INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

IMPLEMENTING A SERVICE ORIENTED PLM ARCHITECTURE USING PLM SERVICES 2.0

D. Bergsjö, A. Ćatić and J. Malmqvist

Keywords: PLM, product lifecycle management, SOA, service oriented architecture, PLM Services 2.0

1. Introduction

The integration of both processes and products within a product data management (PDM) system has been difficult to achieve in industry. In particular integration of different disciplines that have been allowed to evolve separately such as the software and mechanical engineering discipline. This integration is however essential in order to increase time to market and innovation abilities; particularly in the automotive industry. The general direction of product lifecycle management (PLM) and product development has implied a single source strategy for gathering and managing product data in a single system. This strategy has proved hard to realize due to the distributed nature of engineering work and systems and so far it has been impossible to develop a system which supports this way of working (Stark 2005).

Challenges when integrating systems involve defining the master source of the information, the level of integration required, and how processes should be managed. Integration depends also heavily on the legacy of information and the traditions of the company (CIMdata 2006). There are primarily two approaches to integrating system and information. The first one is system level integration, where systems communicate with each other through common interfaces and export/import functionalities. The second approach is information level integration where the systems are integrated on a higher abstraction level, with a common information model (Hallin et al. 2004).

Crnkovic et al. (Crnkovic et al. 2003) describe the problems of integration of two classes of engineering IT systems, namely Product Data Management (PDM) and Software Configuration System (SCM) integration. The main cause of the challenges of integration between PDM and SCM system is the fundamental difference regarding the visions, assumptions and underlying technologies in the two separate domains which is recognizable for many IT systems used at large companies. Persson-Dahlqvist concludes that there are three major factors which have to be considered during a successful integration: tools and technologies, processes, and culture and people's behavior (Persson-Dahlqvist 2005).

Service-oriented architecture (SOA) is an approach to design software applications and specifically within the scope of this paper, a PLM system that is not dependent on a rigid server and client architecture of a multi-tier environment. SOA will make it possible to integrate systems that are heterogeneous (that have custom information models and processes, and is therefore a possible approach to bridging gaps between e.g. PDM and SCM systems under the PLM umbrella. The idea is to create services that collect, distribute and even modify information in several databases. These services are then reachable from e.g. the user PLM interface when a function that involves several disciplines has to be carried out. SOA is an architecture that can be enabled by a web-driven architecture by the use of java services that utilizes protocols such as XML, SOAP and WSDL for

communication between independent tiers (Georgiev et al. 2007).
SOA as an industrial application has been evaluated by (Lee et al. 2007). In their view four aspects are important regarding a SOA; the services, the Enterprise Service Bus, Business Process Management, and Enterprise Portal. The services are defined as providers of reusable business functions in an implementation independent function that are loosely coupled to other business functions. The service bus is the integration middleware where applications are connected by services. Business process management's main function is to provide integration of scatted systems where SOA would offer a smooth integration. Finally the enterprise portal is used as the presentation layer where users can take in the information provided by the service oriented PLM system.
Burr et al. (Burr et al. 2005) show that the integration of PLM systems today is not working properly, resulting in data losses, especially when handovers occur in the development process. Integration between systems can take place in different levels. Burr et al. suggest a Best-in-class integration, where the best systems from each engineering discipline are integrated on a corporate level. They also suggest another approach called the All-in-one integration concept, where the master system is directly connected to the applications; this approach is based on one single storage database where terminology and information are standardized and used in order to tie engineering applications together. These approaches are further elaborated and evaluated in (Bergsjö et al. 2006).
In order to standardize the application of web services specifically for PLM systems OMG and Oasis have developed their own set of standards independently. The OMG standard is based on ISO 10303 AP 214 and is recognized under the name PLM Services (Feltes and Lämmer accessed on 24/11/2007), where version 2.0 is the latest edition, still under revision as this paper is written. The second standard, Oasis PLCS PLM web services definition is based on AP 239. (Vec-Hub accessed on 24/11/2007).
So far it has not been shown in a research context how to achieve a proper implementation and evaluation of a SOA based PLM architecture based on a standard such as PLM services 2.0. This paper aims to bridge the gap regarding SOA and the possibilities to standardize such a PLM architecture, and evaluate the standard in order to improve both the standard itself and the practical application of it.
The research questions for this study are the following:

- What are the practical implications from a business perspective of implementing a service oriented PLM architecture?
- What are the user advantages of a service oriented architecture?
- How suitable are the available standards in supporting a Service oriented PLM architecture?

