INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

DESIGN DOCUMENT MANAGEMENT IN NETWORKED PRODUCT DEVELOPMENT USING STANDARD FRAMEWORKS

Paavo Kotinurmi, Jukka Borgman and Timo Soininen

Abstract

Several B2B standard frameworks have been developed to support inter-company information exchange. We study how using such frameworks can support design document management in networked inter-company product development processes. We develop a rough framework for comparing the B2B standard frameworks and analyse several of them with respect to the requirements extracted from a case study of networked product development. Based on the analysis, the RosettaNet standard seems the most suitable. We further define an architecture and design for a system based on RosettaNet to support the required design document management. Our preliminary experiences with the design and its implementation show that using RosettaNet for this purpose is a feasible approach. However, some of the relevant parts of RosettaNet are still under development, and the current standard does not support communicating all the necessary attribute information on design documents adequately.

Keywords: Collaborative design, integrated and distributed product design, product data management, extended enterprise, web-based systems.

1 Introduction

Understanding and supporting design information sharing is an important issue for a successful networked product development (PD) process [1][2]. The information is usually stored in different kinds of documents such as CAD models within each company's Product Data Management (PDM) systems or other tools for managing design documents. Such systems facilitate the PD process in one company by providing up-to-date information to all the product developers who need it. The problems identified in networked PD projects indicate [3] that the same type of support for managing the design documents among the companies in the network is needed.

In this paper we present an approach to design document management in company networks that aims at satisfying the requirements identified in a case study of one PD network (Section 2). One possible solution is to have a web-based portal giving access to a common PDM system for the companies in the network and another is to build a point-to-point integration between the PDM systems of two companies. The requirements according to a case study indicate that a portal or point-to-point integration is not feasible, but using standard frameworks for exchanging design data and documents between companies might be a solution. We have analysed 15 standard frameworks such as RosettaNet, ebXML and the Standard for the Exchange of Product Model Data (STEP) against the requirements from the case study (Section 3). On the basis of the analysis, RosettaNet seems the most suitable framework to support design document management in networked PD. We describe the architecture, design and prototype implementation for a design document management solution for networked product development based on RosettaNet (Section 4) and discuss the benefits and shortcomings of this approach in the light of our preliminary experiences (Section 5). Finally, we present conclusions and topics for further work (Section 6).

2 Requirements for design document management in networked PD

This section first presents the problems and then the requirements related to design document management in networked product development. They are based on a case study of a company network, on which further details can be found in [3].

The case network consists of a customer and its suppliers. The customer designs, manufactures, and markets consumer electronics products and the suppliers supply plastic parts to these products. The inter-company communication is fast paced and documents are exchanged intensively, even several times a day in a project. The companies work in parallel to cut lead times in the overall process. It is typical that the combination of companies varies between projects as customer uses many suppliers and the suppliers have multiple customers. The design data exchange should also not require learning to use multiple PDM systems.

Two main problems in networked PD rose from the case study. Those are extra work caused by old document versions and extra delays due to lack of documents.

The latest version of a document (e.g. 3D CAD model, project plan, task lists) was not always available in time. There were three main reasons for this. (1) The document was not sent to all the designers in the network that should have received it. (2) Distribution between companies by email was done from project manager to project manager. Sometimes one of the project managers was absent (e.g. on a business trip or ill) or just forgot to forward new documents inside the company. (3) It was not always possible to determine to which component or to which project the CAD file belonged to. The use of email was not a very secure way for transferring documents (e.g. CAD files), so the companies used special data-transfer directories in certain computers for one-to-one communication using FTP (File Transfer Protocol). This meant that there was one special directory per supplier company, where a designer could save the document that needs to be exchanged. The document was automatically encrypted and transferred to the destination company. However, since these were just plain files, their names were the only way to distinguish between the transferred files. Sometimes these files could not be delivered to e.g. the supplier's mould designer. All the above reasons led to situations where work was done based on an old version of a design document. When the latest changed version of the document finally reached its destination, which could take days, some of the work done had to be done again.

Extra delays due to lack of documents. The projects were controlled through weekly meetings between the project managers of the companies, where a task list was updated. Sometimes, the memo of the meeting failed to be sent to a person responsible for an item on the task list.

Both of the problems described above led to a decrease in the controllability of product development projects, that is, lead-times could not be estimated accurately, which caused serious delays for the whole PD process. Based on the problem analysis we now present requirements for document management in networked PD. The main requirements are that delivery of new design documents should happen in a controlled and secure way.

