



**HAL**  
open science

## Ontology for the PSS Lifecycle Management

Azadeh Hajimohammadi, Juliana Cavalcante, Lilia Gzara

► **To cite this version:**

Azadeh Hajimohammadi, Juliana Cavalcante, Lilia Gzara. Ontology for the PSS Lifecycle Management. The 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems, Jun 2017, Copenhagen, Denmark. pp.151-156, 10.1016/j.procir.2017.03.016 . hal-01702247

**HAL Id: hal-01702247**

**<https://hal.science/hal-01702247>**

Submitted on 6 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems

## Ontology for the PSS lifecycle management

Azadeh Hajimohammadi<sup>a</sup>, Juliana Cavalcante<sup>b\*</sup>, Lilia Gzara<sup>b</sup>

<sup>a</sup>*Motivate Me, 51 Avenue Jean Kuntzmann, 38330 Montbonnot-Saint-Martin, France*

<sup>b</sup>*Univ. Grenoble Alpes, CNRS, G-SCOP, 38 000 Grenoble, France*

\* Corresponding author. E-mail address: [juliana.cavalcante-da-silva@grenoble-inp.fr](mailto:juliana.cavalcante-da-silva@grenoble-inp.fr)

### Abstract

Currently, customer expectations are moving more towards global solutions, incorporating both a physical product and a non-physical service. In order to support these new expectations, a new offer has emerged from both manufacturing and service industries to propose what is commonly named Product-Service Systems (PSS). The PSS lifecycle is defined by a composition of Product and Service design and delivery activities, which involve many stakeholders (actors inside the company but also outside like customers and suppliers). A variety of information is exchanged between these stakeholders during the different phases of the PSS lifecycle. Then, in order to manage the PSS, products and services need to be considered in a lifecycle perspective. To fulfill this requirement, an integrated information management associated with the physical product and non-physical related services provided throughout their lifecycle is essential. In this regard, it is necessary to know how to enhance Information System interoperability for PSS lifecycle management. To provide this functionality, ontology can be a good feature for defining a PSS semantic model. The aim of this paper is to use the advantages and features of ontologies in PSS models. In this regard, firstly a generic semantic model of PSS is proposed and then validated on a case study.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems.

*Keywords:* Product-Service System; Lifecycle Management; PSS Modeling; Semantic Model; Ontology

### 1. Introduction

The improvement of information systems and the increasing need to fulfill customer demands are transforming product-based industries into service-based economies through the development of product-service systems (PSS).

As mentioned in Annamalai *et al.* [1], “PSS lifecycle is an integration and connection of the life cycles of services and products to a common lifecycle”. The lifecycle of PSS is defined by a composition of product and service structure and the interaction of all stakeholders including providers, suppliers and customers during design and delivery phases. A variety of knowledge is exchanged between stakeholders during PSS lifecycle. In this context, a knowledge based lifecycle management of PSS provides all stakeholders the appropriate knowledge and help them in decision making processes. Indeed, all stakeholders should have the same perspective of PSS lifecycle and many interdependent aspects of both products

and services should be considered throughout of the PSS lifecycle. Then, an integrated information management system associated with the physical product and non-physical related services throughout their lifecycle is essential. One of the most important goals of the PSS lifecycle management would be the integration of different domain-specific systems. Regarding this goal, relevant information should be gathered and exchanged efficiently. The PSS Lifecycle Management is the central approach for the integration of PSS engineering data, processes, IT-tools, and actors involved in the entire lifecycle, both on the provider’s and the customer’s sides [2]. Ontological representation of PSS can be a solution. The purpose of developing a standardized ontology is that it could help communicate and share knowledge without ambiguity. Using ontology makes the model dynamic and allows to record, store and process data-information about a number of systems in a single source [3].

Based on the literature, the objective of this paper is to answer the following research question: how to enhance Information System interoperability by means of ontology for PSS lifecycle management?

