Proceedings of the 1st CIRP Industrial Product-Service Systems (IPS2) Conference, Cranfield University, 1-2 April 2009. pp32

A Product-Service System Representation and Its Application in a Concept Design Scenario

Y. S. Kim, E. Wang, S. W. Lee, Y. C. Cho
Creative Design Institute, Sungkyunkwan University, 300 Chunchun, Jangan, Suwon, 440-746, Korea
yskim@skku.edu

Abstract
Product-service systems (PSS) can give diverse value provision to consumers reflecting their individual needs, while also addressing multiple issues from manufacturers' viewpoints. We propose graph and ontological representations of PSS, consisting of values, product and service elements, and their relations. PSSs may also be included as subsystems in a larger PSS. We illustrate a case scenario of PSS concept design starting from an existing product. Diverse requirements of stakeholders of the product life-cycle are transferred to persona generation. From state parameters of these personas, we identify operations to improve the personas' values, and implement each operation as a sub-PSS.

Keywords:
PSS Design, PSS Representation, Case Scenario, Visualization and Manipulation of PSS Concept Design

1 INTRODUCTION

In the past decade, product-service systems (PSS) have received much research effort as a means of innovative value proposition through the integration of products and services. The European Union (EU) has sponsored various projects on PSS for sustainable consumption and production [1]. Considerable research has also been done in developing the tools and methodologies for effective development of PSS, as reviewed by Baines *et al.* [2]. Neely has studied the potential advantages of PSS [3], and noted that industrialized countries have shown higher numbers of combined manufacturing and service firms than other countries.

The concept of PSS was first introduced in 1999 by Goedkoop *et al.* to address the challenges of environmental and economical issues [4]. They defined PSS as a marketable set of products and services, jointly capable of fulfilling a client's need. In addition, they addressed several advantages of PSS such as: creation of value for clients through quality and comfort; customization of offers; delivery of offers to clients; decrease in the cost of initial investment through sharing, leasing and hiring; decrease in environmental load; etc. Mont also defined PSS as a system of products, services, supporting networks, and infrastructure that is designed to be competitive, to satisfy customer needs, and to have a lower environmental impact than traditional business models [5][6]. He proposes a theoretical framework for PSS reflecting societal infrastructure, human structures and organizational layouts to enhance environmental values. Manzini and Vezolli present a strategic design approach for sustainability by describing potential benefits of PSS with some examples of eco-efficient PSS [7].

In the area of PSS design, Morelli has studied a methodological framework that considers designers' views [8]. He presents a concrete case study of an urban tele-center, in which the major functions and requirements are first extracted, and then linked to the elements of products and services for the development of a PSS. Aurich *et al.* researched the life-cycle oriented design processes of products and services [9]. They proposed a systematic design process for technical services associated with products, which is later integrated with the product design process. They also introduced the concept of process modularization for integration of product and service design processes by selecting, combining and adapting appropriate process modules [10]. Matzen and McAloone introduced the activity modeling cycle (AMC) model [11] as a tool for conceptualizing the development of PSS, and investigated the effectiveness of the AMC model by conducting a case study on service delivery in the container ship industry. More recently, they describe a structured modeling scheme to differentiate and categorize development tasks during a migration toward service orientation, with a case study of the maritime equipment industry [12].

Shimomura *et al.* have contributed significant research on service engineering [13][14][15][16][17]. They propose the use of receiver state parameters (RSPs) as the underlying representation of values and costs to be managed throughout the service design process. They introduce the service model, which includes the sub-models of flow model, scope model, view model and scenario model. They have also implemented a prototype computer-aided design tool for service design called Service Explorer. Maussang *et al.* have developed a PSS design process by incorporating Shimomura's service model into an engineering product design process, considering functional analysis and agent-based value design [18]. Maussang *et al*. have also proposed the evaluation of PSS during the early design phase using an optimization methodology, considering economical and environmental factors [19]. Meier and Völker identified challenges and opportunities in adapting existing product supply chain techniques to support service supply chains, and illustrated a scheme to maintain supply chain networks autonomously through the application of multi-agent systems [20].

Although considerable research for the effective design of PSS have been conducted, there has not yet been an equivalent effort in developing a comprehensive, machine-understandable representation of PSS itself and its elements, including values, product elements, service

elements, and their relations. To properly support the development of new computer-aided tools and frameworks for PSS design, we see a need for a formal ontological representation of PSS.

This paper is part of an ongoing research effort toward the implementation of an intelligent PSS design framework, which will be an integrated development environment for the PSS design process itself. In this paper, we propose an ontological representation of PSS, which will form a core part of the data model for the PSS design framework. Section 2 describes our approach for conceptual PSS design, using the case scenario of a PSS for a meal assembly kitchen. It also introduces a graph representation of PSS. Section 3 presents an ontological representation of PSS, including product and service elements, their associated values, and relations between them. In Section 4, we illustrate how this PSS ontology could be used to model one sub-PSS of the meal assembly kitchen scenario.