Email is not a good option because of the problems mentioned above. A more systematic and controlled way for document exchange is needed. Document management should be a planned, predefined, communicated and documented process between companies in networked PD, not just accidental events between companies. The design documents drive the supplier's process in a very straightforward way: When a supplier receives a new version of e.g. a CAD-model from the customer it triggers immediately some design and manufacturing work for the supplier. Equally, when a supplier sends a document (e.g. a measurement log sheet) to the customer it triggers e.g. a new component approval process. Documents actually synchronise processes within the companies of the network. Thus, document exchange in a network should itself be considered and treated as *a systematic process*. This process can be triggered by a predefined schedule or an event within one company (e.g. a new version of a document becomes available). If any changes to the schedule should occur, these changes should be communicated to all relevant companies immediately preferably without unnecessary human interaction causing delays.

It is also very important to ensure that changed documents can be distributed inside the companies to the engineers whose work depends on them. Therefore, the relevant information about document relationships to product components and projects needs to be communicated with the document files. In addition, the document versioning information and relationships to other design documents are important.

Design documents are obviously highly confidential. There must be a means to *securely distribute* the documents to only the intended recipients and be certain that the delivery is successful. They should be encrypted if this takes place over non-secure communication channels such as the Internet. However, the engineers should not have to do additional work to encrypt documents; this should happen transparently and automatically.

The requirements identified are mostly the same as for any PDM system support within one company. In networked PD, the company borders and competition between the suppliers require good controllability and security for the design documents distributed. A proper solution supporting design data exchange between the systems should be extendable to many partner companies using different PDM systems and therefore based on standards.

3 Comparison of standard frameworks for supporting networked PD

In this section, we first present a framework for comparing standard frameworks based on the requirements identified in the case study. We further identify five potentially useful standard frameworks. On the basis of our analysis and comparison, the RosettaNet framework seems the most promising one. It is chosen as the basis for our support of networked produce development and described in more detail.

3.1 Comparison and analysis of standard frameworks

To achieve general system interoperability industry standard-setting consortiums have recently developed *frameworks*, that provide standards and specifications enabling businesses to communicate efficiently over the Internet [4]. Based on the case study, we defined a rough *comparison framework* for analysing potential standard frameworks (table 1). The comparison framework is the set of dimensions of requirements together with the possible values from standard frameworks. The brackets in the cell value mean that there is no concrete specification, but some assistance or guidelines are provided to define the content. The empty cell "-" means that nothing related is specified in the framework. The requirements are:

- *Processes* that specify activities that are to be carried out in a given order. Standard frameworks specify processes, participant roles in communication and the messages involved. For instance, when a message should be sent and how they are answered.
- PD related *messages*, which specify the allowable format and contents of the message documents exchanged commonly understood between the collaboration partners.
- *Messaging* that specifies secure communication when exchanging the messages over Internet. Messaging specifies the packaging, encrypting and responding to basic network problems, such as a message lost in the delivery. Messaging makes sure that the design documents delivery is non-repudiated.
- Industrial support and usage that are obviously important for the companies involved

As the starting point for selecting suitable standards frameworks for comparison, we analysed the documentation and specifications of 15 standard frameworks including e.g. BizTalk framework, commerce XML (cXML) and CommerceNet's eCo framework (eCO). Due to limited space, we will concentrate on the five most promising frameworks:

- EbXML. Electronic Business using XML (EbXML) aims at providing an XML-based infrastructure enabling the use of electronic business information in an interoperable and secure manner. To enable this ebXML have issued number of specifications. UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) and OASIS (Organisation for the Advancement of Structured Information Standards) sponsor ebXML activities. [5]
- RosettaNet. RosettaNet is an industry-driven consortium working to create, implement and promote open e-business process standards. RosettaNet is a subsidiary of the Uniform Code Council (UCC), the organisation behind EAN codes. [6]
- OAGIS. The Open Applications Group (OAG) is an industry-driven non-profit consortium with the purpose to integrate business applications inside an enterprise or among enterprises. They specify OAG Integration Specifications (OAGIS). [7]
- PDX. The Product Definition eXchange standard group consists of corporations and standards organisations in electronics manufacturing. PDX focuses on supporting communicating product information. [8]
- STEP. STEP is a set of ISO standards, which aim at helping the exchange of engineering product data. An international community of academic institutions and companies have being developing STEP standards. [9]

