FEDERATED THROUGH-LIFE SUPPORT, ENABLING ONLINE INTEGRATION OF SYSTEMS WITHIN THE PLM DOMAIN

Jonas Rosén

Abstract

Working on product data across organizational boundaries through out the lifecycle of the product is a reality today. However, from a computer system and information management perspective this is not yet fully supported. A major obstacle seems to be that computer systems stores information in ways that are dependent on the governing business process, thus sharing and exchanging information is difficult and usually costly. In order to leverage existing investments of computer systems and without changing existing business processes there is a need for information sharing services that manage shared information and enables information integration, which requires the information to be represented independent of which business process is used to manipulate the information. Hence, information is structured into context and content.

This paper describes the ongoing development of a Service-Oriented Architecture framework (SOA) implementing Web Services based on the Product Lifecycle Support (PLCS) ISO standard. To verify the usability the Web Services have been used to integrate commercial PLM systems, i.e. Dassault Systemes SmarTeam, UGS TeamCenter Engineering, IBM Requisite Pro, Telelogic DOORS and Share-A-space.

Keywords: STEP, PLCS, information integration

1 Introduction

Product Lifecycle Management, PLM, is nowadays the collective concept that targets product information management in various aspects. From an information context, PLM should cater for the management of the information throughout the lifecycle of the product, including multiple domain views, different business processes scattered across enterprises and different representations of a multitude of native product-, resource- and process-models [1,2].

From a computer system perspective, PLM implies management of information that is created in dispersed systems, each providing their own management scope, implementing their own business rules and many times hiding the information from data exchange capabilities. In a virtual organization it is necessary that each participating company can still keep its competitive advantage by; a) owning their own business processes, b) have their own local master PLM information base, and c) use the computer systems being most suited for the company. In order to be agile, a virtual organization should operate in an environment where PLM information can be exchanged across company business processes and computer systems without imposing new business processes for each virtual organization setup. Otherwise, the overhead for each new virtual organization will increase the burden for participating companies. Thus, a PLM computer system environment must facilitate data exchange capabilities for the information that needs to be shared, both from legacy-, commercial-, and future computer systems. Also, since information is shared by different systems, there needs to be collaborative PLM management systems that controls the shared information, otherwise exchanged and shared information can not be aggregated, or put into context of other information.

2 Computer systems integration

Seen from a technology perspective, there exist many standards that are commercial mainstream, e.g. XML, Web Services, J2EE etc. Most systems, commercial and legacy, adhere to some method for data exchange, or Application Programming Interface, API. Integration between systems is also a domain of its own that includes technical concepts such as Business Process Integration, BPI [3], which have opened a market for software products that provide an environment that leaves out the technical plumbing and enables a higher level of abstraction to be addressed [4,5,6]. However, this new generation of integration tools focus on abstracting technology and facilitating mapping of different message formats into one another, it does not say anything about how information is consolidated when being exchanged. Nowhere can be seen the complete picture of the information that is, or actually should be, aggregated. Its more providing means necessary for peer-to-peer networking.

3 Information integration

Integration of computer systems is comprised partly of technical facilities, such as exposing an API for a number of contemporary programming languages, e.g. C++, Java etc., or distributed object-oriented architectures, such as CORBA, J2EE, or Web Services. Further, systems integration provides technical facilities to manage secure transport, loosely coupled messaging and message broadcasting. However, systems integration also comprise information integration, i.e. different representations of product information needs to be mapped, information that is integrated between systems should not deteriorate, information that lives in different integrated systems must be preserved according to the original semantics. An integrated systems environment needs to take into account that pieces of information lives and is governed in many different systems at the same time and in a collaborative environment. The value of information is increased when one piece of information can be seen in the context of other pieces of information. For example, a virtual product is defined by information in many lifecycle domains, where each domain governs its view of the product, i.e. have their product-in-focus view, however, the traceability or relation, between, for example, the design- , the in-service- and the requirements domain should be the glue for a global configuration and must also be managed.

3.1 PLM from a content and context perspective

Looking closer at the information being used in a PLM environment, it is comprised of content and context (assumptions). Context is mostly implicit and managed informal. The involved computer systems each apply their own assumptions, rules and interpretations on the governed information. If the information should keep its integrity and quality it must have a formal representation for the context. In a PLM environment then information must keep its integrity, but information being used by different systems must also be used in different contexts. Today the major issue for successful integration between different systems is that the representation of information also explicitly includes the context dependent information. If all involved systems use the same shared assumptions then information might be exchanged, or if the information that needs to be exchanged is separated from the context. However, applying an information representation that separates the content from the context would provide further capabilities for semantic applications, such as reasoning.