2 GENERATION OF PRODUCT-SERVICE SYSTEMS

2.1 Generation of PSS from a Product

A PSS includes several product elements and service elements that are closely related to other. A schematic diagram of the generation of a PSS from a single product is shown in Figure 1. A product P is sub-divided into its constituent product elements P_i. To these, we may add new product elements P_i, and identify new values V_k. Then, new service elements S_j are added and combined with these product elements so that all values are achieved. The resulting system is a PSS.

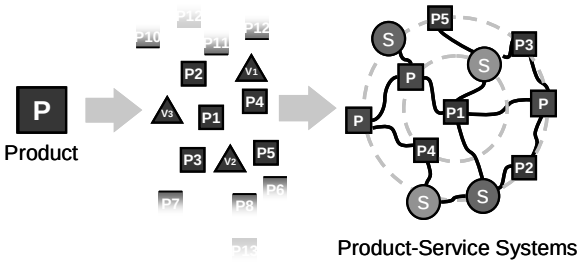

Figure 1. Generation of PSS

2.2 Case Scenario for Conceptual Design of PSS

In this section, we illustrate our approach to the conceptual design of a PSS by presenting a case scenario. Starting from an existing product, which is a typical kitchen oven, we will generate a PSS for a meal assembly kitchen.

A meal assembly kitchen is an innovative concept in meal preparation that moves the meal assembly process out of people's kitchens and into specially equipped stores. It offers the benefits of simplified menu planning, support and instruction in cooking, and elimination of shopping, food preparation, and cleaning. Other social benefits include increased opportunities to engage in mutual family activities and make new friends.

We develop the conceptual design case scenario by the following steps. First, we generate the user requirements of an oven from the oven's life cycle. We identify stakeholders throughout the oven's life cycle, and by considering their requirements and needs, we extract stakeholders' values. Second, from these values, we generate several personas, where each persona represents a target group of users that share one set of values. Third, we identify operations that could improve customers' values, and model each operation as a sub-PSS of product and service elements that achieve the value improvement. Finally, these sub-PSSs are merged and expanded to obtain the complete PSS for the meal assembly kitchen. These steps are further explained in the following subsections.

Proposition of Values and Requirements

We adopt Matzen and McAloone's activity modelling cycle (AMC) model [11] to represent stakeholders and their activities and values. In accordance with the AMC approach, we divide the entire life cycle of the oven into three phases: pre-usage, during usage, and post-usage. In each phase, we list activities and associated stakeholders, values and requirements.

The pre-usage phase includes the production, selling, buying, delivering and installation of the oven itself. The during-usage phase includes normal cooking and cleaning activities. The post-usage phase includes removal of the oven, its disposal (including recycling), and elimination of traces that the oven may leave behind in the environment, such as odors. For each activity, we obtain the values and requirements associated with the stakeholders through usage observation, interviews and surveys.

Persona Generation

A persona is an abstraction of a target user who should be satisfied with the designed PSS; hence it represents a group of users, all of whom are characterized by the same set of values. The set of personas is chosen to cover a significant portion of the intended customers of a PSS. Each persona has diverse and different values, which may be positive or negative. PSS design then proceeds by devising ways to improve the negative values of all personas.

In this case scenario, we have developed three personas: an elderly woman, a housewife who is recently married, and a businessman who lives alone. For example, the businessman persona has one positive value: he knows how to cook with the oven. However, he has four negative values. First, he possesses a second-hand oven that needs frequent attention and maintenance. Second, he does not have proper oven maintenance skills. Third, he has insufficient kitchen space for the oven. Fourth, shopping for ingredients is a burden in distance and time. These four negative values become the targets for value improvement, described next.

Value Improvement and PSS Generation

By considering the negative values defined for all personas, we devise a set of operations that collectively address and overcome all of the negative values. For the case scenario, we devise three operations: *provision of chef's advice*, *provision of wide space and several ovens*, and *management of ovens and ingredients*. Each operation generally improves a different value, and demands different product and service elements.

We implement these three operations as three sub-PSSs, shown in Figure 2, using a graph representation of PSS. Each PSS graph is composed of product elements, service elements, value nodes, and edges. The edges denote the relations among the nodes. This PSS graph representation can quickly show how many product elements, service elements and values exist, and whether any two elements have some relation.

For example, the operation of *Provide wide space and several ovens* is shown as a sub-PSS in Figure 2, lower left. Two values, *V21 wide space* and *V22 variety of ovens*, are identified. These are related to three product elements: *P1 wide space*, *P22 various ovens*, and *P23 utensils*, and two service elements: *S21 provide space for cook* and *S22 provide tools for cook*. These element nodes are joined by edges to make this sub-PSS.

These three sub-PSSs can be expanded by linking them with other sub-PSSs to make the complete PSS, as shown

in Figure 3, using the graph representation. The expansion of the sub-PSS can be made to link the values, product and service elements in the sub-PSS and those in other sub-PSSs. For example, consider the elements *P22: various ovens* and *S22 provide tools for cook*. Since we need many tools for cook such as a burner, micro-oven, mixer, and so on, P22 and S22 can be associated with the sub-PSSs including these tools as elements. In this way, the sub-PSSs are linked together and expanded to make the complete PSS for the meal assembly kitchen.

