Semantic Web Service Wrappers as a foundation for interoperability in closed-loop Product Lifecycle Management

Marco Franke*a, Konstantin Klein*a, Karl Hribernika, Dennis Lappe*b, Marius Veigta, Klaus-Dieter Thobena

aBIBA - Bremer Institut für Produktion und Logistik GmbH, Hochschulring 20, D-28359 Bremen, Germany

* Corresponding author. Tel.: +49 421 218 50094; fax: +49 421 218 50007. E-mail address: fma@biba.uni-bremen.de

Abstract

Closed-loop PLM (Product Lifecycle Management) is an approach to facilitating closing information loops between the individual phases and processes of the product lifecycle. These systems are heterogeneous, decentral and distributed throughout the network and over multiple lifecycle stakeholders. Stakeholders, and thus IT systems, may enter or leave the value chain of the product lifecycle. This means, many different applications, tools and corresponding data formats and standards are required to be integrated in a dynamic way, creating challenging data integration problems. The pervasive availability of relevant product information throughout the lifecycle for all stakeholders is required for the realisation of an efficient closed-loop PLM. Consequently, interoperability between the different IT systems is key to closed-loop PLM. Current interoperability approaches have several drawbacks in this regard. This paper focuses on the design and delivery of semantic interoperability services to meet the requirements of closed-loop PLM. The objective is to establish a general and highly flexible information exchange between different stakeholders. For this purpose, an overview over different approaches to interoperability, including Web Services, is given. Then, interoperability problems in closed-loop PLM are discussed. Subsequently, the paper details how these syntactical and semantic integration problems are currently addressed by Web Services. To resolve the hitherto unsolved problems, an extension of Web Service technology called Web Service Wrappers is proposed. Finally, the benefits of the extension are presented on basis of an exemplary closed-loop PLM use case. The exemplary use case addresses the information flow between the test process of an aircraft and its maintenance activities, bridging the gap between beginning-of-life and middle-of-life phases of the product lifecycle. A critical examination of the research results and an outlook towards future work conclude the paper.

Keywords: Product lifecycle management, data integration, semantic data integration

1. Introduction

Product Life-Cycle Management (PLM) defines the integration of different kinds of activities, from a technical, organizational and managerial point of view. These activities are performed by engineering staff throughout the entire lifecycle of industrial products. A wide range of heterogeneous IT systems hold relevant information to the management of the product lifecycle [1]: "...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 from traditional technologies or document management systems, it is critical to the usefulness of these applications to the organization that they share information between them..."

The given heterogeneity of tasks within PLM and the underlying IT-infrastructure requires new services which enable a pervasive availability of relevant product information at any point in the product lifecycle, both the exchange and aggregation of information between different product lifecycle phases are necessary. Up to now, there are a lot of data formats and standards which should enable the exchange of information between applications. But these data formats, like STEP for storing CAD models and QLM standards for exchanging product data cover only specific regions of the product lifecycle. There is no standard which enables an information exchange over the whole product lifecycle.

This article motivates a method that enables an information exchange over the whole product lifecycle. This method extends a web service to both an additional semantically layer and a transformation layer. This method is implemented as wrapper for a mediator based data integration approach. The objective is to establish a general and highly flexible information exchange method between different stakeholders in the product lifecycle. For this purpose, the technology Web Service is presented. In so doing, the approach “Web Service as a mechanism to achieve the interoperability” will be evaluated and the unresolved data integration issues will presented. On basis of the unresolved data integration problems an extension of the technology Web Services so called Web Service Wrapper is presented. Finally, the benefits of the extension are presented on basis of a use case. The use case addresses information flow from the test process of an aircraft to its maintenance activities, which follows on from previous work by the authors [2].

2. State-of-the-Art

2.1. The Product Lifecycle

Product lifecycle literature broadly differentiates between organizational/marketing and production engineering/ICT lifecycle models [3]. In the former, the focus is on sales and the lifecycle is accordingly divided into five phases: product introduction, growth, maturity, saturation and degeneration. The economic success of a product is the main concern of these models [4]. The product is understood as a model, type or category.

This paper adopts the model widely used in production engineering and ICT [5] which is shown in Figure 1. This model differentiates three main phases which describe the lifecycle of the product from the “cradle to grave” [6]:

- **Beginning-of-Life (BOL):** processes related to development, production and distribution
- **Middle-of-Life (MOL):** processes related to a product’s use, service and repair
- **End-of-Life (EOL):** processes related to reverse logistics like reuse, recycle and disposal.

