Advanced Engineering Informatics 28 (2014) 479–498

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

A representation scheme for digital product service system definitions

Alison McKay *, Saikat Kundu [1]

School of Mechanical Engineering, University of Leeds, Leeds LS2 9JT, UK

ARTICLE INFO

Article history:
Received 21 August 2012
Received in revised form 9 July 2014
Accepted 11 July 2014
Available online 14 August 2014

Keywords:
Service definition
Product service systems (PSS)
PSS representation
PSS development
Computer-aided engineering

ABSTRACT

The growing trend for delivering physical products to customers as parts of product service systems (PSS) is creating a need for a new generation of Computer Aided Design (CAD) system to support the design of PSS: so-called "PSS-CAD". Key research issues in the development of such systems include building understanding of the kinds of applications that designers of PSS might need and the establishment of well-founded representation schemes to underpin and support communication between PSS-CAD systems. Recent literature includes numerous descriptions of integrated PSS development processes, PSS-CAD tools to support these processes and early meta-models to provide information support. This paper complements this work by proposing a representation scheme that is a key prerequisite to achieving the interoperability between PSS-CAD systems which would be necessary to support the deployment of integrated PSS development processes in industry.

The representation scheme, a form of meta-model, draws on learning from the product definition community that emerged in the 1970s in response to a need for interoperability between the different shape-based CAD systems that were being developed at the time. The initial focus on shape representation has developed to digital product definitions that define the design of a product coupled with meta-data recording details of processes by which the design was created and, more recently, supported through-life. Similarly, PSS-related information includes both PSS definitions, to support the lifecycles of physical products and associated services, and meta-data needed to support the management of PSS development processes.

This paper focuses on information requirements for the definition of service elements of PSS and relationships with product elements and service actors. These requirements are derived from earlier work on the use of service blueprinting for the visualisation and mapping of service activities to deliver different types of service contract. Key information requirements addressed include the need to represent service process flow and breakdown structures, relationships between service and product elements, substitution relationships, and service variants. A representation scheme is proposed and demonstrated through application to a PSS case study. The representation scheme is built on a generic information architecture that has already been applied to problems of product definition; as such there is an underlying compatibility that offers real promise in the future realisation of integrated PSS development processes.

1. Introduction

The transition from the delivery of products to product service systems (PSS) is driving companies to focus on the performance of not only the products they develop and deliver to customers but also the services used to provide through-life support for these products. A consequence of this transition is that the role of the physical product is changing. Where once the development of products was a goal in its own right, increasingly products are parts of PSSs where the goal is to support and operate products through their lifecycles. This has led to an increasing interest in both the integration of products and services [64] and innovation in service offerings [37,53]. In response, the description of integrated service development processes is growing [20,30,47,59] and the need to consider both product and service lifecycles, and interactions between them, early in PSS development processes when the cost of change is at its lowest, is increasingly recognised [31,72]. In their broadest sense, PSS-CAD systems will support PSS developers in understanding these issues and many authors describe service design and development tools [23,51].

* Corresponding author. Tel.: +44 113 343 2113.
E-mail addresses: a.mckay@leeds.ac.uk (A. McKay), s.kundu@mmu.ac.uk (S. Kundu).
[1] Present address: School of Engineering, Manchester Metropolitan University, ll Saints Building, All Saints, Manchester M15 6BH, UK. Tel.: +44 161 2471636.

http://dx.doi.org/10.1016/j.aei.2014.07.004
1474-0346/© 2014 The Authors. Published by Elsevier Ltd.

Like product development processes, integrated PSS development processes will require underlying information architectures through which the integration of these tools can be achieved. A key to the realisation of such architectures lies in the establishment of well-founded representation schemes to support the definition of PSS through their lifecycles. This paper introduces such a representation scheme for service elements of PSS. It has been validated through application to case studies on the definition of contracted services in high value manufacturing systems [34]. The resulting digital service definitions have been used to support risk management of an industrial service [35] and the articulation of information requirements in the defence sector [14]. In the future such definitions could increase the use of computer-based evaluations of service concepts, for example by integrating them with simulation models [15], and more human-centred service design activities such as those discussed by Meiren et al. [47].

The representation scheme draws on learning from product definition both on the kinds of functionalities that might be required in PSS development processes and how such information might be represented. Parallels between the definition of product and service elements of PSS are introduced in Section 2.1 and used to inform requirements for digital service definition. The model has been implemented to support the definition of a range of industrial services that are subject to confidentiality restrictions. For this reason, a fragment of a fictitious case study that exhibits key characteristics of these real-world PSS, a coffee making machine repair service with two types of contract (spares only and availability [71]), is introduced in Section 3. The representation scheme, in the form of an information model, is introduced in Section 4 and its efficacy demonstrated through application to the coffee maker case study through population (in Section 4) and through implementation in a prototypical PSS definition system (in Section 5). An analysis of its efficacy in comparison to PSS meta-models available in the literature is provided in Section 6 and areas for further research are outlined in Section 7.

2. Literature review

Information requirements used to inform the development of product data representation schemes result from analyses of (i) the kinds of information that need to be captured to support product-related engineering processes and (ii) the kinds of tools and techniques the representation scheme is required to support. This section provides information requirements for the representation of service elements of PSS (Section 2.3) based on reviews of approaches to the definition of service products when compared with physical products (Section 2.1), and tools and techniques used for the definition of services (Section 2.2).

2.1. Definitions of physical products and service products

For the purpose of this paper, a PSS is a system composed of a physical product and associated services that support the product through-life. Thinking on the dual nature of technical artefacts argues that technical artefacts have both designed physical structures and intended functional structures. On intended functional structures, Vermaas and Houkes, in their ICE (Intentionalist, Causal-role, Evolutionist) theory [74], assert that when engineers ascribe functions to artefacts they have to consider explicitly the goals for which agents use artefacts and the actions that constitute their use; the agents' actions are captured in a "use plan". A number of papers resulting from this work, for example [33], include discussions on the distinction between function, behaviour and capacity of physical artefacts. Mumford [49] provides the following definitions for function and capacity: capacity is a property of an artefact that is understood according to what it can do or what function it can play in relation to other properties; function is a capacity plus the use plan that exploits it for an intended purpose. In this paper we take the view that, in a given PSS, the service elements are forms of use plan for the product elements.

On designed physical structures, Simons [61] uses mereology to provide a theoretical basis for the definition of physical product structures, of which bills of materials are a common manifestation. Key elements for a physical product definition are geometry, material specification and process plan. Design rationale, as captured using tools such as Rationale™ [5,55,73] and DRed [11], is a means by which designed physical structures are related to intended functional structures. Design intent, for example as captured using advanced requirements management techniques [2], enables intended functional structures to be related to stakeholder intents and so aspects of what Vermaas and Houkes refer to as use plans. An initial analysis of requirements for the definition of physical products and services is provided in Table 1.

It can be seen that there are alignments in the lifecycle stages of products and services and potential commonality in the representation schemes that might be used to represent requirements and design rationale. A key difference is that a core aspect of a physical product definition is its shape whereas the core aspect of a service definition is the service delivery process. There are many approaches to the definition of processes in the literature and some include digital process definitions that are used to support process evaluations. For example, Wynn et al. [76] report work related to the definition and simulation of product development processes. However, the underlying representation schemes for these process definitions are typically not published and, for this reason, the extent to which they might be integrated into PSS definitions is uncertain.

2.2. Service definition tools and techniques

A number of tools and techniques are being developed to support the definition of services in the fields of social and behavioural sciences, business, design and information technology; for the range of information to be supported see [69] and for examples of the tools and techniques see [19]. Tassi [65] presents a collection of these tools and techniques according to the following categories:

(i) the design activities they support (e.g. envisioning, designing/co-designing, testing and prototyping);
(ii) the kind of representation they produce (e.g. text, graphs, narratives, models, games);
(iii) the recipients they address (e.g. stakeholders, professionals, service staff, users); and
(iv) the contents of the project they convey (e.g. context, system, offering, interaction).

Service blueprinting approaches have traditionally been used to capture service-only products such as those in the hospitality and financial sectors [21]. Kim et al. [29] and McKay and Kundu [45] report applications of service blueprinting to PSS. In addition to these approaches, which focus on the definition of services, there is a growing body of work on systems that support service development activities and result in service definitions. These systems, which include both PSS-CAD tools such as those described by Tomiyama and Tassi [65,68] and more focussed service development tools, fall into Tassi's first category of tools and techniques. Kim et al. [29] provide an overview of PSS-CAD functionality and the representation tools used in PSS design in the context of a PSS design process framework that includes stakeholder, requirement, product and activity modelling and scenario planning, and which

Table 1
Initial analysis of requirements for the definition of physical products and services.

Product (artefact–goods) definition	Product (service) definition parallels
Product Specification: Requirements and links to stakeholder needs [2]	For services, Service Level Agreements, Performance Indicators (PIs) and Key Performance Indicators (KPIs) constitute a service specification
Product Definition: What is defined depends upon the kind of product and the product development process being used: specifically, the information required at each stage gate and through key process steps	Service designs (including "to be" service processes) are equivalent to product definitions. Drawing parallels with physical product development, understanding of the Service Development Process, for example ABB Full Service® [1], which provides process phases and decision points is needed to determine the scope and coverage of service definitions
Product Definition/Representation: Product structure and relationships are key information [44]. Product definitions include shape and material	Service definitions include process definitions and structures. Services have no physical shape or material specifications; research identifying key properties of service products is in its early stages
Actual Products: These are the physical artefacts that are delivered to customers	A key difference between goods and services is that the "manufacture" and delivery of services are done at the same time, and services are transient. However, although data related to service delivery is frequently collected, issues related to poor service quality are frequently attributed to the poor and variable quality of such data. The equivalent to a physical artefact definition would be an "as delivered" service definition that would couple an "as delivered" service structure with data collected during service delivery, eg, related to customer satisfaction

results in a modified service blueprint that includes a blend of product and service structure elements.

The two most widely reported PSS-CAD systems are "Service CAD" [58] and "Service Explorer" [23]. Cavalieri and Pezzotta [13] refer to a PSS engineering process which can be regarded as a core business process within which PSS design, along with other activities, sits. Cavalieri and Pezzotta highlight the importance of creating PSS whose lifecycle processes are considered in a systematic way during the design process. Many research groups report developments of such systems [57]. As with the development of product CAD in the 1980s, the reporting of these systems tends to focus on system functionality rather than the underlying representation schemes on which the systems are built. However, an analysis of the system outputs presented in these papers highlights a need for the representation of both functional and service activity structures and, in each kind of structure, both part-whole and connection relationships. For example, the representation of Hara et al.'s PSS function structure [23] requires a functional part-whole structure whereas their behaviour blueprint would require a functional structure with connection relationships, and their activity blueprint a service activity structure with connection relationships. Similarly, for the representation of PSS to support Sakao et al.'s system [58], the outputs presented include service function structures, service flow structures, product breakdown structures and relationships between elements of product and service structures.

Other PSS-CAD researchers focus on supporting later stages of PSS design and development processes. For example, "Service Explorer" and "Service CAD" both include means of visualising service activities. Visualisation draws information from the system's underlying representation of a service and presents it in ways that are useful to system users. In addition to supporting visualizations by people, CAD systems also provide information to support application packages such as, for product-CAD, engineering analysis. Rese et al. [56] recognise the different needs of individual user groups and propose a framework for the definition of information strategies that are sensitive to these needs. Dill and Schendel [17] propose an approach to PSS design that supports the generation and evaluation of PSS alternatives; the generated variants are represented using a simplified version of a service blueprint where activities are positioned within swim lanes but not connected to each other flow-wise. This supports evaluation of variants based on ownership and transaction costs. Service simulation can also be regarded as a form of service system application. Cuthbert et al. [15], for example, describe an application of discrete event simulation to maintenance services. Supporting such applications requires a service representation that includes both service and product related aspects of the PSS.

The representation scheme introduced in this paper supports the definition of product and service elements of PSS and both part-whole and connection relationships between them. In addition, it has the flexibility to support Dill and Schendel's system because process/activity steps can be defined without relationships between them – this is because both elements and relationships are treated as first class objects. As a result the representation scheme could be used to underpin the exchange and sharing of data between users of the two systems. From the available literature, the "Service Explorer" and "Service CAD" systems each supports detailed PSS definitions at a single point in time. When used in live PSS engineering processes, there will also be a need to support how PSS definitions change over time and in response to the needs of individual customers. The representation scheme in this paper can be used to support this additional information through substitution relationships and the definition of PSS variants (see Sections 4.4 and 4.5).

2.3. The definition of service information

Zeithaml et al. [77] identify five quality gaps in service delivery that may result in service failure. Managerial strategies to close the service quality gaps are recommended by Zeithaml et al. and distilled by Lovelock and Wirtz [38]. Several are related to improving the management of service information. Effective delivery of service demands access to high quality service information (i.e. complete, correct, minimal and available to the right people at the right time). McFarlane [40] asserts that the information requirements for support service solutions are multifaceted and highly dependent on the nature of the offering and the underpinning service agreement. Berkeley and Gupta [9] survey information required to deliver quality services involving high levels of customer contact. In high customer contact services, a firm's ability to deliver a quality service depends on its capacity to collect, process, distribute and use information. According to Berkeley and Gupta, service delivery processes can be broken down into three broad categories: input information, process information and output information. The representation scheme introduced in this paper defines the service itself; this service definition could include the definition of service processes that collect and operate on data in all three of Berkeley and Gupta's categories. In this context, a number of researchers propose ways of representing services although the focus is on service concepts rather than the service definition itself; these systems support early stages of PSS design and development processes. Becker et al. [8] provide a review of these approaches. Given their focus, it is not surprising that the representation schemes within such systems tend to support the definition of service concepts and requirements. Welp et al. [75] propose three planes (function, process and object) which must be addressed through service representation schemes. The representation scheme introduced in this paper sits on Welp et al.'s process plane.