Table 1. Comparison of standard frameworks

	EbXML	RosettaNet	OAGIS 8.0	PDX	STEP
Processes	(ebBPSS)	PIP	(Example scenarios)	-	-
PDM related messages	(Core Components)	PIP DTD	BOD XML Schemas	PDX DTD	STEP AP
Messaging	EbMS	RNIF	-	-	-
Industrial usage	No/Yes	Yes	Yes	No/Yes	Yes/No

The ebXML Business Process Specification Schema (ebBPSS) provides guidelines for defining inter-company processes, but no process definitions to take into use. RosettaNet

Partner Interface Processes (PIPs) specify about 110 processes in detail (as of May 2003). OAGIS 8.0 provides about 60 example scenarios for message exchange, but the descriptions are on a very high level showing only examples, not definitions, of messages to use. PDX and STEP do not define processes.

EbXML specifies Core Components as general building blocks for messages, but so far no standard *Document Type Definitions* (DTD) or *XML schemas* that describe the allowable syntax and structure of messages. Each RosettaNet PIP provides DTDs for messages. OAGIS specifies *XML Schemas* for each *Business Object Document* (BOD). PDX standard specifies a DTD for messages. STEP *Application Protocols* (APs) specify industry specific messages.

The ebXML Messaging Service (ebMS) and RosettaNet Implementation Framework (RNIF) both define how the messages are packaged, signed and sent securely between collaboration partners over the Internet. They also provide the actions needed when exceptions happen during the message delivery. The other standards do not specify messaging.

EbXML has reported support for some of the specifications by application vendors and other standard bodies, but it still lacks reported industrial implementations [5]. RosettaNet shows industrial support, having many reported implementations and supporting products [6]. OAGIS standard has also show industry support [7]. The PDX standard is supported by products such as Agile software and OpenVision software, but the reports of industrial usage are missing. With STEP, there are many signs of support for certain parts of the standard. Still, the development of some application protocols of STEP has stalled. The case companies do not use any of these frameworks for PD.

3.2 Discussion on comparison

We chose the RosettaNet standard framework as a basis for our support. It is the most promising standard framework, as it covers all the needed features. Therefore, we have chosen it as the framework on which we base our system. Although especially the product data related standards PDX and STEP contain significantly more detailed specifications for representing product data, they lack definitions for processes and messaging.

The comparison table could have contained also many other standards that focus on standardising exchanged messages or messaging solution. For instance, the ENX consortium (http://www.enxo.com/) provides just the secure messaging by network operators. This would make the use of messaging according to RNIF or ebMS unnecessary. The Common Object Request Broker Architecture (CORBA) standard provides some services similar to the ones needed in messaging, but does not contain all the specifications needed in secure inter-company messaging over Internet.

3.3 RosettaNet standard

The most important components standardised in RosettaNet framework are PIPs, dictionaries and RNIF. The RosettaNet PIP defines a common inter-company *public process* with which a company specific internal, *private process* interacts. The PIP does not specify anything about how the content is extracted in the private process from the companies' own applications to form the business messages. Each PIP contains a *specification document*, DTDs and *message guidelines* (MG). A specification document defines the process, the roles of the participants and necessary conditions to initiate messaging. Message guidelines, based on RosettaNet *business dictionary (RNBD)*, introduce additional constraints and guidelines for allowable content in messages to the ones specified by the DTD. RNBD define and explain the common terms to be used in the PIP messages.

RosettaNet PIPs are divided to 7 clusters noted by numbers and the clusters further divided to segments noted by letters. The cluster 2 of RosettaNet PIPs deals with Product information. It is divided to four segments e.g. 2C contains PIPs for "Product Design Information". Segment 2D "Collaborative Design" PIPs have not been released and cannot thus yet be utilised.

The RosettaNet Implementation Framework (RNIF) specifies the messaging of RosettaNet PIPs. RNIF defines the RosettaNet *business message*. This business message contains the message specified by PIP DTD and MG and the necessary headers and security features needed to process the messages. RNIF also defines how *attachments* are encoded in the business messages. These attachments can be any document files, such as AutoCAD or PDF. Many application server vendors like BEA, webMethods, Tibco and Microsoft support RNIF.

4 A system architecture for supporting design document management in networked PD

In this section, we first present an architecture and implementation of a system to support design data sharing in networked PD, and then discuss briefly its implementation status.

4.1 System architecture

The architecture of our system for exchanging design data documents has the following three components (Figure 1):

- *An application server* that is capable of messaging according to RNIF. This manages the actual interaction with partners over the Internet.
- A system that manages the content exchanged in the business messages. This is typically a PDM system or some other system for managing design documents.
- *An adapter* to manage communication between the two above-mentioned systems.

