Ontology Mediation to Rule them all:

Managing the plurality in Product Service Systems

Uri Shani
IBM Haifa Research Lab, Haifa University
Mount Carmel, Haifa, Israel 31905
shani@il.ibm.com

Marco Franke, Karl A.Hribernik
BIBA - Bremer Institut für Produktion und Logistik GmbH
at the University of Bremen line 2-name of organization,
Hochschulring 20, 28359 Bremen, Germany
fma,hri@biba.uni-bremen.de

Klaus-Dieter Thoben
Faculty of Production Engineering, University of Bremen
Badgasteiner Straße 1, 28359 Bremen, Germany
tho@biba.uni-bremen.de

Abstract—**The lifecycle of a product is managed not only through the Product Lifecycle Management (PLM), but needs to integrate with product services into a Product Service System (PSS). The related activities are performed throughout the entire lifecycle and require sharing information among tools of the different product lifecycle phases. When extending collaboration of PLM with services as integrated within a PSS, the physical product is linked with a vastly extended universe of information during the PSS lifecycle. To achieve robust and maintainable PSS the interoperability must be fulfilled between the physical products related data sources and the relevant services. To meet that end, we use ontologies to define a formal semantic for information sources and targets. Each tool or data source can use its own ontology independently of the other tools and sources, creating the potential of an unmanageable universe of data. Yet, the benefit is that components of the PSS have weak dependencies among them which leads to an open and flexible system that can easily evolve and adapt. This paper focuses on the provision of ontology driven services including the transformation of product related data into different ontologies and the aggregation of different data source specific ontologies to a holistic PSS universe with no specific ontology in its core. We present two approaches that implement ontology mediation (also termed "semantic mediation") as a variant of ontology matching since the level of matching can be rather complex. The application of this technology is also demonstrated in related domains, showing its potential when applied in PSS that is presently an ongoing research within the PSYMBIOSYS EU project. In consequence, the applicable data integration and ontology matching approaches are the hand tools to instantiate sustainable PSS into the market**

Keywords—Semantic Mediation; Ontology Mediation; Heterogeneous data management; Ontologies in Product Service Systems; large-scale systems Integration

I. Introduction

Megatrends like globalization, an increasing number of product variants, and the fasten integration of new technologies into products require that companies move to more agility and sustainability, especially in the provision and usage of product relevant knowledge. The lifecycle of a product is managed through the Product Lifecycle Management (PLM) including the integration of Product Service Systems (PSS). The given heterogeneity both of tasks within PLM and the underlying IT-infrastructures forestalls the aggregation and exchange of information between the product lifecycle phases in the scope of a large-scale systems Integration. In fact, when viewing the interplay of product and its services, important dichotomies arise [1] that need special address to mitigate and mediate these points into collaboration and interoperability. The PSYMBIOSYS [2] EU project addresses exactly these collisions, which are identified at the following points: Design and Manufacturing, Product and Services, Knowledge and Sentiments, Service-Oriented and Event-Driven architectures, and Business and Innovations. Hereby, each lifecycle phase covers specific tasks and generates/requires specific information. The Information differs between phases, for instance, in content and format but it is not disjoint. All information views together result in a holistic view of the product.

A. Amount and Heterogeneity of Data to be considered

The overall amount of data sources to support the product lifecycle management is high considering the internal data sources like manufacturer/supplier specific data sources as well as external data sources like Tweeter, Facebook or Blogs. The set of internal relevant data sources contains [3]: "Most organizations of middle to large size have hundreds or, more probable, thousands of applications, each with its own various database and other data stores. Whether the data stores are

978-1-5090-4623-2/17/$31.00 ©2017 IEEE

from traditional technologies or document management systems, it is critical to the usefulness of these applications to the Phases of a closed-loop product lifecycle organization that they share information between them" The amount of social media related content for PLM is also high. The current available 'blogosphere' is more than 100 million blogs, and their interconnections have become an important source of public opinion [4]. In the application of microblogging, the leading company Twitter has achieved more than 145 million users who send more 90 million 'tweets' per day, each consisting of 140 characters or less [5]. Consequently, a huge amount of information available is related to a wide range of consumer products and corresponding services. The data to be used contains product specific feedbacks including usage scenarios, technological hurdles from the perspective of the daily usage, best practises or the quality of product specific services like individualisation or maintenance. Thus, the data and the knowledge contained in it have a high value for manufactures and could be used to improve not only the product but also the overall product service system. The provision of knowledge within all product lifecycle phases of PSS is only possible if the interoperability is achieved in a sustainable manner, which includes the automatic extraction of information and an efficient approach for maintaining the varying set, and internal structure of data sources.

