INTEGRATING DESIGN DOCUMENT MANAGEMENT SYSTEMS USING THE ROSETTANET E-BUSINESS FRAMEWORK

Paavo Kotinurmi, Hannu Laesvuori, Katrine Jokinen, Timo Soininen
Software Business and Engineering Institute, Helsinki University of Technology, POB 9210, FIN-02015 HUT, FINLAND

Keywords: B2B Applications, Document Management Systems Integration, RosettaNet

Abstract: E-business frameworks providing standards and specifications that aim at enabling integrating the enterprise applications of business partners with relative ease have been recently proposed. This paper reports on experiences gained from developing a prototype system for integrating design document management systems based on the RosettaNet e-business framework. We present the requirements for the prototype, extracted from a case study of a product development network, and the design and implementation of the system. We then discuss the experiences gained in the light of the feasibility of applying such frameworks and their supporting technologies as foundations for e-business. The RosettaNet e-business framework was found to be relatively easy to implement and use. However, the RosettaNet specifications for product development processes and the related business document definitions, e.g. for design document delivery, are not sufficient in all respects. As a consequence, two implementations of the same RosettaNet standard process may be incompatible, and thus the aim of providing easy integration may fail. Furthermore, the effort required to build the system and fill in the missing parts in RosettaNet to integrate the product design activities may risk the goal of easy integration.

1 INTRODUCTION

It is expected that the volume and scope of e-business grows fast and the needed systems integration among the e-business partners extended to new areas (Lee and Whang, 2001). One way of facilitating inter-company application integration, offering potential cost and extensibility benefits, is to develop and apply e-business frameworks that provide standards and specifications enabling businesses to communicate efficiently over the Internet (Shim et al., 2000). The aim of such frameworks is to facilitate integration with less implementation effort for each e-business partner.

One potential area where such frameworks could be beneficial is product development (PD) projects conducted in company networks in a concurrent and collaborative manner (Borgman and Sulonen, 2003). Supporting design information sharing is an important issue for successful PD (Clark, 1991). This information is usually stored in different kinds of documents, such as CAD models within each company's Document Management System (DMS), which is typically a part of a Product Data Management (PDM) system. Such systems facilitate the PD process in one company by providing up-to-date information to all the product designers who need it (Liu and Xu, 2001). The same type of support needs to be extended to cover the whole PD network (Kotinurmi et al., 2003). Thus, there is a need to integrate DMS systems. With DMS integration using an e-business framework, information delivery could become faster, less error-prone and more transparent than in manual processes of using e.g. e-mail. End-users in companies could use their own systems, which they know, and still be able to collaborate with other companies.

As the research results on implementations applying e-business frameworks, particularly to PD, are few (Sayal et al., 2002) (Nurmilaakso, 2003), this paper contributes by presenting an actual case of building a prototype implementation and critically discussing the lessons learned.

This paper is structured as follows: Section 2 defines the requirements for DMS integration extracted from a case study and introduces the RosettaNet e-business framework. In Section 3, we describe an architecture and design of a prototype system for DMS integration supporting RosettaNet. Section 4 discusses the experiences and lessons

Kotinurmi P., Laesvuori H., Jokinen K. and Soininen T. (2004).
INTEGRATING DESIGN DOCUMENT MANAGEMENT SYSTEMS USING THE ROSETTANET E-BUSINESS FRAMEWORK.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 502-509
DOI: 10.5220/0002622405020509

learned from building the prototype system and presents related work. Finally, in Section 5 we present conclusions and topics for future work.

2 REQUIREMENTS AND MOTIVATION

This section presents the requirements and motivation for DMS integration and introduces the RosettaNet e-business framework.

2.1 Requirements

The requirements for a system supporting design document management in networked PD are based on a case study of a PD network, which consists of a customer and its suppliers (Borgman & Sulonen, 2003). The customer designs, manufactures, and sells consumer electronics products and the suppliers supply plastic parts to these products.

According to the case study, document management should be a planned, communicated and documented process between companies in networked PD. The design documents drive the supplier's process in a very straightforward way (Figure 1): When the supplier receives a new version of e.g. a CAD-model from the customer, it immediately triggers some design and manufacturing work for the supplier. Equally, when the supplier sends a document (e.g. a measurement log sheet) to the customer, it triggers e.g. a new component approval process. Documents synchronise the 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 automatically. These repetitive information delivery processes can be automated with the help of e-business frameworks.

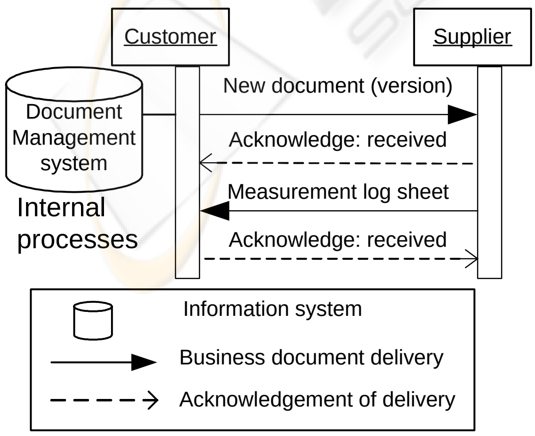

Figure 1: Information delivery in networked PD

It is important to ensure that changed documents are distributed inside the companies to the engineers whose work depends on them. Therefore, the delivered document files must be accompanied with metadata, i.e. information about the document, such as its author, version, and relationship to product components and projects. This should be carried in the *business documents* exchanged in the process steps.