This paper is organized as follows. First, current PSS models are studied and the capabilities of IT advancements for implementing PSS model are verified (Section 2). Second, regarding the research question and according to the adopted research methodology (Section 3), a semantic model of PSS is proposed (Section 4). This semantic model is then transformed into ontology according to two layers called *Generic Ontology* and *Domain Ontology* (Section 5). Finally, a case study is described to test and validate our propositions (Section 6). This case study is validating the *Generic Ontology* in a particular *Domain Ontology* by considering how much the model is generic and applicable for different kinds of PSS.

## 2. Theoretical background

Oriented by the achievement of the interoperability between different information systems supporting PSS, two main topics are considered in this research: PSS information modeling and ontology.

### 2.1. PSS Information Modeling

PSS information modeling has been tackled in literature according to three classical axes of modeling in information science: static, dynamic and functional.

- Static models represent the PSS structure and include the allocation of resources [4], process flow description [4], modularization management [5], formalization of dependencies between products and services [4], knowledge provision and sharing [6] and PSS categorization [6, 7].
- Dynamic axis focus on simulation [7] and modeling of PSS behavior when submitted to different constraints [4], during the different lifecycle phases [6], according to the different stakeholders' point of view [6] and for mass customization [5].
- Functional models describe the PSS functions including economic, environmental [7], customer segmentation [7, 8], and value proposition aspects.

Most of existing works focused on each axis separately whereas the 3 axes are interdependent (for example: the PSS structure artifacts described in static view are referenced in the dynamic view). In this paper, the aim is to propose a PSS model covering the 3 axes in order to enhance communication among modeling artifacts and to facilitate impact analysis when PSS information is modified.

### 2.2. Ontology

Many times the meaning of the word "Ontology" is taken as a branch of philosophy that is the science of what is, of the kinds and structures of objects, properties, events, processes

and relations in every area of reality [9]. The aim of developing a standardized ontology is to help practitioners to communicate and share their views without ambiguity and thus encourage the conception and implementation of useful methods and tools. An ontology can be described as an explicit specification of a shared conceptualization, which can be taxonomically or axiomatically based [1], Ontology reasoning reduces the redundancy of information in knowledge base and finds the conflicts in knowledge content [10].

In this research, we base our propositions on Protégé tool, an open source platform that provides a rich set of knowledge-modeling structures and actions that uses Semantic Web Rule Language (SWRL) and Jess rule engine in order to extract knowledge about the OWL-model for problem solving [3]. Then, a special focus is put on OWL for data representation. OWL is used to define ontology for particular domain, through the definition of classes, properties, individuals and reason about these classes [5]. According to Matsokis and Dimitris [3] OWL can create classes and properties, defining instances and its operations. There are two types of properties: data properties that are attributes of class instances, and object properties which create relationship between classes.

### 2.3. Ontology based approach for PSS

In the past few years, different ontologies have been proposed for product and service systems. For Pagoropoulos *et al.* [15] it is justified by the ontology's capacity of providing methods for integrating fragmented data into a unique model without losing their individuality.

Since the traditional service development concepts, such as service engineering, were not suitable for the new manufacturing industry, in which product and service are bundled, Shen and Wang [16] presented a new architectural perspective of service. The generic ontology they proposed is focused in the adaptation of service to PSS context, but it does not discuss the product perspective or the product-service interface. Identifying the same lack of research in methods to support service, Doultsinou *et al.* [17] developed a knowledge structure based on findings from industrial cases as well as an ontology oriented to service and service concepts that relate to product, without any particularly discussion about the product itself.

PSS being very industrial driven, Annamalai *et al.* [1] conducted explorative interviews with experts to elicit the practical PSS root concepts and compare them with the ones proposed in literature in order to propose a more suitable ontology. This ontology is focused on the top-level concepts of PSS, corresponding to a first global structure from the design perspective.

One important part of the framework presented by Zhang *et al.* [18] for knowledge management for PSS is a product-oriented ontology focused on design. Its main function is to improve product design by integrating knowledge from manufacturing, usage and maintenance phases; however, the service concept is not further developed.

Aiming to integrate the importance of the end-user perspective in PSS modeling and to evaluate completely the

service contents of PSS solutions, Pagoropoulos *et al.* [15] proposed a new PSS ontology that focuses on main concepts of business - such as contracts and customer channel; and operation management - such as inventory management and quality of equipment. However, it is mainly addressed to service design phases when it is driven by product design.