In section 2 the methodology for how the service oriented PLM architecture was deployed and the way the demonstrator case was chosen are described. Section 3 explains more deeply what is meant by a service oriented PLM architecture. Section 4 explains the technology behind the service layer. Section 5 describes how the demonstrator case is supported by the PLM Services 2.0 and finally sections 6 and 7 contain the discussion and conclusions, followed by future work.

2. Methodology

In order to demonstrate and implement the ideas which have been discussed in several works mentioned in this paper the work with defining a suitable case for demonstration was initiated. Along with this an extensive search for different ways of realization of a service oriented PLM architecture was conducted in order to find other implementations and standards which could be applicable. The concept for the demonstrator was discussed and the general idea was that it should demonstrate the implications of service oriented PLM, from a business, implementor and user point of view. In order to make the demonstrator as realistic as possible it was decided that an industrial case addressing an existing problem where this integration could have a substantial engineering benefit was to be chosen.

Demonstrator case

In the search for a reference case it was clear that some kind of multi-domain and multi-organizational issue needed to be addressed in order to demonstrate the idea of integrating heterogeneous environments to support a common cause. The chosen case is engineering change management in a multi-domain product thus including involvement from several departments who deal with this issue at an OEM and also including a supplier. The case is about how a change in the turbo charger, in a turbo diesel engine, affects both mechanical and electrical components in the rest of the engine. The affected components may or may not be in the geometrical vicinity of the turbo charger. The turbo charger is developed and produced by a supplier to the OEM. Parts of the final application are shown in Figure 1.

Realization of a service oriented PLM architecture

The area of service oriented architecture within the computer programming domain is relatively mature with solutions such as web services based on communication using XML messages according to the SOAP standard. This provides a good base for making sure that the communication of data is assured. This is however not sufficient enough to alone provide the complete solution for a service oriented PLM architecture since it provides integration at system level but what is needed is integration at the information level. An extensive search for standards and reference cases provided a standard specifically addressing this issue. The standard, called PLM Services 2.0, is provided by the standardization body Object Management Group (OMG). This standard was chosen and implemented in the demonstrator case. It was chosen partly due to the fact that its origin is in the automotive industry and partly due to the fact that it is the only standard which is released and documented (compared to Oasis which is not officially released as a standard yet).

3. Service oriented PLM architecture

Up until recently, the introduction of commercial PDM/PLM software solutions the approach towards their implementation tended to rely on the idea of a single source for all the data. In the beginning there was limited need to integrate different disciplines since e.g. mechanical and electrical design could be separated. In the automotive industry this was possible due to the fact that most functions were realized by physical systems where electronics played a minor importance. However, the idea of a single source database was further promoted by the fact that the in-house developed legacy systems were and still are perceived as a single system. At the observed companies however the legacy system has one name but usually consists of many different databases and applications on top of them which process the data. This has also been observed by e.g. Svensson (Svensson 2003) who points out that the legacy PLM environment tends to be made up of a variety of applications and databases which were implemented every time a business need was recognized. This way of expanding the PLM environment also meant that much of the information was duplicated and that a lot of time is lost in feeding the same information into different systems. As the legacy PLM environment usually has a common name it is natural to think of it as being a system which can be replaced with another system fulfilling the legacy's functions. As the commercial PLM software solutions began to gain functionalities which were further beautified by the vendor's promises of functionalities some companies decided to replace their legacy PLM environments with commercial solutions (Zimmerman et al. 2008). This shift from legacy to commercial PDM/PLM systems was driven by the increased globalization which in some cases meant integrating suppliers in the product development activities and in other cases meant mergers of companies through acquisitions or partnerships. No matter of the cause; increased globalization required a redefinition of parts of the information models and the process models concerning they corporate traditions and legacy, also known as the business logic. The attempt to replace legacy systems with commercial solutions might also lead to the replacement of those parts of the legacy which were efficient from a company specific process support point of view leading to less efficient commercial solutions for those processes (Zimmerman et al. 2008).