II. Interoperability

The necessary interoperability between and within enterprises is considered by ATHENA Interoperability Framework on four levels [6]:

- Data/information (for information interoperability),
- Services (for flexible execution and composition of services)
- Processes (for cross-organizational processes)
- Enterprise/business (for collaborative enterprise operations)

The interoperability on the first level is the precondition to enable the interoperability on the other three levels. The PSS related services are located on the second level and therefore, the interoperability of the first level is a precondition to be fulfilled. For that purpose, the interoperability must be achieved for a wide range of data sources. In so doing, the interoperability can be established on different levels of understanding. Oren et al. [7] defined different levels of understanding, which are

- Lexical understanding
- Syntactical understanding
- Morphological understanding
- Semantic understanding
- Pragmatic understanding

The lexical, syntactical and morphological understanding enables the recognition of the structure of the information and the grouping of relevant entities together, but the meaning is still unclear. The semantic understanding is the key to understand the meaning of the data and close the gap from a data view to an information view. This level of understanding and corresponding requirements to the information exchange is necessary to be solved to establish ontology based PSS services.

The transformation of the data beyond the heterogeneous data sources into an ontology's Abox would allow to gain a holistic logical view over all product related information. The corresponding ontologies, as discussed in Artificial Intelligence, are formal, partial specifications of an agreement over the description of a domain [8]. The corresponding Tbox should include all relevant concepts and properties which are now and in the future relevant for general PSS. The vision of an ontology capturing all different kinds of information is called a monolithic ontology and is not convincing either under many respects [9]. As a consequence, ontologies exist on different levels whereby each level has a specific aim. There are aims to cover basic concepts and properties in so called foundation/upper ontologies. Mascardi mentioned that "Upper ontologies are quickly becoming a key technology for integrating heterogeneous knowledge coming from different sources" [10].

The availability of different ontologies, the possibilities to transform data into ontologies and the common proceeding regarding the creation of foundation and domain specific ontologies requires an additional step to achieve the interoperability on the data/information level (to reach level two of ATHENA interoperability). The mediation between ontologies in which each of them has formulated knowledge is mandatory to enable an information exchange between terminologies and therefore between different kinds of PSS and application domains.

A. *Interoperability in the model-based system engineering*

Model-based system engineering (MBSE) presents significant problems in tools interoperability [11] where the lack of formal semantics of the tools modeling languages gave rise to a technology of semantic mediation. With this approach, the modeling languages of tools are modeled as ontologies and the models of systems developed in the tools are exposed as RDF structures. In short, whatever the modeling language and modeling representation used by the tool, to participate in semantic-mediation interoperability the tool must expose its content using RDF representation, and the concepts of that RDF must abide by an OWL ontology developed specifically for that tool.

MBSE tools in the systems and application engineering domain have adopted the use of limited ontologies defining domain specific concepts for such domains as requirements management, quality management, lifecycle, architecture and so on. That has been the OSLC (Open Services for Lifecycle Collaboration) initiative which evolved into an OASIS standard [12]. OSLC adopted the semantic web linked data [13] concept as means to facilitate sharing of modeling information among tools as open web documents that can be referenced via hyper-links among these modeling elements. With OSLC, modeling elements of PLM and ALM tools are exposed as RDF structures, and they use concepts defined in the OWL specification as an ontology.

The lack of formal and shared semantics among architecture, simulation and analytic tools of system design, has been dealt in the SPRINT [[14]and DANSE [15] EU FP7 projects working on complex cyber-physical system design, and in system of systems design (respectively), and developed the concept of semantic-mediation as a means to augment the RDF model representation of design models in various design tools through ontology as the formal representation of the semantics of these tools' design and modeling languages. Within OSLC, these tools were considered "architecture" design tools, and had no OSLC predefined concepts to use. Hence, the problem of semantic representation for these tools to enable interoperability among them went beyond what OSLC presented as tools interoperability. With this semantic mediation technology, models could be shared among different types of tools having different languages.