In a closed-loop perspective for PSS lifecycle management, the information flow must be possible in all directions (forward and backward) between all lifecycle phases of both product and service. In order to cover these interactions, the proposed PSS ontology should elicit the information flow between product-related activities (e.g. design, manufacturing, use) and service-related ones (e.g. design, delivery, use).

### 3. Methodology

Several ontology development methodologies are proposed in literature. In general, they start with the scope or domain definition [19, 20, 21, 22, 23], followed by the identification of needed and existent taxonomies through literature analysis [13, 21, 23] and root concepts through experts' interviews [23]. The ontology is then built or adapted through the definition of classes, properties, constraints and their relationships [13, 20, 21, 22, 23]. Testing and refinement are also considered by [13, 19, 20, 21, 22, 23].

In order to integrate the closed-loop perspective in PSS lifecycle management, an explicit specification of the underlying concepts in PSS lifecycle must be initially established. The methodology followed in this work to develop the ontology is inspired by the one proposed by Annamalai *et al.* [1], which includes the most important aspects of the methodologies previously mentioned. We suggest a layered structure to build PSS ontology: *Generic Ontology*, giving the general concepts for PSS lifecycle management, and *Domain Ontology*, specializing the general concepts in a particular business field and PSS offer.

The development of the *Generic Ontology* is driven by the following steps; in the order they are presented.

(1) Decide which domain the ontology will cover and refer to the appropriate resources to determine what terms to use from the domain. The scope of the ontology is to explicitly describe the relationships between product elements and service elements during different phases of both product and service lifecycles.

(2) Identify the existing taxonomies and the root concepts through and extended literature review on Service Modeling, Product Modeling and PSS lifecycle.

(3) Build the semantic model for PSS. For Peckham and Maryansky [14], a Semantic Data Model is a category of data model that aims to provide increased expressiveness to the modeler incorporating a richer set of semantics into the database.

(4) Create the ontology following the semantic model. The main steps to derive the ontology consist of defining class hierarchy, associate properties and constraints [1, 11, 13].

(5) Test the ontology and refinement. This phase consists of creating the classes' instances in the hierarchy, implementing the model in Protégé software and evaluating its functionality

by comparing the result with the capabilities of the initial model.

(6) Evaluation. This phase is performed through practitioners' interviews to test the relevance and applicability of proposed ontology.

The methodology to derive the *Domain Ontology* from the *Generic Ontology* consists in acquiring information from physical product, inferring new knowledge by ontology inference engines, predicting and creating necessary PSS rules and enriching the proposed PSS model according to new criteria.

### 4. PSS Semantic Data Model

A Product-Service System is a combination of products and services, resulted respectively from product and service portfolios. Besides the physical structure, a product has at least one function and one behavior, being these three classes susceptible to documentation. On the other hand, a service is triggered by one or more events, and it is defined by special parameters and one or more functions, having one or more performance indicators and consuming resources that may be material, human or software resources.

The PSS Semantic Model, represented in Fig. 1, is composed of two main and related parts: Service and Entity concepts. Entity corresponds to the class from what Product, Product Function and Product Behavior inherit. These Entity subclasses were defined to make possible the assignment of different services to a specific product function or product behavior.

The service class corresponds to the intangible part of the PSS that aims to fulfil the customer's need. Connected to this class, there are the main concepts: Service Functionality, indicating the objective of the service action; Event representing its triggers; Service Parameter to describe it; Performance Indicator to evaluate it; Business Rule to present its constraints; and Resource shared in three different concepts – Material Resource, Human Resource and Software Resource.

Complementing the Material Resource class, there are the concepts of Resource Usage Method, with the instances Consumable and Non Consumable, and Stock when the Resource Usage Method has Consumable as instance. In case of having other properties than quantity for Stock, the connection between Stock and Material Resource is done by using a third concept called `MaterialResource_Stock`. The same idea reappears between Service and Measurement concepts: in order to make possible the existence of different kinds of service measurements, like quantity and duration, a new class called `Service_Resource` is created. Other concepts are finally used to precise the links between different services, e.g. Compatibility, Recommendation and Service Succession.