Each of the papers reviewed in the previous paragraph supports the positioning of PSS representation schemes with respect to both their scope and level of detail. In discussing more detailed aspects of PSS knowledge and information management, the literature refers to a range of knowledge and information models, ontologies and languages. Fig. 1 provides a framework that is used in this paper to create a coherent overview of these PSS knowledge and information management solutions. The figure provides a schematic that shows the underlying information infrastructure needed to support knowledge intensive activities such as PSS engineering. It can be seen that each organisation (which could be different functions within one company or separate companies in a supply network) has a different manifestation (indicated by the different colours used in the figure) of a PSS engineering process. Each PSS engineering process interacts with (taking information from and adding information to) its own PSS definition. In turn, each of these definitions is defined in terms of a PSS representation scheme which will be tailored to suit the needs of the engineering process it is intended to support. These representation schemes can be viewed using the ANSI/SPARC three layer model for database management systems [16]. This model includes three perspectives for considering database schemas: multiple external views which address the needs of specific applications and users, a single conceptual view which includes information structures that support all of the external views, and one or more internal (or physical) views which specify how the conceptual view is implemented on different computational platforms and/or using different implementation methods. Finally, each representation scheme is underpinned by one or more ontologies which specify the concepts and relationships captured through the representation scheme. The central section of Fig. 1, labelled (b), shows a similar structure for supporting communication of PSS definition data between the organisations represented in sections (a) and (c) of the figure. It can be seen that the underlying infrastructure has a common structure but different manifestations to either of the two organisations; this is because the purpose of the central portion (to support communication) is different to the purpose of the other two sections (to support the engineering of PSS offerings).

PSS definition data is generated and used through PSS engineering processes. Maussang et al. [39] describe a PSS design methodology that encompasses both service and product structures. A core aspect of the theoretical basis upon which the service definitions of this methodology build lies in process definition and modelling. For example, Maussang et al. show connections between service and product breakdown structures and Stanicek and Winkler [62] report an application of conceptual modelling to the behaviour of a service system. The representation scheme in this paper enables the definition of multiple service breakdown structures of a given service. The nature of these [PSS lifecycle] processes determines the information content (both scope and level of detail) in the PSS definition and, therefore, the necessary representation schemes and underlying theoretical frameworks and ontologies. Hepperle et al. [24] define a PSS lifecycle model that includes key PSS lifecycle activities connected to a process model. This model is equivalent to an Application Activity Model in a STEP Application Protocol [25] and could be used to identify PSS engineering activities and information flows that are in (and out of) scope for the underlying information system. Hepperle et al.'s model could be coupled with Muller et al.'s [48] framework for the elicitation of PSS requirements to create information requirements for the PSS definition needed to support particular PSS lifecycle activities. In this vein, Goh et al. [22] give an example of how in-service data collected through the delivery of a PSS might be made available to inform future design activities.

The implementation and population of representation schemes with sample data is a widely accepted means of validating representation schemes. For example, in the STEP standard, Abstract Test Cases are used to provide such data. Such test cases, in their development, can be used to validate the underlying representation schemes and, in use, can be used to test software tools that implement the representation schemes. Ontologies can be tested in the same way, either through implementation and use in knowledge engineering systems such as Protégé or through implementation in data models that can be populated using methods similar to

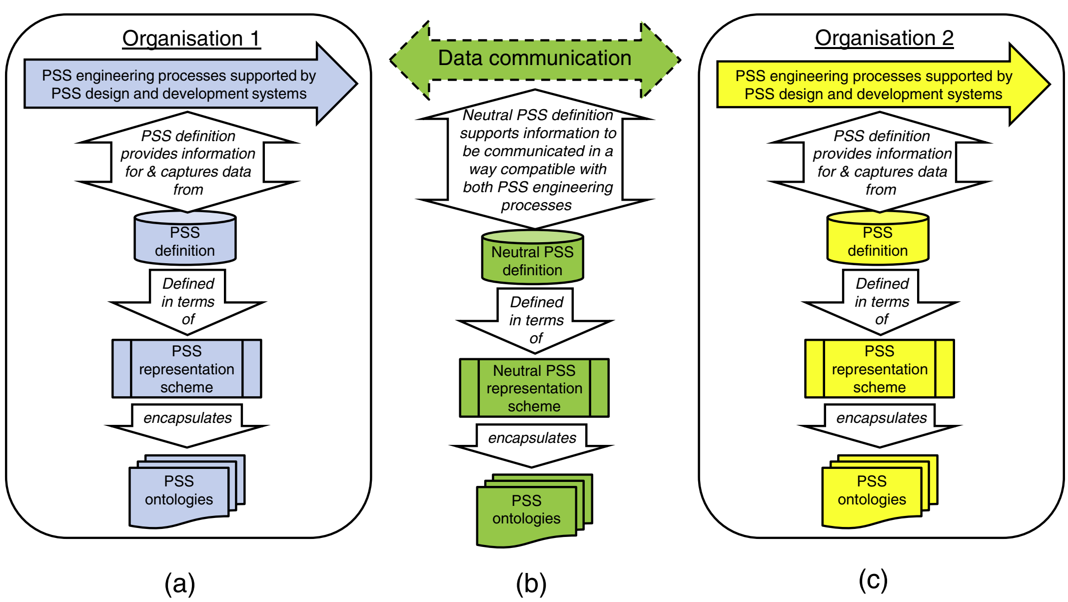

Fig. 1. PSS information architecture.

those used in this paper. A number of authors introduce ontologies that could be used to underpin PSS knowledge and information management systems. However, from information available in the literature, these are in the very early stages of development and require substantial verification and validation efforts before being translated into wide practice. For example, Annamalai et al. [4] present an ontology that was developed through engagement with a wide PSS community and contains a potentially useful collection of attributes for PSS definitions but without detail on how such attributes, or PSS definitions more generally, might be defined and structured for specific purposes derived from PSS engineering process needs. Annamalai et al. cite Bullinger et al. [12] who introduce a service model (a representation scheme in the terminology of Fig. 1) where service activities are defined as parts of service structures which include flow, composition, alternative, substitution and integration relationships. Bullinger et al.'s model represents these relationships as attributes of service activities in contrast to the representation scheme introduced in this paper which supports the explicit definition of these kinds of relationship and, because the relationships are treated as first class objects, the addition of attributes and relationships to other PSS elements. More details on the implications of this are given in Section 6. Similarly, Doultinsou et al. [18] present a UML model for the capture of service information for use in engineering design and Zhu et al. [79] provide an ontology for maintenance, repair and overhaul services. Like Annamalai et al., both of these models have been developed using systematic processes, and screenshots from early implementations are provided, but neither paper provides detail of how the models have been validated or might be used in engineering processes such as design (Doultinsou et al.) or maintenance, repair and overhaul (Zhu et al.). Hara et al. [23] highlight different types of PSS structure and their importance in the configuration of PSS offerings. Two key service structures are shown in Service Explorer screenshots and highlighted in their behavioural and activity blueprints. The representation scheme introduced in this paper could be used to define these different kinds of structure and, because they use the same underlying structure as product definitions, these definitions could be explicitly related to elements of product structures which are represented as hardware entities in the Service Explorer system.

3. Case study – Coffee maker machine maintenance and repair service

PSS definitions are typically not presented in the literature because of commercial sensitivities. This is comparable to product-related literature where, again, design details are rarely published. In such situations, researchers tend to use synthetic case studies that reflect key characteristics of the PSS they are working with. For example, Kim et al. [28] use a case study based on the laundry industry. The efficacy of the representation scheme presented in this paper is demonstrated by populating it with data from a case study based on maintenance services that draw on examples from the high value manufacturing sector but, for confidentiality reasons, using a fictitious coffee making machine PSS. The representation scheme introduced in this paper has been used to capture service blueprint information in the form of digital service definitions. In this paper it is applied to a fragment of the coffee maker machine maintenance and repair service case study introduced in [45]. A labelled fragment of this case study, used in this paper, is given in Fig. 2.

The whole case study defines a service for two types of contract: availability and spares only. For the availability type contract, the coffee maker manufacturer supplies coffee maker machines to its customer and takes responsibility for their maintenance and repair in an availability type contract. The service level agreements (SLAs) and key performance indicators (KPIs) for the availability type contact are shown in the middle column of Table 2. On the other hand, in the spares only type contract, the coffee maker manufacturer supplies coffee maker machines to its customer and takes responsibility for their installation and repair in the event of breakdown in a spare only type contract. The service level agreements and key performance indicators for the spares only type contact are outlined in the right-hand column of Table 2.

The key process steps carried out in the coffee maker machine maintenance and repair service for both availability and spares only type contracts are illustrated in the flow chart in Fig. 2. The process steps carried out only in availability type service contracts are shown by the hatched shapes and the process steps carried out only in the spares only type service contract are shown by shapes filled with dots. Substitute or alternate process steps in both types of contract are tagged using S_x and A_x respectively. For example, the process step tagged with S_1 is used in the spares only type contract and is replaced by the process step tagged with A_1 in availability type contracts. Process steps carried out by the service provider are represented by shapes with solid outlines whilst those done by the service customer are shown using shapes with dashed outlines. The process steps are logically connected by solid arrows to represent the process sequence. The process flow is in the arrow direction. The dashed arrows represent flow of physical artefacts during service delivery. Other labels in this figure are used for clarification later in the paper.

4. Underlying information model and its application to the case study

An information model for the definition of PSS structures is given in Fig. 3. It is an adapted version of the general model for product, process and supply network structures which was originally presented in [42]. The model is based on a theoretical foundation that combines mereology with learning that has evolved from work on the definition of product structure models through process structure to supply chain and enterprise network models. A key feature is that elements and relationships (nodes & arcs) are treated equally, as first class objects, and, as a result, each can exist independently of the other. This capability is exploited in this paper to represent PSS. The model in Fig. 3 is described in this section through application to the case study introduced in Section 3.

The information model shown in Fig. 3 is drawn using the Express-G notation ([25]). Express is used because the representation scheme presented in this paper is a conceptual model (in an ANSI/SPARC sense) and, therefore, not directly related to a specific implementation form or method; the STEP standard provides a series of standards (ISO-10303:20 series) which specify how Express data specifications might be translated into a range of implementation methods. In addition, the representation scheme introduced in this paper is validated through population using a graphical notation of Express-I [27], Express-I-G, that allows instance data to be presented with explicit reference to constructs (entities and relationships) in the underlying data model. Most of the ontologies and representation schemes reviewed in Section 3 are defined and implemented using object oriented approaches (forms of internal schema in an ANSI/SPARC sense). Zhao and Liu [78] give an example of the translation of an Express data specification into the OWL Web Ontology Language. The boxes in Fig. 3 represent concepts (Express entities) and the lines represent relationships between concepts. The circles on the ends of lines indicate direction; in an instance, a concept at the plain end of a fine line has one and only one concept at the circled end and the concept at the circled end has, unless otherwise stated, zero, one or many concepts at the

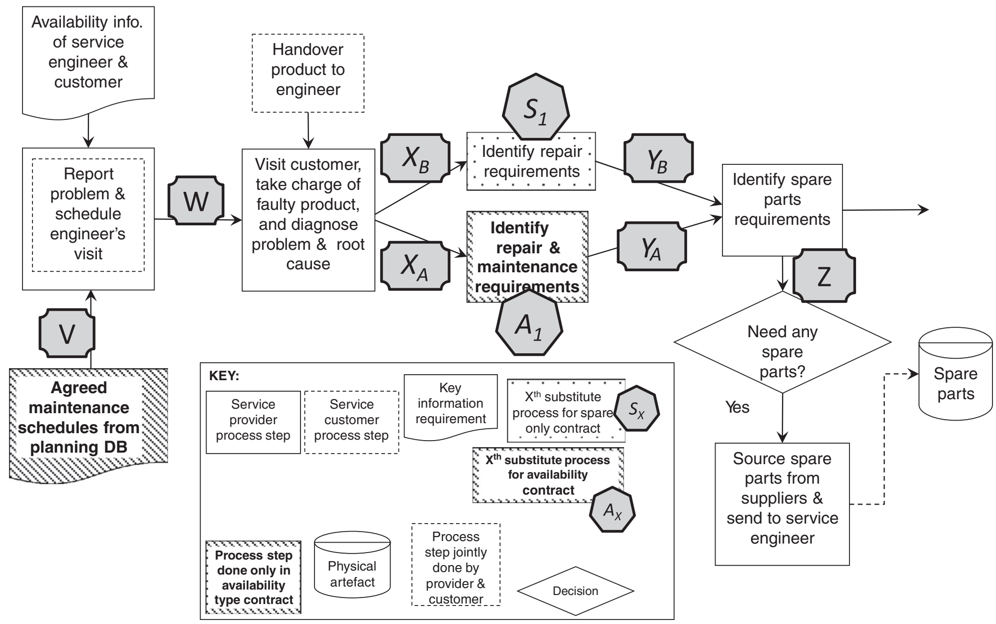

Fig. 2. Labelled fragment of service to demonstrate the capability of the information model introduced in this paper. Adapted from [45]

Table 2
Service level agreements (SLAs) and key performance indicators (KPIs) for availability type and spare only type contracts for coffee maker machine's maintenance and repair service.