In this model, PLM is commonly understood as a concept which “seeks to extend the reach of PDM” (Product Data Management) “[...] beyond design and manufacturing into other areas like marketing, sale and after sale service, and at the same time addresses all the stakeholders of the product throughout its lifecycle” [7]. Thus, PLM is a cross-functional and multi-domain task spanning, for example, business, marketing and manufacturing [8]. Classic PDM functionality encompasses object, component and document management, classification and search, change management and tools for system administration and configuration [9]. PLM also includes strategically modeling, capturing, exchanging and using information in all decision-making processes throughout the product lifecycle [10], [11]. It implements an integrated, cooperative and collaborative management of product data along the entire product lifecycle [12].

2.2. Closed-loop and item-level PLM

Conventional views of PLM tend to stress the first phase of the product lifecycle, due to its beginnings in PDM and CAD (Computer Aided Design). In these views, processes such as product design, development, production and sales are core. Approaches such as closed-loop PLM [13] take a more holistic view upon the entire product lifecycle, from the ideation of a product to its end-of-life processes. Ideally, the end of one lifecycle and its effect on the next should be considered, taking into account processes of recycling, refurbishing and reuse. It thus puts forward a paradigm shift from „cradle to grave” to “cradle to cradle” [14] (see Figure 3).
as models, types or categories, but as individual items (“item-level”).

2.3. Heterogeneity of IT systems in the closed-loop product lifecycle

Closed-loop PLM relies on the pervasive availability of information throughout the product lifecycle. The ICT landscape involved in the lifecycle of a single product is highly complex, distributed and heterogeneous. It involves many heterogeneous enterprise systems distributed across multiple stakeholders, many of which either operate proprietary systems or are small enterprises with no notable ICT infrastructure. Furthermore, stakeholders in the product lifecycle may enter or leave the value chain, making the addition or removal of data sources necessary. An example is maintenance, where independent maintenance providers may enter the product lifecycle in an unforeseen manner. The data generated by these providers can be valuable to other stakeholders in the value chain (e.g. in product design and test processes) and vice-versa, making the integration of their IT systems prudent to a closed-loop PLM approach. Consequently, a suitable and flexible integration of IT systems is required for information exchange between lifecycle phases and processes across multiple stakeholders. The interoperability approach needs to support data acquisition, integration and aggregation in a distributed and dynamic stakeholder network throughout the product lifecycle.

2.4. Tendency towards decentralized systems

Since 1996, the product lifecycle IT architectures have undergone a shift from monolithic, central systems to highly decentralized, loosely coupled and autonomous systems. This trend has been identified by the research initiative “Industrie 4.0” of the high-tech strategy announced by the German government [16]. From a technical perspective, the capabilities of the Internet of Things are being extended to Cyber-Physical-Systems (CPS), which divide systems into modular and autonomous entities, which are able to communicate, to recognize the environment and to make decisions. This development needs to be recognized by viable approaches to closed-loop PLM. Today, business is complex, dynamic and fast with a focus on distributed networks of enterprises throughout the product lifecycle, including design, manufacturing and logistics [17]. Global sourcing is realized by different companies adopting different roles in the product lifecycle. Each involved company has its own IT-landscape which often uses its data formats, protocols and semantics. An extract of the living IT-infrastructure for product lifecycle management of the development of an aircraft is given in Fig.2.

2.5. Approaches to Data Integration

In order to manage the interoperability of heterogeneous systems throughout the product lifecycle, different methods have been presented in literature [18], [19]. The following sections discuss the most relevant of these methods.

Interoperability approaches are broadly categorized in literature into tightly and loosely coupled approaches.

Tightly coupled approaches generally implement federated schemata over the systems to be integrated. This means that a single schema is used to define a combined (federated) data model for all involved data sources. The definition of such a schema requires considerable effort [20]. Access to the data is only possible via the federated schema. Any modification of any of the individual systems’ data models needs to be reflected by a corresponding modification of the entire federated schema. That means federated systems are quite inflexible due to the need to extend, contract or modify the federated schema manually each time a data source is added, removed or changed. This makes tightly coupled approaches such as federated database management systems unsuitable for the management of closed-loop PLM data.

Object-oriented interoperability approaches are closely related to tightly coupled ones. Different types of these approaches are described in [21]. The similarities of object oriented interoperability approaches to tightly coupled ones mean that they are also unsuitable for the problems discussed in this paper.