As presented before, from the Entity part derives Product, Product Function and Product Behavior concepts. Considering that a product can have different behaviors in different conditions, the class Product Behavior Condition is necessary. In the same sense, a condition can result in different behaviors and vice-versa, so an intermediate class named `Product_Behavior` must be created.

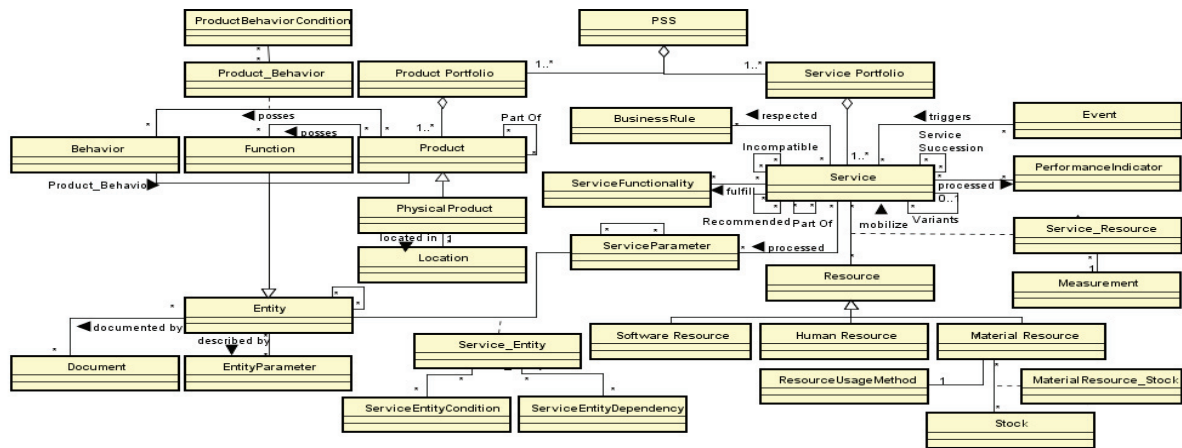


Fig. 1. PSS Semantic Object Model

The Physical Product class is used to define a real product and it has an associated Location concept. Directly associated to the Entity concept there are Document and EntityParameters classes, respectively representing the documentation and the parameters describing the Entity.

The connection between the Entity and the Service parts is assured by ServiceEntityCondition and ServiceEntityDependency. The first defines which criteria will be used on assigning services to products and the second, the kind of dependency between the two. This dependency can be expressed by instances as substitution, recommendation, necessity and exclusiveness.

5. PSS Ontology Model

Defined the PSS Semantic Model, the next step is to translate the model previously presented to a generic ontology. The ontology model aims to create a common model for PSS, including the definition of its concepts and their hierarchy as well as the identification of the relationship between them. For modeling PSS in ontology, it was selected the free and open-source software Protégé, developed at Stanford University.

5.1. Representing relationships and attributes

Both relationships and attributes in UML are defined as properties in ontology. Relationships, that identify the link between two individuals, become Object Properties; while attributes, corresponding to the link between an individual and data values, are called Data Property. Each property relates an individual from a Domain class to an individual or data type from a Range class. Domain and Range classes are used to define the property.

In terms of formalization, properties were named on Protégé starting with lower case and the abbreviation of the domain they refer to, e.g. phpProductType is an object property having PhysicalProduct as domain and Product as range.

5.2. Representing associations

Although there are no associations in ontology like in UML, they can be represented by Object properties relating the different classes of the model. The relationship between classes will be described separately for Entity, Service and Service-Entity class associations.

- Entity Class associations

The classes Product, ProductFunction and ProductBehavior are represented as subclasses of Entity, while the range classes EntityParameter and EntityDocument are related to the domain class Entity through the properties entityParameter and entityDocument, respectively.

Product, inherited from Entity, has a subclass defined PhysicalProduct, as can be seen in Fig. 2. It also corresponds to the domain class related to ProductFunction and PhysicalProduct range classes through pdrFunction and pdrProductTypeOf properties.

