INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

AN INTEGRATED PRODUCT, PROCESS AND RATIONALE MODEL FOR THE PROVISION OF THROUGH-LIFE INFORMATION IN PRODUCT SERVICE SYSTEMS

Alison McKay[1], Saikat Kundu[1], Peter Dawson[2], Alan de Pennington[1]
(1) School of Mechanical Engineering, University of Leeds, Leeds, LS2 9JT, UK
(2) The Keyworth Institute, University of Leeds, Leeds, LS2 9JT, UK

ABSTRACT
Many organisations are moving from the supply of products to the delivery of product-service systems. As a result, the potential value of product-related information created through the whole life-cycle, from design through to operation and support, of large complex products is increasing dramatically. This transition is creating new requirements for information solutions that enable access to reliable product data, both from design and reflecting how the product has changed over its life, in a form that is well suited to the purpose to which it is to be put.
This paper focuses on changes in engineering information system requirements caused by the need to support both physical artefacts and associated services. Key characteristics that differentiate services from artefacts and their impact on the design of engineering information systems that facilitate the delivery of product-service systems are outlined and approaches to integrating product and service definitions are explored. We argue that the strategy of establishing future-proofed product information systems to support future life-cycle processes will fail in situations where the information requirements of the processes are not anticipated far enough in advance. To address this weakness the paper proposes an integrated product, process and rationale model that allows the definition of multiple product structures for a given artifact. The applicability of the model is demonstrated through a case study that has been implemented in a prototypical software tool. The software tool incorporates e-drawings from a commercially available computer aided design package (Solidworks), design rationale maps from a rationale editor (DRed) and allows users to visualise structures and relationships within and across product, process and enterprise network structures.

1 INTRODUCTION
As increasing numbers of companies move from product delivery to the provision of through-life support services, the need to be able to access reliable product definition data, both from design and reflecting how the product has changed over its life, increases. For example, the business model in the aero-engine sector is changing from the provision of physical engines to the provision of power by the hour [1,2]. Consequences of the transition from product to product-service delivery include the role of the physical product changing (for example, from an end in its own right to a means of delivering functionality and performance) and the required scope of engineering information systems extending beyond design and manufacturing to cover the use, maintenance, support, refurbishment and retirement of products.
Where once the design, manufacture and delivery of a product to a customer was a goal in its own right, increasingly these products are now parts of product-service systems where the goal is to deliver supported and operational products through the entire life of the product. As with the physical product itself, networks of organisations are typically involved in the delivery, operation and support of product-service systems. Key challenges in supporting delivery of such systems include the need to deal with individual artefacts (identified, for example, by serial number rather than drawing number) and the need to support and accommodate the activities of a wider range of users and the tasks they carry out. Many of these are far less easy to control and anticipate. In part this is because of the nature of the activities, the people carrying them out and their priorities, and in part it is because they are geographically and organisationally dispersed; increasingly products are operated and services are

delivered in and by global networks of organisations where ownership and location of resources are not necessarily aligned [3].

These characteristics impact the context within which the information systems that support product-service systems will be used and so should be designed for. A specific need is for access to reliable product data, both from design and reflecting how the product has changed over its life. As with any information, the context within which it was created is critical to how it may be used. In contrast to design information, which has typically been through a review process before release, through-life information comes from a wider range of less systematically controlled sources. A key issue in through-life support lies in the fact that through-life information is often based on the product definition provided in an information pack with the product on delivery. This has a number of consequences that have an adverse impact on later life-cycle processes. Firstly, the product definition is typically structured to support product realization processes that need to be completed prior to the delivery of the product to a customer such as design, production and testing. Secondly, the product definition cannot include all information needed since some is only generated during use, after the product definition has been created.

This paper proposes a solution that allows users throughout the life of the product to define product structures that best suit their purposes and then link them into relevant aspects of the product definition: either as-designed or as-defined at an earlier point in the life of the product. It involves superimposing product structures (with associated process, enterprise network structures) onto available product definition data such as shape and life-cycle rationale information. Benefits of the approach include that new product structures can be created to suit the life-cycle stage where it is to be used, the people who will work with it and the systems with which it will interact. The inclusion of enterprise network, process and rationale information allows the context within which information was created to be captured.

2 STRUCTURE OF PAPER

The first part of the paper focuses on how requirements for the design of engineering information systems are changing as the transition from product to product-service delivery progresses. Key features of service products, that distinguish them from physical products, are outlined in Section 3.1 and theories that underpin, or can be used to explain, current product definition schemes are discussed in Section 3.2. Information system related problems related to the need for long term knowledge retention are summarised in Section 3.3 along with an information architecture that might contribute to addressing these problems. A key element of this architecture lies in the information model that underpins the life-cycle definition that is introduced in Section 3.3. An integrated product, process and rationale model designed for this purpose is presented in Section 5. The applicability of the model is illustrated using the case study described in Section 4; the case study is defined in terms of the integrated model in Section 6 and implemented using a software prototype that enables the capture of through-life product information in Section 7.

3 BACKGROUND

The research reported in this paper draws from two distinct bodies of previous work, on the definition of service and physical products (see Sections 3.1 and 3.2 respectively), with a view to addressing the problem of long term retention of product-related knowledge and information (see Section 3.3).