The mediation engine developed in these projects could infer a model in a new ontology from the model defined with the concept of another ontology. The inference mechanism interprets matching rules among these too ontologies. Matching could be simple conceptual compatibility. Yet, the differences among modeling languages could be much more subtle and present challenges to bridge them. As a result, an matching needs more complex ontology statements to bridge and the mediation engine needed to be customized to work and properly interpret these statements.

In this section of the paper, we will review the status of adoption of this semantic mediation technology within the system engineering MBSE tools, and idea behind the semantic mediation to enable interoperability where it seems impossible, and the mediation rules patterns that the semantic mediation engine can interpret.

B. *Semantic mediation in MBSE*

When matching two different modeling languages, such as Modelica [16] and SysML [17], the issue of completeness comes to mind and makes the task impossible. Practically, the languages have significant differences, but they also have significant overlaps. The structure of systems modeled by Modelica and SysML would be represented similarly as component sub-component structure. That can certainly be shared. The OPM modeling tool [18] too, models the structure of systems, but using a different representation than that of Modelica or of SysML. The OWL ontologies for a Modelica tool (SystemModeler) and for SysML (Rhapody) were developed in the SPRINT project [19], while OWL ontology was developed for Opcat separately [20],[21]. The common overlap of such modeling languages can be viewed as yet another ontology that represent concepts shared by these ontologies, so that models in any of them can be mediated to this common ontology. In SPRINT and DANSE, that ontology was termed "Basic Structure Ontology" (BSO), and the mediation among three different tools in SPRINT was working through three matching sets that connected the common structured ontology with each of the tools: Modelica tool, SysML tool and a 3rd party proprietary tool. That is shown in the Fig. above.

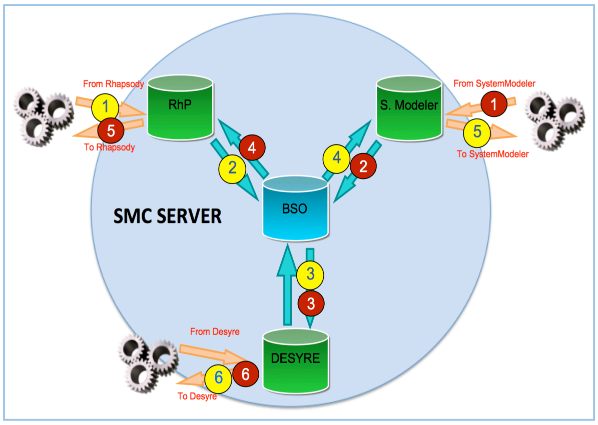


Fig. 1. Mediation in design tools

The generalization of this idea is that as such commonality can be found among additional languages, an elaborate network of relations among ontologies can be built and make sharing of modeling information among tools more open and span over domains that would otherwise be distinct and isolated from each other. One possible benefit would be an increased reuse of models (in the system engineering design domain for instance) over tools to increase the value and benefits of the model-based engineering [22].

1) *Semantic mediation rules*

The mediation rules constitute the "SMC rules language". That "language" constitutes six categories of rules templates, which the mediation engine is, programmed to apply. So, given an RDF model where resources are described with properties and classes from certain ontology, applying the rules will result with the same resources, possibly some new resources, described with properties and classes from a different, the "target" ontology.

The rules are defined within a structure which is an ontology by itself, which may also define some new terms as intermediary in the relations constituting the rules. In addition, there are some terms which are defined in the "rules language" – an ontology which is maintained as part of the semantic mediation container platform and which have special semantics related to the mediation, so that the mediator engine is programmed to properly interpret them.

To describe the rules categories, we will use the following namespaces *bso:*, *sysml:*, and *modelica:*, to refer to modeling ontologies we indend to mediate models from one to another. The namespace *resource:* will refer to model resources being mediated, and *smc:* to the mediation "language" rules terms.