The PDM system manages the design documents within a company and provides an interface for accessing the documents and their *metadata.* Metadata is descriptive attribute information on a document. The adapter communicates with the PDM system and the application server. It contains and evaluates the rules for information exchange needed to manage and initiate the PIPs that are supported. The application server is able to create and manage communication of RosettaNet business messages and it communicates with the adapter. The system supports so far two PIPs - "2A1 Product Information Notification" and "2C5 Notify of Engineering Change Order".

4.2 The system managing the design documents

In our implementation, the component for design data management is a PDM system called engineering document management system (EDMS) developed in our research group and in production use at KONE Corporation [10]. It manages the design documents and the metadata such as document life-cycle status (e.g. draft, ready, approved, obsolete), author and document type (e.g. drawing, test plan). It provides the functions needed to query and store documents and/or their metadata through a Java API. In the EDMS system, there is a way to define a triggering mechanism to react to changes made to particular documents. E.g., when a designer makes a change to a document and stores it in the EDMS system, we have defined triggers to send notification of this with the document identification (ID) to the adapter.

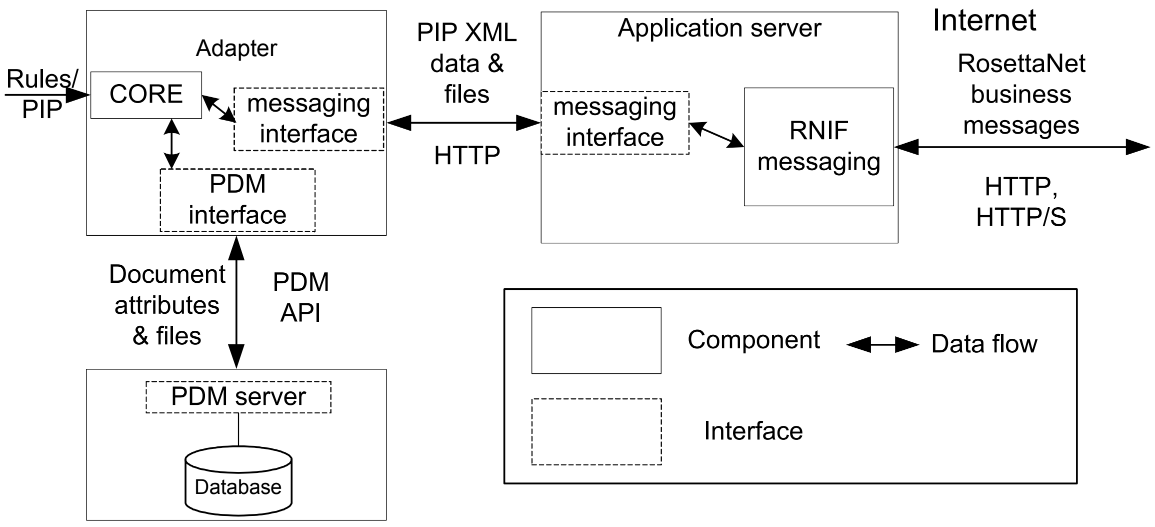

Figure 1 System architecture

4.3 Adapter

Our adapter runs on a Linux server and is implemented using Java programming language. The adapter manages the internal processes necessary to form, send and receive PIP messages. The adapter has interfaces to the PDM system and the application server. The adapters interfaces are designed to be independent of the PDM system and the application server used. This is to make the functionality in the adapter *core* independent. In the core the rules for managing the PIP processes are defined. For instance, on can define a rule that says that all the documents related to a given project that are in status "ready" are to be sent to a certain partner using a certain RosettaNet PIP e.g. 2A1 Product Information Notification.

The rules are based on event-condition-action type of functionality. The rule conditions use the document metadata such as document identifications, version identifiers, life-cycle statuses, generation times, partners and project information in the evaluation. There is a web-based tool for defining the rules in the adapter. The rules can be set time-based so that e.g. the rule applies only during a specified period.

A notification from PDM initiates rule processing in the adapter. The rules evaluation decides based on the conditions the resulting action. The action is e.g. that the adapter queries for relevant document metadata and the document files from EDMS, and forms an XML document according to the RosettaNet PIP 2A1 DTD and MG. This document is then sent to the application server with the document files or file references. The adapter manages also the interaction needed when the application server sends a PIP message to the adapter coming from a partner. The adapter gets the incoming information and is able to store it to the EDMS.