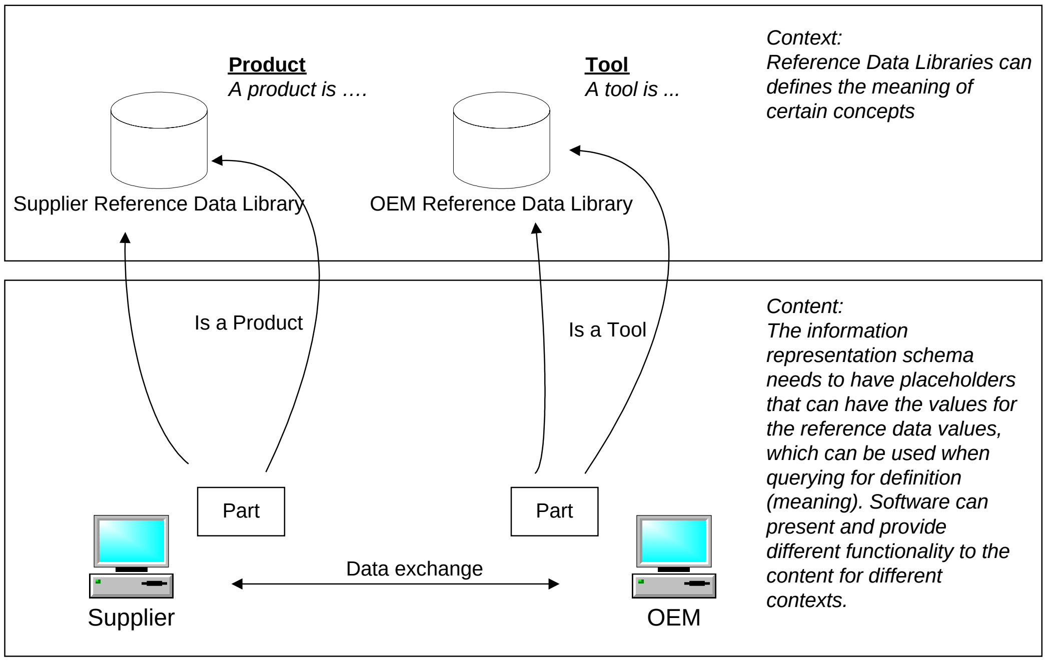

Figure 1. Information can be exchanged and used in different ways depending on the context, which defines the rules, presentation and behaviour. Computer applications manipulate content according to which context is applied.

4 An approach for PLM systems integration based on standards

To met the needs for the information that should be exchanged, shared, or by other means communicated across company borders and between dispersed computer systems in a heterogeneous environment, there needs to be means to represent PLM information such that semantics originating from various life-cycle disciplines can be integrated in one information model.

As a solution to enable cost efficient integration between PLM systems, there should be a core information model that allows various business processes to exchange, or integrate, PLM information independent of which business rules that governs a particular piece of information. Such a core information model should then be used to derive models for implementation. Within ISO 10303 [7] (STEP) a number of information models have been standardized with this in mind. Hence, a core information model would benefit from referencing the STEP standards, or simply be developed on top of existing STEP standards. The core information model represents the conceptual business domain model and downstream implementation models might need to change the logical structure (for a better fit into computer system artefacts such as database, APIs etc.), but not the semantics. The obvious reason for this need is simply because the information entities in the STEP information models are networked and suited for file-based data exchange. Analyzing the information entities in a STEP information model reveals that many of the pieces needed to acquire the full characterisation of a certain business entity, e.g. a *Part* and its *Part versions*, are associated via relationship entities, which results in a networked set of instances where its not possible to directly navigate from the characterized business entity. An implementation model can be derived via transformation of the core information model into, for example, a XML Schema definition. Such an implementation model defines a logical structure of the conceptual information model, which will add attributes in order to support easier navigation. A suitable implementation model is a XML Schema [8] that is used as a part of a Web Service Description Language [9] (WSDL) API. A WSDL is platform neutral and can be implemented by any systems that implement the W3C standards SOAP [10] and WSDL. The derived implementation model can be used for exposing the API for new computer systems, but more important is that it can be used as an API wrapping up existing legacy systems.