For lack of space in this paper, we can demonstrate only some of the rule categories. In the following the rules are presented as triples.

1. Equivalence replacements. That will transform a resource type or a property of a resource based on class or property equivalence OWL relations. Hence, these OWL relations:

TABLE 1 EXAMPLES FOR EQUIVALENCE REPLACEMENTS RULES

Subject	Predicate	Object
rhapsody:Block	owl:equivalentClass	bso:Componen
dcterms:title	owl:equivalentProperty	bso:name

.

When applied to the *bso:* relations here:

```
resource:1    a          sysml:Block         ;
       dcterms:title "GateTester" .
```

Will conclude the following relation for that resource in the *rhapsody:* ontology:

```
resource:1    a          bso:Component       ;
       bso:name "GateTester" .
```

2. Replacement with a super-property (or a super-class). This rule generalizes the equivalence rule above with subclass and subproperty OWL relations to achieve similar goals as in category 1 above. There asymmetric nature of this OWL relation poses theoretical problems to apply it in both directions of mediation between the two related ontologies, but due to the nature of the mediation transformation we define, the practical application of these rules serves in both directions. Simply stated, we define the inverse application of the rule to mean "mediate to a relation which when applied with the mediation relation, the original relation can be inferred." Unfortunately as this may sounds confusing.

Our rule would define an intermediate property of the mediation rules ontology for that as follows:

TABLE 2 EXAMPLES FOR REPLACEMENT WITH A SUPER-PROPERTY RULES

Subject	Predicate	Object
sysml:hasAttributeType	owl:equivalentProperty	sys-bso:attributeType
sys-bso:attributeType	rdfs:subPropertyOf	bso:type

3. Replacement with a sub-property (or a sub-class)

When two different relations in one ontology is mediated to the same relation in the other ontology, but to resources having distinct types, that can be used to properly recover the original properties when mediated back. This is the case in the next rule which mediates the properties *bso:subpackages* and *bso:declared* to the property *modelica:localClass*. Yet, the range of that relation in the target ontology modelica distinguish that with the classes *modelica:MPackage* and *modelica:MModel*. The resulting rule to use is therefore:

TABLE 3 EXAMPLES FOR REPLACEMENT WITH A SUB-PROPERTY RULES

Subject	Predicate	Object
bso:subpackages	rdfs:subPropertyOf	modelica:localClass
bso:subpackages	rdfs:range	modelica:MPackage
bso:declared	rdfs:subPropertyOf	modelica:localClass
bso:declared	rdfs:range	modelica:MModel

4. Complex equivalence replacements. This rules applies restrictions on an equivalence relation. For instance,

```
resource:1 a bso:Port ; bso:direction bso:IN ;
bso:type bso:Integer .
```

That would be the modelica connector of a compound type:

```
resource:1  a  modelica:MConnectorComponent  ;
modelica:type modelice:IntegerInputConnector .
```

The rule involves restrictions on the *modelica:type* property such as

```
[owl:intersectionOf  ([  a  owl:Restriction  ;
owl:onProperty   bso:type   ;   owl:hasValue
bso:Integer ]
[a    owl:Restriction    ;    owl:onProperty
bso:direction ; owl:hasValue bso:IN ] ) ]
```

5. Type-inference from property domain or range. The type of a resource can be inferred from its being the domain, or range, of a property, and that property exists for the resource. Then, the inferred type can be subject to further equivalence rules as seen above. For these mediation rules:

TABLE 4 EXAMPLES FOR TYPE-INFERENCE FROM PROPERTY DOMAIN OR RANGE RULES

Subject	Predicate	Object
sysml:hasAttributeType	owl:equivalentProperty	sys-bso:attributeType
sys-bso:attributeType	rdfs:domain	sysml:FlowProperty
sysml:FlowProperty	owl:equivalentClass	bso:Flow

This leads to assume that resources having the *sysml:hasAttributeType* predicate to be assigned the *bso:Flow* type, even though they did not originally had the *SYSML:FLOWPROPERTY* TYPE ORIGINALLY.

