Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Procedia CIRP 25 (2014) 361 – 368

www.elsevier.com/locate/procedia

8th International Conference on Digital Enterprise Technology - DET 2014 – "Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution

Process-Oriented Semantic Knowledge Management in Product Lifecycle Management

Artur Felic[a,b]*, Birgitta König-Ries[a], Michael Klein[b]

[a]*Friedrich-Schiller-Universität Jena, Institut für Informatik, Ernst-Abbe-Platz 2, 07742 Jena, Germany*
[b]*CAS Software AG, CAS-Weg 1-5, 76131 Karlsruhe, Germany*

* Corresponding author. Tel.: +49 721 9638 103. *E-mail address:* artur.felic@uni-jena.de

Abstract

Product Lifecycle Management (PLM) focuses on management of product data and processes across their lifecycle stages. One of the key features of PLM is the management of knowledge collected during product development. Reduced time to market, a better collaboration and savings are expected benefits of PLM amongst others. The implementation of a PLM System inside a company can be a hurdle because of the heavy change of the company structure during introduction.
Most PLM solutions are based on one integrated product model that stores product data and shares these data with all contributors. However, the access of product data by different expert domains can be challenging when domain expert knowledge is necessary to understand it. This leads to a communication overhead that increases cost, product development time and thus time to market due to the need for contact to experts.
To deal with comprehensible knowledge throughout the product lifecycle phases and thereby eliminate communication overhead, this paper presents a process-oriented and integrated semantic solution that supports interoperability of knowledge during all phases of the product lifecycle. Based on shared ontologies and product models, collaborators of product chains have the ability to define their own extensions to the underlying models and ontologies. Collaborators are thus able to use their own modeling methodology, which reduces inhibitions to use the solution. The ability to automatically infer information between partial product models of different process chains enables a better collaboration during product development. Since the access of inferred information for specific process chains can be permitted or restricted, collaboration between multiple departments inside and outside the enterprise is supported.
For a better understanding, the solution will be exemplarily applied to the aviation industry. This use case will also be used for evaluation and further improvement. To give a brief outlook on future activities besides PLM, the OSMOSE Project is introduced.

Peer-review under responsibility of The International Scientific Committee of the 8th International Conference on Digital Enterprise Technology - DET 2014 – "Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution"
Keywords: Product Lifecycle Management; Knowledge Management; Semantic Web Technologies

1. Introduction

Product Lifecycle Management (PLM) can be seen as a composition of methodologies to manage product information and processes over the whole product lifecycle. Many companies that manage complex products that are developed over a long period of time use PLM Systems. PLM Systems encompass the development, planning, design, production, and use of a product [1]. Some of the major issues addressed by PLM are data integrity and completeness as well as interoperability and collaborativity of experts from different knowledge domains [2]. The benefits that are expected from PLM are primarily reduced time-to-market and product cost and improved product quality and competitiveness.

However, there are several challenges that need to be overcome in order to achieve this aim. In this paper, we propose a novel solution that addresses several of them. The first challenge is to allow collaborators to use their own strategies and mechanisms. Designers, engineers, sales personell and many more use different software solutions and thus mechanisms to represent their knowledge. Changing the way they use to work is a burden and may lower the expected benefits of PLM solutions. Simultaneously, knowledge has to be shared and distributed across borders of product lifecycle

Peer-review under responsibility of The International Scientific Committee of the 8th International Conference on Digital Enterprise Technology - DET 2014 – "Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution"
doi:10.1016/j.procir.2014.10.050

phases in an accurate and meaningful way. Thus, the second challenge is to provide a suitable solution for knowledge distribution.

In order to deal with these issues and achieve the desired benefits, it is necessary to manage knowledge that is produced, stored and exchanged during all of the product lifecycle phases [3]. Furthermore, knowledge has to be understood accurately since it is critical to competitiveness [3] [4]. Especially knowledge that is produced from different expert domains is not comprehensible for everybody, though the access to knowledge from these expert domains by others from outside the expert domain is inevitable. Knowledge Management (KM) focuses on the denoted challenges and comes up with solutions to satisfy the need for systematic knowledge creation, sharing and reuse [5]. Ontologies are useful to represent knowledge formally and to create a shared knowledge base, from which knowledge can be inferred automatically.

Although a shared knowledge base is desirable, obstacles from inside and outside the company could prohibit a shared knowledge base. Especially in supply chains or business networks, knowledge is reluctantly revealed to business rivals. To overcome these obstacles, a knowledge base where only selected information is published to others is introduced in this paper. Information can be defined as public or private.