	Availability type contract	Spare only type contract
KPI 1	Call-to-Repair response time-24 h (max)	Call-to-Repair response time-48 h (max)
SLA 1	Provider supplies, installs, maintains, repairs and supports spare parts and the whole machine	Provider supplies, installs and repairs spare parts for the coffee maker during breakdown
SLA 2	Service package includes planned preventive and predictive maintenance, and unplanned breakdown maintenance	Provider repairs the coffee maker back to safe working condition after it has malfunctioned or broken down
SLA 3	Provider is responsible for customer training	Provider responsible for customer training
SLA 4	Provider supplies user manuals/training materials to the customer	Provider supplies user manuals/training materials to the customer
SLA 5	Customer pays an annual fixed price to the provider for availability of the product (coffee maker) and services	Customer pays for each new or replaced spare part supplied by the provider
SLA 6	The price includes both spares and services (i.e. complete availability and ready for use) for a fixed period of time	Customer pays an annual fixed price to the provider for installation and repair services that cover a whole calendar year
SLA 7	24 month minimum contract period (which can be extended after end of the contract)	12 month minimum contract period (which can be extended after end of the contract)
SLA 8	Services are provided on both as-planned and on-demand bases for the duration of the contract	Services are provided on an on-demand basis for the duration of the contract

plain end. For example, every *structured_element_relationship* has two and only two *structured_elements* and, every *structured_element* can play the role of zero, one or many *relating* and *related structured_element_relationships*. The heavy lines represent specialisation relationships. For example, in Fig. 3 a *structured_element* can be either an *item* or an *item_relationship* and a given *item* is either a *part*, a *process_step*, an *information_requirement*, a *physical artefact or* a *decision*[2]. It can also be seen that a given *item_relationship* can represent either a *composition* or a *connection* relationship.

The remainder of this section reports the application of this information model to represent key facets of service elements of the PSS case study introduced in Section 3.

4.1. Definition of service process flow structures

A service process flow structure is a collection of process steps (shown by rectangles and diamonds in Fig. 2) and the connections between them that are needed to deliver the service (shown by solid arrows in Fig. 2). With respect to product definitions, and taking a view that, like physical products, services have lifecycles, the service flow structure in the case study can be regarded as an as-designed service definition. An instance of the information model given in Fig. 3, for the part of the service definition given in Fig. 2, is shown in Fig. 4. It can be seen that three process steps (labelled with black stars) are shown. Each is defined as a *process step* which is a subtype of *item*, and each flow between process steps (*V*, *W* and X_A) is represented using the *connection* entity which is a subtype of the *item relationship* entity. The Express-I-G also includes an information requirement (labelled with an open star) which, like process steps, is modelled as a subtype of the *item* entity and can therefore be connected to process steps in the same way as process steps are connected to each other. This flexibility, in a relatively small information model, results primarily from both elements and relationships being treated as first class objects.

A key reason for developing the representation scheme introduced in this paper was to support the creation of digital service blueprints. A service blueprint is built around the principal stages

[2] Note: only *process_steps* and *information_requirements* are used in this paper.

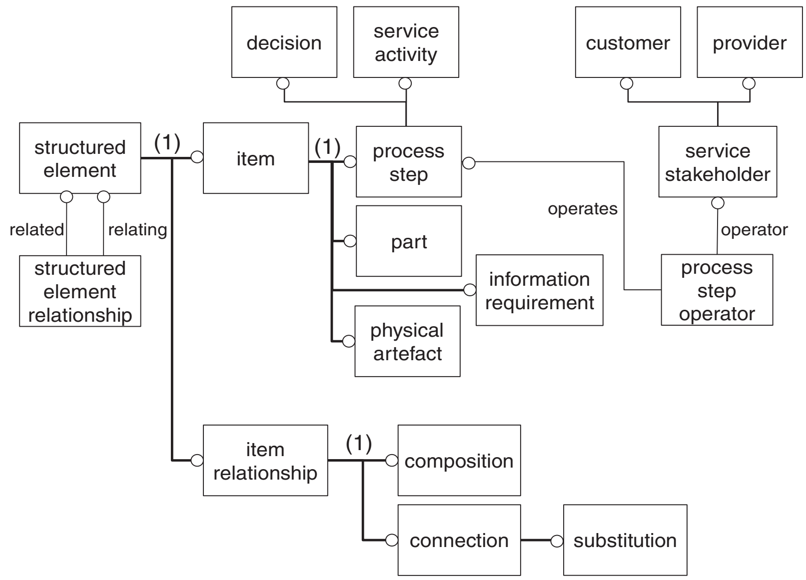

Fig. 3. An adaptation of the general purpose information model from [42] to support process flows and part breakdown structures.

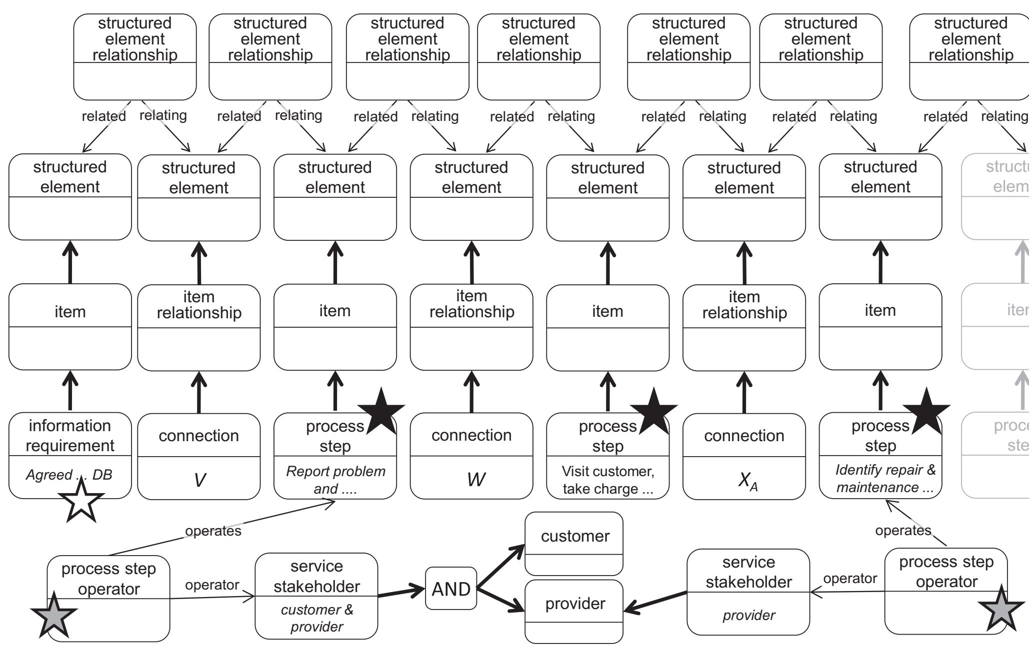

Fig. 4. Express-I-G of selected connection relationships from the case study.

(i.e. key process steps) of the service and two axes on the blueprint: (i) a horizontal axis representing the chronology of actions conducted by the service customer and the service provider and (ii) a vertical axis that distinguishes between different areas of actions – these areas of actions are separated by different lines of visibility and interaction. The information represented in Fig. 4 supports the positioning of service process steps on the horizontal axis of a service blueprint.

4.2. Identification of who carries out each process step

For the vertical axis of a service blueprint, two kinds of process need to be captured: principal actions and support processes. Principal actions can be of three types: onstage principal actions by the customer, and onstage and backstage principal actions by the service provider's customer contact personnel. A service provider's principal actions and support processes interact with IT resources. For technical services these could include engineering information systems. The visibility of the sub-processes that form processes in the service definition is governed by their positioning on the blueprint relative to a number of visibility and interaction lines. If the enactment of a service blueprint is seen as a simultaneous production and consumption of the service then these lines govern who sees which parts of the delivery of the service. For this reason, the representation scheme needs to support information to determine the visibility and interaction level of each process step. This is achieved through the *process step operator* entity which allows the specification of who carries out the step: customer, provider or both (achieved through the default ANDOR supertype relationship on the *process step operator* entity). The instance data that defines who operates each of the process steps in the case study can be seen at the bottom of Fig. 4, highlighted with grey stars.

In its current form the information model only allows operator information to be associated with process steps and therefore (because they are subtypes of the *process step* entity) process activities and decisions. This is because the service blueprint that the information model was designed to support only has this as a requirement. In the future the need to position other kinds of service element on a service blueprint, such as information requirements and parts, may emerge. If this were to happen then the information model given in Fig. 3 would need to be adjusted. A key characteristic of the information model is that a *process step* can be defined without being associated with a *service stakeholder* and vice versa. Drawing on learning from the STEP standard [26], the association of people and organisations with a service definition is application specific. For example, in this case it is needed to support the visualisation of a service as a service blueprint. The information model has been designed to allow both stakeholders and service processes to be defined independently of each other.

4.3. Definition of service breakdown structures

The phrase "service breakdown structure" is widely used in the literature but covers a range of possible meanings. Drawing from research in product definition, a product breakdown structure has elements of a given kind related to each other through composition (part-whole) relationships. Simons [61] uses mereology to provide a theoretical basis for the definition of physical product structures, of which bills of materials are a common manifestation. Given the earlier assertion (Section 4.1) that the process steps form a service definition, it can be argued that the service breakdown structure is a process breakdown structure where the elements are process steps, related to each other through composition (part-whole) relationships. It will be seen later that this logic enables the definition of service variants and substitution relationships.

Examples of service breakdown structures, shown as indented [service element] parts lists, for the two kinds of contract introduced through the case study are given in Fig. 5. It can be seen that each service (spares only and availability) has its own structure composed of process steps that are largely common across the two. Italics are used in Fig. 5 to highlight differences between the two breakdown structures. As with product breakdown structures, a key benefit from these structures is that they highlight service elements needed to realise each service. For example, the spares-only service includes a process step called "Identify repair requirements"; this also exists in the availability service but at a lower level of decomposition as part of the "Identify repair & maintenance requirements" process step.

A schematic of the spares only service breakdown structure is given in Fig. 6 and a population of the information model for a part of this structure is shown in Fig. 7.

A key difference between Figs. 4 and 7 lies in the kind of *item relationship* entity that is used: *connection* relationships in Fig. 4 and *composition* relationships in Fig. 7. The process steps are again labelled with black stars; the numbers in the stars indicate the level of decomposition in Figs. 5 and 6 with zero indicating the top level of structure. However, since both process steps and relationships between them are treated as first class objects the same service elements (in this case process steps and relationships between them) can occur in multiple service process flow and breakdown structures.

4.4. Definition of substitutions within service process definitions

As with product definitions, the definition of a service is unlikely to remain constant through its lifecycle and, for this reason, there is a need to be able to capture substitutions made as a result of change processes. The definition of substitutions within service process definitions is supported through a substitution entity based on the STEP substitution entity [26] (see Fig. 8).

A benefit of treating elements and relationships as first class objects is the additional expressive power that is achieved in comparison with models, such as the STEP integrated resources, where relationships are dependent on the elements that they relate. For example, the information model in Fig. 2 captures *substitution relationships* as a kind of connection relationship that relates a composition relationship with a process step in Fig. 10. An example of a service element substitution is given in Fig. 9 and the associated information model that supports its representation is given in Fig. 9. It can be seen that a *substitution* is defined as a subtype of a *connection* which, in turn, is a subtype of an *item_relationship*; meaning that a substitution is a kind of connection that is used to connect parts of the service in a given time and/or context rather than flow-wise. An example instance is given in Fig. 10 where one process step is substituted for another. The example relates to process step substitutions because these occur in the case study. However, the information model is more general and could be used

⇒ **Spares only service**	⇒ **Availability service**
⇒ ⇒ Report problem and schedule engineer's visit	⇒ ⇒ Report problem and schedule engineer's visit
⇒ ⇒ Visit customer, ... diagnose problem and root cause	⇒ ⇒ Visit customer, ... diagnose problem and root cause
⇒ ⇒ <u>Identify repair requirements</u>	⇒ ⇒ *Identify repair & maintenance requirements*
⇒ ⇒ Identify spare parts requirements	⇒ ⇒ ⇒ <u>Identify repair requirements</u>
⇒ ⇒ Source spare parts ... & send to service engineer	⇒ ⇒ ⇒ *Identify maintenance requirements*
⇒ ⇒ ⇒ Source spare parts	⇒ ⇒ Identify spare parts requirements
⇒ ⇒ ⇒ Send spare parts to service engineer	⇒ ⇒ Source spare parts ... & send to service engineer
	⇒ ⇒ ⇒ Source spare parts
	⇒ ⇒ ⇒ Send spare parts to service engineer

Fig. 5. Examples of service breakdown structure from the case study.