4.1 Standards for PLM information models

Within the ISO STEP organization there exists standardized information models that are used for (file-based) data exchange of product data. A number of different information models exist targeted for different business domains. Of those standardized information models, also known as application protocols (AP), there are some, which are of particular interest for the PLM domain. ISO 10303-214 (AP214), ISO 10303-203 (AP203) and PDMSchema [11,12,13] have been acknowledged for transfer of PDM data between commercial PDM and CAD systems, specifically for applications within the automotive and aero business segment. Recently the ISO 10303-239 [14,15] (AP239), or PLCS, Product Lifecycle Support, has started to be used for a wider range than just PDM and CAD. ISO 10303-233 [16] (AP233) is also a good candidate to further complement the standardized PLM information model landscape. Since the AP239 have a significant overlap with PDM found in the AP214, AP203 and PDMSchema, and also extends the information representation to cover other product lifecycle phases and being based on the same information constructs used by the AP233, then AP239 can be used as the reference information model for PLM. Moreover, AP239 explicitly have separated content and context information. Within OASIS [17] (standardization organization) there are a number of usage guides, Data Exchange Sets [18] (DEX), that defines how to use the AP239 for different business applications. Each of these DEXes use a number of capabilities (AP239 modules) that are shared by the different DEXes, c.f. Figure 2.

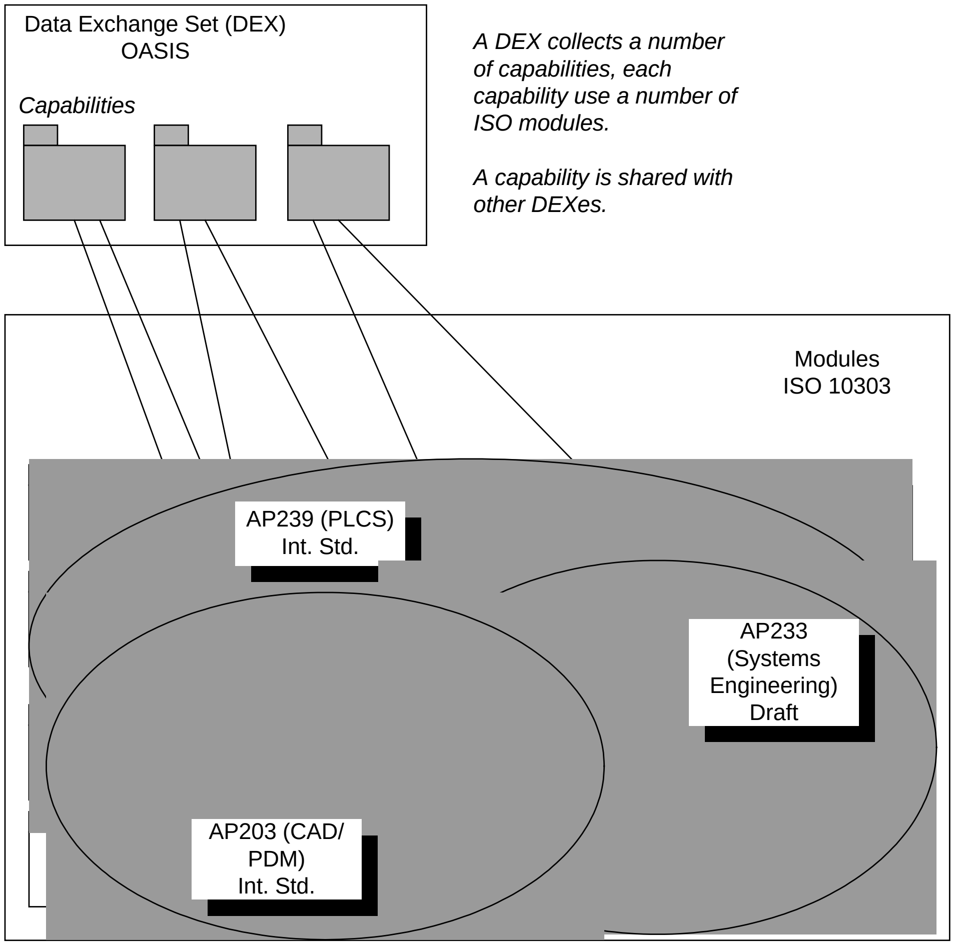

Figure 2. The architectural relation between ISO modules, ISO Application Protocols (AP233, AP239 and AP203), capabilities and DEXes.