Today, many companies incorporate in business networks or similar groups to generate business value. Some companies also outsource inner processes for value generation. Also, partners can change often during a products lifetime. Even a technology change could involve changes in heterogeneous systems. It is necessary to adapt to the resulting changes as quickly as possible. Changing the whole knowledge base and data model can be difficult. Many other partners or departments need to adapt. In this paper, a modular and process-oriented approach is introduced. Process chains are used to modularise the data produced during product lifecycle, as well as the data and knowledge produced inside the process chains itself. Where data from other extensions (i.e. process chains) is needed, interfaces are defined that are accessible from other process chains. Furthermore, the use of ontologies allows to infer information from the inside of a process chain. This inferred information can thus be used to translate knowledge from one expert domain to another.

This modularization approach also comprises the possibility for each process chain to use its own underlying models and knowledge base due to the fact that only interfaces between process chains need to be defined. Exchanging process chains does not have an impact on the existing knowledge base. Additionally, contributors to the knowledge base do not need to change their methodologies, which reduces inhibitions to implement the provided solution.

This paper presents an approach to solve the problem of incompatible knowledge between lifecycle processes in PLM. Thus, the paper is particularly interesting for industries that have running PLM systems or plan to do so. Additionally, important gaps of current solutions are identified in the present work and a possible solution to fill these gaps is presented. Thus, it is also particularly interesting for researchers in the same research area.

The approach presented in this paper is applied to PLM, although it is not limited to the purpose of PLM. In the OSMOSE Project [6], the main goal is to interconnect the real, virtual and digital world in a semi-permeable manner. The presented approach could also be adapted to exchange knowledge between those worlds appropriately.

This paper is structured as follows: Section 2 is an exploration of work related to this paper. It also distinguishes the solution provided in this paper from the related work. Section 3 explains the developed architecture and components and how they work together to provide the solution. Section 4 introduces *Knowledge Links* as a solution for knowledge distribution across borders of lifecycle phases. Section 5 critically discusses the provided solution and states advantages and disadvantages that come along with the presented work. Section 6 summarises the work in this paper and gives a brief enumeration of possible questions that needs further research.

2. Related Work

A lot of effort has been made to provide solutions for collaborative product development. Plenty of them concentrate on the collaboration of partners that contribute to a specific process in product lifecycle management.

In [6], the lack of seamless interoperability of current systems is identified. Patil et al. denoted that current systems are not suitable for collaborative product development and standard methods are missing. The Product Semantic Representation Language (PSRL) is introduced as a transformation between different application domain ontologies. PSRL supports one-to-one mappings in DAML+OIL.

Heterogeneous systems lead to collaboration problems since various stakeholders and systems typically need to be integrated. Plenty of proprietary software tools cannot be combined seamlessly. The authors of [7] address these problems during design phase by introducing the Assembly Design (AsD) ontology and Assembly Relation Model (ARM) to describe assemblies and their relationship globally with OWL [8] and SWRL [9].

The main problems identified in [10] are missing inflexibility in provided solutions, the missing ability to provide different views on information, the lack of interoperability between competitive software systems, the provision of a shared knowledge base and thus meaning and the overhead that comes along with the implementation of such systems.

The authors of [11] depict research issues on closed-loop PLM. In closed-loop PLM, knowledge is exchanged back and forth along the product lifecycle. The purpose is to use product embedded information device (PEID) technology, with which products embed their information with an information device over the whole product lifecycle.

In [12], Kiritsis applies closed-loop PLM to the Internet of Things. By using the PROMISE Architecture, product knowledge loops are closed and lifecycle information is seamlessly transformed to knowledge. PROMISE Data Services act as middleware component which is responsible for the communication and data gathering. PROMISE PDKM/DSS (Product Data Knowledge Management/Decision Support System) supports decision making with analysis of gathered product knowledge and thus aims to integrate product data during product lifetime.

Zhang and Yin [13] apply Semantic Web technologies, i.e. ontologies, to collaborative engineering design. Along five layers, in particular the knowledge elicitation, product modelling, ontology modelling, knowledge reuse and knowledge application layer, knowledge from different sources and with heterogeneous formats is being acquired, uniformly modelled and stored, semantically annotated and distributed by means of a query interface.

A process-oriented PLM implementation approach is introduced by Schuh et al. [14]. The authors identified the need for customisable process models that have to be chosen according to the companies characteristics. Therefore, reference models of the machinery industry are presented and classified according to different company characteristics.

A framework for collaborative product development has been introduced in [15]. Based on a product-process-organization model (PPO), with which product and organisational classes are modelled, their presented framework for Integration of Product Process and Organization for Performance Enhancement in engineering (IPPOP) enhances interoperability between expert tools.

P^4LM [16] is a methodology that focuses on projects, products, processes and proceeds rather than being too product oriented. Each abstraction level has its own model. The project related model represents organisational entities and resources, whereas the product related model incorporates product characteristics. The approach is ontology based and also provides strategies for data security.