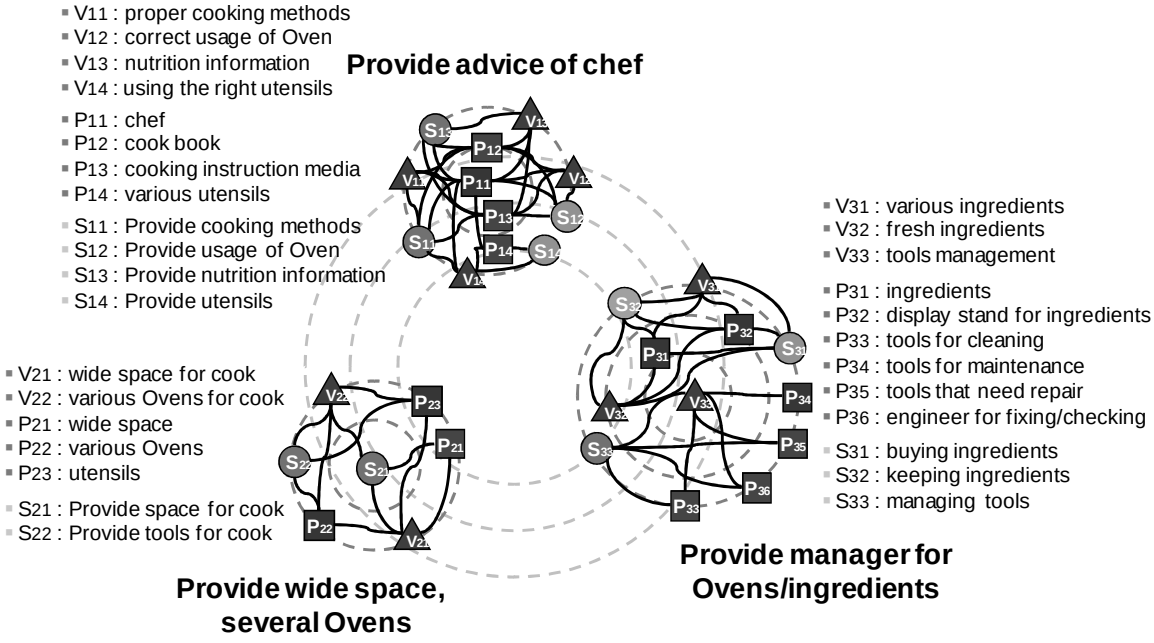

Figure 2. Sub-PSS Graphs

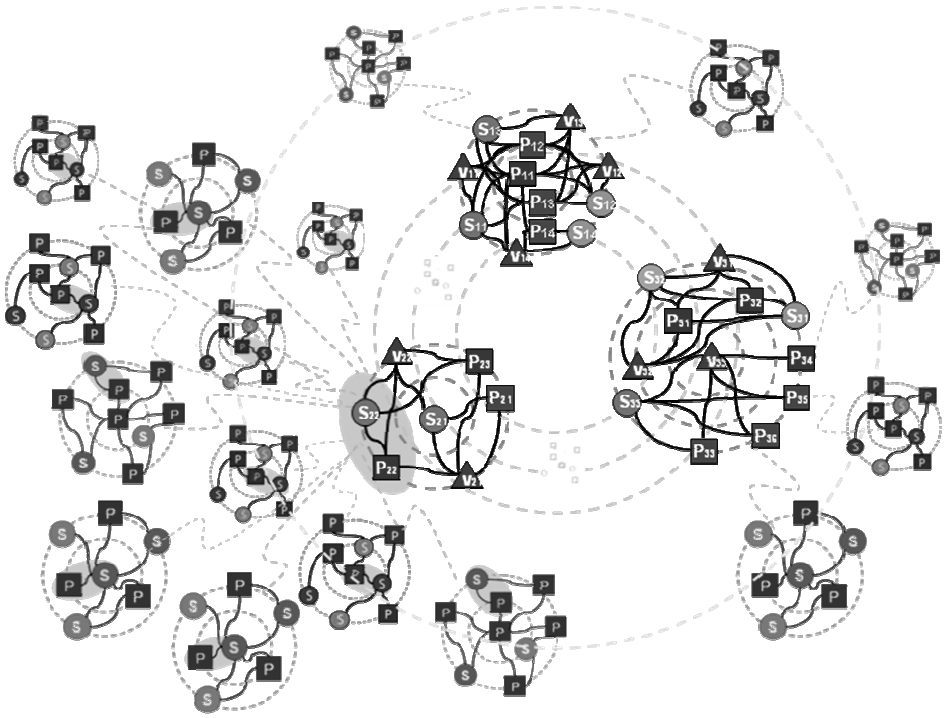

Figure 3. Expanded PSS Graphs

3 ONTOLOGICAL REPRESENTATION OF PSS

The graph representation of PSS, as shown in the preceding sections, provides a simple, visual way to relate entities to each other. This representation could be useful for some applications. However, it omits considerable detailed data. Other non-graph perspectives of PSS data could also be useful for different applications, e.g. tabular representations may be more suitable for editing and sorting. To represent all data of a PSS in a format that could support future automated reasoning applications, we present an ontological representation of a PSS. Our PSS model describes values, product elements, and service elements, and their relations, as shown in Figure 4. We present the ontology in UML format, while we have also modeled it in OWL using Protégé, with conversion to Jess.

