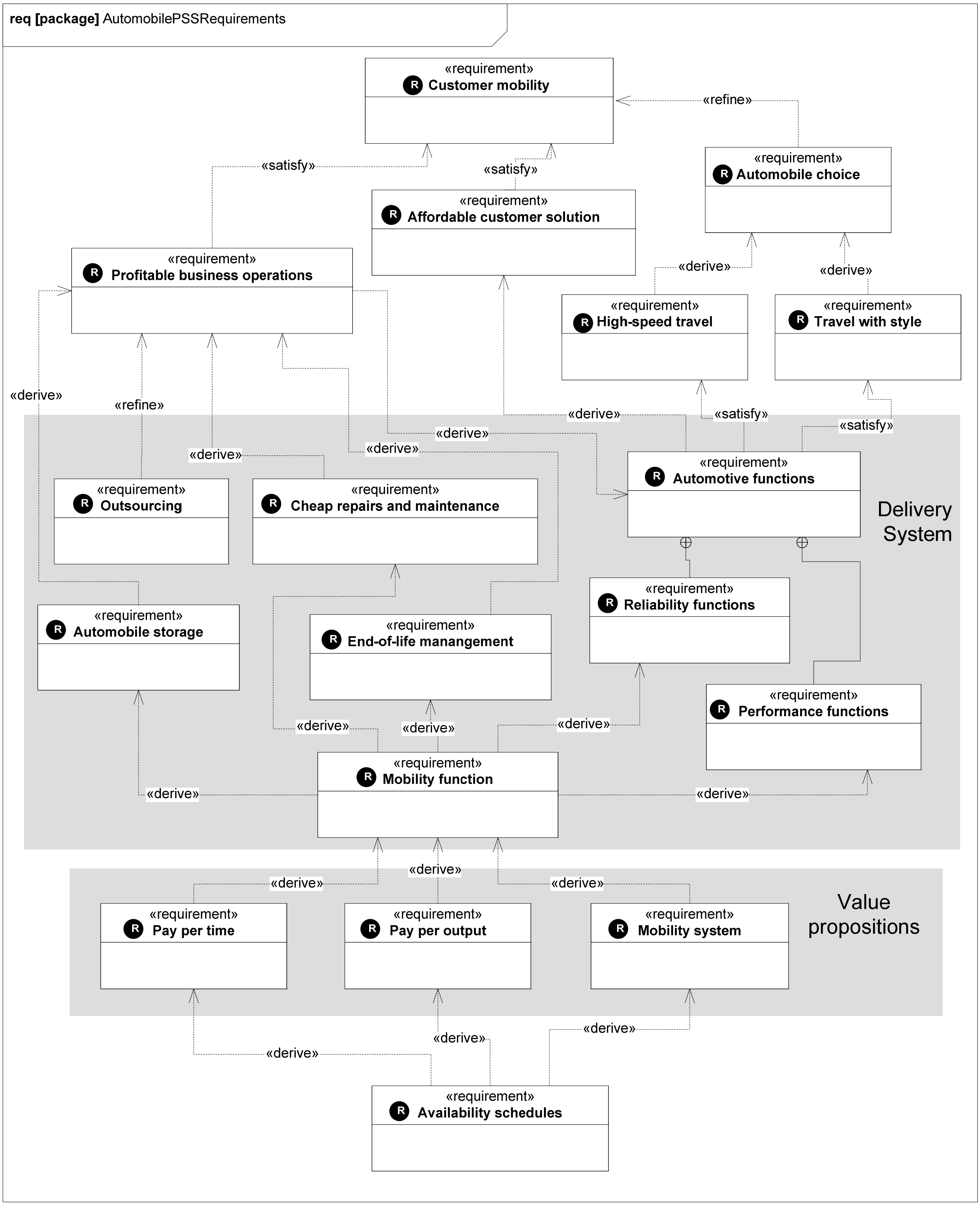

Figure 6: Requirements diagram for a result-oriented product-service system in the automotive industry

Nonetheless, SysML provides *visual representations* that are useful for modelling systems and *semantics* that show different aspects of a system such as structure, behaviour and requirements.