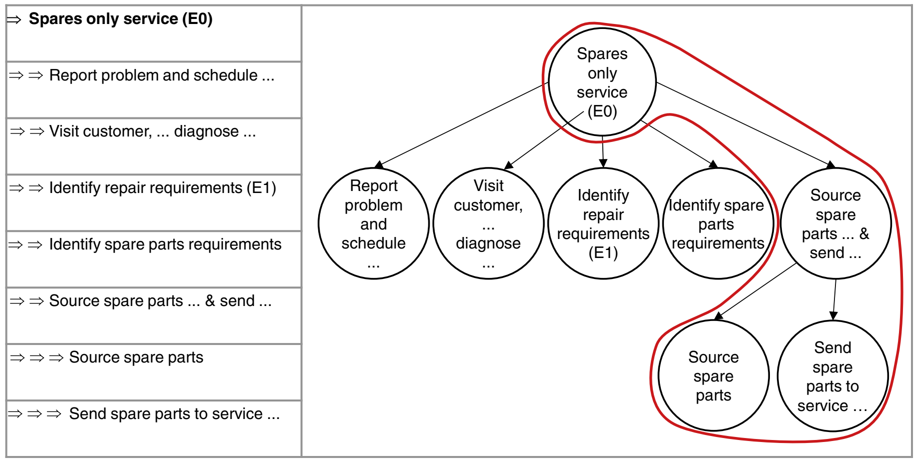

Fig. 6. Example spares only service breakdown structure.

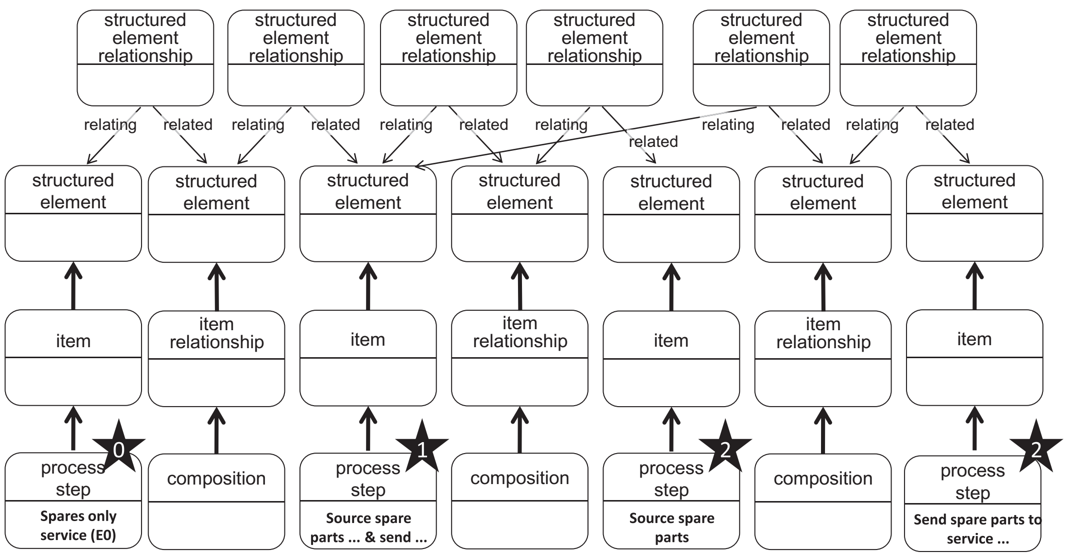

Fig. 7. Express-I-G of the right-hand branch (outlined in red) of the service breakdown structure in Fig. 6.

```
ENTITY product_definition_substitute;
    description: text;
    context relationship: product definition relationship;
    substitute definition: product definition
WHERE
    WR1: context relationship.related product definition
:<>: substitute_definition;
END_ENTITY;
```

Fig. 8. The STEP substitution entity [26].

to support substitution of any kind of item rather than just process steps; further work is needed to explore the implications and applicability of this generality. The context of the substitution is the part-whole [breakdown structure] relationship that connects the step being replaced to the rest of the service structure. An alternative approach could be to substitute the relationships, which is possible because arcs and nodes are treated equally as first class objects in the underlying information model, in which case no context would be needed. However, in the longer term, if process connections are shared then this may not be feasible; more research is needed to understand the possible range of instances and their meanings and implications. An example instance of a service substitution relationship from the case study is given in Fig. 11.

4.5. Definition of service variants

Substitution relationships are used to specify design changes across versions of a design. There is also a need to support the definition of variants of a design. The key difference between versions and variants is that versions reflect how a design (intended to meet a common set of requirements) changes over time whereas variants are responses to different sets of design requirements using a common solution architecture. Tidstam and Malmquist [66] report research on physical product variants and the design of product platforms as a common architecture on which designs delivering different functional requirements are built. In this paper we demonstrate how the definition of variants might allow variety and so customisation of service offerings. An example of service variants from the case study is given in Fig. 12.

It can be seen that there are two service variants, one spares-only and the other availability, which both share a generic architecture (see Fig. 12(a)) where the details of one step, “Identify requirements”, depends on the kind of contract being delivered. If the variant is a spares-only service (see Fig. 12(b)) then the “Identify requirements” step is E1, “Identify repair requirements”. On the other hand, if the variant is an availability service contract (see Fig. 12(c)) then the “Identify requirements” step is E2,

⇒ **Spares only service (E0)**		⇒ **Spares only service (E0)**
⇒ ⇒ Report problem and schedule ...		⇒ ⇒ Report problem and schedule ...
⇒ ⇒ Visit customer, ... diagnose ...		⇒ ⇒ Visit customer, ... diagnose ...
⇒ ⇒ Identify repair requirements (E1)	**Replace E1 with E1-rev1 in the context of part-whole relationship with E0**	⇒ ⇒ Identify repair requirements (E1)
⇒ ⇒ Identify spare parts requirements		⇒ ⇒ Identify spare parts requirements
⇒ ⇒ Source spare parts ... & send ...		⇒ ⇒ Source spare parts ... & send ...
⇒ ⇒ ⇒ Source spare parts		⇒ ⇒ ⇒ Source spare parts
⇒ ⇒ ⇒ Send spare parts to service ...		⇒ ⇒ ⇒ Send spare parts to service ...

Fig. 9. An example of service substitution relationship.

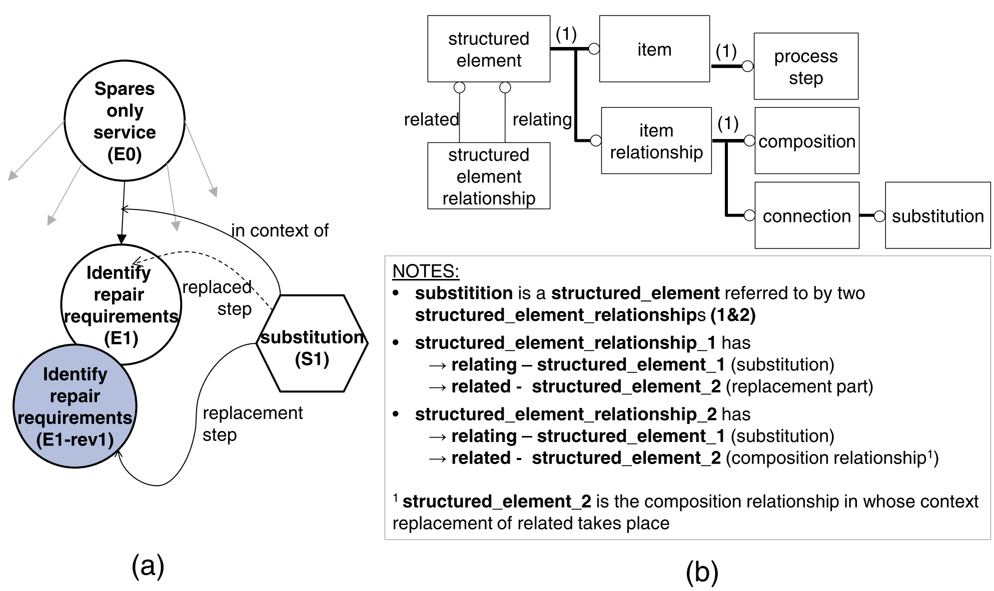

Fig. 10. Example of a substitution and associated information model.

"Identify repair & maintenance requirements", which includes both step E1 and a second step, E3, "Identify maintenance requirements". The structure of these two service variants and the generic service architecture are defined using a functional notation in Fig. 13. Implementation of this notation would require a functional capability with the characteristic of lambda calculus where the [data] types of both functions and their parameters can be of any kind. For example, in Fig. 13, the *service_definition* function is of type *service_definition;* the function has two parameters, one is text, a string of characters, and the other is a *process_step*. As can be seen in the function invocations on the right-hand side of Fig. 13, the value of this second parameter can be either a simple process step or a process step that is composed of other process steps.

In this section product structure relationships have been used to demonstrate the principle of service variants. For a full implementation, more sophisticated modelling methods would be needed such as the ability to represent instances as types [43] or reflective computing [63]. For this reason, an Express-I-G instance is not provided for the definition of service variants.

5. Computer implementation

The representation scheme to support the definition of PSS structures was validated through population (see Section 4) and through application to support a prototypical PSS definition system. In this section screenshots taken from this system are presented and used to highlight key features of the representation scheme and the benefits it provides.

The screenshot given in Fig. 14 shows a product model that includes two design definitions: a bill of materials product structure (containing part-whole relationships) in the top left-hand corner of the figure and a Solidworks eDrawing in the top right-hand corner of the figure. These two definitions (the product structure defined using the representation scheme from Section 4) and the CAD model are related to each other by relationships between the bill of materials and the eDrawing. These relationships can be seen by the highlighted (red[3]) parts at the bottom of the figure which change as the bill of materials is navigated. This functionality is the same as a Solidworks assembly tree, of which there are equivalents in other 3D modelling systems; however, including

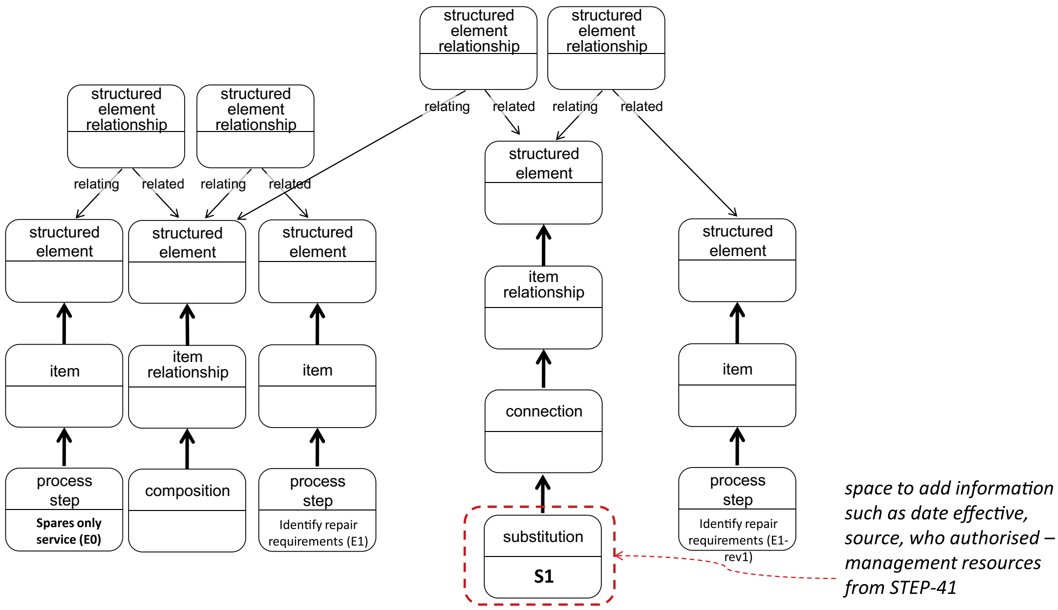

Fig. 11. Express-I-G of substitution example given in Fig. 9 in terms of the information model given in Fig. 10.

⇒ Generic support service	⇒ Spares service (E0)	⇒ Availability service
⇒ ⇒ Report problem and schedule ...	⇒ ⇒ Report problem and schedule ...	⇒ ⇒ Report problem and schedule ...
⇒ ⇒ Visit customer, ... diagnose ...	⇒ ⇒ Visit customer, ... diagnose ...	⇒ ⇒ Visit customer, ... diagnose ...
⇒ ⇒ Identify requirements	⇒ ⇒ Identify repair requirements (E1)	⇒ ⇒ Identify repair & maintenance requirements (E2)
		⇒ ⇒ ⇒ Identify repair requirements (E1)
		⇒ ⇒ ⇒ Identify maintenance requirements (E3)
⇒ ⇒ Identify spare parts requirements	⇒ ⇒ Identify spare parts requirements	⇒ ⇒ Identify spare parts requirements
⇒ ⇒ Source spare parts ... & send ...	⇒ ⇒ Source spare parts ... & send ...	⇒ ⇒ Source spare parts ... & send ...
⇒ ⇒ ⇒ Source spare parts	⇒ ⇒ ⇒ Source spare parts	⇒ ⇒ ⇒ Source spare parts
⇒ ⇒ ⇒ Send spare parts to service ...	⇒ ⇒ ⇒ Send spare parts to service ...	⇒ ⇒ ⇒ Send spare parts to service ...
(a)	(b)	(c)

Fig. 12. An example of service variants.

the bill of materials in the PSS definition allows other relationships to be created as well. For example, the network shown in the bottom right-hand corner of Fig. 14 shows a maintenance process definition of the kind that would be found in a service blueprint swim-lane. The red[3] highlighted area in this part of the screenshot highlights a relationship between the parts in the bill of materials and the Solidworks eDrawing and steps in the maintenance process (item to item relationships). A small number of these relationships are shown in the bottom left-hand corner of Fig. 14.