3.1 Representation of Value

What is Value?

A key idea in our approach for PSS representation, in both graph and ontological representations, is to model values explicitly as first-class elements, i.e. at the same level of abstraction as for product elements and service elements. "Value" has been variously defined or used in previous research; hence we first clarify exactly what we mean to be a value. We take the definition of value as used in economics: value is *the market worth or estimated worth of products or services* [21][22]. We also take Maussang *et al.*'s approach that values within a PSS are deduced from the needs of stakeholders [18].

Examples of Value in PSS

Human stakeholders are very flexible in perceiving different kinds of value within a PSS; hence, we need a flexible representation. Some examples of values in PSS include *availability of X*, *condition of X*, and *usage rights for X*, where *X* = bicycle [18]. Similarly, in our meal assembly kitchen scenario, we have identified values such as (*availability of) V21: wide space for cook*, as shown in Figure 2. *X* could be any resource, which may be a tangible object, but it is composed with a discriminator such as *availability of* to realize the value. That is, *X* is not, by itself, a value.

Ontological Representation of Value

We model a **Value** class, and its component classes, as shown in Figure 4, left side. A **Value** object has one **ValueNature** discriminator, such as **AvailabilityOf**, which indicates that this value is related to the *availability* of a resource. "Available" implies numerous conditions: that the resource exists, is within easy reach, is in working condition, the receiver has adequate permissions, the receiver must actively get the object himself, etc.

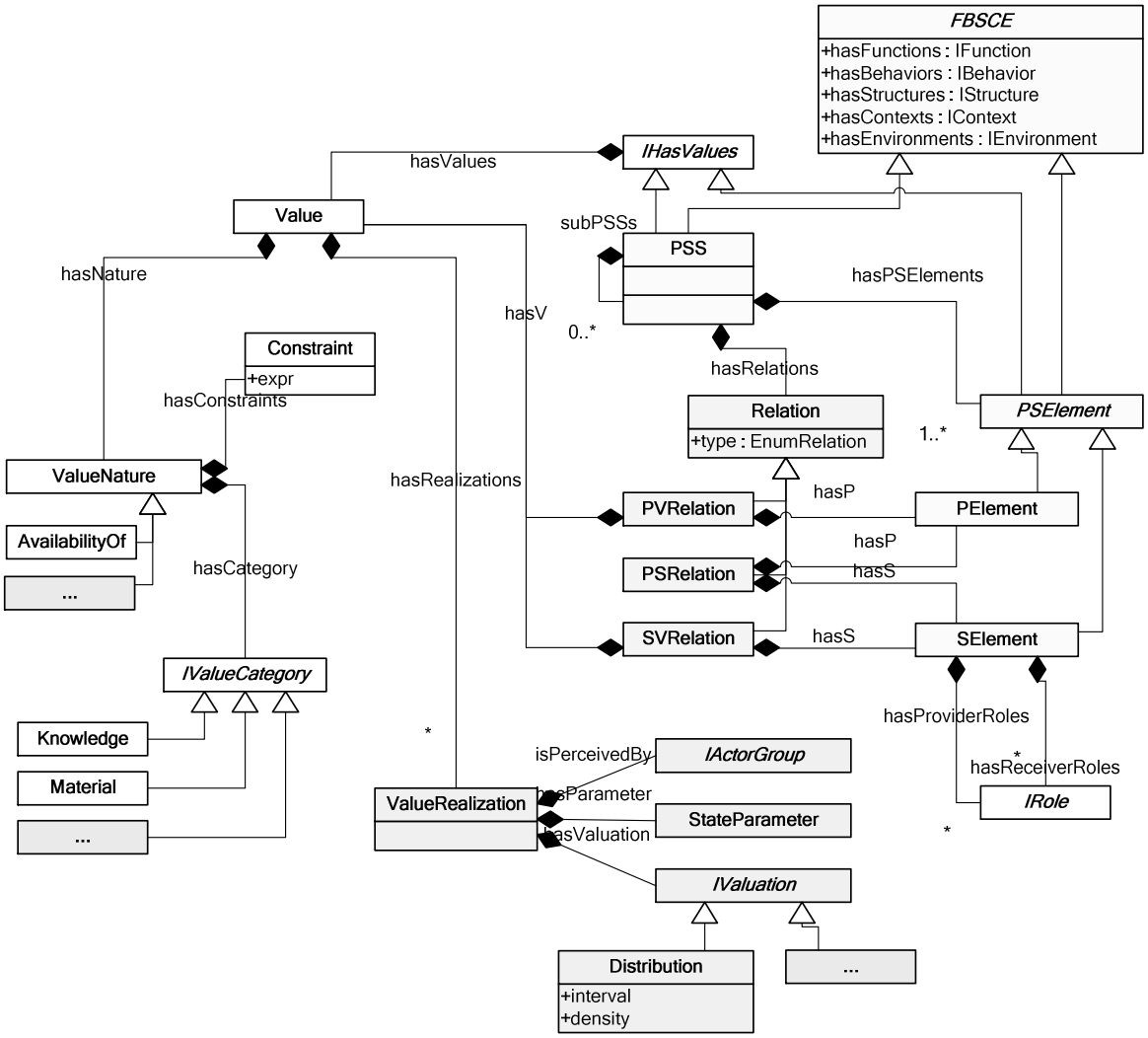

Figure 4. Ontological Representation of Value and PSS

The **ValueNature** object is composed with one ***IValueCategory***, which could describe a tangible resource (such as *ingredients*), or intangible resource (such as *cooking methods*). It also has 0 or more **Constraint**s.