For companies today who need to implement PLM it is important to realize the value of their processes and business logic on which their complete business and uniqueness rests. Even though it may sound attempting to cut costs through outsourcing the PLM system by the implementation of commercial presumably easy to manage software on a global extended enterprise involving suppliers in all of the products lifecycle stages it is of critical business value to control and maintain the PLM architecture. This means that companies need to move from a situation of old tools realizing a complex business logic towards new tools required for the new business setup but to keep the essence of the business logic that has shown to be successful in the past. In Figure 2 this transfer from a complex legacy towards a single storage solution is depicted. However in this figure the problems with a single storage solutions is shown illustrated as that the processes (A-D) are not fully supported anymore and require changes to them (As illustrated by process B turning to B'' and C to C'). In order to save costs when introducing a commercial PLM system short term savings can be made by changing the processes instead of customizing the PLM system. It can also be declared that the PLM system should be kept as standardized as possible in order to reduce future maintenance costs, this will however further require process "work arounds" and adaptations.

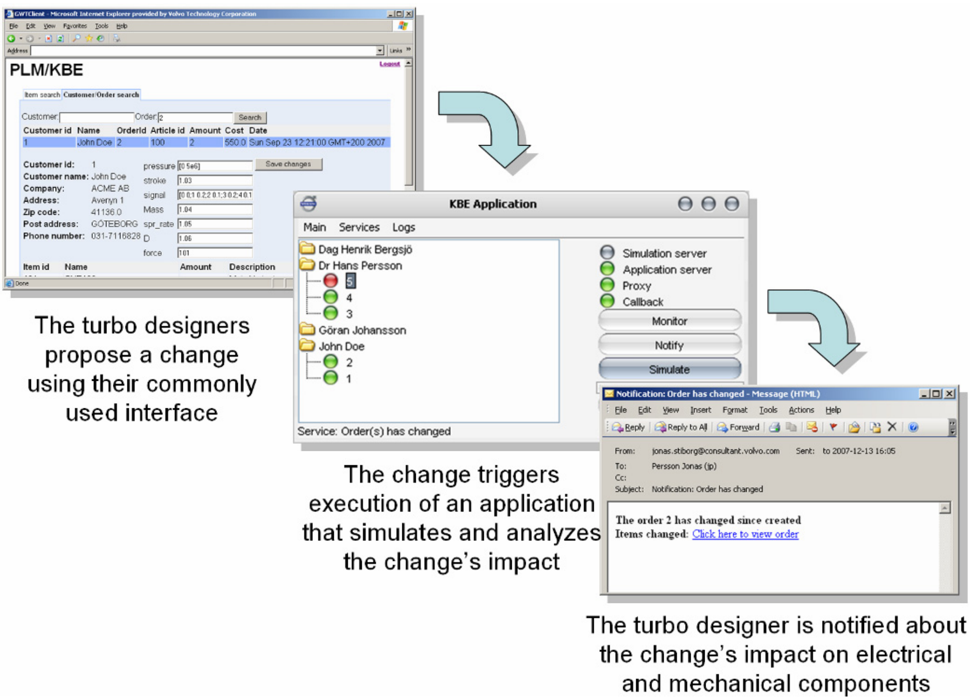

Figure 1. Demonstrator application

An other difficult task for a single storage solution is to provide suitable integration for all engineering disciplines and processes. In Figure 2 this is illustrated by the dashed lines that require manual labor. For example it could be that the PLM suppler locks in the customer by not providing open API:s to competitor software. It could also mean that the PLM suppler have not realized a specific need of a specific business and have decided not to support this features in their products.

Therefore a service oriented PLM architecture (to the very right in Figure 2) seems to be a promising solution for creating flexible integration and full process support; hence the business logic which the company has built up for decades around its products can be kept.

A service oriented PLM architecture has been described by [Bergsjö et. al. 8] as a solution to integrate information from different product domains in mechatronic products. Bergsjö et. al. extend the idea of a service oriented PLM architecture as a support for the extended enterprise as a more effective management of the PLM architecture and information architecture. The concept of a service oriented PLM architecture means that the applications and database layer is separated from the business logic and processes which should not be dependent on the IT-tools used.

This separation of business logic and processes from the tools is realized by considering applications as providers of information elements and processes as consumers of these information elements. The layer in between (the middleware) is based on a common contract according to which information elements are expected to be delivered. The contextualization of these information elements is done in the processes according to the business information model. This provides the processes with the independence from the database layer by a loose integration. At the same time the service oriented PLM architecture provides the IT governing organ with a possibility to control the IT environment in terms of choosing the best tools for the processes with the flexibility to change when so is needed.