6 SUMMARY

A product-service system (PSS) is delivered as a system made up of intermixed product and service components for added customer value. The design for PSS delivery involves analysis, concept and implementation phases for realising value propositions based on product-, use- and result- orientations. As shown by the examples in this paper, the requirements diagram can be used to analyse the three orientations of a PSS. The requirements diagram is an integral part of SysML that can be used to trace initial requirements during the design and development of a system and capture design rationale in terms of how design decisions were made. SysML for PSS design provides standardised language for design, traceability of requirements, and the support for communication among PSS collaborators during development and delivery of a PSS.

7 ACKNOWLEDGMENTS

The authors would like to extend their sincere thanks to the EPSRC, for its support via the Cranfield IMRC, towards the work carried out in the preparation of this paper.

8 REFERENCES

[1] Aurich JC, Fuchs C and DeVries MF, 2004, An Approach to Life Cycle Oriented Technical Service Design, CIRP Annals - Manufacturing Technology, 53(1), pp. 151-154.

[2] Balmelli L, 2007, An Overview of the Systems Modeling Language for Products and Systems Development, Journal of Object Technology, 6(6), pp. 149-177

[3] Object Management Group, 2009, OMG Systems Modeling Language (OMG SysML™) Version 1.1, online at: http://www.omg.org/spec/SysML/1.1, accessed 15 October, 2008

[4] Holt J and Perry S, 2006, SysML: Describing the system, Information Professional, 3 (4), pp. 35-37.

[5] United Nations Environment Programme, 2001, 'The Role of Product Service Systems in a Sustainable Society', online at:

http://www.uneptie.org/pc/sustain/reports/pss/pss-brochure-final.pdf, accessed 19 October, 2008

[6] Durugbo C., Tiwari A., Alcock J.R., 2009, An Info-dynamic Engine Approach to Improving the Efficiency of Information Flow in a Product-Service System in Proceedings of the 1st CIRP IPS2 Conference, pp. 107-112

[7] Baines T, Lightfoot HW, Evans S, Neely A, Greenough R, Peppard J, Roy R, Shehab E, Braganza A, Tiwari A, Alcock JR, Angus JP, Bastl M, Cousens A, Irving P, Johnson M, Kingston J, Lockett H, Martinez V, Michele P, Tranfield D, Walton IM, & Wilson H, 2007, The state-of-the art in Product Service Systems, Proceedings of the I MECH E Part B Journal of Engineering Manufacture, vol. 221, no. 10, pp. 1543-1552

[8] Tukker A, 2004, Eight types of product-service system: eight ways to sustainability? Experiences from SusProNet, Business Strategy and the Environment, vol. 13, no. 4, pp. 246-260

[9] Durugbo C., Tiwari A., Alcock J.R., 2010, A Review of Information Flow Diagrammatic Models for Product-Service Systems. International Journal of Advanced Manufacturing Technology (*Accepted for Publication*).

[10] Morelli N, 2006, Developing new product service systems (PSS): methodologies and operational tools, Journal of Cleaner Production, vol. 14, issue 17, pp. 1495-1501.

[11] Becker J, Beverungen DF and Knackstedt R, 2009, The challenge of conceptual modeling for product-service systems: status-quo and perspectives for reference models and modeling languages, Information Systems and e-Business Management, pp. 1-34.

[12] Friedenthal S, Moore A and Steiner R, 2008, Practical Guide to SysML: The Systems Modeling Language, Morgan Kaufmann.

[13] Williams A, 2007, Product service systems in the automobile industry: contribution to system innovation?, Journal of Cleaner Production, 15 (11-12), pp. 1093-1103.

[14] Vanderperren Y and Dehaene W, 2005, SysML and Systems Engineering Applied to UML-Based SoC Design in Proceedings of the 2nd UML-SoC Workshop at 42nd DAC, Anaheim (CA), USA.