Multiple Realizations of Value

A Value could have an absolute (non-subjective) measurement, such as a time in seconds. However, different stakeholders could perceive or appreciate the value differently. To support many different subjective interpretations of a Value by different actor groups, we introduce a **ValueRealization** class to represent one specific instance of an actor group and its valuation of the Value. A **Value** then has any number of **ValueRealization**s.

Each **ValueRealization** consists of an *IActorGroup*, a **StateParameter**, and an ***IValuation*** that specifies how this group appraises the value. Each state parameter could be quantitative, qualitative, or both. We define one specific kind of valuation as a **Distribution** subclass, which presumes that its state parameter is quantitative, and defines a mathematical function of state value to the actors' degree of satisfaction. While this has some similarity to the approach of Shimomura *et al.* [15][16], we explicitly model the notions that a Value can have multiple subjective interpretations, and could be non-quantitative, or could have valuations that aren't expressed as distributions.

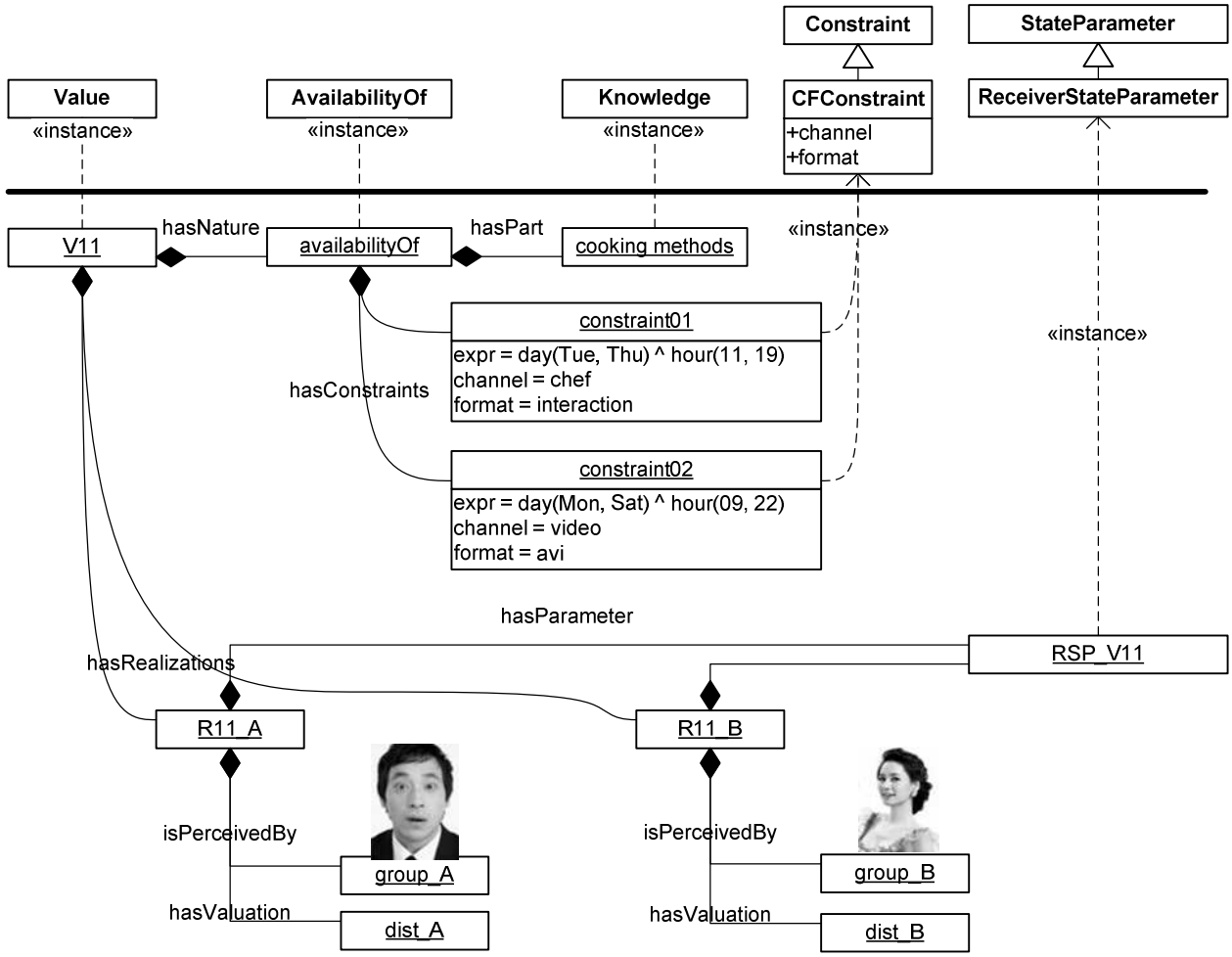

Figure 5. Example of Value with Multiple Conditions and Realizations

An example of Value modelling, including multiple constraints and realizations, is shown in Figure 5. We choose the value of *(availability of) V11: cooking methods* from the meal assembly kitchen scenario in Figure 2. It defines two constraints that refine the availability: constraint01 denotes that, during certain hours only, a chef is present to provide cooking methods in a highly desirable way; while constraint02 indicates that, for other the times, only videos are provided. We model two realizations of this value, using the businessman and housewife personas of Section 2.2. They both share the same RSP, but define different distributions, reflecting different personal valuations of the cooking methods. For this example, we could say that the businessmen do not know how to cook, so they greatly appreciate every cooking method; but housewives are more familiar with cooking, so they already know many of the cooking methods, and they react with boredom.

3.2 Representation of PSS

Our ontological model of PSS is shown in Figure 4, upper right. A **PSS** is represented as a class which aggregates three constituent classes, representing a set of **Value**s, a set of ***PSElement***s, and a set of **Relation**s of this PSS. Furthermore, **PSS** uses a notion of an abstract base class that includes function, behavior, structure, context, and environment attributes, which is named ***FBSCE***. PSS also has a **subPSSs** relation to 0 or more other PSSs, which allows a PSS to be composed from sub-PSSs in a recursive manner.

Element subclasses

The ***PSElement*** class is an abstract base class, whose subclasses represent the elements of a PSS. It conveniently extracts common operations from its subclasses, which simplifies numerous algorithms. ***PSElement*** also inherits function, behavior, structure, context, and environment attributes from the ***FBSCE*** base class.

- **PElement (product element)** describes a product design.
- **SElement (service element)** describes a service, including its provider and receiver roles.

Values in PSS