Chen [17] identifies the security aspect of knowledge sharing as a factor of success in PLM of virtual enterprises (VEs). In the authors approach, knowledge can only be accessed by privileged users. The knowledge access control policy (KACP) is responsible for the access control of the knowledge base.

To support interoperability and collaboration in the design phase of product development, the authors of [5] introduce a product data warehouse (PDW) based on ontologies. An extensible core-ontology represents basic concepts of product, process, storage and description. Additionally, decision-making processes are recorded to provide user experience for particular situations.

In [18], an integrated ontology-based knowledge management mechanism is proposed for collaborating enterprises. Partners use their own ontology extension to represent own models. The collaboration roles are mapped via product lifecycle ontology. The integrated global knowledge base is mapped and merged from all local enterprise ontologies.

Panetto et Al. propose the generic product ontology ONTO-PDM [19] emerged from existing standards for technical product description. Semantic translations from standards are the key to interoperability in their work.

The combination of a supply chain ontology derived from the Supply-Chain Operations Reference-model (SCOR) and the product ontology ONTO-PDM is introduced in [20]. Product concepts are mapped to enterprise process inputs and outputs to overcome the problem of interoperability in supply chain environments.

PRONTO [21] is an ontology for product modelling that can deal with product variants and families. Products are modelled on two different hierarchy layers. The structural hierarchy is used to model structural product information, whereas the abstraction hierarchy is used to model product families and variants.

A socio-technical and human-centric approach is proposed in [22]. Knowledge sharing is done by accessing knowledge agents provided by partners or collaborators inside and outside a company of a VE. Each agent has a particular knowledge base and an access interface. Users can request knowledge of registered agents. Agents can also communicate to ask other agents for knowledge that is requested. Access to knowledge of agents can be restricted in terms of privacy protection.

A different approach is presented in [23]. By using semantic annotation, collaborators can use their own models and kind of data representation. Data in different formats is semantically annotated and distributed with a Semantic Annotation Structure Model (SASM), a meta-model for semantic annotations used for knowledge discovery.

In [24], an ontology-based model-driven approach is proposed. It is based on different hierarchy levels representing abstraction levels. Knowledge models from different model hierarchies are transformed and converted to each other. Interoperability is reached through sharing of base semantics. With this fundamental basis domain knowledge can be evaluated and verified.

The work discussed in this section can be categorised as product-centric (e.g. [21]), process-oriented (e.g. [14] or [16]) or human-centric (e.g. [22]) knowledge management strategies. This paper focuses on a process-oriented approach. Furthermore, this paper focuses on a shared knowledge base and extensions to this knowledge base by process collaborators to permit every process chain to have its own model structure. Emphasis is placed on semantic interoperability and a shared understanding. Modelling mechanisms provide a solution to link information accurately and meaningfully.

3. Process-Oriented Semantic Architecture

In this section, we introduce a novel architecture of process-oriented semantic knowledge management that allows management and distribution of appropriate knowledge across product lifecycle borders without forcing collaborators to change methodologies and structures.

For the purpose of an integrated shared knowledge base and the suggestion of a possibility to use ones' own modelling methodologies and process structures, the architecture is based on a shared and extensible knowledge base. *Knowledge Link*s between knowledge base extensions are used to model transformations between knowledge domains. Thus, knowledge from different knowledge domains is made accessible in a comprehensible way and semantic interoperability can be achieved.

To achieve a clear understanding of the architecture and its particular components, a running example taken from the aviation industry is presented across all component explanations. In this example, we assume that the knowledge about the range of an aircraft is needed by a sales person, while the sales person has no direct access to knowledge from engineering or construction.

The architecture of the approach, shown in Figure 1, is divided into three layers: the *Ontology Layer*, the *Data Layer* and the *Process Chain Layer*. The *Core Ontology* represents underlying general common knowledge, whereas the *Domain Ontology* extends this knowledge with domain specific concepts. *Process Chain Ontology Extensions* are attached to the *Domain Ontology* and uses its concepts while concepts of *Process Chain Ontology Extensions* are not distributed among other *Process Chain Ontology Extensions*. Process chains inside the *Process Chain Layer* may access their particular *Process Chain Ontology Extension.* The *Data Layer* stores data from the process chains in a suitable way and links the data to the *Ontology Layer*.