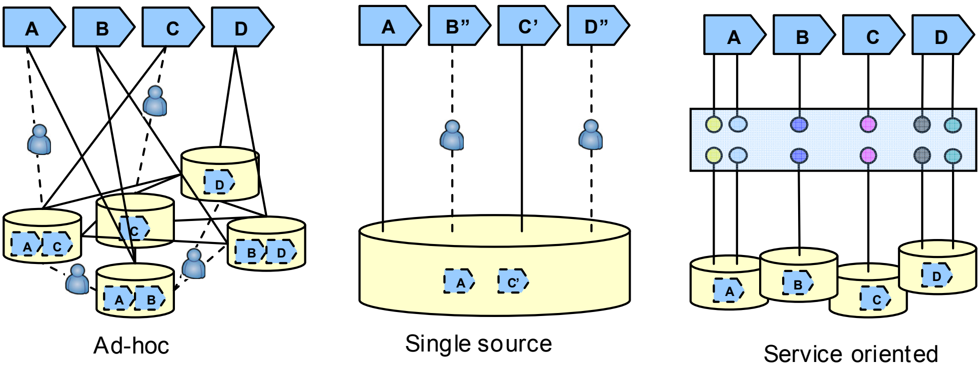

Figure 2. Different PLM architectures and integration concepts

A service oriented architecture implies certain requirements on the organization and management capabilities. Important aspects to consider in order to be successful with a SOA PLM system are: modularity (services depend on each other), central coordination (central governance of the service layer), standard communication (facilitates re-use and modularity), use of general modeling constructs (meta data in the service layer that facilitates it's management) and minimum process redundancy (services are re-used for same tasks) (Bergsjö et al. 2007). These aspects would enable management capabilities of a potential SOA chaos. Therefore in order to make a SOA based PLM system work continuous maintenance is required. Governance rules and a responsible organization needs to be assigned in order to maintain the services throughout their lifecycle (Bergsjö et al. 2007; Lee et al. 2007).

4. Application of PLM services 2.0 standard

The PLM services 2.0 is a framework that will give the implementation support in different stages and it is schematically illustrated in Figure 3. The service layer consists of a server application which has a common application programming interface (API) that communicates with the different databases and applications in the bottom layer. The server application communicates with a client application through HTTP sending XML messages according to a standard called SOAP (Simple Object Access Protocol). The client application communicates with the user applications (such as CAD, CAE, word processors, spreadsheets etc) or a graphical user interface (GUI) which finally communicate with the user. This is depicted in Figure 3. The implementor is provided with a description of the web services which make up the service layer. The description is provided both through a document which explains what the different web services are supposed to perform and how it is supposed to perform it and through WSDL documents (Web Service Description Language) from which the different web services can be generated automatically. The WSDLs exist for implementation of web services in the major programming languages such as Java and C++ and are readily available to download from OMG's website [OMG].

In order for the service layer to be able to communicate information elements there needs to exist an information model according to which the information is structured. Due to the fact that the standard PLM Services 2.0 has been developed within an automotive sphere the information model used in the service layer is AP214 edition CC21. The lead of the standardization is also taken by the major

German automotive manufacturers. This means that the standard is a way for the automotive companies to show what kind of integration capabilities the automotive industry want PLM suppliers to provide. In the extension this would also enable other players to use and benefit the standard.
In practice the information model is based on specific objects inheriting from generic objects. These generic objects are few and instantiated to cover the whole area of PLM which has led to them being quite general and thus vague. PLM object, PLM query, PLM container and PLM exception are the main information carrying objects. In addition to this the implementor is also free to add or neglect elements in the information model in order to adapt the service layer to fit with the business logic.