3.1 Key characteristics of service products

Johne and Storey [4] provide a review of literature related to new service development and identify five key characteristics that distinguish service products from physical products: intangibility, perishability, non-ownership, inseparability of production and consumption, and variability.

- **Intangibility:** Services are predominantly performances of actions rather than physical objects. For this reason they are more difficult to perceive using the physical senses than physical products.
- **Perishability:** Services must be consumed as they are provided. In general, they cannot be saved, stored, returned or carried forward for later use or sale.
- **Non-ownership:** Largely as a result of their intangibility and perishability, customers do not obtain ownership of services, instead, they experience their delivery.

- **Inseparability of production and consumption:** Service products are typically produced and consumed at the same time - consumption cannot be separated from the means of production.
- **Variability:** Service product quality is subject to variability because services are delivered by people to people. Two dimensions of variability have been identified [4,5]: the extent to which delivery standards vary from a norm, and the extent to which a service can be deliberately varied to meet the specific needs of individual customers. Parallels between these variabilities and those of physical products can be drawn. The extent to which a delivered service varies from a norm is akin to the extent to which a dimension on a physical product varies with respect to its nominal dimension and tolerance band on an engineering drawing. On the other hand, the variation of a service to meet the needs of individual customers has parallels with mass customisation and the delivery of customised products.

Engineering information systems to support the life-cycles of product-service systems need to accommodate these distinctions without compromising the need to preserve commonalities between physical and service products.

3.2 The definition of physical products

Current 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 & Houkes, in their ICE (**I**ntentionalist,**C**ausal-role,**E**volutionist) theory [6], 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 authors discuss the distinction between function, behaviour and capacity of physical artefacts. Mumford [7] 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.

Design rationale, as captured using tools such as the D-Red software tool [8], is a means by which designed physical structures might be related to intended functional structures. Design intent, for example as captured using advanced requirements management techniques [9], enables intended functional structures to be related to stakeholder intent and so aspects of what Vermaas & Houkes refer to as use plans. On designed physical structures, Simons [10] uses mereology to provide a theoretical basis for the definition of physical product structures, of which Bills of Materials are the most common manifestation. McKay et al [11] propose a collection of relationships needed to support the definition of physical products. Three groups of relationships needed for the definition of a product are identified:

- those needed to describe a product at a point in its life-cycle and time;
- those needed to support configuration management; and
- those needed to support product realisation.

Relationships needed to describe a product at a point in its life-cycle and time are composition, constitution, inherence, quantification, designation and qualification. Fuller details of each of these kinds of relationship can be found in the source paper; a brief summary, for the purposes of this paper, is provided in this paragraph. Composition relationships are part-whole relationships from mereology; a common occurrence of these relationships are the part-part relationships in Bills of Materials. Constitution relationships identify the medium through which a product is realised; a given product can have only one constitution. Inherence relationships relate products to their properties, for example, the shape of a part or the hardness of a material. Quantification relationships relate product definitions with physical quantities. For example, a quantification relationship could be used to say how many engines were on a given aeroplane in a Bill of Materials (BOM) or the size of a dimension in a shape definition. Designation relationships relate products with names or codes that are used for identification purposes. For example, a designation relationship could be used to assign a part number to a part. Qualification relationships relate parts of product definitions with conditions that govern their existence. For example, a part of a product may have been replaced during the life of the product. A user wishing to look at the product definition defining the product that was delivered to the customer after manufacture would need to see the product definition that included details of the original part whereas a maintenance engineer might wish to see a product definition that included details of the replacement part; a qualification relationship would be used to ensure that each person saw the correct product definition for their intended purpose.

Relationships needed to support configuration management are equivalence, alternation, variation, order and transformation. Fuller details of each of these kinds of relationship can be found in the source paper; again, for the purposes of this paper, a brief summary is provided here. Equivalence relationships are used to define substitutes whereas alternation relationships capture options. Variation relationships represent diversity. For example, they define relationships between baselines (for example, the generic BOM of a product family) and variants of the family. Order relationships are used to define the relative positions of entities with respect to each other. For example, a process plan has an ordered sequence of steps to define, for example, an assembly process. Transformation relationships show development over time. They capture states, for example, developments of a product through its life.

Relationships needed to support entity realisation are articulation, factorisation and consolidation. Again, fuller details of each of these kinds of relationship can be found in the source paper but a brief summary is provided here. Factorisation relationships allow physical entities to be conglomerated; for example, sheet metal parts might be nested when cut from a sheet of material. In contrast, consolidation and articulation relationships allow physical entities to be aggregated and disaggregated respectively. For example, a disassembly process needs a product structure that defines the initial assembly and at least two product structures that define the disassembled pieces; articulation relationships enable the definition of disaggregations such as this. Consolidation relationships, on the other hand, allow parts to be aggregated, for example, into assemblies.

If services are regarded as products, or parts of product-service products, then a number of questions arise given the discussion in this section. These include the following.

- What are the intended functional structures of service products and how might they be represented?
- Which product definition relationships apply to service products and how might they be represented in an engineering information system?
- For through-life support, are additional kinds of relationship needed to support life-cycle processes after product realisation?