6. EXPANSION (AND CONTRACTION, BY DOMAIN, CLASS, AND RANGE. FOR THIS CATEGORY, WE INTRODUCE THREE NEW CONCEPTS DEFINED IN THE MEDIATION "LANGUAGE" ONTOLOGY AS *RDFS:OBJECTPROPERTY*:

SMC:DOMAINPROPERTY – IS A DOMAIN PROPERTY OF A PREDICATE

SMC:PROPERTYCLASS – IS A ROPERTY CLASS OF A PREDICATE

SMC:RANGEPROPERTY – IS A RANGE PROPERTY OF PREDICATE .

FOR EXAMPLE, GIVEN THE FOLLOWING MODEL TRIPLE:

```
Resource:1 sysml:hasEndPoint_1 resource:2 .
```

With the following rules:

TABLE 5 EXAMPLE FOR EXPANSION RULES

Subject	Predicate	Object
sysml:hasEndPoint_1	smc:domainProperty	bso:interconnectionEnds
sysml:hasEndPoint_1	smc:propertyClass	bso:InterconnectionEnd1
sysml:hasEndPoint_1	smc:rangeProperty	bso:part

Results with a new resource link and definition:

```
resource:1  bso:interconnectionEnds  resource:t
.
resource:t    a   bso:InterconnectionEnd1 ;
         bso:part resource:2 .
```

2) *Semantic mediation for semi structured data sources*

The first step to achieve the interoperability on data information level is to transform the data beyond the data sources which are semi structured or unstructured into ontologies, subsequently to enable the aggregation of knowledge over the boundaries of ontologies. To achieve the first step of interoperability, the developed semantic mediator approach uses the concept of a **wrapper** and a **reasoning mediator** defined in [23][24] to enable the mapping of data to elements of Tbox. Applicable Tbox elements are datatype or object properties and concepts of an ontology of PSS related data sources which could spread over different companies. To support the autonomy criteria of data sources, a virtual data integration approach, more precisely, a mediator based approach was chosen. This approach implements a global as view (GAV) mapping over multiple data source specific ontologies in which each aggregated ontology concept is a merged version according to the available data properties and axioms in the data source specific ontologies. This capability is necessary to enable a natural join of knowledge over the boundaries of data sources.

The generic wrapper implements the linkage to the data sources. Each wrapper follows a defined interface and implements for a specific kind of data source a corresponding semantic data integration approach. Common kinds of data sources are e.g. relational database, XML files, CSV files or web services. All listed kinds of data sources are covered by corresponding existing wrapper implementations.

The linkage between a specific data source and a wrapper is established through a configuration. Such a configuration includes an ontology as well as corresponding mapping file. The ontology defines the amount of information which are available beyond the data source. The mapping file defines the proceeding how to transform the data schema into an ontology. For that purpose, it contains semantic mediation rules. Each semantic mediation rule includes transformation rules to describe the transformation from a piece of data into an ontology specific structure. The execution of a wrapper does not always apply all available semantic mediation rules and therefore, it does not transform everything into an Abox. The requested amount of information is defined by a revolved SPARQL query. Each time a wrapper shall extract knowledge, a resolved SPARQL query and an existing Abox is passed as input. The task of the wrapper is to complete the Abox corresponding the concepts and properties which are included in the resolved SPARQL query. The role of a wrapper is mainly to transform data into concepts and properties as fast as possible. For that purpose, a wrapper may not apply inferring or reasoning technics but rather apply rules for completing the Abox. The consideration of taxonomy and equivalence according to concepts and properties is only done by the semantic mediator on basis of a user specific SPARQL query. The semantic mediator applies the resolution of taxonomy and equivalence and send a resolved SPARQL query to a wrapper. The following example shall illustrate this capability.

An ontology contains the concepts student, employee which are inherit from the concept person. Moreover, the concept robot is set to equivalent to the concept person. If a user requests all known students, each configured wrapper which could deliver corresponding individuals would receive a request not only for students but also for the concepts human and robots. This approach accelerates the data acquisitions, because the inferring and reasoning work load is only done by the reasoning mediator once for all linked data sources.