5. Demonstrator

In order to test the standard PLM Services 2.0 a demonstrator applying a service oriented PLM architecture as described in section 2 was designed. The demonstrator is to a high extent defined from a business and a user point of view where the process for a turbo change is followed and the different user actions are supported by the service layer. The different actions performed in Figure 4 are performed within the user applications depicted in Figure 3 which means that the user does not experience any difference when working with the service layer as compared with working towards the legacy PDM system. The user advantage here lies in that a higher extent of more relevant information

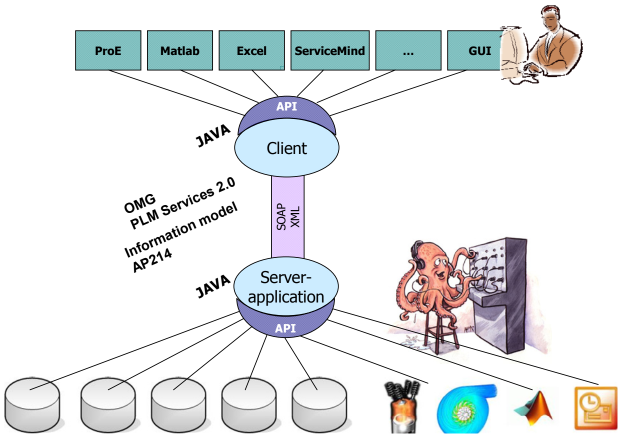

Figure 3. Service layer architecture

can be accessed (without the need for additional systems and interfaces). Further more this creates a basis for some of the processes to become automated. The user needs only to initiate the change, and then the information needed is supplied to the user through the service layer, rather than independent databases that has to be accessed independently. The user hence has to spend less time on information management issues and can rely on the service layer to supply the correct information. For demonstrator purpose there is also a web-based graphical user interface (GUI) available which directly accesses the service layer and provides an entrance for the user to find information. In a future industrial extended implementation of the demonstrator, this GUI would be integrated in the applications used by the engineers on a regular basis, such as the CAD system or the PDM system. From a business point of view this means that the applications and databases in the application and database layer (Figure4) are easily maintained since they are not directly connected to several end user applications but simply have one active integration link towards the service layer. As long as the communication fulfils the initial contract of service changes will not be noticed by the end users.
The demonstrator process is initiated with a suggested change of the turbo geometry performed by an engineer, as shown in Figure 4. The succeeding three steps in the business process layer are then performed automatically. The simulation and analysis application is triggered and collects the required information and performs a simulation and analysis of the new turbo geometry. Finally an e-mail

notification is sent to the affected end users that contains the change impact analysis. From the service layer view this task sequence is initiated as the users check in a new turbo charger and initiate a change request in the tool which they use for this task. The tool is connected to the client application which calls the server application to perform a save operation of the turbo charger object and initiate a change order in the applications which support these activities. In this specific case both of these operations are handled by a PDM system which is executed through its API. In other words; the PDM system provides the information services of storage of certain components and handling of the change order. In the same way the rest of the operations are performed by the other applications and databases depending on their respective domain. The executions of these operations are thus services performed to support the process and the end user.