3.3 The problem of long term knowledge retention

Long term knowledge retention involves the keeping (curating) of data so that it can be used in the future for purposes that have not necessarily yet been defined. For engineering data it inevitably includes the applications and computational operating platforms (hardware, software) that were used to create and develop the data since these are essential in enabling raw data to be seen as it was seen when previously created or used. The situation is made more complicated by the fact that a product definition is rarely self-contained, it usually refers explicitly to reference data such as standards and library data, and it is defined in formats that require readers to be trained in a range communication methods such as engineering drawing standards and conventions.

These technical challenges of providing archived product data, with guaranteed fidelity, to end users mean that, for the foreseeable future, many people operating and supporting large complex products will be working with presentations of product definition data (for example a B-Rep generated from a CAD model) rather than the structured data (for example the CAD model from which a B-Rep is generated) from which the presentation will have been generated. A second important factor in the use of engineering data is awareness of the context within which it was created. This includes provenance, organizational, social/political, economic, legislative/regulatory, environmental situations at the time and the reasoning behind decisions that led to the data in question[1].

Current approaches to the design of engineering information systems are based on definitions of (and assumptions about) predicted future usage scenarios, users, usage contexts and information needs [12,13]. Meta-models are built to enable the capture and use of the data concerned. Whilst proven in a range of scenarios, this approach is limited when the future uses of the data are unclear or unanticipated, or when information that is relevant through-life was not known during product

[1] A wider discussion of these issues is available at the web site of the Atlantic Workshop on Long Term Knowledge Retention (LTKR) held at the University of Bath, UK, 12-13 February 2007 http://www.ukoln.ac.uk/events/ltkr-2007/programme.html.

realisation. During product realisation, it is only possible to capture information that exists or is planned or intended. Functional information is a good example of this problem. Vermaas and Houkes [6] argue that products have capacities (things that they can be used for) but only those that were intended during product definition can be captured as functions; functions are special kinds of capacities that were intended by the designer. However, all capacities are potentially as important as each other during use and many products are used for purposes other than those intended – either deliberately or inadvertently. As a result, design intent/rationale/process capture cannot capture all capacities since some emerge through use and do not exist at the design phase of the life-cycle. The focus of this paper is on providing access to product data in these kinds of circumstances. Such data is likely to have been preserved in some form of long term knowledge retention system but not defined for a specific use. The research in this paper envisages a "what you see is what you get" type scenario. In such a scenario a product definition, typically a shape-based model using today's product definition technology, will have through-life information superimposed on it: analogous to embedding in shape computation but the things being embedded are life cycle data rather than shapes [14].

An information architecture showing the proposed approach is given in Figure 1. Current approaches to the definition of product data are depicted in Figure 1(a); the proposed information architecture adds the life-cycle definition given in Figure 1(b) and the references represented by the arrows between the two parts of Figure 1.

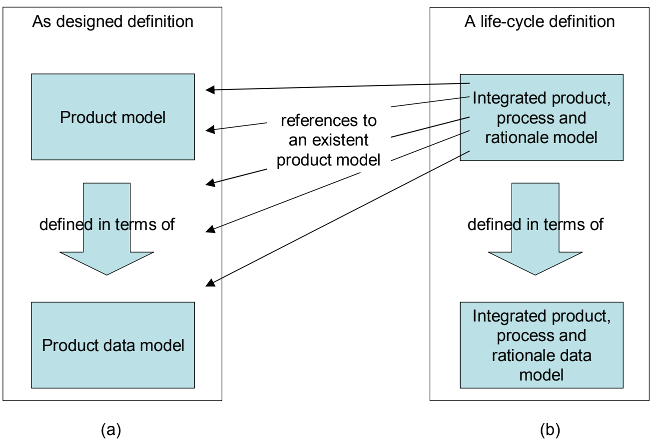


Figure 1: Proposed information architecture

It can be seen that the requirements on the design definition are limited in that it needs to be possible to refer to its elements but there is no need for the design definition to change. This is important in situations where, for example, a design has been approved and certified and so cannot be easily changed. In the proposed solution, a new life-cycle model takes the form of an integrated product, process and rationale model which refers to an existent product definition: initially the as designed definition but potentially also a previous life-cycle model in the form of Figure 1(b). As the life cycle proceeds further life-cycle definitions may be added that refer to both the design definition and earlier life-cycle definitions.