Values can be associated with product elements, service elements, and PSSs themselves. To ensure a consistent interface for accessing the Values associated with any object, we define an abstract interface *IHasValues*, which carries a hasValues property to 0 or more Value objects. We then add *IHasValues* as an additional base class of both **PSS** and **PSElement** classes, using multiple inheritance. This provides the standard benefits of interface inheritance: (a) it ensures that both PSS and PSElement inherit the same property, and (b) it allows algorithms to access all PSS, PElement, and SElement objects in a uniform manner.

Relation subclasses

Relation represents a relation between two elements, or between a value and an element. It corresponds to an edge between two nodes in the simpler graph representation of PSS. We define three specific subclasses, in order to exploit exact type information in modeling these subclasses.

- **PVRelation** represents a relation between a product element and a value.
- **SVRelation** represents a relation between a service element and a value.
- **PSRelation** represents a relation between a product element and a service element.

Types of Relations

Borrowing UML terminology for relationships, we use *supplier* to denote the value element of a P-V or S-V relation, or the product element of a P-S relation, and *consumer* to denote the other element. We characterize the following types of relations, based on Shimomura *et al*.'s terminology for service categories [13].

- **Enable**. An "enabling service" is one that "makes the receiver easily achieve its aim". We generalize it to indicate that the supplier element satisfies a strong requirement, necessary dependency, or prerequisite of the consumer element.
- **Enhance**. An "enhancement service" is one that "helps, supports, or enhances the achievement" of the receiver's aim. We generalize it to indicate that the supplier element improves or otherwise contributes to the consumer element, but is not a requirement for it.
- **Proxy**. A "proxy service" is one where an agent performs an activity on behalf of the receiver. We take the meaning that the supplier element performs an activity for the consumer element.

4 CASE SCENARIO: MEAL ASSEMBLY KITCHEN

We apply our PSS ontology to the meal assembly kitchen scenario, as described in Section 2.2. For this example, we focus on the second of three sub-PSSs in Figure 2, which is labeled "Provide wide space, several ovens". This sub-PSS is shown again in Figure 6, with the dashed circles elided for clarity.

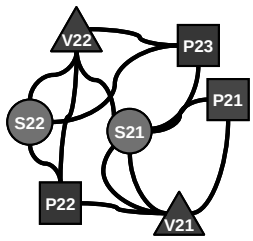

Figure 6. Sub-PSS of Meal Assembly Kitchen (Graph Representation)

In the simpler graph representation of this sub-PSS, the edges highlight the relations between the elements, but do not provide any attribute or other information, such as the types of the relations. Using our PSS ontology, we model this same sub-PSS as shown in Figure 7. Each **Value** node is represented in more detail, with a greater degree of self-documentation. Edges are modeled as **PVRelation**, **SVRelation**, and **PSRelation** objects, as appropriate. The types of the relations are explicitly represented as **enable** or **enhance** types.

The sub-PSS itself is explicitly represented as a PSS object named PSS-B, shown at the top of Figure 7. PSS-B has five **hasPSElements** relations and two **hasValue** relations, as shown. It also has eleven **hasRelations** relations, but these are omitted for clarity.