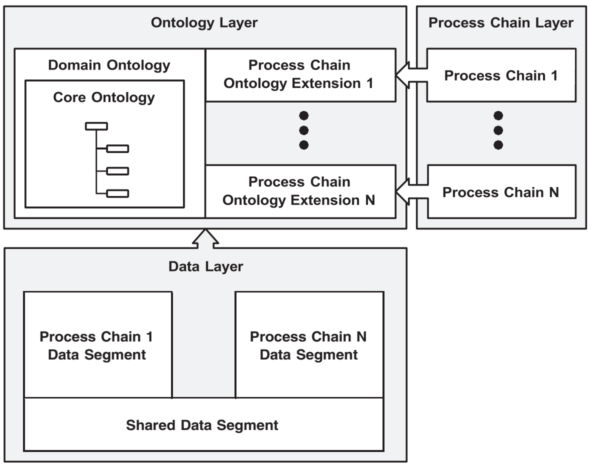

Figure 1: Basic Architecture

3.1. Ontology Layer

The *Ontology Layer* contains the semantic descriptions of product and process models. The ontology for the shared knowledge base consists of a *Core Ontology* and a *Domain Ontology*. The separation of the knowledge base makes it exchangeable and reusable, which can save time and effort for companies that manufacture products with the same generic product and process representation across sectors.

Core Ontology:

The *Core Ontology* semantically represents generic product and process information. For instance, this ontology structures information about the structure of a product, mathematical or physical units and measurements, product metadata and process representation concepts. Also, standard ontologies for generic product information like ONTO-PDM [19] or PRONTO [21] can be applied.

The *Core Ontology* of the ongoing example is indicated in Figure 2. Very basic aspects are derived according to the recommendation of the W3C Product Modelling Incubator Group [25]:

- The common meaning of units and scales
- A shared meaning of parts of a product

The structure of these ontologies is of any importance for this example. Another fact that needs to be modelled in the *Core Ontology* is:

- Weight is associated with the product and its parts.

Processes are basically defined as activities with inputs and outputs.

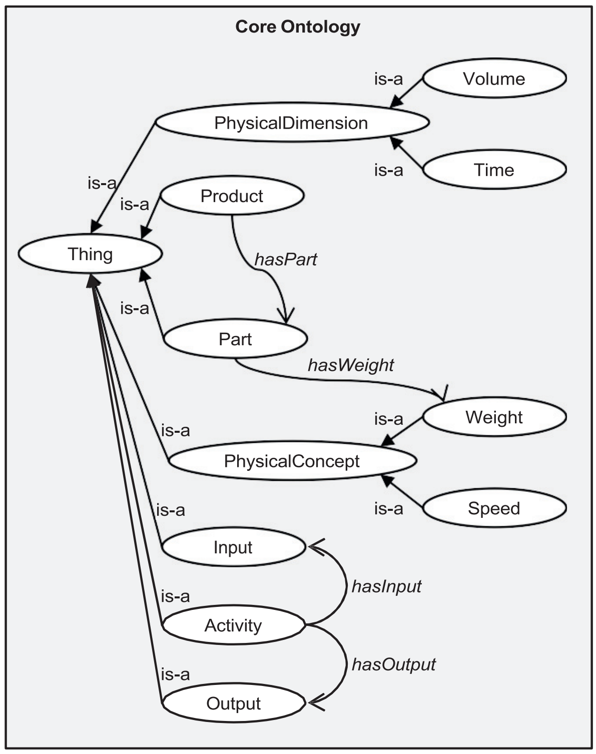

Figure 2: Core Ontology of the example

Domain Ontology:

The *Domain Ontology* comprises semantic descriptions of domain specific knowledge. Depending on the domain, this ontology extends the *Core Ontology* with domain specific vocabulary and relations of model components. For instance, in the aviation industry the representation of an aircraft with its assemblies and aerostatics is modelled inside this ontology.

Concepts in the *Domain Ontology,* shown in Figure 4, that are of importance for our example are basically concepts of aircraft parts like the engine, the fuel tank and the aircraft itself. Additionally, relationships to concepts like speed or volume are applied to the aircraft parts for a shared aircraft domain knowledge base.