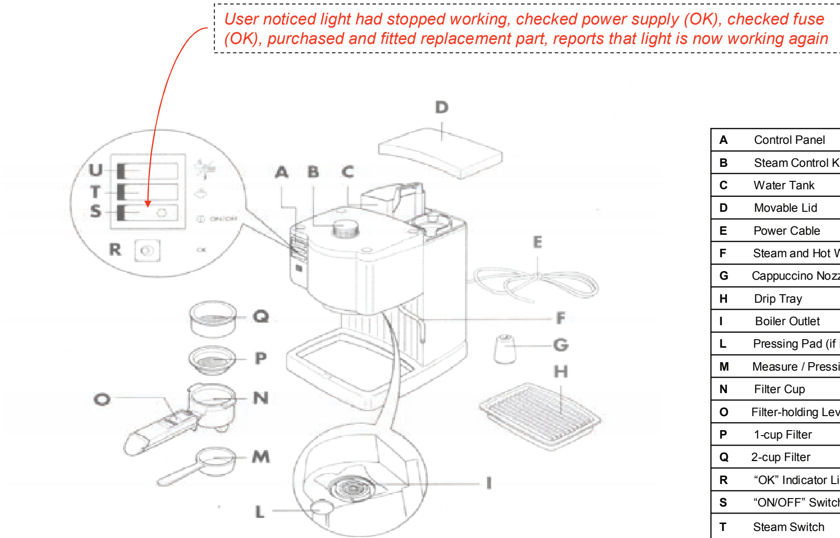
4 CASE STUDY

A case study that will be used in the rest of this paper is given in Figure 2. The data shown in black is the as defined definition and the data in the dashed box is life-cycle data that would be captured through an integrated product, process and rationale model. It can be seen that, although the shape is the thing that is attributed, the data added is not shape data, rather it is product identification, process identification and through-life rationale.

There are two unlabelled parts in Figure 2:

- the Coffee Machine Base Unit onto which the other parts are assembled; and
- the Switch Housing which houses the switches and indicator lights (Parts R – U).

Given that this paper is focusing on the text in the dashed box in Figure 2, only the Control Panel is relevant here. The main body of the machine is composed of a number of parts including the Control Panel (A). The Control Panel itself is composed of the following parts: the Switch Housing, the “OK” Indicator Light (R), the “ON/OFF” Switch with Indicator Light (S), the Steam Switch (T) and the Coffee and Water Dispensing Switch (U).



A	Control Panel
B	Steam Control Knob
C	Water Tank
D	Movable Lid
E	Power Cable
F	Steam and Hot Water Spout
G	Cappuccino Nozzle (if included)
H	Drip Tray
I	Boiler Outlet
L	Pressing Pad (if included)
M	Measure / Pressing Pod
N	Filter Cup
O	Filter-holding Lever
P	1-cup Filter
Q	2-cup Filter
R	“OK” Indicator Light
S	“ON/OFF” Switch & Indicator Light
T	Steam Switch
U	Coffee & Water Dispensing Switch

Figure 2: Example exploded assembly (reproduced from a de Longhi user manual) with superimposed life-cycle data

5 AN INTEGRATED PRODUCT, PROCESS AND RATIONALE DATA MODEL

A product, process and rationale model is presented in Figure 3. It uses the general purpose model first presented in [15] to integrate the product and process structures and a simplified version of the *part aspect schema* to relate these to design rationale. In this section of the paper, key concepts from the model are described. These are illustrated through population with the case study data in Section 6. The software demonstrator described in Section 7 also includes an enterprise network, in the form of a supply chain. Enterprise network definitions are realized using a parallel approach to that for product and process steps; they have been excluded from Figure 3 to save space. The grey dashed lines in Figure 3 are used to delineate three key areas of the model: design rationale, linkages to product definition and product, process and enterprise network structuring.

5.1 Design rationale

The design rationale model is based on the results of an analysis of the information requirements of the DRed software, as presented in [16]. It should be noted that this model captures key DRed concepts but not the full richness that it is possible to define in a DRed model. Three key concepts are shown in Figure 3: an *issue*, a *proposed answer* and an *argument* that either supports or counters a given answer. The heavy lines in Figure 3 represent specialization relationships between concepts. For example, *supporting argument*s and *counter argument*s are kinds of *argument*. Similarly, there are two kinds of *proposed answer*: *accepted* and *rejected*. Each *argument* is related to a *proposed answer* that it either supports or counters, and each *proposed answer* is related to a given *issue*.

5.2 Linkages to product definition

An analysis of published DRed charts, such as those in [16], resulted in the observation that the textual definitions of *issue*s, *proposed answer*s and *argument*s often refer to parts of products and processes. This is akin to the aspect of part schema that was introduced in [17] and is reproduced in this part of Figure 3. The original purpose of the *part aspect schema* was to provide as structured a context as possible for definitions of product requirements; the context of each product requirement was defined as either an aspect of a part (for example, the height or material of the water tank (Part C) in Figure 2) or a relationship between two aspects of two parts (for example, a diametrical clearance between the