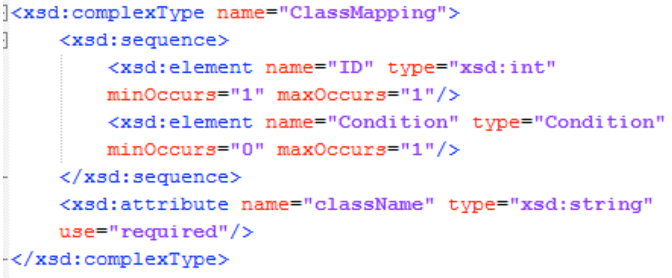
a) Semantic mediation rules

The wrapper specific mediation rules cover the concepts and properties of the wrapper specific ontology which shall be populated. Each mapping file foresees the presence of inverse functional property in the ontology and also a corresponding mediation rule within the mapping. This property is mandatory and used to determine how many different individuals are located within a data source as well as to create a link to the occurrence of the same individuals in other data sources. Common inverse functional properties of the real world could be the MAC address for a network device, the national insurance number within one country for humans or the IPV6 address for IOT objects. The listed set of examples demonstrates that a datatype property for individualization could be found on the global scale or at least on the local scale to specific regions.

Apart from this specific precondition, the transformation approach follows the following generic rules which are defined in a proprietary XML format. In the following, the semantic mediation rules are presented for a CSV data source as well for a web service data source.

b) CSV related transformation rules

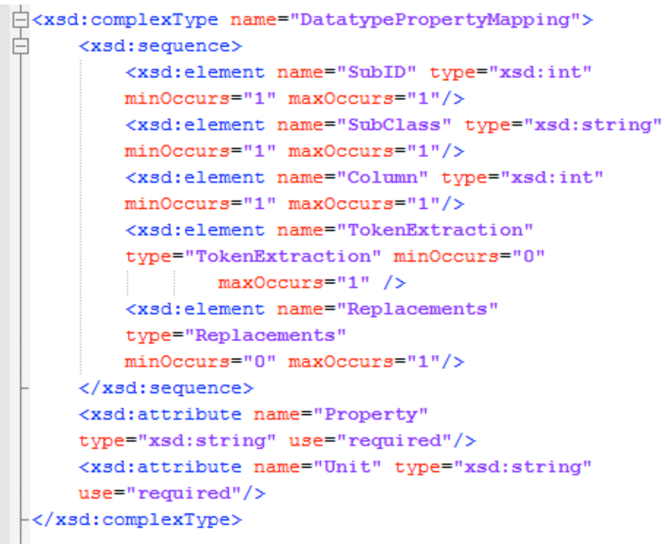
The mapping of a concept is shown in Fig. 2. The mapping maps the column of a CSV file to a concept of an ontology. The optional constrain can be used to define whether a specific row could be assigned to a specific concept. This capability allows to extract different concepts from the same CSV file.



```
<xsd:complexType name="ClassMapping">
    <xsd:sequence>
        <xsd:element name="ID" type="xsd:int"
        minOccurs="1" maxOccurs="1"/>
        <xsd:element name="Condition" type="Condition"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="className" type="xsd:string"
    use="required"/>
</xsd:complexType>
```

Fig. 2. Mapping of column to Concept

The mapping of a data property is shown in Fig. 3. It assign a column to a datatype property. Apart of the name of the datatype property the corresponding concept must be addes as SubClass and SubID. The TokenExtraction and the Replacements could be applied to add a string pre-processing if necessary.



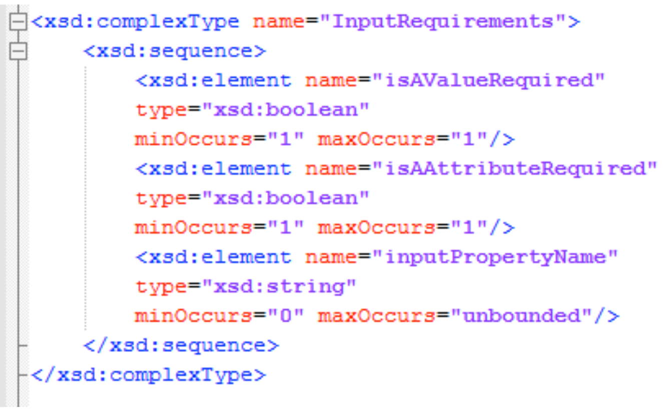
```
<xsd:complexType name="DatatypePropertyMapping">
    <xsd:sequence>
        <xsd:element name="SubID" type="xsd:int"
        minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SubClass" type="xsd:string"
        minOccurs="1" maxOccurs="1"/>
        <xsd:element name="Column" type="xsd:int"
        minOccurs="1" maxOccurs="1"/>
        <xsd:element name="TokenExtraction"
        type="TokenExtraction" minOccurs="0"
                maxOccurs="1" />
        <xsd:element name="Replacements"
        type="Replacements"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="Property"
    type="xsd:string" use="required"/>
    <xsd:attribute name="Unit" type="xsd:string"
    use="required"/>
</xsd:complexType>
```