Context information is defined separate from the AP239 schema. Certain allowed values of things in the AP239 schema are defined as an ontology that is managed within OASIS by the DEX organization. Thus, AP239 provides an information model, or schema, that is targeted for PLM in terms of content, and also have an explicit information architecture that separates context dependent information from the content model c.f. Figure 3.

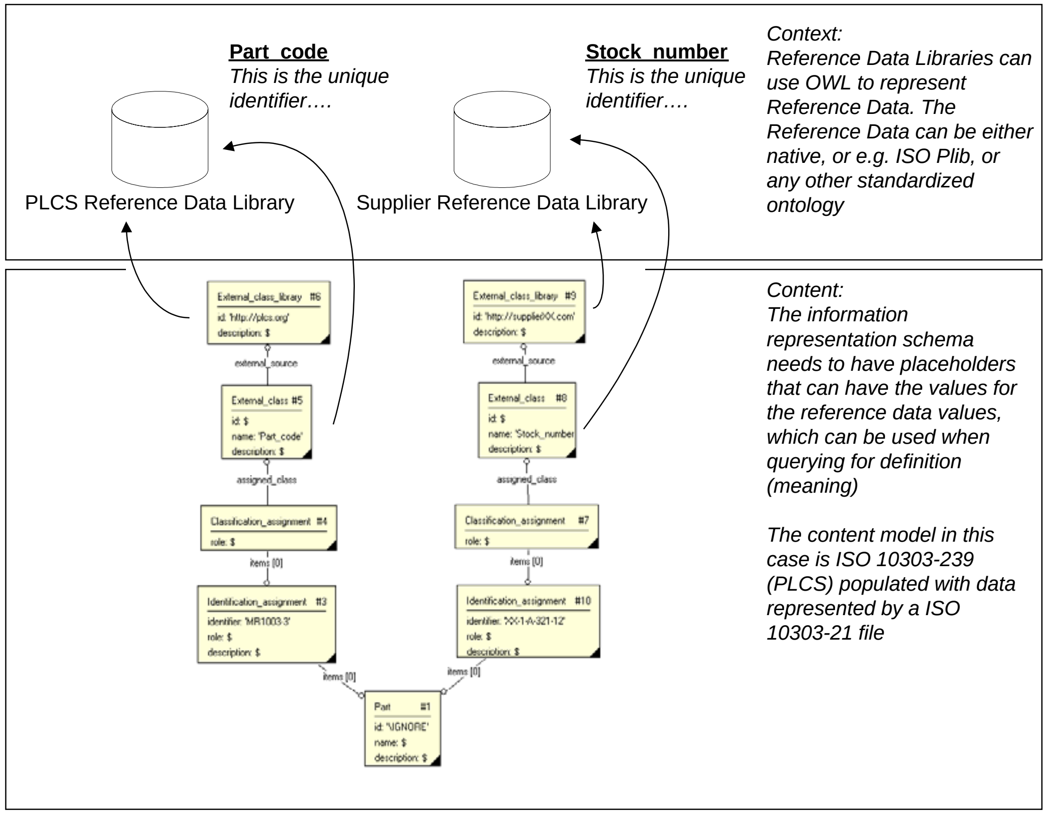

Figure 3. The content model, or schema, in this case ISO 10303-239 (PLCS) represents references to external reference libraries where the definition for the attribute "identification" can have different semantics, or even be represented with different names. This means that the content model is not dependent on business specific semantics, but can represent place holder, or references, to pieces of information that are governed by business specific semantics. The actual definition of the business specific semantics is managed in a Reference Data Library (RDL).

4.2 An integration platform based on a service oriented architecture

Conceptually, an application integration framework needs to provide three building blocks: (a) *"what"* - describe, the services format (API), (b) *"where"* - discover, how to find available services, and (c) *"how"* - transport, the wiring, or connection between the involved systems. The concept Service Oriented Architecture [19] (SOA) is now the collective term used with the increased use of Internet technology, such as http, XML and Web Services. There is development of standards for the three different integration building blocks taking place in different organizations, many times overlapping each other.