Loosely coupled interoperability approaches are more suitable to achieving scalable architectures, modular complexity, more robust design, supporting outsourcing activities, and integrating third party components. [22] The most widely used such approach today are Web Services. A Web Services is a communication method between two electronic devices over the internet. The W3C standardization organization describes Web Services as “…a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards…” [23]. Web API over representation state transfer(REST) uses the simpler communication protocols which not requires XML-based protocols SOAP for interfaces and communication establishment. Nowadays, Web Services are one of most common communication standard in business ICT. The semantic meaning of a service is described in platforms that use OWL (Web Ontology Language). This standard provides the semantic description over third party ontologies [24]. Web Services described in this way are called Semantic Web Services.

Service Oriented Architectures (SOA) describes a software architecture pattern which makes use of Web Services to provide functionality. At first glance, SOA seem suitable for the solving the interoperability problems addressed by this paper. Some problems, however, remain and discussed in the following sections.
3. The problem of data integration in closed-loop product lifecycle management

As discussed above, the requirements towards interoperability in closed-loop PLM include the ability to dynamically integrate heterogeneous software applications distributed over stakeholders in the product lifecycle. Each application offers interoperability functionality via various interfaces, data exchange formats and standards. For example, CATIA V5 allows access via Component Object Model (COM). This allows full access to all information in CAD models, including rules for knowledge based engineering. Similar situations can be found in other applications, which enable interoperability and collaboration over a network. Whilst the exchange of information between product lifecycle phases is necessary as motivated in [2], the information exchanged between companies is needs to be carefully selected to protect the companies’ business, research and innovation properties. A common approach is to enable an additional web interface which enables access to a subset of information, e.g. via Web Services. A Web Service offers a set of public methods whereby the information can be requested. The description of the available Web Service’s methods is codified using the Web Services Description Language (WSDL) and is deployed with every Web Service. A Web Service method description contains the method’s name, the input parameter and the output parameter. The structure of the parameter can range between primitive parameters and complex data structures. The definition of the parameter is represented as XML Schema (XSD). The interpretation of a WSDL description enables the formalization and a corresponding call of a Web Service method. In conclusion, a Web Service description covers all syntactic aspects of interoperability.

A Web Service description does not however contain any semantic description of which information can be reached via a Web Service method. This circumstance is currently addressed by applying human readable names to Web Service methods and their corresponding parameters. Thus, skilled employees interpret the names of Web Service methods and try to understand its meaning. The impact of a possible misunderstanding is investigated in [24]. A similar approach is given by semantic Web Services which include a reference to an ontology element for each Web Service.

A semantic description of a Web Service contains more than only the above mentioned mapping between Web Service method and a concept name. The dependencies (e.g. taxonomical, functional criteria like symmetry or transitivity) between information and corresponding Web Service methods and the semantic restrictions of the input and output parameters (e.g. value range or discontinuity) are not contained in the Web Service descriptions. To enable an automated data integration approach, a full semantic description of the information model beyond the Web Service is required.

4. Approach: A Wrapper for Web Service

A Wrapper, as motivated in [25], connects a data source to a data integration solution and therefore implements a loosely coupled, service-oriented interoperability approach. The main objectives of a wrapper are to provide a self-description of the information model beyond the data source and a transformation mechanism for the data structures. In the case of a Web Service, the information model is represented as an ontology. This kind of wrapper needs a special transform mechanism which links a Web Service method to a data property of the information model. For this purpose, the Web Service wrapper developed in the authors’ research contains an XML-based mapping file. The mapping file defines the information which can be requested by calling a specific Web Service method. Thus, a Web Service method can require an input which is the result of another Web Service method. In such a case, sequence of different Web Service methods has to be called to gain the final result. To determine the sequence, the input of a Web Service method is classified in three types: void, constant and data property. In the following the application of this transformation mechanism is illustrated on basis of the procession of an information request.

An incoming information request consists of concepts and properties of ontology. As the first step, the wrapper identifies on basis of its own OWL ontology which data properties are requested. Then, the wrapper creates a list of Web Service methods which can deliver the specific information. For this purpose, the wrapper iterates through the mapping file and searches the data properties. Subsequently the wrapper identifies for each Web Service method whether an additional Web Service method is required. This estimation is based on the classified input parameters of the transformation file. Finally, it calculates a sequence in which the Web Service methods must be called. The following example illustrates it on basis of an information exchange between the results of a test process and a maintenance activity.