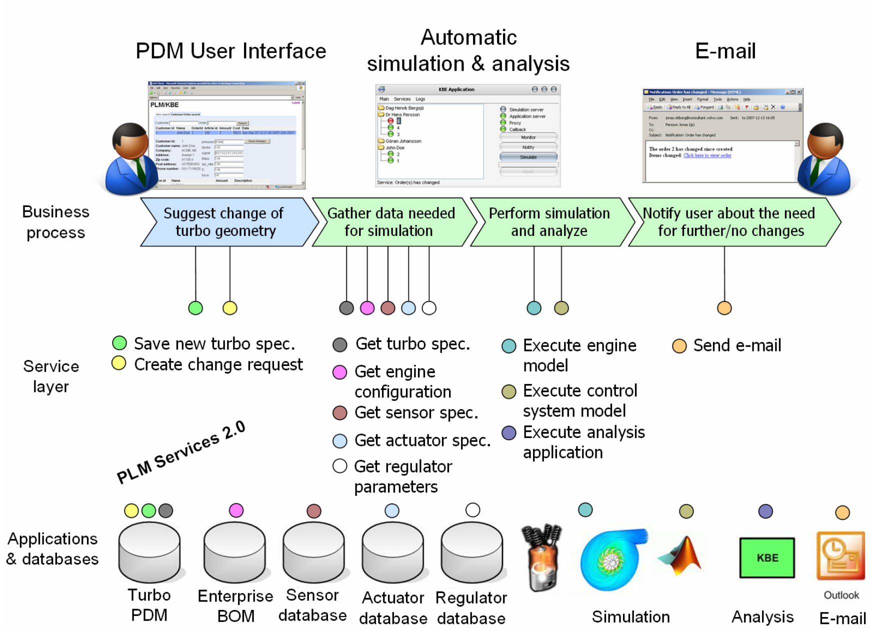

Figure 4. Process, information elements and data sources

6. Discussion

When reflecting over the study and the initial research questions it can be said that one of the practical difficulties with the PLM services standard is that it lacks detail e.g. a couple of queries were added in order to manage change orders. This implies that the standard had to be expanded to support the case we tried out. This means that the current service layer is not fully covered by the standard and integration to other PLM services 2.0 service layers are not likely to work right out of the box.

An extended implementation guide is beneficial in order to more quickly start to work with the implementation of the standard. OMG has chosen not to specify the interfaces which imply difficulties in order to use PLM services in the extended enterprise. The interface between customer and supplier is not fully developed either. PLM services is more focused on internal exchange of product information e.g. from product development to production. It is easy to adopt PLM services to your internal systems but it is difficult to assure integration within the extended enterprise.

Documentation is sparse throughout the standard. In the beginning it was difficult to assure required information (required annotation). A good knowledge in AP214 and expert programming skills is a prerequisite for understanding the implementation fully. The unspecific standard would make it

possible for different dialects of the implementations to occur, which is not preferable when different SOA's are being integrated.
The applicability of the PLM Services 2.0 has been tested practically with the use case from the change management within a turbocharged diesel engine. This demonstrator has been successfully developed using OMG PLM Services 2.0, which has been shown to be a feasible standardization effort, especially when considering the alternatives such as supplier single storages and software suites.
A problem with this standardization effort as with many similar efforts is its future use as an industry standard. We believe that OMG's effort here is a possible candidate especially in the automotive industry. The fact that it is based on ISO/STEP AP214 and being developed by and for especially the German automotive industry is a good sign that it will be used in the future. Competing standards are those that are being developed by large IT-suppliers where IBM and Oracle are suppliers that genuinely seem to be developing their own open standards for SOA focused on their particular fields e.g. finance human resources and enterprise resource planning systems. The disadvantage of those giants is that they are not the big players in the PLM field and the future will have to show how transparent their solutions are to work with for example engineering tasks and, more importantly, engineering tools.
One of the main advantages with SOA as discussed in this paper is that it enables every company to customize and standardize the IT-environment using a loose integration concept that would simulate a single storage towards the user. The users are not forced to work directly towards the database layer, but are working through their ordinary GUI's and applications towards the service layer. Problems with loose integration that have to be managed are aspects related to a more complicated governance and maintenance functions. The services exist in a layer outside the traditional PDM/PLM system which makes it more complex to manage. Related research as well as this research has shown that administrative tools with management and documentation capabilities need to be developed to support both the implementation and governance processes of service oriented PLM systems.
Traditional supplier focused SOA, suites and single source solutions have often been attempts to lock in the customer to use applications and systems from one single supplier. A different approach has been an attempt to duplicate information in a new location that later can be accessed in a standardized fashion, these so called hub solutions show instead data redundancy and data integrity problems. These different types of supplier lock-ins have in reality led to that companies would have to change their way of doing business in order to integrate and share information within and across the extended enterprise. With an open SOA standard this can be avoided. And with a large effort such as OMG PLM Services it is going to be possible to influence the IT suppliers to comply with the new standard, or even to keep legacy systems or develop new internal PDM/PLM systems that can communicate with external systems through the services they supply.
The usability issues of a SOA is basically that engineers would continue to work with the applications they like, but at the same time get customized services for performing time consuming information management tasks. A single interface towards a wide variety of databases and applications could also be developed. As SOA (Java) and the internet are working integrated most information management tasks could in this future be performed from a web interface.

7. Conclusion and future work

In this paper it has been shown that a service oriented architecture can benefit both user and business perspectives of PLM. These ways include, but are not exclusive to, issues regarding PLM architecture, control of the business logic and superior usability. The applicability has also been tested practically with the use case from change management in a turbocharged diesel engine. This demonstrator has been developed using OMG PLM services 2.0, which has been shown to be a suitable standardization effort.
PLM architecture is improved since a SOA allows transparency and flexibility to IT integration where supplier suites and single source solutions actively work against this principle. In a SOA that is based on an open standard such as OMG PLM Services 2.0 the principles of a SOA of modularity, central

coordination, standard communication, general modeling constructs, and minimum process redundancy can be managed.