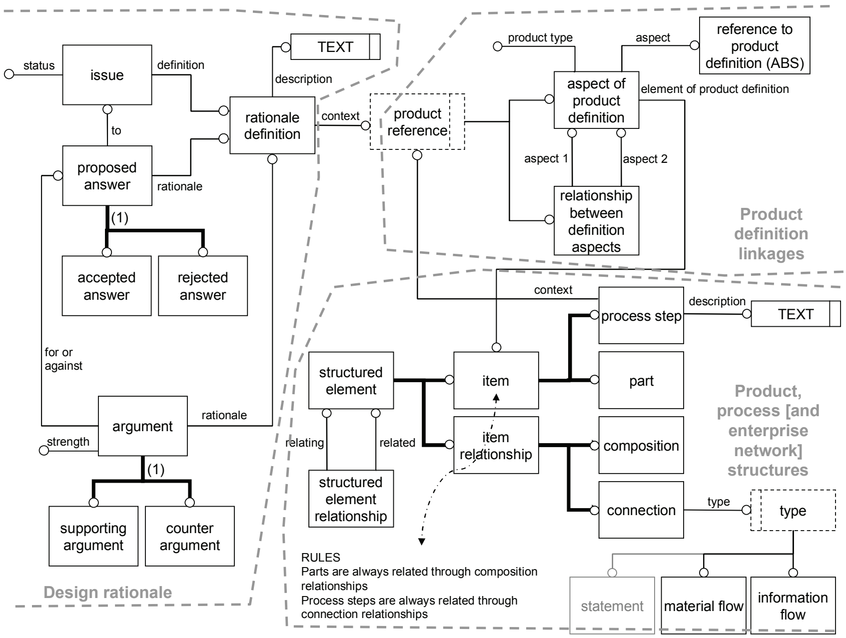


Figure 3: Integrated product, process and rationale model

filter cup (Part N) and each of the filters (Parts P and Q) in Figure 2). A detailed discussion of the different kinds of *aspects of part*, and issues in their implementation, is provided in [17]. Since the research reported in [17] was carried out (in 1990-1994), the capabilities of shape-based computer aided design systems have developed substantially and multi-dimensional geometry has become a norm in commercially available systems. However some of the issues in the implementation of the *part aspect schema* remain. Specifically, some of the *part aspects* that are referred to do not exist in any form in the product definition; they are emergent properties of the definition. For example, it is unlikely that a diametrical clearance between the filter cup (Part N) and each of the filters (Parts P and Q) in Figure 2 would be explicitly represented in the product definition, especially if the people creating the definition were unaware that it would be needed in the future. Since emergent properties do not exist in the product definition it is not possible to refer to them.

The information architecture introduced in this paper provides a means by which emergent properties relevant to a particular task can be superimposed onto an existent product definition. The types of entity being related come from the product definition and are not therefore needed in the integrated model; for this reason the specializations of the *aspect of part* entity in the *part aspect schema* are not included. The *part type* attribute of the *aspect of part* entity allowed the user to state whether the part was a part of the product of interest or the a part of a product with which it might interact. For example, a key dimension for users of the coffee machine in Figure 2 is the height of the cup that is to be filled with coffee; although not a part of the coffee machine, for ease of use, the height of the cup must be less than the distance between the bottom of the filter cup (Part N) and the top of the drip tray (Part H) – less than 7.5cm.

5.3 Product, process and enterprise network structures

In this region of Figure 3 it can be seen that the *process step* and *part* entities are defined as specializations of the *item* entity. Rules associated with the *item* entity ensure that *part*s can only be related to each other through *composition* relationships and *process steps* through *connection* relationships. A *process step* is defined by some descriptive text (through its *description* attribute) and a reference to a product (through its *context* attribute). The rationale model is related to the process and product models through a new *rationale definition* entity. This entity replaces the textual definitions of *argument*s, *proposed answer*s and i*ssue*s with a collection of a textual description and a reference to an *aspect of a product definition*. The *aspect of product definition* entity is based on the *aspect of part* entity in the *aspect of part schema* but has been renamed because it refers to items that can be either *part*s or *process step*s. As it stands this allows elements of rationale to be linked to parts and process steps in a product definition; if in the future there is a need to relate elements of rationale to relationships within product definitions then this could be easily achieved by changing the type of the *element of product definition* attribute from an *item* to a *structured element.* Each *composition relationship* refers to two parts where the *parent* part is the part that contains the *child* part; the concept of a composition relationship is based on mereology[2] [10].

A new entity, *reference to product definition*, has been added to allow references to elements of a product definition to be created; this provides the same capability as the aspect attributes of the specialisations of the *aspect of part* entity in the *aspect of part schema* that were removed for the purposes of this research. The *reference to product definition* entity is defined as an EXPRESS-G abstract supertype because its detailed definition can only be created during implementation when details of the product definition to which relationships will be established are known.

Key capabilities of the model lie in its ability to support the definition of multiple product, process and enterprise network structures where a given structure contains elements related by either part-whole or connection relationships. Once such structures have been established, relationships between the structures can be defined where the following kinds of linkage are supported: element to element, relationship to relationship and element to relationship. For example, it is possible to define a product breakdown structure and a supply network and then relate parts in the product breakdown structure to flows [usually represented as relationships between organisations] in the supply network. Given such structures, it is then possible to relate them, through either elements or relationships, to pre-defined data such as digital product definitions and design rationale. These capabilities are illustrated in Sections 6 and 7. The model provides flexibility in the kinds of structure that can be defined; its main limitation arises from this flexibility in that it is achieved through a weak typing system (with respect to the engineering data being defined) that would make it difficult to integrate models defined in terms of it tightly with commonly used engineering applications. For example, the link to the shape model shown in Section 7 is to a Solidworks e-drawing but the relationships are to graphical elements of the presentation rather that the underlying shape definition.