4.1. Exemplary Use Case

The following exemplary use case from the avionics domain is based on the following closed-loop PLM scenario, and illustrates how the Web Service Wrapper approach can support interoperability between systems in the product lifecycle. It shows how the approach can be applied to exchange information between MOL and BOL processes and the relevant stakeholders in an efficient and flexible way. The scenario is based on work carried out in the WIB-funded project BreTeCe, in which the test process of the high lift system optimized. The Integrated Diagnostic System (IDS) of an aircraft had detected errors during the operation time of the aircraft. The corresponding maintenance activities couldn’t resolve the failure source. To identify possible failure sources further information of test process according to tested behavior and monitored signal curves were inquired. The access to such kind of information is provided by the aircraft manufacturer for third parties as a Web Service.

The example of a Web Service of Fig 3. offers the following methods:
The first method delivers all executed test cases for an aircraft type as a list of identifiers. With an identifier, the details of a test case can be requested. As details, the expected value of measuring point and signal curves are available. To get the signal curve of a specific sensor of a specific test case, an engineer has to call Web Service methods in the sequence 1;2;4. In so doing, he has to consider the meaning of the input and result parameters of the Web Service methods to determine the right order.

Beyond the Web Service the following information model is implicit given, which is illustrated in Fig.4.

In addition similar information model of Fig.4 is required by the Web Service wrapper and would be represented as an ontology. In this ontology each property like id or expected value would be represented as a data property.

To determine the above mentioned sequence of Web Service calls, the Web Service wrapper requires two kinds of information:
1. Which data properties can be requested by which Web Service method?
2. Which data properties are required to be used as input information for a Web Service method?

This information is configured in the XML-based mapping file of the wrapper. Fig.5. shows an example how the mapping is structured for the second Web Service method GetListOfAllMeasuringPointsForATestCase.

On basis of this mapping information the Web Service wrapper determines whether additional information is necessary. In the example of Fig. 5 the wrapper would identify that the id of a test case is required as input information for calling the Web Service method GetListOfAllMeasuringPointsForATestCase. To get this input information the wrapper would identify that the Web Service method GetListofTestCasesByTestCampaign is required.

This example demonstrates that the integration of a web service for PDM is available. The self-description of a web service (information model, transformation file) in combination with the Web Service Wrapper enables a plug & play integration mechanism. This circumstance simplifies the collaboration over the boundaries of companies in PLM.

5. Conclusions and Outlook

In this paper a design and delivery of a semantic interoperability service via Web Services to meet the requirements of a closed loop PLM was described. This kind of services enables the exchange of knowledge without addressing one specific standard. Particularly the information flow between the test and the maintenance processes of an aircraft were technically and functionally described and
depicted. The remaining relationships between product lifecycle phases are not researched.

The described approach was developed and is in evaluation in an industry use case in German national funded research flagship project CyProS and now part of a Future Internet initiative FITMAN. In this initiative the shown approach is described and used as a generic enabler. Within these activities following future work were identified.

Scalability: The amount of possible exchanged data must be researched to consider the applicability of this semantic data integration approach.

Security Aspects: nowadays security aspects are becoming more and more important. Particularly, security is an important factor in business services. Important functions in this environment have encryption, access restriction and secured authentication methods.

Rights and Roles: in business environments a right and role management is an important requirement. This concept realizes the function of defined access areas for specific user roles.

Model Transfer: usual methods for requirement specification and development in product development are modelling languages. Particularly in closed loop PLM it is important to ensure that the approached communication method covers the transfer of test, specification and development model. The enhancement of the described approach towards a model transferring method is one of the most important future task.

Divergent Data Formats: information container like video and audio are become more important today. To evaluate the described approach in this direction will ensure its sustainability.

Acknowledgements

The paper is based on research that is done in the research project “CyProS”, funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept “Research for Tomorrow’s Production” (fund number 02PJ2461)