The control of the company's business processes means that the company does not outsource the way it is doing business to an IT supplier, who usually does not understand this. The service oriented PLM architecture allows for flexible integration of the current business processes and instead puts demands on IT suppliers to support standardized interfaces rather than to force every company to work according to their PDM system logic.

Superior usability is achieved since information services are created focusing on a specific need of an engineer or a development process. These services do not change the way people used to work with the applications, but rather add a new service layer for those who benefit from that, and those are most likely engineers and managers working cross functional with new and innovative products.

OMG PLM services needs to be improved regarding documentation, a higher degree of support for the implementation, and a greater structure and detail level. This is in particular important for communication within the extended enterprise where a transparent implementation of the standard is a prerequisite in order to make different implementations of the standard to communicate with each other.

Future work includes work with a focus on general IT architecture. Focusing on governance functions as well as modeling and maintenance issues with a PLM architecture and the integration of both service oriented and legacy PLM architectures. It would also be interesting to further study the application of a SOA in an extended enterprise context.

Acknowledgements

The authors would greatly acknowledge the support from ProViking, Vinnova, Dr. Hans Persson from Volvo Technology, the interviewed engineers from Volvo Powertrain and Jonas Persson, Martin Sjöblom, and Jonas Stiborg.

References

Bergsjö, D., Malmqvist, J. and Ström, M. Architechtures for Mechatronic Product Data Integration in PLM Systems. Design 2006, Dubrovnik, Croatia.(2006)

Bergsjö, D., Vielhaber, M., Malvius, D., Burr, H. and Malmqvist, J. Product Lifecycle Management for Cross-X Engineering Design. ICED'07, Paris, France.(2007)

Burr, H., Vielhaber, M., Deubel, T., Weber, C. and Haasis, S. "CAx/engineering data management integration: Enabler for methodical benefits in the design process." Journal of Engineering Design 16(4): 385-398. (2005)

CIMdata (2006). PLM and ERP Integration: Business Efficiency and Value. Ann Arbor, Mi, USA.

Crnkovic, I., Asklund, U. and Persson-Dahlqvist, A. (2003). Implementing and Integrating Product Data Management and Software Configuration Management. Norwood, MA, Artech House.

Feltes, M. and Lämmer, L. (accessed on 24/11/2007) "Introduction to PLM services 2.0, http://www.prostep.org/en/standards/plmservices/." DOI:

Georgiev, I., Ovtcharova, J. and Georgiev, I. "Modelling web services for PLM distributed system." International Journal of Product Lifecycle Management Vol. 2(No. 1): pp. 30-49. (2007)

Hallin, K., Zimmerman, T. and Malmqvist, J. Towards a Framework for Integrated Information Management in Mechatronic Product Development. Proceedings of the NordDesign 2004, Tampere, Finland.(2004)

Lee, T., Lim, J., Shin, J., Myung, S., Choi, M., Baek, S., Kim, J., Oh, J., Lee, D. and Han, Y. An Implementation Methodology of SOA Based PLM System. PLM'07, Bergamo, Italy.(2007)

Persson-Dahlqvist, A. (2005). Product Data Management and Software Configuration Management Integration, Licentiate Thesis, Mälardalen University, Västerås, Sweden.

Stark, J. (2005). Product Lifecycle Management: 21st century paradigm for product realisation. Springer, USA.

Svensson, D. (2003). Towards Product Structure Management in Hetrogeneous Environments. Engineering and Industrial Design. Göteborg, Chalmers University of Technology.

Vec-Hub (accessed on 24/11/2007) "Virtual Enterprise Collaboration Hub (VEC-Hub), www.vec-hub.org." DOI:

Zimmerman, T., Malmqvist, J. and Ćatić, A. "Implementing PLM in Practice - Findings from an Empirical Study in the Commercial Vehicle Industry." To be Submitted to International Journal. (2008)

Dag Bergsjö MSc, Lic. Eng.
PhD Candidate
Chalmers University of Technology
Department of Product and Production Development
SE – 41296 Göteborg, Sweden
Tel.: +46 (0)31 772 1378
Email: dagb@chalmers.se