4.4 Application server

The application server has an interface to the adapter and handles RNIF messaging. We have built our solution on BizTalk server and Microsoft BizTalk Accelerator for RosettaNet. It receives the PIP message instances from the adapter with the possible document files, and forms and secures the RosettaNet business message from this information. It sends and receives business messages of different PIPs and takes care of possible exceptions in them. When the application server receives a business message sent by a partner, it interprets it and sends the PIP DTD part of the message and possible attachment files to the adapter.

4.5 Implementation status

We are able to send and receive RosettaNet business messages and query and store the information in the PDM system used. A web-based tool for defining the rules is ready and tested. The testing of the adapter with different PDM systems has been so far limited to the EDMS system. The use of EDMS system has not required major modifications to EDMS.

The necessary functions to support the PIPs used in the application server are ready and have been tested with an outside organisation. The testing involved both sending and receiving RosettaNet business messages with and without security features defined in RNIF.

5 Discussion and previous work

In this section, we discuss our experiences on applying the RosettaNet framework and implementing the system architecture, and our approach in relation to previous work.

5.1 Using RosettaNet framework

The available tools and available information of RosettaNet implementations helped in designing and setting up the system. The RosettaNet standards are understandable and quite well documented. Especially the RNIF messaging seems well thought out. However, the standard processes for cluster 2 did not fully meet the requirements, as the data models of those PIPs do not provide sufficient support for most of the document metadata attributes we wish to exchange in the messages and thus information is lost in the delivery. For instance, in current PIPs there are no standard document life-cycle statuses "e.g. draft, ready, approved". In addition, it is not possible to specify that document A has subdocuments A1 and A2. In the future, this might be rectified by the forthcoming standards for collaborative design (PIP 2D).

RosettaNet does not even try to standardise the content and structure of e.g. a CAD file exchanged as attachment. This obviously means that the tools used for viewing and modifying these documents in the company network must be compatible. Our RosettaNet solution helps to transport the files in a controlled and secure way as in the requirements. The case network PD companies need to manage also e.g. purchasing processes as the suppliers after the PD phase also supply the plastic parts to the customer. Purchasing seems to be well supported by RosettaNet. If the companies use RosettaNet also in this collaboration, they can benefit from using the same messaging infrastructure and standard terminology in the exchanged messages.

5.2 System architecture

In the system architecture, RosettaNet dictated only the use of RNIF and the contents of the business messages exchanged in the PIPs. The application server part was implemented in less than a man-month, while the effort needed to build functionality in the adapter has been the most time demanding activity. In addition, the work needed to define triggers for EDMS took only a couple of weeks. Two computer science under-graduates using the tools and techniques partly unfamiliar to them have implemented most of the system in about 10 man-months.

We have noted one potential performance problem in RosettaNet messaging when the attachment files are very big. Attachment encoding and encryption cause overhead, which can require a substantial amount of computation in the servers at both ends of delivery. The whole process of sending the business message, network delivery, receiving and validating the incoming message and sending acknowledgements of the delivery needs to be completed in a specified time frame, typically 2 hours. Therefore, in some cases it might be more feasible to send only the metadata of the changed document and provide a reference to an address to get the document file. This is managed in the adapter by providing a rule to take care of this.

There are standard architectures that could have been used as a basis of our system, such as CIM Open Systems Architecture (CIMOSA). However, it seemed too complicated for the problem at hand, as it encompasses the design of the whole enterprise architecture instead of building support for inter-company processes. It also lacks the tool support.

5.3 Related work

Cutkosky et al [11] describe an agent-based infrastructure for concurrent engineering in which the communication relies on Internet protocols. These are also used in RNIF as the underlying layers of messaging. The solution is similar to ours in the way the different applications communicate by translating internal concepts of applications to a shared language (grammar, vocabulary, meaning), for which we use messages offered by standard frameworks. Domazet et al [12] present an infrastructure for collaboration based on an event-driven software component framework using CORBA and STEP while Borland and Wallace [13] describe an approach for collaboration between product designers and environmental experts, facilitated by modelling capabilities distributed over the Internet using CORBA. Thus, these infrastructures are partially based on standards that are not used in our case network and would be hard to introduce into. Salminen et al. present a framework for networked PD [14] consisting of a strategic process layer, a models layer, a tools (applications) layer and a physical IT-layer. They did not define any concrete system architecture, but our architecture seems to fit their framework. In our prototype, the strategic process layer is formed by the PIPs and RNIF corresponds to the physical IT-layer. The model layer information is in the messages. As in our solution, the tools used for handling the documents must be compatible.