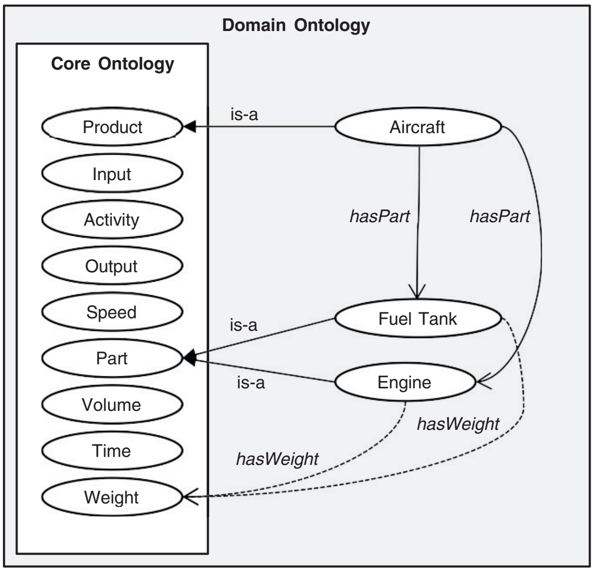

Figure 4: Domain Ontology of the example

Process Chain Ontology Extensions

Process Chain Ontology Extensions extend the shared knowledge base. They are modelled by contributors inside or outside a company that build a process chain to generate a particular outcome. In the context of PLM, departments or business partners that are responsible for product lifecycle phases are able to model enterprise specific product and process ontology extensions to fit their internal model and processes. Process chains are a modular and hierarchical paradigm. Therewith, contributors of a process chain are also able to subdivide their internal process chain to structure or delegate specific processes. For instance, the design specific ontology extension of a business partner that is responsible for product design in a business network is included in the *Process Chain Ontology Extension* of this particular business partner. Thus, the business partner can adapt its ontology extension to the output of its applications like CAD software.

Documenting knowledge is designated to be done via the Process Chain. Knowledge must be mapped to appropriate concepts of the Process Chain Ontology Extension or underlying and thus more generic concepts.

Process Chain Ontology Extensions can be queried to acquire the knowledge that is demanded. Queries can refer to concepts of the domain specific ontology extension included in the Process Chain Ontology Extension or to concepts of the Domain or Core Ontology. In the latter case, access restriction on the result set of the query must be checked at Data Layer. Queries thus always take concepts of permitted Process Chain Ontology Extensions into account and ignore those of restricted Process Chain Ontology Extensions.

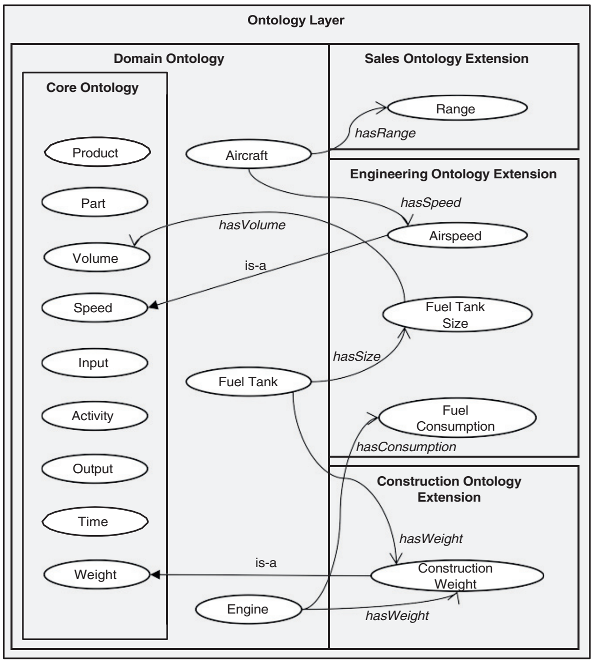

Figure 3: Process Chain Ontology Extensions of the example

For the running example, three process chains with their *Process Chain Ontology Extensions,* visualised in Figure 3, are of importance:

Engineering Process Chain:

The Engineering Process Chain extends the *Domain Ontology* with concepts needed during engineering. The resulting knowledge base is only shared by members of the process chain. This extension contains concepts for data like CAD drawings, tolerance calculation, quality parameters, kinematics and so forth. To keep the example simple, we consider the Engineering Process Chain Ontology Extension to extend the engine, tank and the overall aircraft by specifying fuel tank size, fuel consumption and speed of individual parts and aircrafts.

Construction Process Chain:

The *Process Chain Ontology Extension* of the Construction Process Chain includes concepts that the construction department uses to exchange information among constructors and other collaborators within this particular process chain. Concepts about data that is needed to construct specific parts, e.g. CNC programs, device properties or facility location and workflows, are located here. For the running example, the actual weight of each part after construction is located in the Construction Process Chain's Ontology Extension.

Sales Process Chain
Concepts for sales relevant data may include sales units, pricing, return of investment. Additionally, sales departments may need data for distribution to the customer like number of passengers, payload or the range of an aircraft. Concepts for data sales data are modelled in this particular *Process Chain Ontology Extension*. The example is limited to the range of an aircraft to keep it simple.

3.2. Data Layer

The *Data Layer* is the data representation of the underlying *Ontology Layer*. Concepts and relations are stored in an appropriate way. The structure heavily depends on internal enterprise or business network structures. For instance, small companies could rely on annotated file structures that are locally stored, while collaborating business networks with a high parallel communication activity rather rely on databases.

The Data Layer is divided into two segment types. A Shared Data Segment, which is shared by all contributors. There are no access restrictions on data stored in this data segment. In contrast, Process Chain Data Segments handle data from process chains. The access can be restricted to specific contributors or process chains.

For the running example, we assume that the data is stored locally and centralised. All contributors have access to the product model and access is not restricted. To keep the example simple, the data that is of importance for the example is stored in a database without going into detail about how the database is structured.