The screenshot shown in Fig. 15 includes all of the definition elements given in Fig. 14 plus a linkage to a supply network shown in the centre of Fig. 15. The definition of supply networks, which are defined as structured entities in a similar way to product and process elements of PSS, is outside the scope of this paper but is used to illustrate the value of representing relationships as first class objects. In this case the part in the product definition (the "OK indicator light") highlighted in the top half of Fig. 15 is related to a flow in a supply network map. This allows users of the PSS definition to find out who supplied the part, for example, to support the acquisition of a replacement part, and demonstrates how the representation scheme presented in this paper can be used to define relationships between items in one structure (in this case, parts in a product structure) and item relationships in a second structure (in this case, a flow in a supply network). This is possible because, like the STEP integrated resources, relationships are represented explicitly in the underlying conceptual model.

The third kind of relationship across structures supported by the representation scheme presented in this paper is between pairs of item relationships. An example of this kind of relationship is given in Fig. 16 where a process step in the process model (Check fuse. Not OK") is related to a flow in the supply chain definition ("fuse supplier – power connector" relationship). This flow, in turn, is related to an element in the product definition (a fuse). The definition of these kinds of relationship is possible because

[3] For interpretation of color in Fig. 14, the reader is referred to the web version of this article.

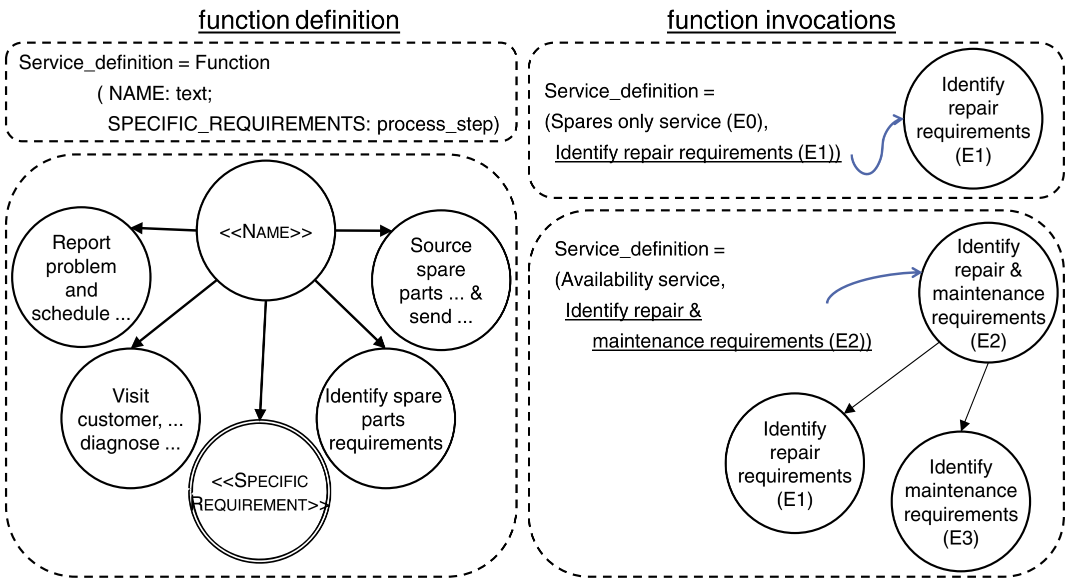

Fig. 13. Spares-only and Availability service definitions defined using functions.

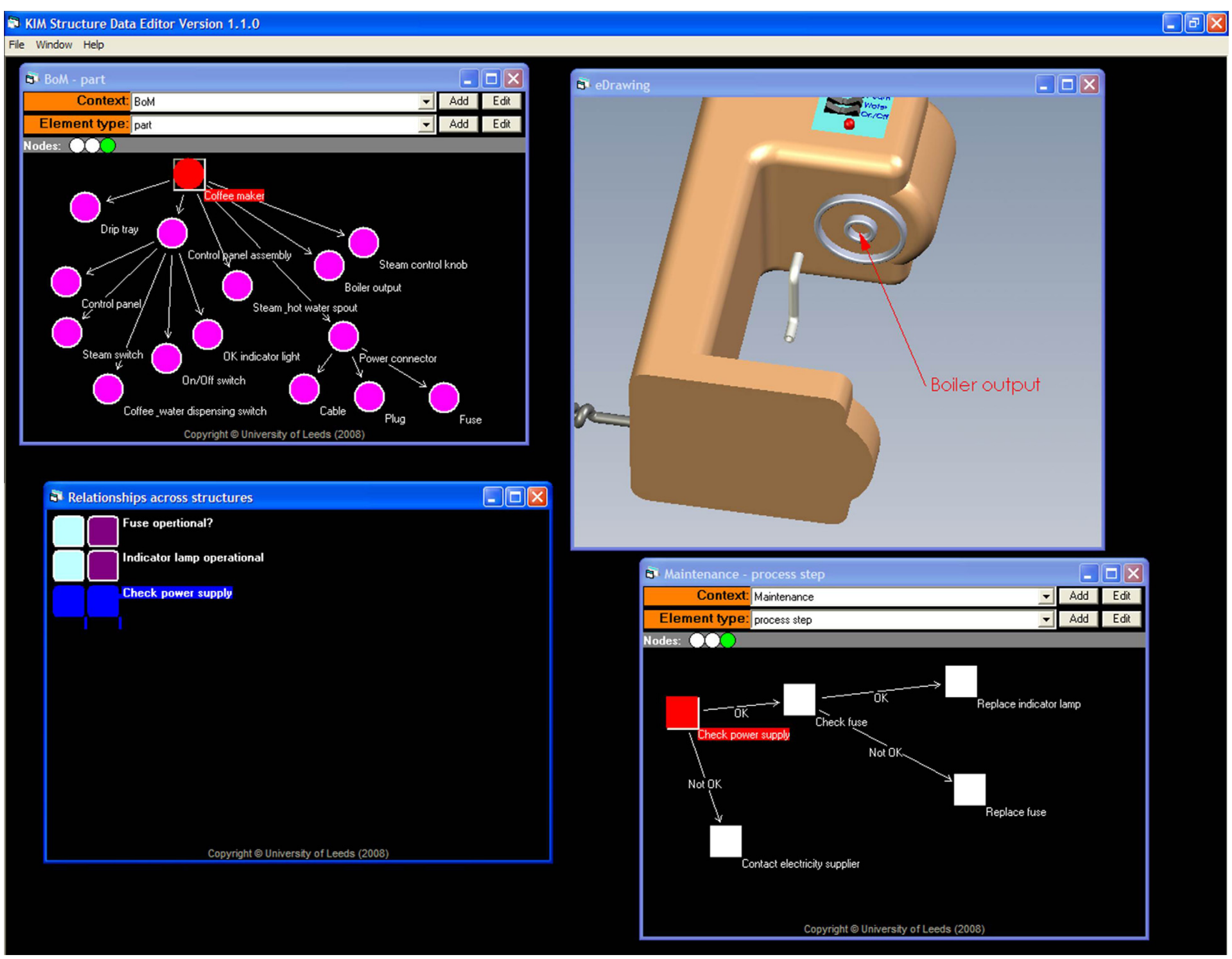

Fig. 14. A PSS definition including product and service structures – node to node relationships.

relationships and elements are treated as equal, first class, objects in the underlying conceptual model. This contrasts with the STEP integrated resources where relationships can only be related to each other through elements of product structures.

6. Discussion

Nøhr Hinz et al. [52] provide a general framework for PSS development and use it to position a range of PSS development methods

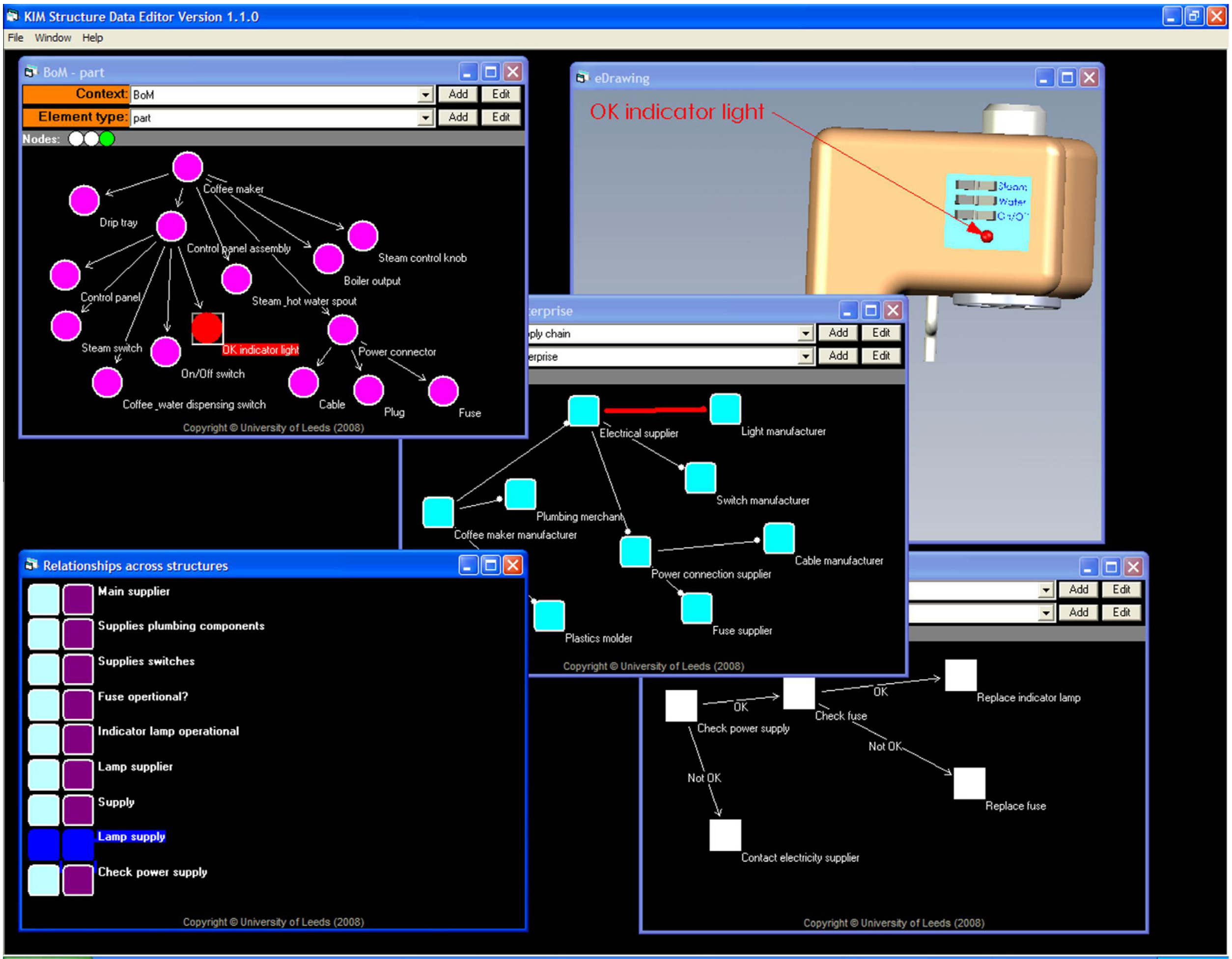

Fig. 15. support for maintenance activities – node to arc relationships.

and tools. Bochnig et al. [10] provide a similar generic PSS development process as does Laurischkat [36] who then describes a general architecture for PSS-CAD systems. A common theme in these and other PSS development processes lies in the decomposition of the development process into series of process steps and stage gates which require key information elements. Information requirements elicited from representative PSS development processes, methods and tools are discussed in Section 6.1 with a view to informing the design of underlying PSS information support systems. Like product-CAD (P-CAD), industry strength PSS information support systems are likely to include PSS-CAD systems from multiple vendors used across organisational and project boundaries. The role of the representation scheme introduced in this paper is to support these information flows, especially at the later stages of the PSS development process, what Bochnig et al. refer to as "IPS^2 architecture specifications and definitions of product service modules", or the concept development and detailing stages of Nøhr Hinz et al.'s framework for PSS development. The efficacy of the representation scheme introduced in this paper is compared with similar schemes available in literature in Section 6.2.

6.1. Information requirements of PSS development processes, methods and tools

Approaches developed in product definition communities can be used as a basis for the definition of information requirements for PSS development processes. Bochnig et al. [10] specify a collection of thirteen requirements for a "general IPS^2 data model". The focus of the data model introduced in [10] covers two later stages of their general PSS development process and includes configuration, embodiment and detailed definition of PSS-CAD solutions. A detailed comparison of Bochnig et al.'s data model and the representation scheme introduced in this paper against selected requirements is given in Table 3.

Bochnig et al.'s other requirements are not included in Table 3 because detailed comparison between the models is not feasible given the available information. For this reason, these requirements are discussed here.

6.1.1. Represent all the different elements of IPS^2 [4]

As with a number of Bochnig et al.'s requirements, further detail and precision in their formulation is needed to enable a detailed evaluation of different solutions. For example, like the first requirement in Table 3 ('*Support the entire planning and development process across all process phases*'), systematic assessment against this requirement needs a clearer definition of what constitutes 'all the different elements'. If the key elements are taken to be product–service modules where a given module is either a product or a service element then the representation scheme in this paper could be used to support the definition of service modules in a

[4] 'IPS^2' is used in the headings below because they are quoted from [10]. 'PSS' is used in the text to give consistency with the rest of this paper.