References

[1] Reeve, A. (2013). Managing Data in Motion Datadntegration Best Practise Techniques and Technologies. USA: Elsevier Inc
[2] Franke, M., Hribenik, K., Thoben, K-D. Hans C.,(2013). Improving Maintenance Activities by the Usage of BOL Data, Procedia CIRP, Volume 11. pp. 62-67, ISSN 2212-8271
[3] Sundin, E. (2009). Life-cycle Perspectives of Product/Service-Systems: In Design Theory. In Introduction to Product/Service-System Design, London, Springer, 2009, pp. 31-49.
[4] Meffert, H., Burmann, C., &and Kirchegeorg, M. (2008). Marketing: Grundlagen marktorientierter Unternehmensführung. Konzepte- Instrumente- Praxisbeispiele, Wiesbaden: Gabler.
[5] Kiritisis, D.; Bufardi, A. and Xiroouchakis, P. (2003). Research issues on product lifecycle management and information tracking using smart embedded systems. In Advanced Engineering Informatics, 2003.
[6] Stark, J. (2005). Making Progress with PLM. A Manual to Support PLM Initiatives and PLM Projects. John Stark Associates.
[7] Golovatchev, J. & Budde, O. (2007). A holistic Product Lifecycle Management framework facing the challenges of 21st century. International Conference on Lifecycle Management, Zurich.
[8] Garetti, M., Terzi, S., Bertacci, N. &and Brianza, M. (2005). Organisational change and knowledge management in PLM implementation. International Journal of Product Lifecycle Management, 1(1), pp. 43-51.
[9] Abramovic, M., &and Sieg, O. (2001). PDM - Technologie im Wandel - Stand und Entwicklungsrichtungen. Orientierung für die Praxis. Industrie Management, 5, pp. 71–75
[10] Stark, J (2011). Product Lifecycle Management - 21st Century Paradigm for Product Realisation. (Springer, Ed.) (2nd ed.). London Dordrecht Heidelberg New York. doi:10.1007/978-0-85729-546-0
[11] Moorthy, A., &and Vivekanand, S. (2007). Integration of PLM with other concepts for empowering business environments. Product Lifecycle Management Special Publication 3, p. 93.
[12] Terzi, S.; Panetto, H.; Morel, G. &and Garetti, M. (2007). A holonic metamodel for product traceability in Product Lifecycle Management. International Journal of Product Lifecycle Management, 2(3), pp. 253 – 289.
[13] Hong-Bae, J., Kiritisis, D., &and Xiroouchakis, P. (2007). Research Issues on Closed-loop PLM. Computers in Industry, 58(8–9), pp. 855-868.
[14] Pokharel, S., and Mutha, A. (2009). Perspectives in reverse logistics: A review. Resources, Conservation and Recycling, 53(4), 175-182. doi:10.1016/j.resconrec.2008.11.006
[15] RENMARE, A. (2013). Evolution of PLM and SE inside Airbus. Abgerufen am 24. 03 2014 von Evolution of PLM and SE inside Airbus: http://www.pdeurope.com/media/5569/day_1_2_the_aerospace_industry__and_plm_c_and_se.pdf
[16] Bundesministerium für Bildung und Forschung, Industrie 4.0, http://www.hightech-strategie.de/de/59.php, 2013, [accessed 16.07.2013]
[17] Rennemann, T. (2007). Logistische Lieferantenauswahl in globalen Produktionsnetzwerken: Rahmenbedingungen, Aufbau und Praxisanwendung eines kennzahlbasierten Entscheidungsmodells am Beispiel der Automobilindustrie. Katholische Universität Eichstatt.
[18] Jun, H., Shin, J., Kiritiss, D., &and Xiroouchakis, P. (2007). System architecture for closed-loop PLM. International Journal of Computer Integrated Manufacturing, 20(7), pp. 684-698.
[19] Hans, C., Hribenik, K., &and Thoben, K.-D. (2010). Improving Reverse Logistics Processes Using Item-level Product Lifecycle Management. International Journal of Product Lifecycle Management, 4(4), pp. 338-359.
[20] Tatarinov, I. & Halevy, A. (2004). Efficient query reformulation in peer data management systems. SIGMOD '04 Proceedings of the 2004 ACM SIGMOD international conference on Management of data.
[21] Plouchre, E., Bukhres, O. & Elmagarmid, A. (1995, June). Object Design-Pattern, API and Reference Implementation”. J Biomed Semantics 359. doi:10.1186/2041-1480-2-8. PMC 3212890. PMID 22024447
[22] Rzewik, J. (2007). Data-Oriented Architecture: A Loosely-Coupled Real-Time SOA. White Paper. Real-Time Innovations, Inc. http://rtcgroup.com/whitepapers/files/RTI_DataOrientedArchitecture_WhitePaper.pdf. Last accessed: 28.04.2024
[23] W3C. February 11, 2004. Retrieved 2014-04-01.
[24] Wilkinson, Mark; Vandervalk, Benjamin; McCarthy, Luke (2011). “The Semantic Automated Discovery and Integration (SAIDI) Web service Design-Pattern, API and Reference Implementation”. J Biomed Semantics 2: 8. doi:10.1186/2041-1480-2-8. PMC 3212890. PMID 22024447
[25] Franke, M., Pirvu B.-C., Lappe D. and other (2013). 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, 2014 pp. 10-14