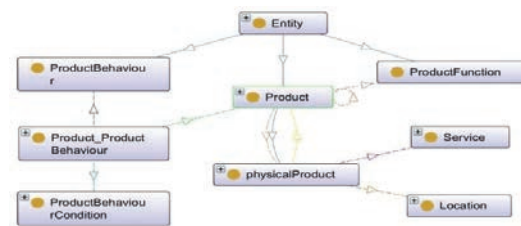


Fig. 2. Product schema in ontology model

The relationship between Product and ProductBehavior depends on another condition entitled ProductBehaviorCondition class. The class Product\_ProductBehaviorCondition represents this relationship between a product and its behavior following a specific condition. In this case, the Product\_ProductBehaviorCondition is the domain class and Product, ProductBehavior and ProductBehaviorCondition



are the range classes. The physical product is equally associated to Service and Location concepts, which are their range classes.

- Service Class associations

The Service class is related to itself through the concepts of: (1) Succession, represented by the `svcSucessionService` property and defining which service should start when another ends successfully; (2) Recommendation, represented by `svcRecommendedService` property and proposing compatible other services; (3) Incompatibility, represented by `svcNonCompatibleService` and defining the services that are not compatible with the current one; (4) Homogeneity, represented by `svcVariet` and regrouping multiple services having the same functionality and different parameters; (5) Partiality, represented by `svcPartOf` and defining which services are part of others.

As presented in Fig. 3, service class is also related to different classes, being the domain of: Event through `svcTriggerEvent`; Performance Indicator through `svcPerformanceIndicator`; Service Functionality through `svcObjective`; Service Parameter through `svcParameter`; and Business Rule through `svcConstraint`.

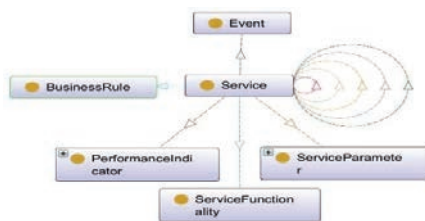


Fig. 3. Service schema in ontology model

Service and Resource classes can be connected through a third domain class called `Service_Resource`, which is also linked to the `Measurement` concept, corresponding to the use duration of the resource.

From Resource inherit three subclasses: `HumanResource`, `MaterialResource` and `SoftwareResource`. Resources can be defined for their `ResourceUsageMethod`, whose possible values are `Consumable` and `Non Consumable`. In addition to that, if a service requires using consumable material resource, it may be represented by the `Stock` class. The resource associations are presented in Fig. 4.

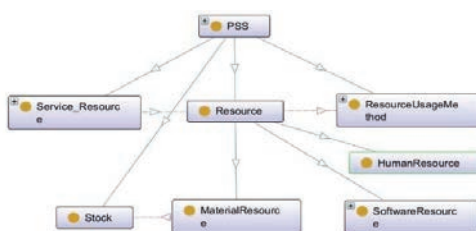


Fig. 4. Resource associations

- Service-Entity Class associations

Through `Service_Entity` class it is possible to represent associations between Entity and Service. Besides the object

properties `s_eService` and `s_eEntity` linking them, there are also `s_eDependency`, `s_eMeasurement` and `s_eCondition` associating `Service_Entity` with the classes: `ServiceEntityDependency`, that includes the values of recommendation, necessity, excluding and exclusiveness; `Measurement`, that specifies the frequency of a service use by an entity; and `ServiceEntityCondition`, which defines this use conditions.

The `ServiceEntityCondition` class, on the other hand, is settled depending on three other concepts: `ServiceParameter`, `EntityParameter` and `Location`. Both `Service-Entity` and `Service-Entity Condition` can be seen in Fig. 5.

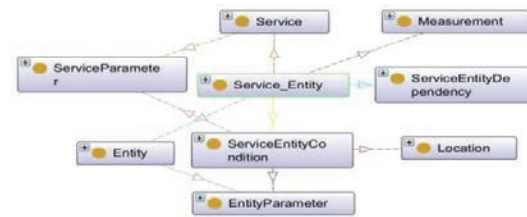


Fig. 5. Service-Entity schema in ontology model

## 6. Case Study

In order to prove the effectiveness of the Generic Ontology Model proposed in section 5, a case study will be implemented in this section under the Domain Model using the example of a bike system which includes rental and maintenance services.