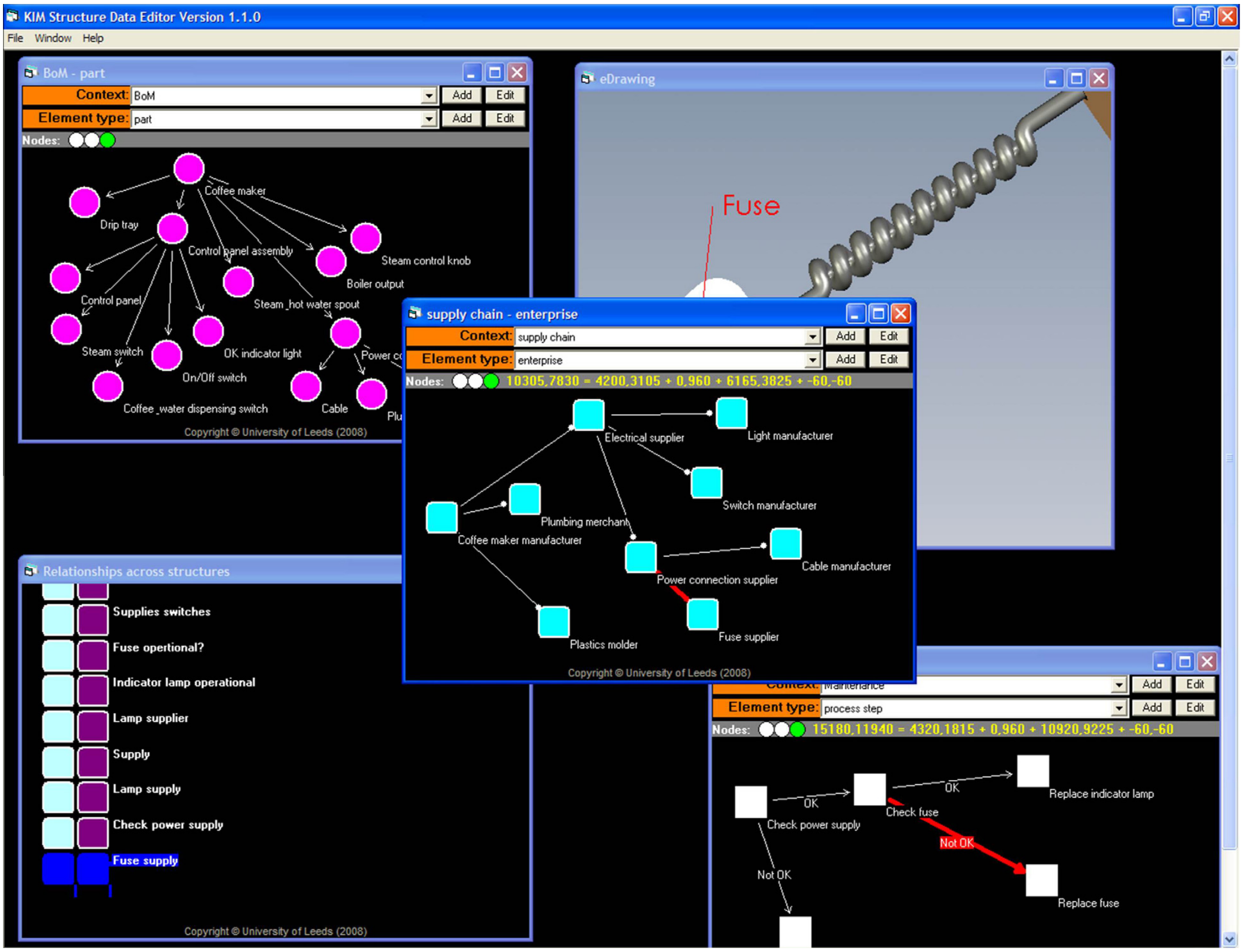

Fig. 16. arc to arc relationship.

manner that is compatible with existing product definition schemes.

6.1.2. Allow for combining IPS^2 elements to modules and thus modularisation of IPS^2

More detail is needed of the kinds of modules and the nature of relationships between them for a detailed assessment against this requirement. Bochnig et al.'s model includes a '*Product–Service module*' entity that is a composition of a '*Product Element*' and a '*Service Element*' but the description of what constitutes a Product–Service module, how modules are selected and how modules are related needs further elaboration. Learning from the product definition community indicates that substantially more detail is needed before reliable data models can be created [67]. In this paper, the idea of a module is not explicitly covered but comparable functionality to Bochnig et al. could be achieved through the addition of a *PSS module* subtype to the *item* entity although more detail would be needed to determine appropriate attributes.

6.1.3. Be generic, not domain specific AND Preferably be simple, not too complex AND Be flexible and extensible

These requirements relate to general issues to consider in the development of any information model. Trade-offs, and associated benefits and drawbacks, between being generic and specific are widely reported and clearly visible in the STEP family of standards and associated literature. Without a clear application domain, which neither this paper nor Bochnig et al. provide, it is not possible to comment on whether an appropriate trade-off has been made in either case. Requirements for simplicity, flexibility and extensibility are obviously desirable in the design of information models. Batini et al. [7] used three criteria in their comparison of database schemas (minimal, complete & correct, and understandable). These were extended by McKay et al. [41]) for product data models to include conceptual (in an ANSI/SPARC three layer model sense), extensible (able to accommodate new information requirements without corrupting existing instances) and structural integrity. Both this paper and Bochnig et al. introduce conceptual models with core entities that promise structural integrity and extensibility but neither is evaluated against these criteria. The population of the model presented in this paper coupled with the use of the model to support a computer implementation provide confidence that the model meets all three of Batini et al.'s requirements. Similar assertions for Bochnig et al.'s model would be more difficult to support because a systematic population of the model is not provided.

6.1.4. Support different viewpoints

Both Bochnig et al. and this paper include examples that are derived from different viewpoints but the range of viewpoints supported is not discussed in either case. Implementation in a database management system that supports the ANSI/SPARC three layer model would enable the definition of application viewpoints but detailed information requirements for specific viewpoints are outside the scope of both papers. Any data model will support

Table 3
Bochnig's information requirements for PSS definitions.

Requirement (direct quotes from [10])	Commentary	[10] general IPS^2 data model	Data model presented in this paper
Support the entire planning and development process across all process phases	This requirement is aspirational but not testable, primarily because delivering this demands a clear definition of what constitutes "the entire planning and development process across all process phases" which is not yet evident from the literature	Supports the definition of IPS^2 architectures comprised of product–service modules where a given product–service module is either a product or a service element	Supports the definition of PSS architectures comprised of product and service elements
		The model is presented across stages of an IPS^2 process but ways in which instances will be related to specific stages of specific development processes are unclear	Relationships between PSS definitions and PSS development process elements can be defined as instances of the *process_step* entity
Allow for development of concrete IPS^2 instances for particular customers		Model does not explicitly support the identification of specific customers but this might be possible using features of the file system within which an implementation would sit	Model includes support for the identification of customers and their association with process steps that may be parts of PSS offerings
Allow the management of data for IPS^2 authoring	Again, management requires a process and tools; for a data model the requirement would be that data needed to support these processes and tools can be captured	Neither model includes management data. However, using STEP-like interpretation methods it would be possible to associate STEP management resources to elements of the data models and, through these resources, include management data in instances	
Allow the modelling of the interdependencies between the IPS^2 elements (essential due to the focus on an integrated development)	Both models support the definition of relationships between PSS elements but a systematic assessment against this requirement would need more detail on the kinds of interdependencies to be represented	Data model supports the definition of instances that are collections of product service modules. Relationships between modules are implicit. Details of the cardinalities of relationships between modules are provided but not the kinds of relationship that can be defined	Data model supports the definition of relationships between service and process elements. Two kinds of relationship are explicit in the model: composition and connection. Cardinalities of these are not limited because this depends on the number of relationship entity instances that are defined. The paper demonstrates how these core relationship types can be used to support the definition of substitution relationships and considers their use to support the definition of PSS variants
Carry along results across phase borders	This demands the definition of data exchange interface requirements	The entities provided cover all phases of the process presented but the efficacy of the model against this requirement is not addressed	The data model presented in this paper could be used as an underlying conceptual model for PSS data exchange formats and standards

the definition of instances but more detailed understanding of specific application views, and the PSS development activities each is intended to support, is needed to enable the development, verification and validation of proposed PSS representation schemes and ontologies. Substantial further work is needed in this area and could begin by building on research on PSS development processes, methods and tools cited elsewhere in this paper.

6.1.5. Allow traceability of results from preceding phases and process steps

Learning from product definition indicates that support for this requires management type data within the PSS data model and links between the PSS definition and PSS development process workflows. The data model presented in this paper does not address this requirement but anticipates future development of PSS lifecycle management systems as parallels to widely used Product Data and Lifecycle Management systems.

6.1.6. Allow links between partial models

Existence dependence is an important factor in creating solutions that address this requirement and is elaborated later in this section.

6.2. Data models for PSS definition

A number of authors distinguish between processes and activities. For example, (Nøhr [52] relate generic process to PSS development activities supported by individual PSS-CAD systems that they review and Shimomura et al. [60] use PSS development activities to structure a discussion of emerging PSS design and development tools such as service engineering methods and their integration with engineering simulation tools. In the future, both PSS development processes and the PSS process definitions that result from them could be used to inform the development of work systems; Alter [3] distinguishes between processes and activities in a meta-model for the definition of work systems. These work systems will be underpinned by information systems and the representation schemes embedded within them.

The majority of representation schemes for PSS definition available in the literature are in the form of data models. As with any model, validation and verification are important aspects of their development. McKay et al. [41]) identify two ways in which data models might be validated: through instantiation and application. The data model introduced in this paper has been validated using both methods (see Sections 4 and 5 respectively). In contrast to literature reviewed, the emphasis of this paper lies in the data model being reviewed rather than the PSS-CAD system to be supported. One limitation of all papers is the lack of benchmark PSS/IPS^2 examples that can be used to compare the capabilities of different models and systems or the ontologies that underpin them. In this section we introduce a benchmark example based on a case study introduced by Umeda et al. [72] (see Fig. 17) to compare the functionality of the data model introduced in this paper (see Fig. 18) and a range of recently published PSS data models. The example was chosen for the following reasons:

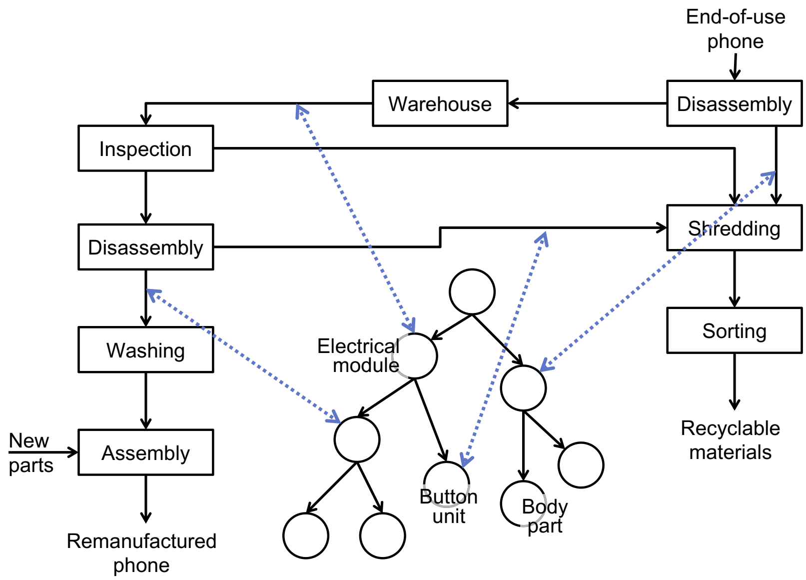

Fig. 17. Benchmark example derived from literature (adapted from Umeda et al. [72]).

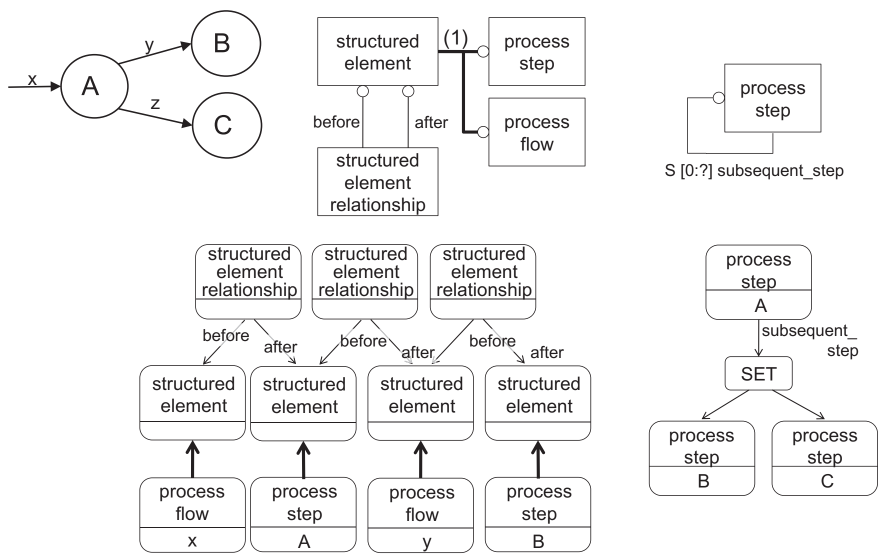

Fig. 18. Example highlighting the implications of existence dependence.