Fig. 3. Mapping of column to datatype property

The mapping of an object property has a similar mapping like the datatype property. The exception is that the assigned column in CSV file must contain the value of the inverse functional property of the concept which is mentioned in the range of the object property. In cases, where no corresponding individual is contained in the current Abox, an individual of the specific concept including the given value for the inverse functional property is created.

c) Web service related transformation rules

The mapping of a concept and datatype property is similar to the mapping for a CSV file. The exception is that the extraction of all necessary information is done through a couple of web service methods invocations. The necessary sequence of web service methods invocations is calculated on basis of the input and output schemas of the WSDL operations as well as the modelled input information of the datatype property mapping. The extension of the mapping schema to add input information of a datatype property is shown in Fig. **4**, in which the provision of a constant or the dynamic value of a datatype property could be assigned.



```
<xsd:complexType name="InputRequirements">
    <xsd:sequence>
        <xsd:element name="isAValueRequired"
        type="xsd:boolean"
        minOccurs="1" maxOccurs="1"/>
        <xsd:element name="isAAttributeRequired"
        type="xsd:boolean"
        minOccurs="1" maxOccurs="1"/>
        <xsd:element name="inputPropertyName"
        type="xsd:string"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

Fig. 4. Mapping of a required input information

III. Summary

The domain of products and services is in need for integration and collaboration over tools and information sources of great variety and size. The PSYMBIOSYS project identifies five points of friction where collaboration through information sharing and reuse can help improve the value of products to the consumers and the manufacturers. In these points, we have specifically focused on the product design and manufacturing domains where our approach has been to adopt linked data among information sources and targets with respect to the emerging standard for open services for lifecycle collaboration (OSLC) and adopt an approach by which ontologies provide the formal semantic of information and models. A semantic mediation technique that is originated in system engineering EU projects SPRINT and DANE, models of various tools can be matched to different tools in the PLM (product lifecycle management) tool chain. In parallel to that, we also look in this paper at the friction between the real and digital worlds where we adopt a similar ontology-based approach to make information coming from these two directions useful and meaningful for new purposes of interoperability and collaboration so that service factors can be introduced as early as possible into the product manufacturing process. Here, we use a technique developed by BIBA to enable the interoperability of heterogeneous autonomous data sources through the transformation of data into ontologies and to mediate them to a unified ontological view that make the resulting information independent to the syntax and semantic of its sources.

We have described these two approaches with some details in light and in comparison to the more general term of ontology matching, showing at the case of semantic mediation container that elaborate mediation rules stated in ontological terms can bridge gaps, which go beyond simple equivalences. Both approaches aligned to a joint mediation enables the provision of information from semi-structured information (e.g. XML files, CSV) as well as formulized information (e.g. UML, Modellica, test models) to be applied under the umbrella of ontologies to solve the interoperability issues for information driven PSS services.

In conclusion, both data integration approaches unified enables the aggregation and integration of all available product service specific data sources to support the information driven product services. The ongoing research will demonstrate the best way to apply the approach in PSYMBIOSYS related business cases and corresponding existing IT Landscapes. .