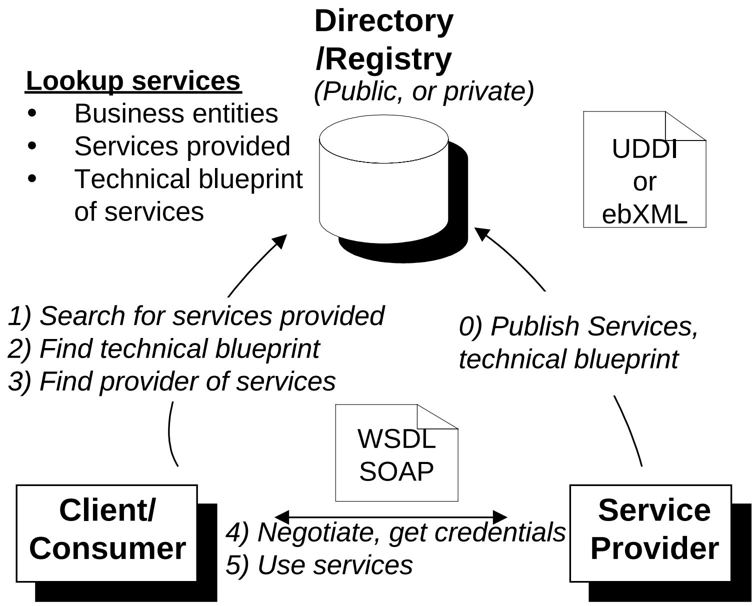

Figure 4. A Service Oriented Architecture, where Service a Provider (0) publish its services to a (public, or private) Directory where Consumers can (1) lookup the available services. Consumers can then (2) get technical blueprints of the services (the service API) and (3,4) find the provider of the service and negotiate (to get user credentials, establish business relationship).

A SOA implies that services are available for use by clients not known on beforehand. From a PLM perspective a SOA can be used to make access to in-house and commercial PLM systems in an integrated environment, thus, a scalable SOA requires the adoption of new business processes with a minimum of impact on the existing PLM infrastructure. As means to deal with a heterogeneous environment the use of one (standardized) PLM interface would make each PLM system user (consumer) independent of which PLM system is used to deliver the PLM service. Further, this standardisation would benefit clients that needs to access PLM information from many sources, thereby aggregating PLM information online. To further accommodate system-to-system integration there is a need to have a standardized API that defines events, which can be sent to a message bus. Hence, if PLM information can be sent (via events) each subscribing system can apply their business (context driven) logic.

4.3 Model driven development of a PLM integration platform

In order to plug in existing PLM systems into the PLM environment they need to expose functionality according to a given standardized PLM API. The proposed PLM API is based on standardized ISO STEP PLM standards. However, in order to produce an API there needs to be implementation models generated from the ISO STEP information models (conceptual models). To make a good fit with the SOA such an implementation model should be expressed as a WSDL API.