From this example, we obtain evidence that our PSS ontology is sufficiently complete to represent the sub-PSSs of this case scenario, and by extension, any PSSs of similar complexity. By construction, it is no less informative than the graph representation. Through additional model instantiation steps, we can model the complete PSS for the meal assembly kitchen example, which combines all three sub-PSSs. Currently, we perform the model instantiation manually, using Protégé. While this task is straightforward, it is also low-level and thus tedious, since standard tools such as Protégé don't provide high-level support tailored to our PSS ontology.

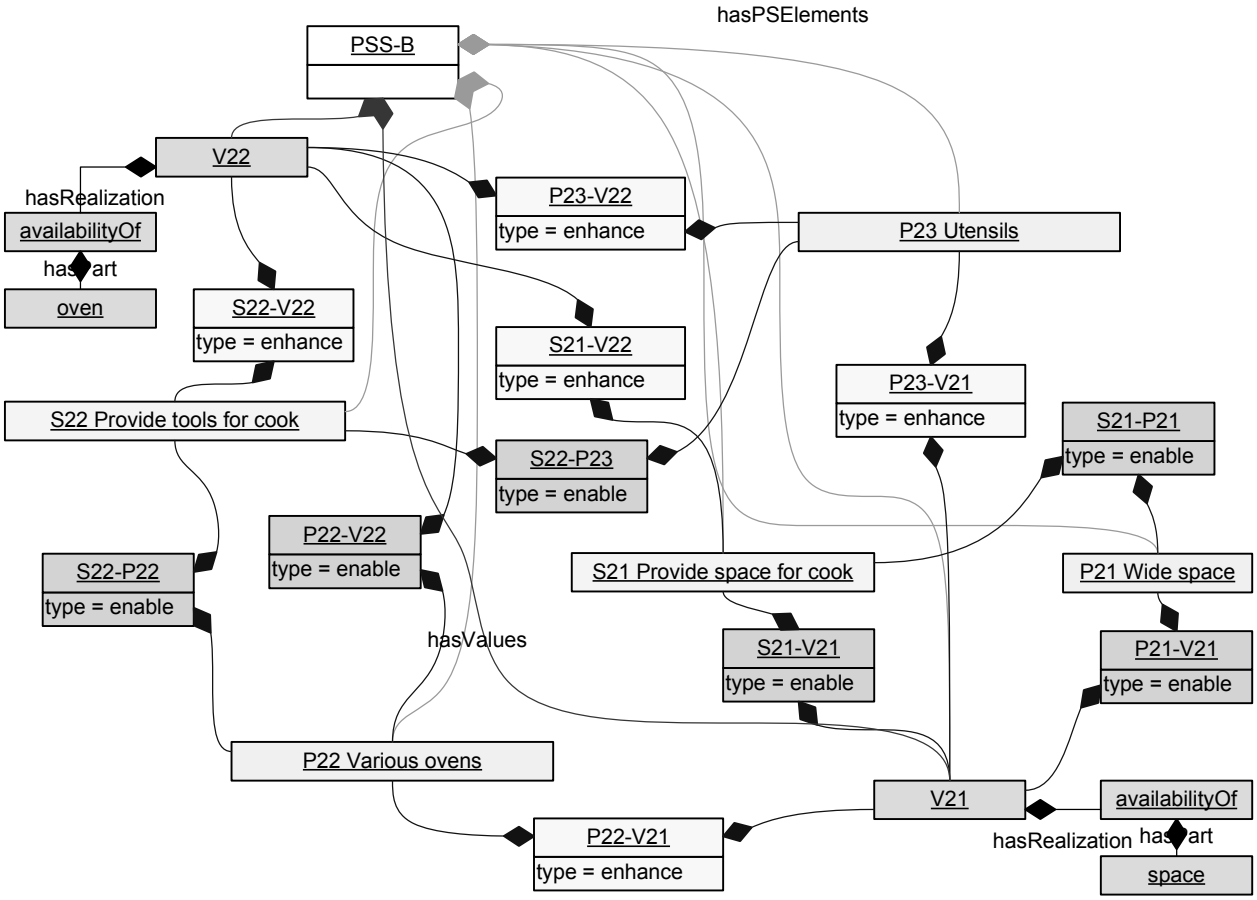

Figure 7. Sub-PSS of Meal Assembly Kitchen (Ontology Model)

5 SUMMARY AND FUTURE WORK

We have described an approach for conceptual design of PSS, which includes the following steps: proposition of values and requirements, persona generation, value improvement, and PSS generation. We illustrate this approach with a case scenario, in which we develop a PSS for a meal assembly kitchen.

We also present graph and ontological representations of PSS. A key idea in our representation is that values are first-class elements, and that they, and their relations with other elements, are explicitly represented in both the graph and ontological representations. We have shown that our ontological representation is sufficient to model a sub-PSS obtained from the case scenario.

In future work, various portions of our ontological representation can be further expanded. In particular, the **ValueNature,** ***IValueCategory***, and ***IValuation*** class hierarchies will be greatly enriched with additional subclasses.