Acknowledgment

This research has been funded by the European Commission PSYMBIOSYS EU H2020 project grant #636804

REFERENCES

[1] M. Seregni, C. Zanetti, A. Canepa, M. Taisch, S. Gusmeroli, “Main obstacles to adopting service innovation by all EU manufacturing industries,” in New approaches to product service systems Workshop of the I-ESA 2016 8-th international conference on interoperability for enterprise systems and applications, March 29, Guimaraes, Portugal.

[2] PSYMBIOSYS EU H2020 Project, Grant #636804, http://www.psymbiosys.eu/

[3] Reeve, Managing Data in Motion Data Integration Best Practice Techniques and Technologies. USA: Elsevier Inc;2013

[4] Kietzmann, J. H., Hermkens, K., McCarthy, I. P., & Silvestre, B. S. 2011, “social media? Get serious! Understanding the functional building blocks of social media.”, 2011, Business horizons, 54(3), 241-251.

[5] Madway, G. (2010, September 14). “Twitter remakes website, adds new features”. Retrieved November 5, 2010, from *http://www.reuters.com/article/idUSN1411135520100915*

[6] Berre, A-J., et al. "The ATHENA interoperability framework." Enterprise interoperability II. Springer London, 2007. 569-580..

[7] Oren, T., Ghassam-Aghaee N., and Yilmaz, L. “An Ontology-based Dictionary of Understanding as a Basis for Software Agents with Understanding Abilities”, Proceedings 2007 Spring Simulation Multiconference, 2007, IEEE Press

[8] Guarino, N.”Formal Ontology in Information Systems” In Guarino, N., ed.: Pro-ceedings of the International Conference on Formal Ontology in Information Sys-tems (FOIS’98), Trento, Italy, 1998, IOS Press, Amsterdam 3–15

[9] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.:The WonderWeb Library of Foundational Ontologies. WonderWeb Deliverable 17(2002)

[10] MASCARDI, Viviana; CORDÌ, Valentina; ROSSO, Paolo. A Comparison of Upper Ontologies. In: WOA. 2007. S. 55-64.

[11] H. Broodney, U. Shani, A. Sela, “Model Integration – Extracting Value from MBSE,” in MBSE, INCOSE International Symposium 2013, Philadelphia, U.S.

[12] Open Services for Lifecycle Collaboration (OSLC), *http://open-services.net/.*

[13] W3C World Wide Web Consortium. Linked Data - W3C. *http://www.w3.org/standards/semanticweb/data*.

[14] SPRINT - Software Platform for Integration of Engineering and Things. *http://www.sprint-iot.eu/*

[15] DANSE - Designing for Adaptability and evolutioN in System of systems Engineering. *http://www.danse-ip.eu/home*.

[16] Modelica Language Specification. Version 3.3. Modelica Association. 2010. May 9. *https://www.modelica.org/documents/ModelicaSpec33.pdf.*

[17] SysML™ specification version 1.3. *http://www.omg.org/spec/SysML/1.3/.*

[18] D. Dori. Object-Process Methodology: A Holistic Systems Paradigm. Springer, Berlin, Heidelberg, New York, 2002.

[19] U. Shani, Final Report. A SPRINT project deliverable. March 17, 2014. http://www.sprint-iot.eu/public_deliverables/D_5_11.PU.Final%20Report.pdf.

[20] S. Jacobs, N. Wengrowicz and D. Dori. Exporting Object-Process Methodology System Models to the Semantic Web. IEEE International Conference on Systems, Man, and Cybernetics. 2014

[21] Uri Shani, Shmuela Jacobs, Niva Wengrowicz and Dov Dori “Engaging ontologies to break MBSE tools boundaries through semantic mediation,” 2016 Conference on Systems Engineering Research,

[22] Uri Shani and Henry Broodney, “Reuse in Model-Based Systems Engineering,” 9-th Annual IEEE International Systems Conference, 2015, April 13-16, Vancouver, CA, pp.77-83

[23] Klein, K., Franke, M., Hribernik, K. A., & Thoben, K. D. Identification of Interface Information for a Virtual Data Integration. In Enterprise Interoperability VI. Springer International Publishing. 89-99; 2014

[24] Franke, M., Pirvu B.-C., Lappe D. and other Interaction Mechanism of Humans in a Cyber-Physical Environment. Proceedings of the 4th International Conference on Dynamics in Logistics. International Conference on Dynamics in Logistics (LDIC-14), 4th, February, Bremen, Germany, LDIC.10-14; 2014