6 ILLUSTRATION OF A USE OF THE INTEGRATED MODEL

The potential applicability of the integrated model is illustrated here through population with sample data taken from the case study given in Section 4. The BoM relevant to this case study, defined as an instance of the data model given in Figure 3, is given in Figure 4 using the EXPRESS-I-G notation [18]. For clarity only the Control Panel, the “OK” Indicator Light (R) and the “ON/OFF” Switch with Indicator Light (S) are included in the EXPRESS-I-G. Note that two parts not in the exploded assembly are included in Figure 4; the fuse and power supply are needed because the text in the dashed box in Figure 2 refers to them. The labels in square brackets in some of the item entity instance boxes (for example, the one labelled with a star in the middle of the figure) are included so that these items can be referred to from other EXPRESS-I-G diagrams later in this section. Three process steps can be inferred from the text in the dashed box in Figure 2: checked power supply (OK), checked fuse (OK), purchased and fitted replacement part. An instance of the data model given in Figure 3 that represents these process steps is given in Figure 5. As in Figure 4, the labels in square brackets in some of the item entity instance boxes are included so that these items can be referred to from other EXPRESS-I-G diagrams. The labels in dashed entity instance boxes are references to the labelled entity instances in

[2] Mereology is the theory of part-whole relationships.

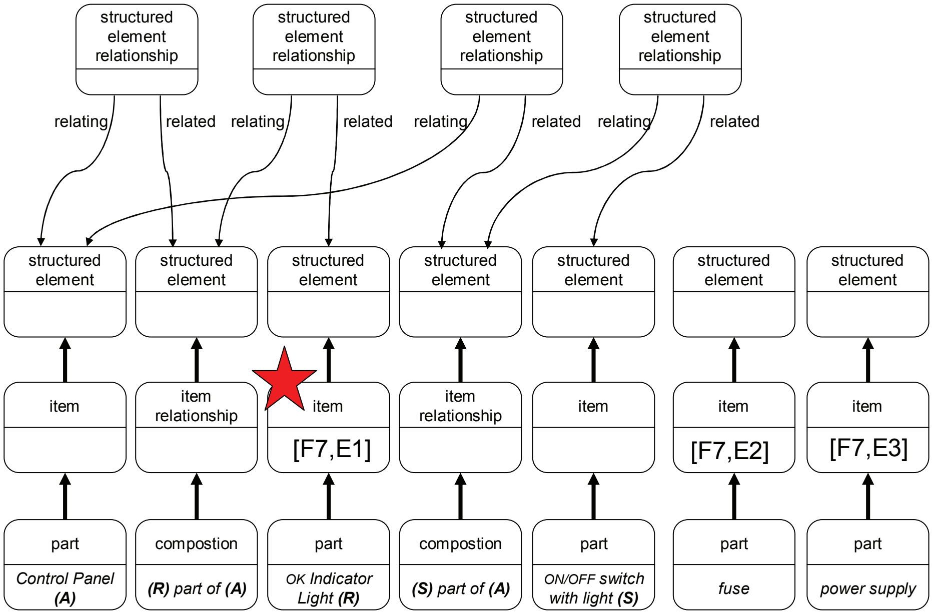


Figure 4: Instance of a fragment of the BoM (Parts A, R and S)

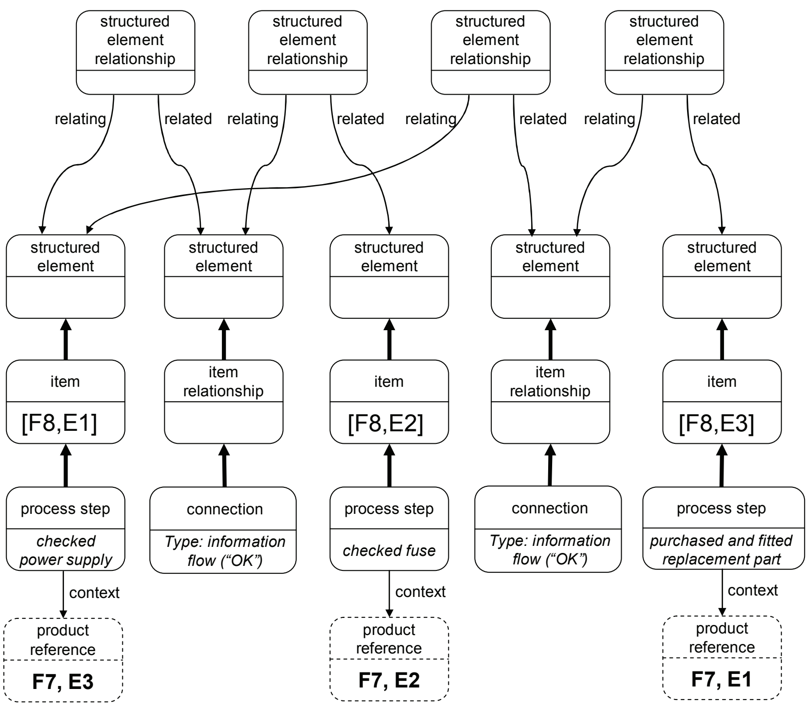


Figure 5: Instance of the process steps implied by the sample data

other diagrams. For example, the process step, purchased and fitted replacement part, refers through the context attribute of the process step entity to the item in Figure 4 that represents the OK Indicator Light in the BoM. Similar references are included to the fuse and the power supply. Some reasoning can also be inferred from the text in the dashed box in Figure 2; this is referred to as life-cycle rationale to avoid confusion with design rationale. For the purposes of this paper an issue (Light has stopped working) has three proposed answers: mend the power supply, mend the fuse, purchase and fit replacement part. An instance of the data model given in Figure 3 that represents this rationale is given in Figure 6. It can be seen from Figure 6 that the rationale refers to both process steps in Figure 5 and elements of the BoM in Figure 4.