3.3. Process Chain Layer

The *Process Chain Layer* divides the whole product lifecycle into meaningful process chains. Process chains are formally represented in *Domain Specific Process Chain* models. Each participant of a process chain acts inside the process chain's privacy protected environment while the shared knowledge base stays accessible to everyone. Due to the separated extensions of the process chain's ontology and data structure, business partners that are responsible for particular process chains are able to implement own process structures that fit to their internal process structures.

The three process chains mentioned at the *Ontology Layer* are modelled here. Since basic concepts of process chains are included in the *Ontology Layer* and the data representation of those is defined in the *Data Layer*, the instances of process chains used to generate the necessary output are settled in this particular layer. Figure 5 shows the modelled process of acquiring the range of an aircraft. To avoid confusion the illustration of other processes in this example has been omitted. Some simple activities for the Engineering Process Chain are:

- *Calculate Airspeed*
  To calculate the airspeed, all properties that are necessary are gathered from the model. Especially, the overall weight and the power of the engines are needed.
- *Calculate Fuel Consumption*
  Some inputs of this activity are fuel efficiency of the engines and expected payload, as well as the weight of the aircraft.

An activity of the Construction Process Chain may be:

- *Calculate Construction Weight*
  Taking into account all manufactured parts and their attachments necessary to attach the part to the airplane, this activity calculates the overall actual weight of an aircraft.

The range of an aircraft is very important for customers of aircraft manufacturers. One modelled activity of a sales process may be to calculate the range of a customised aircraft that the customer configured according to its demand:

- *Calculate Range*
  To calculate the range, properties like weight, speed, fuel consumption and fuel tank size have to be taken into account.

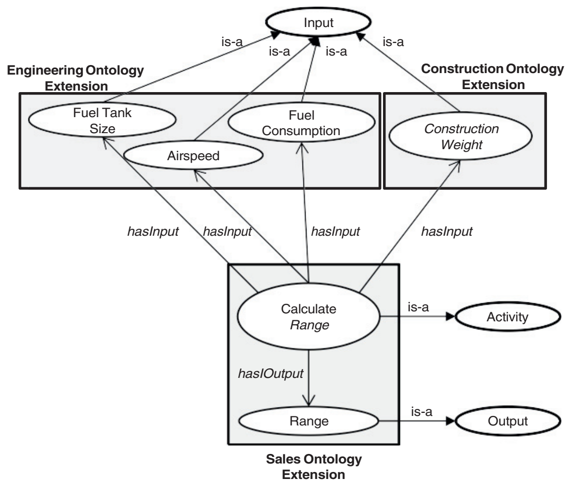

Figure 5: Aircraft Range calculation process

4. Knowledge Links

To provide a way how knowledge between different knowledge domains can be made semantically interoperable for machines and humans, i.e. comprehendible between different experts from knowledge domains, *Knowledge Links* are introduced.

Knowledge Links are connections between different elements of data models from different knowledge domains. Operators are provided to define transformations between data elements. To allow different process chain contributors to use model elements, the model elements need to be specified as public elements. Elements specified as private elements can only be used within their own process chain. For instance, a business partner can use public data elements from different process chains and private or public data elements from its own process chain to transform the elements with various operators into a new data element in its process chain, which is thereupon stored and made accessible.

Knowledge Links can either be established by collaborators with a specific knowledge demand alone or by communication with other collaborators that know about knowledge that is necessary for the establishment. Either way, the new data element is available like other data elements that are not induced by *Knowledge Links* for the demanding process chain after establishment. Furthermore, the definition is also stored in the Process Chain Data Segment of the demanding process chain.

Although data elements created with *Knowledge Links* can be queried in the same way normal data elements can be queried, they are not static. When querying data elements created with *Knowledge Links*, the definition is loaded and data elements from other process chains that are necessary for the calculation are queried. Thus, data from other process chains doesn't need to be maintained and is always up to date.

To exemplify the idea behind *Knowledge Links,* we take a closer look at the activities presented in the *Process Chain Layer*. The inputs depend on outputs of activities from other process chains. See Figure 6 for a visualisation of the exemplified *Knowledge Link.*