A major future direction is to implement automated reasoning services using this ontology, e.g. by converting the ontology and a PSS model to Jess. This can support numerous kinds of validation and checking computations to assist designers and other users in various tasks within a framework for PSS design.

REFERENCES

[1] Tukker, A., and Tischner, U., 2006, Product-Services as a Research Field: Past, Present and Future. Reflections from A Decade of Research, Journal of Cleaner Production, **14**: 1552–1556.

[2] Baines, T. S., Lightfoot, H. W., Evans, S., Neely, A., Greenough, R., Peppard, J., Roy, R., Shehab, E., Braganza, A., Tiwari, A., Alcock, J. R., Angus, J. P., Bastl, M., Cousens, A., Irving, P., Johnson, M., Kingston, J., Lockett, H., Martinez, V., Michele, P., Tranfield, D., Walton, I. M., and Wilson, H., 2007, State-of-the-Art in Product-Service Systems, Proc. IMechE, Journal of Engineering Manufacture, **221**: 1543–1552.

[3] Neely, A., 2007, The Servitization of Manufacturing: an Analysis of Global Trends, Proc. 14th European Operations Management Association Conference, Ankara.

[4] Goedkoop, M. J., van Halen, C. J. G., te Riele, H. R. M., and Rommens, P. J. M., 1999, Product Service Systems: Ecological and Economic Basics, Report for Dutch Ministries of Environment (VROM) and Economic Affairs (EZ).

[5] Mont, O., 2002, Clarifying the Concept of Product-Service System, Journal of Cleaner Production, **10**, 237–245.

[6] Mont, O., 2004, Product-Service Systems: Panacea or Myth?, Ph.D. Dissertation, Lund University.

[7] Manzini, E. and Vezolli, C., 2003, A Strategic Design Approach to Develop Sustainable Product Service Systems: Examples Taken from the 'Environmentally Friendly Innovation' Italian Prize, Journal of Cleaner Production, **11**, 851–857.

[8] Morelli, N., 2003, Product-Service Systems, a Perspective Shift for Designers: A Case Study: the Design of a Telecentre, Design Studies, **24**(1): 73–99.

[9] Aurich, J. C., Fuchs, C., and Wagenknecht, C., 2006, Life Cycle Oriented Design of Technical Product-Service Systems, Journal of Cleaner Production, **14**: 1480–1494.

[10] Aurich, J. C., Schweitzer, E., and Mannweiler, C., 2008, Integrated Design of Industrial Product-Service Systems, Proc. 41st CIRP Conf. on Manufacturing Systems, Tokyo.

[11] Matzen, D. and McAloone, T. C., 2006, A Tool for Conceptualising in PSS Development, Design for X, Beiträge zum 17. Symposium. Lehrstuhl für Konstruktionstechnik, Technische Universität Erlangen, 131–140.

[12] Matzen, D. and McAloone, T. C., 2008, From Product to Service Orientation in the Maritime Equipment Industry – a case study, Proc. 41st CIRP Conf. on Manufacturing Systems, Tokyo.

[13] Tomiyama, T., Shimomura, Y., and Watanabe, K., 2004, A Note on Service Design Methodology, Proc. ASME Int'l. Conf. of Design Theory and Methodology, Salt Lake City.

[14] Lindahl, M., Sundin, E., Sakao, T. and Shimomura, Y., 2005, An Application of a Service Design Tool at a Global Warehouse Provider, Proc. Int'l. Conf. on Engineering Design, Melbourne.

[15] Sakao, T., Shimomura, Y., Comstock, and M., Sundin, E., 2005, Service Engineering for Value Customization, Proc. 3rd Int'l. World Congress on Mass Customization and Personalization (MCPC), Hong Kong.

[16] Sakao, T., Shimomura, Y., Comstock, M., and Sundin, E., 2006, A Method of Value Customization, Proc. Int'l. Design Conference, Dubrovnik.

[17] Sakao, T., and Shimomura, Y., 2007, Service Engineering: a Novel Engineering Discipline for Producers to Increase Value Combining Service and Product, Journal of Cleaner Production, **15**: 590–604.

[18] Maussang, N., Sakao, T., Zwolinski, P. and Brissaud, D., 2007, A Model For Designing Product-Service Systems Using Functional Analysis and Agent Based Model, Proc. Int'l. Conf. on Engineering Design, Paris.

[19] Maussang, N., Zwolinski, P. and Brissaud D., 2008, Evaluation of Product-Service Systems during Early Design Phase, Proc. 41st CIRP Conf. on Manufacturing Systems, Tokyo.

[20] Meier, H. and Völker, O., 2008, Industrial Product-Service Systems – Typology of Service Supply Chain for IPS^2 Providing, Proc. 41st CIRP Conf. on Manufacturing Systems, Tokyo.

[21] Ohtomi, K., 2005, Importance of Upstream Design in Product Development and Its Methodology, Keynote Presentation, Proc. 6th IEEE EuroSimE Conference, Berlin.

[22] ENGINE, Design for Service for Both Service and Manufacturing Businesses, http://www.enginegroup.co.uk.