6 Conclusions and future work

We analysed several standard frameworks for data interchange with respect to the practical needs in a product development network. Of these, the RosettaNet framework seems the most suitable. We further defined a system architecture for supporting the required design data management in the case network. Our preliminary experiences with the design and its implementation show that using RosettaNet for this purpose is a feasible approach. The RosettaNet messaging and processes match the requirements. However, some of the relevant Partner Interface Processes (PIPs) in RosettaNet are still under development. Our experiences indicate that there is a need for RosettaNet to better support design document management by enabling representing more information in PIP messages so that information is not lost in the document delivery. This is e.g. the design document structure and certain document metadata.

Future work includes finishing and further testing our implementation and integrating it with different design document management systems. Furthermore, the feasibility and utility of the system should be tested in practice in the case network and other networks. As the existing RosettaNet specifications did not meet all the requirements for messages, the forthcoming relevant specifications should be tried, and if they are still not suitable, then new specifications tailored for the case companies should be defined.

Acknowledgements

We gratefully acknowledge the financial support from the National Technology Agency of Finland (Tekes). We thank Hannu Laesvuori and Katrine Jokinen for implementing significant parts of the system and Hannu Peltonen for his advice on implementation.

References

[1] Paasivaara, M. and Lassenius, C., "Communication in New Product Development Networks - A Case Study", in Proceedings of 8th International Product Development Management Conference, Enschede, the Netherlands, June 11-12, 2001.

[2] Clark, K. and Fujimoto, T., "Product Development Performance: Strategy, Organisation, and Management in the World Auto Industry", Harvard Business School Press, 1991.

[3] Borgman, J. and Sulonen, R., "A Case Study of the Impacts of Preliminary Design Data Exchange on Networked Product Development Project Controllability", in Proceedings of International Conference on Engineering Design, Stockholm, August 19-21, 2003.

[4] Shim S.S.X., Pendyala, V.S., Sundaram, M. and Gao, J.Z., "Business-to-Business E-Commerce Frameworks", IEEE Computer 33(10): 40-47, 2000.

[5] ebXML, "ebXML deliverables", http://www.ebxml.org/specs/.

[6] RosettaNet, "RosettaNet standards", http://www.rosettanet.org/standards.

[7] Open Applications Group, "Open Applications Group Integration Specification, Release 8.0". http://www.openapplications.org.

[8] PDX standard group, "IPC-2571 Generic Requirements for Electronics Manufacturing Supply Chain Communication", http://www.pdxstandard.org/.

[9] 10303, "Product Data Representation and Exchange – Part 1: Overview and Fundamental Principles", ISO, 1994.

[10] Peltonen, H., "Concepts and an Implementation for Product Data Management." Ph.D. thesis, in Acta Polytechnica Scandinavica, No. 105, Espoo, 2000.

[11] Cutkosky, M.R., Engelmore, R.S., Fikes, R.E., Genereseth, M.R., Gruber, M.R., Mark, W.S., Tenenbaum, J.M. and Weber, J.C., "PACT: An Experiment in Integrating Concurrent Engineering Systems", IEEE Computer 26(1), 28-38, 1993.

[12] Domazet, D.R., Yan, M.C, Calvin, C.F.Y, Kong, H.P.H. and Goh, A., "An Infrastructure for Inter-Organizational Collaborative Product Development", in Proceedings of the 33rd Hawaii International Conference on System Sciences, January 4-7, 2000.

[13] Borland, N., Wallace, D., "Environmentally Conscious Product Design - A Collaborative Internet-based Modeling Approach." Journal of Industrial Ecology, Volume 3, Number 2 & 3, 2000.

[14] Salminen, V., Yassine, A. and Riitahuhta, A., "A Strategic Framework for Collaborative Product Development", in Proceedings of 4th International Conference on Engineering Design and Automation, Orlando, Florida, July 30-August 2, 2000.

Corresponding author:
Paavo Kotinurmi, Software Business and Engineering Institute, Helsinki University of Technology,
P.O. Box 9600, 02015 HUT, Finland, Tel: Int +358 9 451 6224, Fax: Int +358 9 451 4958
E-mail: Paavo.Kotinurmi@hut.fi, URL: http://www.soberit.hut.fi/~pkotinur/