– it was created independently of the research reported in this paper;
– it is relatively small but includes the complexities that typically occur in PSS;
– it includes both product and service elements;
– it is the kind of definition that might result from a PSS development process.

Key features of the example are that it includes a product structure, a service structure and relationships between the two in the form of links between elements in the product structure and relationships in the process structure. In addition, as is often the case in process definitions, the process begins with a flow into an initial process step (in this case, an end-of-use phone into the Disassembly process step). Table 4 provides a comparison of the data model introduced in this paper with a selection of recently published models.

The following meta-models were selected for this comparison: Alter [3] (focussed on the part of the model related to product/service), Averbeck et al. [6], Bochnig et al. [10], and Uhlmann and Bochnig [70]. Each of these models satisfies two criteria: (i) it relates to later stages of a PSS development process where the goal is to create a PSS definition and (ii) the data model is presented in sufficient detail for a comparison to be carried out. The first criterion led to the exclusion of a number of models that relate to early PSS development, such as [50]. The second criterion led to the exclusion of a number of works that are very relevant to this paper such as that by Orawski et al. [54], where the paper includes example instances but with insufficient detail of underlying meta-models used to define them. Similarly, Komoto and Tomiyama [32] describe the integration of data from Service CAD with a lifecycle simulator which, in a full implementation, would require a digital PSS definition, but the paper focusses on the PSS CAD system rather than the underlying meta models that might support such

Table 4
A comparison of the data model introduced in this paper with a selection of recently published models.

Requirement	Averbeck et al. [6]	Alter [3]	Bochnig et al. [10]	Uhlmann and Bochnig [70]	This paper
Physical parts	Out of scope – model is for information flows in service networks	*Product/service* entity	*Product Element* entity[a]	*Product* entity	*Physical artefact* entity[a]
Part-whole relationships between physical parts		A self-referencing composition attribute of the *Product/service* could support these kinds of relationship	Self-referencing attributes of each subtype of the *Product Element* entity with a cardinality of '*' could support these kinds of relationships between product elements of a given type. [b]	A self-referencing attribute of the *Product* entity with a cardinality of '*' could support these kinds of relationships	*Composition* entity in an instance where *items* of type *physical artefact* are related to each other
Process steps	*Process Model* entity	*Process* entity which is linked to the *Product/ service* entity through the *Activity* entity	*Process* entity, a subtype of the *Service Element* entity	*Process step* entity	*Process step* entity
Flows between process steps	By design, the model supports information flows in service networks	Self-referencing attributes of the *Process* (Alter), *Process* (Bochnig et al.) and *Process step* (Uhlmann and Bochnig) entities, each with a cardinality of '*', could support these kinds of relationship			*Connection* entity in an instance where *items* of type *process step* are related to each other
Process steps and parts may be labelled	None of the models show label attributes but the examples provided in the papers include these labels and there is no structural reason why such attributes could not be added.				
Process flows may be labelled	Information flows are explicitly represented in this model and substantially more detail than a label is supported	It would not be possible to label process flows in instances of the models presented in these papers because, as instances of an attribute, it would not be possible to uniquely identify and refer to individual flows. [c]			It is possible to label process flows in this model because they are explicitly represented (using a *connection* entity)
Relationships between parts in a product and flows in a process structure	Not applicable because the model does not support the definition of products	Achieving this requirement requires that both parts and process flows are represented explicitly so that instances of individual parts and flows can be referred to when being related to each other. Since process flows are represented through attributes this would not be possible [c]			The definition of any kind of relationship (in the scope of the model) with any kind of *structured element* is supported (e.g., see Figs. 14–16)
Process flow can exist without a successor/ predecessor	The cardinalities in the model indicate that an information flow could be defined independently of successors & predecessors	If a process flow is an attribute of a process step then it is not possible to define a process flow independently of the process step to which it is related.			The abstraction of the *item* and *item relationship* entities to the *structured element* entity mean that instances of process flow structures can begin and end with either process flows or steps

[a] Bochnig et al.'s model includes three kinds of product (mechanical, physical and software) but does not elaborate on how they are represented or related to each other. The model in this paper refers only to physical artefacts because this is the coverage of the instances it has been used to define.
[b] It is unclear in Bochnig et al.'s model how product elements of different types might be related to each other without modelling them as separate Product–Service Modules because there is no UML composition or aggregation relationship on the Product Element entity as presented.
[c] If the models in these papers were implemented using relational database technology then the normalisation process is likely to make these attributes explicit, in an intersection table. However, it would not possible to add attributes, such as labels or relationships to other service elements, in the conceptual (in an ANSI/SPARC 3-layer sense) model presented in the papers reviewed here.

systems; and Umeda et al. [72] provide a series of rich instances but little detail of the underlying meta model.

7. Concluding remarks

Digital PSS definitions will be a core part of both future generations of PSS engineering systems (including but not limited to CAD) and the methods used to communicate between such systems. Delivering digital PSS definitions demands representation schemes for PSS and associated PSS ontologies on which representation schemes can be built. These representation schemes and ontologies will include means of capturing both product and service structures, and relationships between them. In this paper, we have demonstrated how representation schemes and ontologies used to represent physical product structures might be adapted to represent service structures in PSS. The use of a common underlying approach to the definition of both service and product elements of PSS results in an inherent compatibility that ensures product and service definitions can be related to each other. In addition, as with product data, the service structures introduced in this paper can be used as a framework to which service-related data, both service and product definition data, can be attached.

The representation scheme introduced in this paper is a general purpose model that needs tailoring to provide PSS definitions for specific application contexts. Its efficacy for a small number of relationship types in PSS definitions has been demonstrated: part-whole, connection, variant and substitution relationships within PSS definitions, and relationships between product, service process and supply chain definitions. As can be seen from the cited literature, a large number of researchers are developing PSS engineering methods and systems that provide such contexts as part of an emerging PSS informatics subject area. The research reported in this paper has the following implications for PSS informatics.

- *A given PSS structure has elements of a kind and relationships of a kind*

The representation scheme in this paper is based on the premise, taken from product structuring research, that a given PSS structure has elements of one type and relationships of one type. In this way of thinking, a PSS definition that includes both flow and decomposition relationships would be regarded as containing two structures superimposed on each other. A benefit of considering these as different structures lies in the clarity it provides in creating PSS definitions; a limitation is the lack of a coherent theoretical foundation for the identification of different kinds of relationship that occur in PSS definitions. An emerging subject area that requires further investigation lies in the nature of relationships across product and service elements of PSS; this will be necessary to support Meier et al's definition of industrial product service systems [46] where product and service aspects of the PSS interfere with each other rather than being separate but related entities. In practice the service and product structures of a given PSS have many touch points. Further research is needed to identify the full range of relationships that occur in PSS through their lives. Results of such research could be embedded into ontologies that could form theoretical foundations for future generations of digital PSS definition.

- *Both elements and relationships in PSS structures are best represented as first class objects*

A key characteristic of the representation scheme in this paper is that elements and relationships in PSS structures are represented as first class objects. Keeping both elements and relationships as first class objects ensures flexibility in creating relationships across and within PSS definitions. For example, in the implementation shown in Section 5, relationships between the product CAD model and bill of materials are achieved through relationships that exist independently of either related entity; each relationship is a pair of references to the related entities and neither related entity is treated as a primary or dominant entity. An important benefit of this approach is reduced sensitivity of PSS definitions to changes in the underlying representation schemes in third party systems (such as CAD) which are an inevitable consequence of developments in the technologies on which they are built. In addition, avoiding hard-wiring a dominant viewpoint into a PSS definition increases the long term viability and flexibility of the PSS definition itself.

- *Digital PSS definitions can capitalise on learning from the digital product definition community*

Digital PSS definitions are intended to support PSS development and engineering processes. Like product development and engineering processes, PSS processes will need PSS definitions to support technical activities and information to support the management of such processes. In this paper we have shown how approaches to the definition of technical product data can be applied to PSS definitions. Maintaining compatibility with product definition where possible will allow methods and models to be reused in the development of PSS definitions. For example, the STEP integration methods and management resources could be applied to PSS definitions which, in the longer term, could ease the application of current Product Data Management (PDM) and Product Life Cycle Support (PLCS) systems to PSS. Similarly, methods currently applied in product development processes, such as the capture of design intent and design rationale, could be adapted for use in PSS definitions.

The long term vision of this research is the establishment of digital PSS definitions and records to underpin through-life support of PSS. A digital service definition will be analogous to a product definition and will form the design definition of service elements of PSS. Such definitions will be subject to comparable administrative requirements as a product design, such as change, versioning and effectivity, and they will need a comparable structure, such as the one introduced in this paper. Given such representations it becomes possible to build engineering design tools that can be applied both to service elements of PSS and to PSS as whole engineering systems. An important challenge in the foreseeable future is likely to lie in the integration of PSS definitions that result from the different PSS CAD systems that are being developed. The representation scheme introduced in this paper is compatible with the STEP standard (widely used in the communication of product data) and the service process definitions it can support could be used as starting points for the development of STEP-like application protocols for the exchange of PSS-related product data.

Acknowledgments

The research reported in this paper was carried out through a series of projects including the S4T and KIM projects. The Support Service Solutions: Strategy and Transition (S4T) project was jointly funded by EPRSC and BAE SYSTEMS through Grant No. EP/F038526/1. The KIM project, Immortal Information and Through-Life Knowledge Management project, was an EPRSC/ESRC Grand Challenge project (EPSRC (Engineering and Physical Research Council) Project (EP/C534220/1 and ESRC (Economic and Social Research Council) Grant RES-331-27-0006). Although not directly

related, the research drew on learning from KIM case studies carried out with ABB and Rolls-Royce.

References

[1] ABB, ABB Full Service® Executive Briefing, ABB Limited. Available: <http://library.abb.com/GLOBAL/SCOT/SCOT288.nsf/VerityDisplay/C99BD10938B9E6D0852571F80065CD08/$File/FS%20executive%20briefing_Sept2006.pdf> (accessed April 2010).

[2] V. Agouridas, H. Winand, A. McKay, A. de Pennington, Early alignment of design requirements with stakeholder needs, Proc. Inst. Mech. Eng.: Part B – J. Eng. Manuf. 220 (2006) 1483–1507.

[3] S. Alter, Metamodel for service analysis and design based on an operational view of service and service systems, Service Sci. 4 (2012) 218–235.

[4] G. Annamalai, R. Hussain, M. Cakkol, R. Roy, S. Evans, A. Tiwari, An ontology for product–service systems, in: J. Hasselbach, C. Hermann (Eds.), Proceedings of the 3rd CIRP International Conference on Industrial Product Service Systems, Springer, Braunschweig, Germany, 2011, pp. 231–236.

[5] Austhink, Rationale™, Austhink Software, <http://rationale.austhink.com>, 2010 (Accessed April 2010).

[6] A.-K. Averbeck, T. Bernhold, S. Bräuer, R. Knacksted, M. Matzner, Towards a reference model of information exchange and coordination in facility management networks, in: H. Meier (Ed.), Product–Service Integration for Sustainable Solutions: Proceedings of the 5th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Bochum, Germany, Berlin, Heidelberg, 2013, pp. 107–120.

[7] C. Batini, M. Lenzerini, S. Navathe, A comparative analysis of methodologies for database schema integration, ACM Computing Surveys 18 (1986) 323–364.

[8] J. Becker, D. Beverungen, R. Knackstedt, Reference models and modeling languages for product–service systems – status-quo and perspectives for further research, in IEEE Proceedings of the 41st Hawaii International Conference on System Sciences, 2008, pp. 105–114.

[9] B.J. Berkeley, A. Gupta, Identifying the information requirements to deliver quality service, Int. J. Service Indust. Manage. 6 (1995) 16–35.

[10] H. Bochnig, E. Uhlmann, H. Nguyen, R. Stark, General data model for the IT support for the integrated planning and development of industrial product–service systems, in: H. Meier (Ed.), Product-Service Integration for Sustainable Solutions: Proceedings of the 5th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Berlin Heidelberg, Bochum, Germany, 2013, pp. 521–533.

[11] R.H. Bracewell, S. Ahmed, K.M. Wallace, DRED and design folders: a way of capturing, storing and passing on knowledge generated during design projects. ASME Design Automation Conference, ASME, Salt Lake City, Utah, USA, 2004, pp. 235–246.

[12] H.-J. Bullinger, K.-P. Fahnrich, T. Meiren, Service engineering-methodical development of new service products, Int. J. Prod. Econom. 85 (2003) 275–287.

[13] S. Cavalieri, G. Pezzotta, Product–service systems engineering: state of the art and research challenges, Comput. Indust. 63 (2012) 278–288.

[14] R. Cuthbert, D. Mcfarlane, A. Neely, A framework for service information requirements, in: Irene C.L. Ng, G. Parry, Peter J. Wild, Duncan Mcfarlane, Paul Tasker (Eds.), Complex Engineering Service Systems: Concepts and Research, Springer, 2011, pp. 183–196.

[15] R. Cuthbert, A. Tiwari, P. Ball, A. Thorne, D. McFarlane, Investigating the role of information on service strategies using discrete event simulation, in: Irene C.L. Ng, G. Parry, Peter J. Wild, Duncan Mcfarlane, Paul Tasker (Eds.), Complex Engineering Service Systems: Concepts and Research, Springer, 2011, pp. 197–214.

[16] C. Date, An Introduction to Database Systems, Addison-Wesley Pub. Co, 1995.