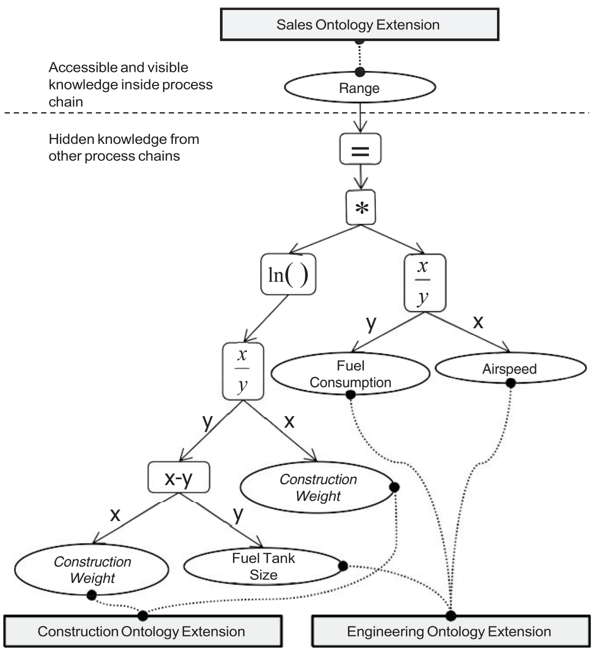

Figure 6: Knowledge Link for the Sales Process Chain

Data has to be made available to others. In our example, we assume that fuel consumption, airspeed, fuel tank size and construction weight is published to others but hidden inside the process chains' data.

However, data like *Fuel Consumption* is not necessarily useful for others outside the Engineering Process Chain like salespersons or even constructors. Hence, a *Knowledge Link* is established to deliver only the information needed by acquiring and aggregating information from outside the process chain. In case of the salesperson, who wants to know the *Range* of an aircraft, a *Knowledge Link* between the Sales Process Chain, the Engineering Process Chain and the Construction Process Chain is established. The output needed from Engineering is the *Airspeed* and *Fuel Consumption.* The output needed from Construction Process Chain is *Construction Weight.* These three values are combined as inputs to the *Knowledge Link* to generate the new output Range. The range is calculated in the following way after Louis Charles Breguet:

$$Range = \frac{Airspeed}{Fuel\ Consumption} \times \ln\left(\frac{Construction\ Weight}{Construction\ Weight - Fuel\ Tank\ Size}\right)$$

5. Discussion

The approach presented in this paper focuses on knowledge interoperability between collaborators of product lifecycle phases. Interoperability is aimed to be reached by using *Knowledge Links* and semantic representation of a shared knowledge base, whereas collaborators can rely on their internal structures. Thus, inhibitions to implement the approach are reduced.

In contrast to other approaches related to this paper, the problem of interoperability is encountered from the perspective of collaborators of process chains rather than from the perspective of the product model. Instead of sharing all knowledge along the product lifecycle, only selected knowledge is shared and additionally aggregated to be comprehensible for the target group. Thus, interoperability is extended from interoperability along data from different software tools to interoperability of knowledge between users along the product lifecycle. Furthermore, the modular approach, which doesn't define a concrete ontology and data model structure, allows other process oriented approaches to adapt the idea of *Knowledge Links.*

6. Summary and future work

Distributing knowledge across contributors and alongside the whole product lifecycle is critical to competitiveness, but making knowledge available doesn't imply that knowledge is comprehensible. This paper presents an approach to link knowledge between contributors of product lifecycle phases in an appropriate way while comprising a shared knowledge base. Internal knowledge, i.e. knowledge owned and produced by collaborators of a process chain, that mustn't be published to others, especially business knowledge of competing business partners in a business network is settled at extensions that are privacy protected. Knowledge that is published to other contributors can be requested and transformed with *Knowledge Links to* understand the knowledge provided by others and to deliver distributed knowledge appropriately.

The kind of operators to transform knowledge from different knowledge domains is not discussed in the present work. There is more effort needed to elaborate the different data sources and types that have to be taken into account during the whole product lifecycle.

Privacy protection inside process chains is a necessity where business partners come together and share a knowledge base. It is not yet specified in this paper how privacy protection is enforced and what exactly has to be privacy protected. Furthermore, which mechanisms comply with the requirements of business partners during the product lifecycle is a critical point.

Currently, we are working on the creation process of *Knowledge Links*. It has to be considered by whom *Knowledge Links* are being created and how they are stored. The critical point is to find an appropriate way how collaborators of different knowledge domains interact to establish those *Knowledge Links*.

Additionally, the approach will be applied to other fields of application besides product lifecycle management. In the OSMOSE Project, the approach could be adapted to handle semi-permeable knowledge management and transfer between the real, virtual and digital world by using *Knowledge Links* to create the inter-connections and settle the intra-world connections in process chains for the particular world.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°610905, the OSMOSE project.