7 RESULTS FROM SOFTWARE PROTOTYPE

A software prototype has been created to provide an example of the kinds of decision support tools that product service system developers and operators might use in their daily work and to demonstrate how such tools might be used to enhance knowledge management within and across product service projects. In this section two screen shots from the software tool are presented. The screenshot in Figure 7 shows two windows – the top left-hand corner of the figure shows a BoM structure for the coffee maker (in the form shown in Figure 7) alongside a digital product definition in the form of a Solidworks eDrawing. The two models are linked to each other. For example, in Figure 8, the steam control knob is highlighted in the BoM structure and the corresponding part is highlighted in the eDrawing.

The two structure windows in Figure 8 show a BoM and maintenance process structure, with relationships between the structures shown in the window in the top right-hand corner of the figure. The “check power supply” relationship between the product and process structures is highlighted in this window and the related elements of the product and process structures are highlighted in the corresponding windows. In the software prototype, similar structures have been created for an enterprise network structure (the spare parts supply chain) and relationships between parts in the BoM and flows in the supply chain. The meta-model for enterprise networks in the software prototype (not presented in this paper) takes the same form as the process structure but with organizations as elements and product flows as relationships. Relationships are established between these flows and elements of the BoM.

8 CONCLUDING REMARKS

Although small, the sample population and software prototype presented in this paper illustrate the potential of the product, process and rationale model and demonstrate its capability to support the integration of product, process and rationale information. Together, the software tool coupled with the data model provide a means by which product, process and enterprise network structures can be defined and visualized. The data model represents both elements and relationships in product structures as first class objects. For this reason relationships across different kinds of structures can be defined by relating elements of one structure to elements in another, relationships in one structure to relationships in another and elements in one structure to relationships in another. A next phase in the development of this work is to evaluate it with data from industrial case studies that reflect more realistically the scale and complexities of real-world life-cycle process and product definition data. We anticipate that this could result in vision demonstrator experiments that recreate the narrative of a life-cycle support process and show how through-life rationale and process data might be captured and linked to product definitions in commercially available CAD packages that are used in industry today.

Early research results indicate that, like physical products, service products have both intended functional and designed (but not physical) structures. Current research is exploring the use of product, process and enterprise network structuring techniques to establish a detailed understanding of relationships that occur in service-artefact relationships.

Today, potentially valuable life-cycle information is typically created and owned by a range of organisations and stored in ways that renders it inaccessible to potential beneficiaries. A real opportunity, in moving to the product-service paradigm, lies in the ability to capture life-cycle information as it is created through the delivery of life-cycle services. Understanding of the nature of the relationships between products and services is a key to unlocking this potential and enabling sharing of more information across a wider range of life-cycle stages.