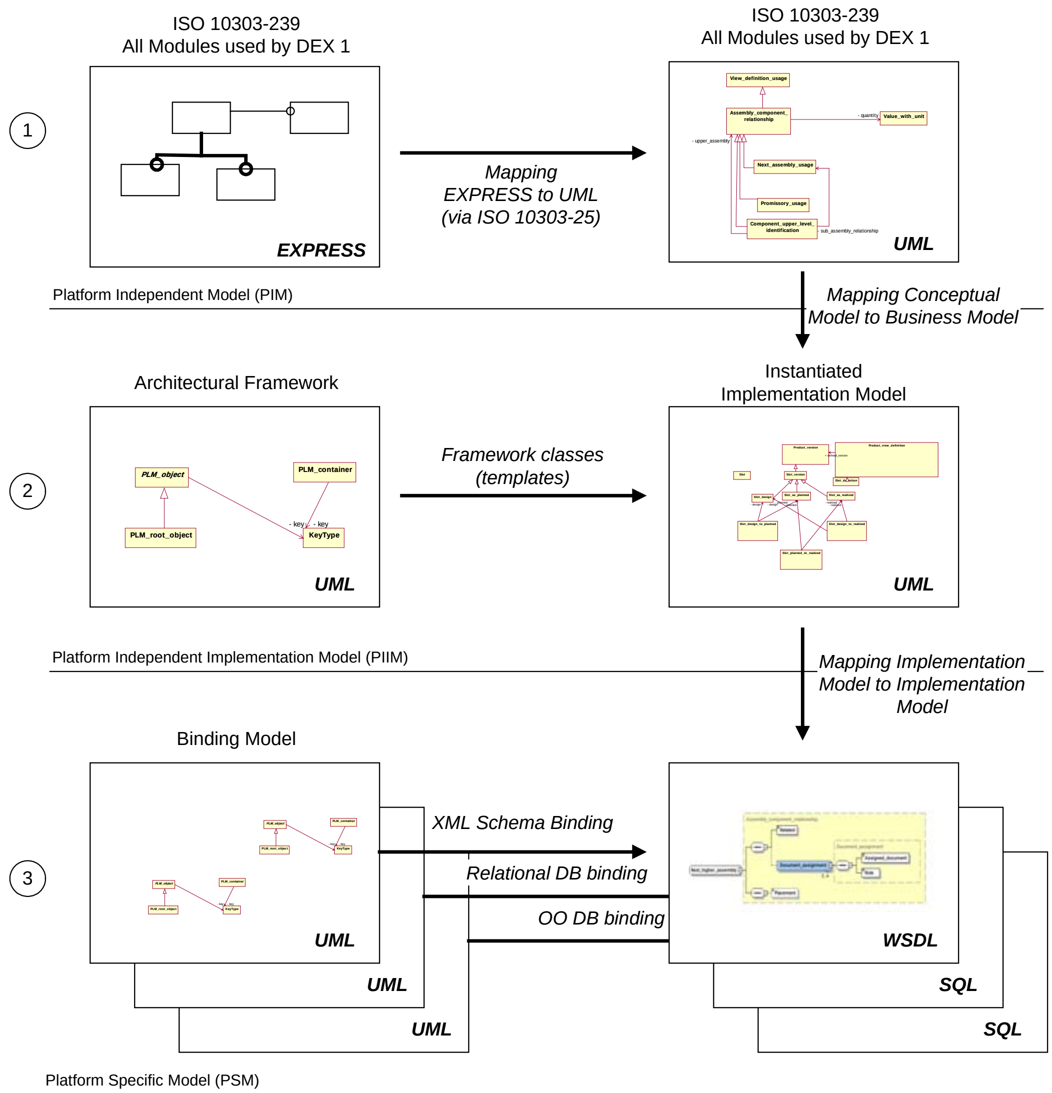

Figure 5. Existing PLM standards, for example ISO 10303-239, can be used to derive implementation models. The PLM API to be used in a SOA is generated as a WSDL. Existing systems that need to expose their functionality can take advantage of the model driven development [20] for developing the bridge between the existing system and the PLM API.

5 Achieved results

The author has implemented the research that has been presented in this paper in a number of commercial and academic projects.

A PLM Web Services API based on the ISO STEP modules is being developed within the European 6th framework project VIVACE [21]. Initial implementations have successfully been developed on the PLM solution Share-A-space [22] and two Requirements Management systems; a) RequisitePro (IBM) and b) DOORS (Telelogic) and also on two PDM systems; c) TeamCenter Engineering (UGS) and d) SmartTeam (Dassault systems) [23,24,25,26]. A PLM Web Services client is developed that aggregates PLM information from the various systems that provide implementations of the PLM Web Services API. The PLM Web Services client look up possible service providers by use of the business repository standard UDDI [27].

6 Conclusion

A core information model exist that represents product models throughout the lifecycle in an integrated way, capturing content and context and thus allows business process dependent characteristics to be separated from the actual product model content. Hence, computer systems that have their own business rules and product model representation, can still exchange and share the product data to other systems without changing the semantics. However, in order to provide a standardized PLM API there needs to be definitions for how to populate the information. This is addressed within the OASIS DEX lib organization, but needs more attention from the system implementers. So far the ISO 10303 STEP modules provides a sufficient information model in terms of representation, which is the starting point to generate implementation models, but the actual system implementation of such a standardized PLM API needs to be governed by explicit rules for how to populate the exchanged, or shared, information.