The implementation of the PSS model in domain ontology demands creating instances and extending the model according to the specific requirements. For querying, SPARQL language is used.

In the proposed example, for the physical product class there are, respectively, three kinds of bikes and two kinds of wheels, represented by the property `has individual`. On the other hand, each kind of bike is associated to a bike product through a `phpProductType` property; the same happens with the different kinds of wheels.

As a second association, wheels are part of bikes, and both products can be connected through the `prdPartOf` property. In addition to the standard associations, the specific kinds of bikes can be enriched with `entityParameters` like size, location and components identification.

As presented before, this bike system is composed by *rental* and *maintenance* services. Rental service includes the activities of: availability management; monitoring bike usage; managing of parking stations (composed by data analysis of stations and moving bikes between stations); and subscribers' management (composed by billing, loyalty and fault management). Maintenance service acts through monitoring bike state (including data logging bike usury and status of cycling evaluation); corrective maintenance (including visual inspection, diagnosis and repair which can be electrical or not), bike preventive maintenance, parking station maintenance (including station state monitoring, maintenance and terminals repair); and bike cleaning. All these activities can be associated to a service through a `svcPartOf` property. As for the Generic

Ontology, the association between products and services on this bike system is made through a service-entity class, represented by individuals such as *S\_E\_BikeCleaning\_Bike*, which is related to *prd\_Bike* and to *svc\_Maintenance\_BikeCleaning*, respectively, through *s\_eEntity* and *s\_eService* object properties. This service-entity, presented in Fig. 6, can also be detailed by other associations, such as *s\_eCondition* for the *ServiceEntityCondition* class and *s\_eMeasurement* for the *Measurement* class, which can have *SEC\_BikeLocation\_Population* and *M\_Week* as instances examples in the bike cleaning example.

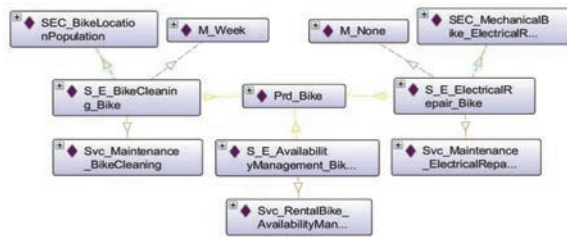


Fig. 6. Individuals of the relationship between Service and Entity

## Discussion-Conclusions

The aim of this research is to use ontology advantage to enhance semantic interoperability of PSS related models. In this regard, first a semantic model is evaluated in UML and then transferred into a generic ontology. Then, a domain ontology specializing the generic one is experimented through a case study (bike rental).

The proposed ontology focuses on information sharing between all lifecycle phases of both product and service, in all directions (forward and backward). It addresses three issues: performance (quality of service and product monitoring), use (consumption cost, consumables to deliver) and MRO (Maintenance, Repair and Overhaul).

Besides considering the whole lifecycle, the proposed ontology explores, through the Entity class, in a more detailed level the interface between Product and Service classes, presenting concepts such as dependency, condition and measurement. Other example of further detailing is the Resource analysis, presenting different kinds of resources, susceptible to different usage methods and stock planning concepts.

In the near future, the proposed ontology will be improved by enriching the PSS semantic model (with cost and resource issues) and defining rules on both generic and domain levels. These rules correspond to business rules or constraints applied inside a company. The aim is to: (1) alert stakeholders when rules are violated (2) calculate indicators or, (3) infer new knowledge. Three kinds of rules are distinguished: (1) Instantiation rules (possible/impossible combination between instances' values, relationships between instances, restrictions on instance' value or relationships between instances), (2) Instance' evolution rules (following the evolution of the model or not), (3) Generic model rules (relationships between

concepts). Future work on the PSS ontology includes industrial validation through practitioners' interviews.