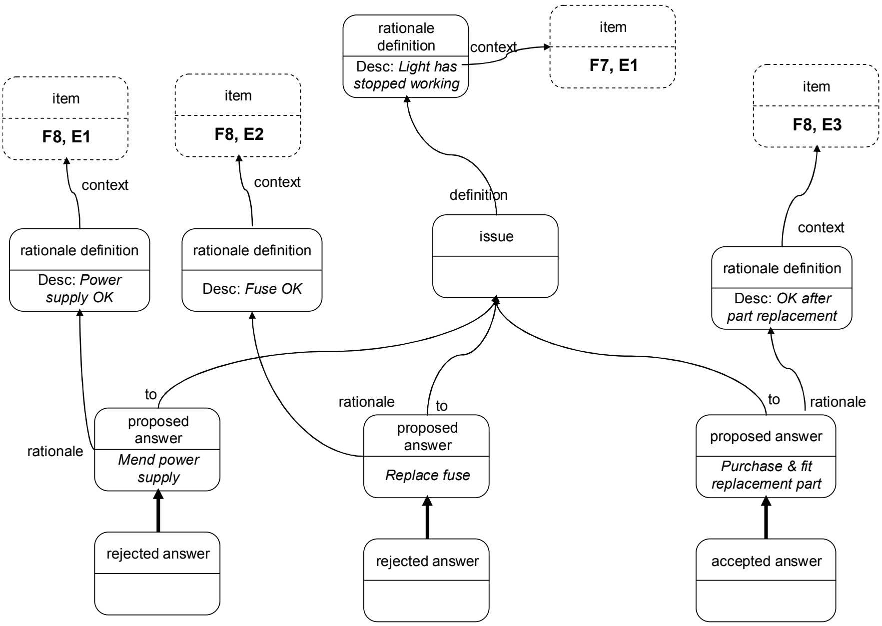


Figure 6: Instance of the life-cycle rationale data implied by the sample data

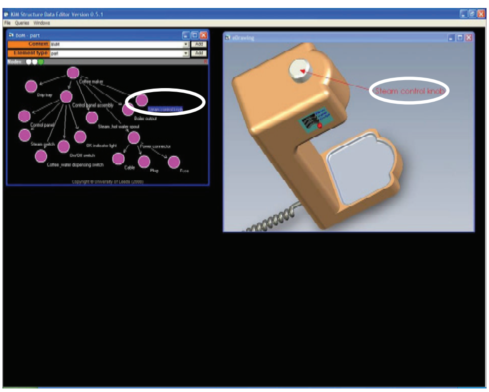


Figure 7: A Bill of Materials and corresponding digital product definition

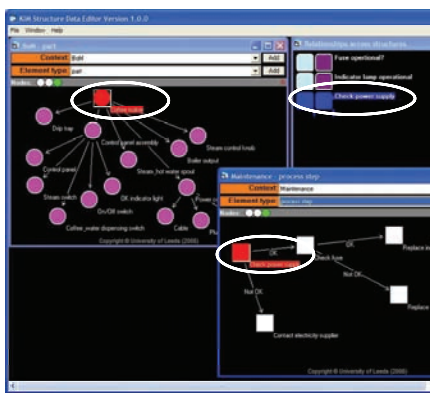


Figure 8: Related Bill of Materials and mainte process structure (element to element relatio

ACKNOWLEDGEMENTS

The paper reports research carried out as part of the Immortal Information and Through-Life Knowledge Management (KIM): Strategies and Tools for the Emerging Product-Service Paradigm 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 papers draws on general learning from KIM case studies being carried out with ABB and Rolls-Royce. Dr R Bracewell at Cambridge EDC provided clarifications that informed the semantic analysis of DRed.

REFERENCES

[1] Wall, R. (2003) Power-by-the-Hour Navy targets E-2C and T-45C engines for a commercial deal to improve reliability. Aviation Week & Space Technology, 158(23), p36.

[2] Anon (2007) 'Power by the Hour': Can Paying Only for Performance Redefine How Products Are Sold and Serviced? Knowledge@Wharton.

[3] Eppinger, S. D. & Chitkara, A. R. (2006) The New Practice of Global Product Development. MIT Sloan Management Review, 47, 22-30.

[4] Johne, A. & C. Storey (1998), New Service Development: A Review of the Literature and Annotated Bibliography. European Journal of Marketing, 1998. 32(3/4): p. 184-251.

[5] Zeithaml, V. and M. Britner, Services Marketing: Integrating Customer Focus Across the Firm. 3 ed. 2003, Irwin, USA: McGraw-Hill. 668.

[6] Vermaas, P.E. & Houkes, W. (2006) Technical functions: a drawbridge between the intentional and structural natures of technical artefacts. Special Issue of Studies in History and Philosophy of Science, 37, 5-18.

[7] Mumford, S. (2006), Function, structure, capacity. Studies in History and Philosophy of Science, 2006. 37: p. 76-80.

[8] Bracewell, R.H., S. Ahmed, & K.M. Wallace. (2004) DRED and design folders, a way of capturing, storing and passing on, knowledge generated during design projects. in Proceedings of DETC'04, ASME Design Automation Conference. 2004. Salt Lake City, Utah, USA.

[9] Agouridas, V., Baxter, J.E., McKay, A., de Pennington, A. (2001) On defining product requirements: a case study in the UK health care sector. in 13th International Conference on Design Theory and Methodology. 2001. Pittsburgh, Pennsylvania: ASME.

[10] Simons, P., (1987) Parts: a study in ontology, Oxford: Clarendon.

[11] McKay, A., Hagger, D,N.N., Dement, D., de Pennington, A., Simons, P. (2004) Relationships in product structures. in 7th Workshop on Product Structuring, The Design Society, Sweden. 2004.

[12] Eastman, C., Lee, G. & Sacks, R. (2002) A new formal and analytical approach to modeling engineering project information processes. International Council for Research and Innovation in Building and Construction, CIB w78 conference 2002. Aarhus School of Architecture.

[13] Eastman, C. & Augenbroe, F. (1998) Product modelling strategies for today and the future. International Council for Research and Innovation in Building and Construction, CIB w78 conference 1998. Stockholm, Sweden.

[14] Stiny, G. (2006) Shape: Talking About Seeing and Doing, MIT Press.

[15] McKay, A. & de Pennington, A. (2001a) Towards an integrated description of product, process supply chain. International Journal on Technology Management, 21, 203-220.

[16] Bracewell, R. (2006) Capturing Designers' Rationale During the Design Process. Knowledge and Information Management: the Challenge of Through-life Support. London, IMechE

[17] McKay, A., de Pennington, A. & Baxter, J.E. (2001c) Requirements management: A representation scheme for product specifications. CAD Computer Aided Design, 33, 511-520.

[18] McKay, A., Erens, F., Bloor, M. S. & Owen, J. (1993) EXPRESS-I-G: A graphical notation for EXPRESS-I. EXPRESS User's Group (EUG'93) Conference.