References

[1] Sääksvuori A, Immonen A. Product lifecycle management, 3rd ed., Berlin; London: Springer, 2008
[2] Fortineau V, Paviot T, Lamouri S. Improving the interoperability of industrial information systems with description logic-based models—The state of the art, Computers in Industry, vol. 64, 2013, no. 4, p. 363-375.
[3] Farhad Ameri DD. Product Lifecycle Management: Closing the Knowledge Loops, Computer-Aided Design & Applications, vol. 2, no. 5, 2005, p. 577-590.
[4] Wang S, Noe RA. Knowledge sharing: A review and directions for future research, Human Resource Management Review, vol. 20, no. 2, 2010, p. 115-131.
[5] Brandt SC, Morbach J, Miatidis M, Theißen M, Jarke M, Marquardt W. An ontology-based approach to knowledge management in design processes, Computers & Chemical Engineering, vol. 32, no. 1-2, 2008, p. 320-342.
[6] OSMOSE Project Website, [Online]. Available: http://www.osmose-project.eu/. [Accessed 5 12 2013].
[7] Patil L, Dutta D, Sriram R. Ontology-Based Exchange of Product Data Semantics, IEEE Transactions on Automation Science and Engineering, vol. 2, no. 3, 2005, p. 213-225.
[8] Kim KY, Manley DG, Yang H. Ontology-based assembly design and information sharing for collaborative product development, Computer-Aided Design, vol. 38, no. 12, 2006, p. 1233-1250.
[9] OWL 2 Web Ontology Language Document Overview (Second Edition), [Online]. Available: http://www.w3.org/TR/owl2-overview/. [Accessed 10 10 2013].
[10] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, [Online]. Available: http://www.w3.org/Submission/SWRL/. [Accessed 10 10 2013].
[11] Young RIM, Gunendran AG, Cutting-Decelle AF, Gruninger M. Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies, International Journal of Production Research, vol. 45, no. 7, 2007, p. 1505-1519.
[12] Jun HB, Kiritsis D, Xirouchakis P. Research issues on closed-loop PLM, Computers in Industry, vol. 58, no. 8-9, 2007, p. 855-868.
[13] Kiritsis D. Closed-loop PLM for intelligent products in the era of the Internet of things, Computer-Aided Design, vol. 43, no. 5, 2011, pp. 479-501.
[14] Zhang WY, Yin JW. Exploring Semantic Web technologies for ontology-based modeling in collaborative engineering design, The International Journal of Advanced Manufacturing Technology, vol. 36, no. 9-10, 2008, p. 833-843.
[15] Schuh G, Rozenfeld H, Assmus D, Zancul E. Process oriented framework to support PLM implementation, Computers in Industry, vol. 59, no. 2-3, 2008, p. 210-218.
[16] Noël F, Roucoules L. The PPO design model with respect to digital enterprise technologies among product life cycle, International Journal of Computer Integrated Manufacturing, vol. 21, no. 2, 2008, p. 139-145
[17] Danesi F, Gardan N, Gardan Y, Reimeringer M. P4LM: A methodology for product lifecycle management, Computers in Industry, vol. 59, no. 2-3, 2008, p. 304-317.
[18] Chen TY. Knowledge sharing in virtual enterprises via an ontology-based access control approach, Computers in Industry, vol. 59, no. 5, 2008, p. 502-519.
[19] Chen YJ, Chen YM, Chu HC. Development of a mechanism for ontology-based product lifecycle knowledge integration, Expert Systems with Applications, vol. 36, no. 2, 2009, p. 2759-2779.
[20] Panetto H, Dassisti M, Tursi A. ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment, Advanced Engineering Informatics, vol. 26, no. 2, 2012, p. 334-348.
[21] Lu Y, Panetto H, Gu X. Ontology Approach for the Interoperability of Networked Enterprises in Supply Chain Environment, On the Move to Meaningful Internet Systems: OTM 2010 Workshops, vol. 6428. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture notes in computer science), 2010, p. 229–238.
[22] Vegetti M, Leone H, Henning G. PRONTO: An ontology for comprehensive and consistent representation of product information, Engineering Applications of Artificial Intelligence, vol. 24, no. 8, 2011, p. 1305-1327.
[23] Liu P, Raahemi B, Benyoucef M. Knowledge sharing in dynamic virtual enterprises: A socio-technological perspective, Knowledge-Based Systems, vol. 24, no. 3, 2011, p. 427-443.
[24] Liao Y, Lezoche M, Loures E, Panetto H, Boudjlida N. Formalization of Semantic Annotation for Systems Interoperability in a PLM Environment, On the Move to Meaningful Internet Systems: OTM 2012 Workshops, vol. 7567, Berlin, Heidelberg, Springer Berlin Heidelberg, 2012, p. 207-218.
[25] Chungoora N, Young RI, Gunendran G, Palmer C, Usman Z, Anjum NA, Cutting-Decelle AF, Harding JA, Case K. A model-driven ontology approach for manufacturing system interoperability and knowledge sharing, Computers in Industry, vol. 64, no. 4, 2013, p. 392-401.
[26] Böhms M, Leal D, Graves H, Clark K. W3C: Product Modelling using Semantic Web Technologies, 8 October 2009. [Online]. Available: http://www.w3.org/2005/Incubator/w3pm/XGR-w3pm-20091008/. [Accessed 12 10 2013].