When PLM information is available to clients from many sources there is also a need to be able to configure chunks of information that is subject for configuration control. Since many systems govern pieces of information that now can be put together in a collaborative context, the assembled information needs to be managed outside each local master system. This will also make it possible to introduce traceability and change management that is distributed across different systems and business processes. Hence, establishing data exchange, or sharing, of information in a virtual organization, or PLM environment, will make information visible outside each local system when using the proposed solution. Moreover, by introducing virtual change management systems, like Share-A-space, configuration control can be applied on information within the virtual organization [28].

References

[1] Starck J., "Product Lifecycle Management 21st Century Paradigm for Product Realisation", Springer, ISBN 1852338105, 2004.

[2] CIMdata about PLM, http://www.cimdata.com/PLM/aboutPLM.html, 2005-04-22.

[3] Linthicum D., "Next Generation Application Integration: From Simple Information to Web Services", ISBN 0201844567, Addison-Wesley Professional, 2003.

[4] Tibco, http://www.tibco.com/, 2005-04-22.

[5] MicroSoft BizTalk, http://www.microsoft.com/biztalk/, 2005-04-22.

[6] IBM WebSphere, http://www-306.ibm.com/software/websphere/, 2005-04-22.

[7] ISO/TC184/SC4, ISO10303-1 Industrial automation systems and integration - Product data representation and exchange, Part 1: Overview and fundamental principles.

[8] Extensible markup Language, XML, http://www.w3.org/XML/, 2005-04-22.

[9] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, http://www.w3.org/TR/wsdl20/, 2005-04-22.

[10] SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation 24 June 2003, http://www.w3.org/TR/soap12-part1/, 2005-04-22.

[11] ISO/TC184/SC4, 1999, ISO/IS 10303-214, "Industrial Automation Systems and Integration-Product Data Representation and Exchange-Application Protocol: Core Data for Automotive Mechanical Design Process".

[12] ISO/TC184/SC4, 1994, ISO/IS 10303-203, "Industrial Automation Systems and Integration-Product Data Representation and Exchange-Application Protocol: Configuration Controlled Design".

[13] STEP PDM Schema, "The STEP PDM Schema Home Page", http://www.pdm-if.org/pdm_schema/, 2000-09-02.

[14] ISO/TC184/SC4, 2005, ISO/IS 10303-239, "Industrial Automation Systems and Integration-Product Data Representation and Exchange-Application Protocol: Product Life Cycle Support".

[15] PLCS at OAIS, http://www.oasis-open.org/apps/org/workgroup/plcs/, 2005-04-22.

[16] ISO/TC184/SC4, WD, ISO/IS 10303-233, "Industrial Automation Systems and Integration-Product Data Representation and Exchange-Application Protocol: Systems Engineering".

[17] OASIS, http://www.oasis-open.org/, 2005-04-22.

[18] PLCS Data Exchange Sets at SourceForge, http://sourceforge.net/projects/dexlib/, 2005-04-22.

[19] "SOA an overview", http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html, 2005-04-22.

[20] "OMG Model Driven Architecture", http://www.omg.org/mda/, 2005-04-22.

[21] "VIVACE", Integrated, intelligent, multidisciplinary design in extended enterprise, European 6th framework programme,

[22] "CIMdata about Share-A-space", http://www.cimdata.com/publications/article05-03_cimdata_commentary.html, 2005-04-22, also at http://www.share-a-space.com, 2005-04-22.

[23] IBM Rational RequisitePro, http://www-306.ibm.com/software/awdtools/reqpro/, 2005-04-22.

[24] Telelogic DOORS, http://www.telelogic.com/products/doorsers/doors/, 2005-04-22.

[25] UGS TeamCenter Engineering, http://www.ugs.com/products/teamcenter/, 2005-04-22.

[26] Dassault Systemes SMARTEAM, http://www.3ds.com/products-solutions/brands/SMARTEAM/, 2005-04-22.

[27] "Universal Description, Discovery and Integration, UDDI", http://www.uddi.org/, 2005-04-22.

[28] Ljung R., Johansson M. and Rosén J., "Sharing Product Data Through Life in the Extended Enterprise", Product Data Journal, Vol. 11, No 2, 2004, pp 40-42.

Corresponding author:

Jonas Rosén
Eurostep AB
Vasagatan 38
SE-111 20 Stockholm
Sweden
Tel: Int +46 8 20 04 50

Fax: Int +46 8 20 03 99
E-mail: jonas.rosen@eurostep.com