[17] A. Dill, C. Schendel, Implications of new institutional economic theory for PSS design, in: T. Sakao, T. Larsson, M. Lindahl (Eds.), Industrial Product Service Systems (IPS^2), Linkoping University Press, Linkoping University, Sweden, 2010, pp. 43–49.

[18] A. Doultsinou, R. Roy, D. Baxter, J. Gao, A. Mann, Developing a service knowledge reuse framework for engineering design, J. Eng. Des. 20 (2009) 389–411.

[19] Engine Group, Engine Service Design, <http://www.enginegroup.co.uk>, 2009 (accessed November 2009).

[20] H. Gebauer, R. Krempl, E. Fleisch, Service development in traditional product manufacturing companies, Eur. J. Innov. Manage. 11 (2008) 219–240.

[21] S. Fließ, M. Kleinaltenkamp, Blueprinting the Service Company Managing Service Processes Efficiency, J. Bus. Res. 57 (4) (2004) 392–404.

[22] Y.M. Goh, M. Giess, C. McMahon, Facilitating design learning through faceted classification of in-service information, Adv. Eng. Inform. 23 (2009) 497–511.

[23] T. Hara, T. Arai, Y. Shimomura, T. Sakao, Service CAD system to integrate product and human activity for total value, CIRP J. Manuf. Sci. Technol. 1 (2009) 262–271.

[24] C. Hepperle, R. Orawski, B. Nolte, M. Mortl, U. Lindemann, An integrated lifecycle model of product–service-systems, in: T. Sakao, T. Larsson, M. Lindahl (Eds.), Industrial Product Service Systems (IPS^2), Linkoping University Press, Linkoping University, Sweden, 2010, pp. 159–166.

[25] ISO10303-1 1994. Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles.

[26] ISO10303-41 2005. Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated Generic Resource: Fundamentals of product description and support.

[27] ISO/TR10303-12 1997. Industrial automation systems and integration – Product data representation and exchange – Part 12: Description methods: The EXPRESS-I language reference manual.

[28] K.-J. Kim, C.-H. Lim, J. Lee, D.-H. Lee, Y.S. Hong, K.-T. Park, Generation of concepts for product–service system, in: T. Sakao, T. Larsson, M. Lindahl (Eds.), Industrial Product Service Systems (IPS^2), Linkoping University Press, Linkoping University, Sweden, 2010, pp. 201–210.

[29] Y. Kim, W. Sang, J.-H. Lee, D. Han, H. Lee, Design support tools for product–service systems, in: S.J. Culley, B.J. Hicks, T.C. McAloone, T.J. Howard, J. Clarkson (Ed.), Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, vol. 1, The Design Society, 2011, pp. 288–298.

[30] D. Kindstrom, C. Kowalkowski, Development of industrial service offerings: a process framework, J. Service Manage. 20 (2009) 156–172.

[31] H. Komoto, Computer Aided Product Service Systems Design: Service CAD and its Integration with Life Cycle Simulation, PhD Thesis, Delft University of Technology, 2009.

[32] H. Komoto, T. Tomiyama, Integration of a service CAD and a life cycle simulator, CIRP Ann. – Manuf. Technol. 57 (2008) 9–12.

[33] P. Kroes, A. Meijers, Special Issue of Studies in History and Philosophy of Science on Dual Nature of Technical Artefacts, Part A 37 (1) (2006) 1–4.

[34] S. Kundu, A. McKay, P.G. Dawson, Understanding information requirements in product–service systems design, in: T. Sakao, T. Larsson, M. Lindahl (Eds.), Industrial Product–Service Systems (IPS2) Proceedings of the 2nd CIRP Industrial Product Service Systems (IPS2) Conference, Linkoping University Press, Linkoping University, Sweden, 2010, pp. 141–149.

[35] S. Kundu, A. McKay, A. de Pennington, N. Moss, N. Chapman, Implications for engineering information systems design in the product–service paradigm, in: Y.U. Shozo Takata (Ed.), 14th CIRP Conference on Life Cycle Engineering: Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses, Waseda University, Springer, Tokyo, Japan, London, 2007, pp. 165–170.

[36] K. Laurischkat, Computer-aided service design for the development of product–service systems – motivation and benefits, in: H. Meier (Ed.), Product–Service Integration for Sustainable Solutions: Proceedings of the 5th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Bochum, Germany, Berlin Heidelberg, 2013, pp. 547–560.

[37] C.S. Lee, Y.G. Chen, J.C. Ho, P.F. Hsieh, An integrated framework for managing knowledge-intensive service innovation, Int. J. Services Technol. Manage. 13 (2010) 20–39.

[38] C. Lovelock, J. Wirtz, Services Marketing: People, Technology, Strategy, Prentice Hall, Upper Saddle River, NJ, USA, 2007.

[39] N. Maussang, P. Zwolinski, D. Brissaud, Product–service system design methodology: from the PSS architecture design to the products specifications, J. Eng. Des. 20 (2009) 349–366.

[40] D. McFarlane, Service Information Systems, DSI Workshop, Pittsburgh, PA, USA, 2007.

[41] A. McKay, M.S. Bloor, A. de Pennington, A framework for product data, IEEE Trans. Knowledge Data Eng. 8 (1996) 825–838.

[42] A. McKay, A. de Pennington, Towards an integrated description of product, process and supply chain, Int. J. Technol. Manage. 21 (2001) 203–220.

[43] A. McKay, F. Erens, M.S. Bloor, Relating product specifications and product definitions in a product model, Res. Eng. Des. 2 (1996) 63–80.

[44] A. McKay, D.N. Hagger, C. Dement, A. de Pennington, Relationships in Product Structures 7th Workshop on Product Structuring, The Design Society, Sweden, 2004.

[45] A. McKay, S. Kundu, A blueprint for engineering service definition, in: Irene C.L. Ng, G. Parry, Peter J. Wild, Duncan Mcfarlane, Paul Tasker (Eds.) Complex Engineering Service Systems: Concepts and Research. Springer, 2011, pp. 215–232.

[46] H. Meier, R. Roy, G. Seliger, Industrial product–service systems-IPS2, CIRP Ann. – Manuf. Technol. 59 (2010) 607–627.

[47] T. Meiren, C. Van Husen, R. Karni, Laboratory support for service engineering and design. American Society of Mechanical Engineers – ASME: Engineering Systems Design and Analysis Conference, ESDA 2008, July 7–9, ASME, Haifa, Israel, 2008.

[48] P. Muller, F. Schulta, R. Stark, Guideline to elicit requirements on product–service systems, in: T. Sakao, T. Larsson, M. Lindahl (Eds.), Industrial Product Service Systems (IPS^2), Linkoping University, Linkoping University Press, Sweden, 2010, pp. 109–116.

[49] S. Mumford, Function, Structure, Capacity, Studies History Philos. Sci. 37 (2006) 76–80.

[50] Y. Nemoto, F. Akasaka, Y. Shimomura, Knowledge-based design support system for conceptual design of product–service systems, in: H. Meier (Ed.), Product–Service Integration for Sustainable Solutions: Proceedings of the 5th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Bochum, Germany, Berlin, Heidelberg, 2013, pp. 41–52.

[51] I.C. Ng, G. Parry, P.J. Wild, D. McFarlane, P. Tasker (Eds.), Complex Engineering Service Systems: Concepts and Research, Springer, London, 2011.

[52] H. Nøhr Hinz, N. Bey, T. Mcaloone, Timing and targeting of PSS methods and tools: an empirical study amongst academic contributors, in: H. Meier (Ed.), Product–Service Integration for Sustainable Solutions: Proceedings of the 5th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Bochum, Germany, Berlin, Heidelberg, 2013.

[53] V. Ojanen, M. Lanne, M. Reunanen, H. Kortelainen, T. Kassi, New service development: success factors from the viewpoint of fleet asset management of industrial service providers, 15th International Working Seminar of Production Economics, 2008.
[54] R. Orawski, D. Kammerl, S.A. Schenkl, C. Hepperle, M. Mörtl, Element visualization ElViz: graphical representation of planning-relevant dependencies between PSS-elements, in: H. Meier (Ed.), The Philosopher's Stone for Sustainability: Proceedings of the 4th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Tokyo, Japan, Berlin, Heidelberg, 2012.
[55] F. Patterson, Provoking students into thinking, Compak 1 (2007) 79–82.
[56] M. Rese, W. Strotmann, J. Gesing, M. Karger, How to educate customers about industrial product service systems – the role of providing information, in: T. Sakao, T. Larsson, M. Lindahl (Eds.), Industrial Product Service Systems (IPS^2), Linkoping University, Sweden, 2010.
[57] T. Sakao, T. Larsson, M. Lindahl (Eds.), Proceedings of 2nd International Conference on Industrial Product Service Systems (IPS2), Linkoping University, Linkoping University Press, Sweden, 2010, pp. 27–33.
[58] T. Sakao, Y. Shimomura, E. Sundin, M. Comstock, Modeling design objects in CAD system for service/product engineering, Computer-Aided Des. 41 (2009) 197–213.
[59] G. Schuh, G. Gudergan, Service Engineering as an Approach to Designing Industrial Product Service Systems. 1st CIRP Industrial Product–Service Systems (IPS2) Conference, Cranfield University, UK, 2009.
[60] Y. Shimomura, F. Akasaka, Toward Product-Service System Engineering: New System Engineering for Pss Utilization, in: H. Meier (Ed.), Product-Service Integration for Sustainable Solutions: Proceedings of the 5th CIRP International Conference on Industrial Product-Service Systems, Springer-Verlag, Berlin Heidelberg, Bochum, Germany, 2013, pp. 27–40.
[61] P. Simons, Parts: A Study on Ontology, Clarendon Press, Oxford, UK, 2003.
[62] Z. Stanicek, M. Winkler, Service systems through the prism of conceptual modeling, Service Sci. 2 (2010) 112–125.
[63] S. Stepney, T. Hoverd, Reflecting on Open-Ended Evolution. ECAL, MIT Press, Paris, France, 2011.
[64] F. Sturm, F. Josefiak, M. Schubert, Combining Products and Services into Solutions – Evidence from the Capital Goods Industry, in: Proceedings of the 14th International Conference on Concurrent Enterprising, Lisbon, Portugal, 2008, pp. 309–316.
[65] R. Tassi, Service Design Tools: Communication Methods Supporting Design Processes, <http://www.servicedesigntools.org> (accessed April 2010), 2009.
[66] A. Tidstam, J. Malmquist, Information Modelling for Automotive Configuration, NordDesign 2010, Göteborg, Sweden, 2010.
[67] A. Tidstam, J. Malmqvist, Obstacles and development of support for translation of configuration rules, in: U. Lindemann, V. Srinivasan, Y.S. Kim, S.W. Lee, J. Clarkson, G. Cascini (Ed.), Proceedings of the 19th International Conference on Engineering Design (ICED13): Design for Harmonies, vol. 4, The Design Society, Seoul, South Korea, 2013, pp. 119–128.
[68] T. Tomiyama, Service engineering to intensify service contents in product life cycles, in: Proceedings of the Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing, IEEE Computer Society, Tokyo, Japan, 2001, pp. 613–618.
[69] T. Tomiyama, Y. Shimomura, K. Watanabe, A note on service design methodology, ASME 2004 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, September 28–October 2, 2004, ASME, Salt Lake City, Utah, USA, 2004, pp. 251–260.
[70] E. Uhlmann, H. Bochnig, PSS-CAD: an approach to the integrated product and service development, in: H. Meier, (Ed.), The Philosopher's Stone for Sustainability: Proceedings of the 4th CIRP International Conference on Industrial Product–Service Systems, Springer-Verlag, Tokyo, Japan, Berlin Heidelberg, 2012, pp. 61–66.
[71] UKCEB, LCIA4 Release Candidate 2- SDM/SOM: Contract Types (Shown by Type), T.U.M.O. Defence (Ed.), 2010.
[72] Y. Umeda, S. Fukushige, E. Kunii, Y. Matsuyama, LC-CAD: A CAD system for life cycle design, CIRP Ann. – Manuf. Technol. 61 (2012) 175–178.
[73] T. van Gelder, The Rationale for Rationale™, Law, Probab. Risk 6 (2007) 23–42.
[74] P.E. Vermaas, W. Houkes, Technical functions: a drawbridge between the intentional and structural natures of technical artefacts, Studies History Philos. Sci. Part A 37 (1) (2006) 5–18.
[75] E.G. Welp, H. Meier, T. Sadek, K. Sadek, Modelling approach for the integrated development of industrial product–service systems, Manuf. Syst. Technol. New Frontier Part 14 (2008) 525–530.
[76] D.C. Wynn, A.M. Maier, P.J. Clarkson, How can PD process modelling be made more useful? An exploration of factors which influence modelling utility, in: D. Marjanovic, M. Storga, N. Pavkovic, N. Bojcetic (Ed.), Proceedings of DESIGN 2010: the 11th International Design Conference, The Design Society, Glasgow, Dubrovnik, Croatia, 2010, pp. 511–522.
[77] V.A. Zeithaml, A. Parasuraman, L. Berry, Delivering Quality Service: Balancing Customer Perceptions and Expectations, The Free Press, New York, 1990.
[78] W. Zhao, J.K. Liu, OWL/SWRL representation methodology for EXPRESS-driven product information model Part II: Practice, Comput. Industry 59 (2008) 590–600.
[79] H. Zhu, J. Gao, D. Li, D. Tang, A web-based product service system for aerospace maintenance, repair and overhaul services, Comput. Industry 63 (2012) 338–348.