## References

- [1] Annamalai, G.; Hussain, R.; Cakkol, M.; Roy, R.; Evans, S.; Tiwari, A. An ontology for product-service systems. *Journal of Engineering Design*; 2011; p. 635-659.
- [2] Abramovici, M.; Aidi, Y.; Dang, H. B. Knowledge Based Lifecycle Management Approach for Product Service Systems (PSS) in Bernard, A.; Rivest, L.; Dutta, D. *Product Lifecycle Management for Society*; 2013; 409: 239-248.
- [3] Matsokis, A.; Dimitris, K. An Ontology-Based Approach for Closed-Loop Product Lifecycle Management. *Computers in Industry*; 2010; 61 (8): 787-797.
- [4] Klingnet, S.; Becker, M. Formal Modeling of Components and Dependencies for Configuring Product-Service-Systems. *Enterprise Modelling and Information Systems Architectures*; 2012; 7:144-66.
- [5] Dong, M.; Yang, D.; Su, L. Ontology-based service product configuration system modeling and development. *Expert Systems with Applications*; 2011; 38(9):11770-11786.
- [6] Zhu, H.; Gao, J.; Dongbo, L.; Tang, D. A Web-based Product Service System for aerospace maintenance, repair and overhaul services. *Computers in Industry*; 2012; 63: 338-348.
- [7] Alix, T.; Zacharewicz, G. Product-service systems scenarios simulation based on G-DEVS:HLA: Generalized discrete event specification/high level architecture. *Computers in Industry*; 2012; 63: 370-378.
- [8] Stark, R.; Grosser, H.; Beckmann-Dobrev, B.; Kind, S. *Advanced Technologies in Life Cycle Engineering*. *Procedia CIRP*; 2014; 22: 3-14.
- [9] Anand, S.; Verna, A. Development of Ontology for Smart Hospital and Implementation using UML and RFD. *International Journal of Computer Science*; 2010; p. 206-212.
- [10] Abburu, S. A Survey on Ontology Reasoners and Comparison. *International Journal of Computer Applications*; 2012; 57 (17): 33-39.
- [11] Akmal, S.; Batres, R. Shih, L. H. An Ontology-based Approach for Product-Service System Design, in Shimomura, Y.; Kimita, K. *The Philosopher's Stone for Sustainability*; 2013;91-107.
- [12] Zhu, H.; Gao, J. A Web-based Product Service System for aerospace maintenance, repair and overhaul services. *Computers in Industry*; 2012;
- [13] Ahmed, S.; Kim, S.; Wallace, K. M. A Methodology for Creating Ontologies for Engineering Design. *Journal of Computing and Information Science in Engineering*; 2007; 7: 132.
- [14] Peckham, J.; Maryansky, F. *Semantic Data Models*. *ACM Computing Surveys*; 1988; 20 (3): 153-189.
- [15] Pagoropoulos, A.; Andersen, J.; Kjaer, L.; Maier, A.; Mcaloon, T. Building and Ontology of Product/Service-Systems: Using a Maritime Case Study to Elicit Classifications and Characteristics. *IFIP Advances in Information and Communication Technology*; 2014; 119-126.
- [16] Shen, J.; Wang, L. A new perspective and representation of service. *Mob. Comput.*; 2007; 3176-3179.
- [17] Doultsinou, A.; Roy, R.; Baxter, D.; Gao, J.; Mann, A. Developing a service knowledge reuse framework for engineering design. *J. Eng. Des.*; 2009; 20: 389-411.
- [18] Zhang, D.; Hu, D.; Xu, Y.; Zhang, H. A framework for design knowledge management and reuse for Product-Service Systems in construction machinery industry. *Comput. Ind.*; 2012; 63: 328-337.
- [19] Zhan, P., Jayaram, U., Kim, O., and Zhu, L. Knowledge Representation and Ontology Mapping Methods for Product Data in Engineering Applications. *ASME Transactions JCISE*; 2010; 10.
- [20] Pinto, H. S.; Martins, J. P. Ontologies: How can they be built?. *Knowledge and Information Systems*; 2007; 7: 441-464.
- [21] Eris, O.; Hansen, P. H. K. Mabogunje, A.; Leifer, L. Toward a Pragmatic Ontology for Product Development Projects in Small Teams. *Proc. International Conference on Engineering Design*; 1999; 1645-1950.
- [22] Uschold, M.; King, M. Towards a Methodology for Building Ontologies. *Proc. Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
- [23] Gruninger, M.; Fox, M. S. Methodology for the Design and Evaluation of Ontologies, *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal; 1995.