Design documents are typically highly confidential. There must be a means to distribute the documents securely to the intended recipients only and to be certain that the delivery is successful. The documents should be encrypted if the distribution takes place over non-secure communication channels, such as the Internet.

A proper solution supporting design document exchange should be extendable to new collaboration partners and processes without requiring major modifications. The solution should be extendable to many partner companies using different DMSs, and therefore based on standards. There are numerous seemingly applicable e-business frameworks such as ebXML, RosettaNet and Standard for the Exchange of Product Model Data (STEP) to support definitions for processes, business documents, and secure and reliable messaging over the Internet. Based on our analyses and comparison of five e-business frameworks, we chose RosettaNet as the basis for our system (Kotinurmi et al., 2003).

2.2 RosettaNet

RosettaNet is an industry-driven consortium aiming at creating, implementing, and promoting open e-business process standards. The most important components standardised in RosettaNet are *Partner Interface Processes* (PIPs), *dictionaries* and *RosettaNet Implementation Framework* (RNIF).

PIPs define common inter-company public processes such as "PIP 2A10 Distribute Design Engineering Information" and the associated business documents. *Trading partners'* internal processes interact with PIPs to initiate or receive business documents. RosettaNet PIPs are divided to eight clusters noted by numbers, and the clusters are further divided to segments noted by letters. Cluster 2 deals with Product information. It is divided to four segments, e.g. 2C "Product Design Information". Segment 2D "Collaborative Design &

Engineering" PIPs have not been released and cannot be utilised yet.

Each PIP contains a *specification document*, *Document Type Definitions* (DTD) and *Message Guidelines* (MG). A specification document defines the process with Unified Modeling Language (UML) activity diagrams and textual descriptions, the roles of the partners, and necessary conditions to initiate messaging. Each PIP defines one or more business documents. The DTD and MG define the PIP *service content* of one business document. The DTD defines the valid Extensible Markup Language (XML) document structure of a PIP service content. The MG introduces additional constraints and guidelines, such as what a modification date means and how the date value should be represented. Figure 2 represents a fragment of a service content XML document.

RosettaNet business dictionary (RNBD) defines common terms used in all the PIPs. In addition to dictionaries, RosettaNet uses certain identifiers, such as Data Universal Numbering System (DUNS) codes to identify companies uniquely.

```
<ProductInformationObject>
  <GlobalActionCode>Add
  </GlobalActionCode>
  <Version>1.2</Version>
  <ObjectName>Cover345</ObjectName>
  <Supplier>DUNS number</Supplier>
  <Description GlobalLanguageCode="EN">
   Note the red areas on the model are
   Work in Progress</Description>
  <modificationDate>
    <DateStamp>20031015Z</DateStamp>
  </modificationDate>
</ProductInformationObject>
```

Figure 2: PIP 2A10 service content fragment

RosettaNet Implementation Framework (RNIF) specifies messaging. It defines the *RosettaNet business message* that contains the *service content* 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 RosettaNet business messages. These attachments can be of arbitrary file format, such as AutoCAD. RNIF contains exception-handling mechanisms and makes sure that the delivery is non-repudiated, so neither the sender nor the receiver can later deny having sent/received the RosettaNet business message. Many vendors, such as BEA and Microsoft, support RNIF in their products.

To set up RosettaNet messaging using a certain PIP, the companies involved set up a *Trading Partner Agreement* (TPA) to specify both the business and technical aspects of the collaboration for each PIP. Example business aspects are conditions for trading, such as how certain elements are used, confidentiality, and when and how the PIP must be answered. Technical aspects include e.g. security features, such as the use of certificates for authentication and the addresses where the RosettaNet business messages are delivered.

3 DESIGN & IMPLEMENTATION

In this section, we will first present the architecture, functionality, and internal data model of a prototype system to support DMS integration. Then we will present the implementation and test setup briefly.

3.1 Architecture

Our prototype system consists of five architectural components (figure 3).

PDM system is a repository for design documents and their metadata. It notifies the PDM adapter of changes in its contents.

PDM adapter is connected to the rule engine and has interfaces to the PDM system and the RN adapter. It notifies the rule engine of events in the PDM system, retrieves design documents and metadata from the PDM system, and sends them to the RN adapter.

Rule engine has a user interface for defining document delivery rules for the design documents, e.g. a rule saying that updates to a specified design document must be sent to a specified trading partner. The rule engine also evaluates these rules each time there are changes in the PDM system.

RN adapter is connected to the RosettaNet messaging server and has an interface to the PDM adapter. Based on the design documents it receives from the PDM adapter, it adds RosettaNet specific delivery information to them.

RosettaNet messaging server constructs RosettaNet business messages from the documents received by the RN adapter. It controls the exchange of business messages with trading partners based on the RNIF specification.

3.2 Functionality

Our prototype system implements the following two use cases, which illustrate its functionality.

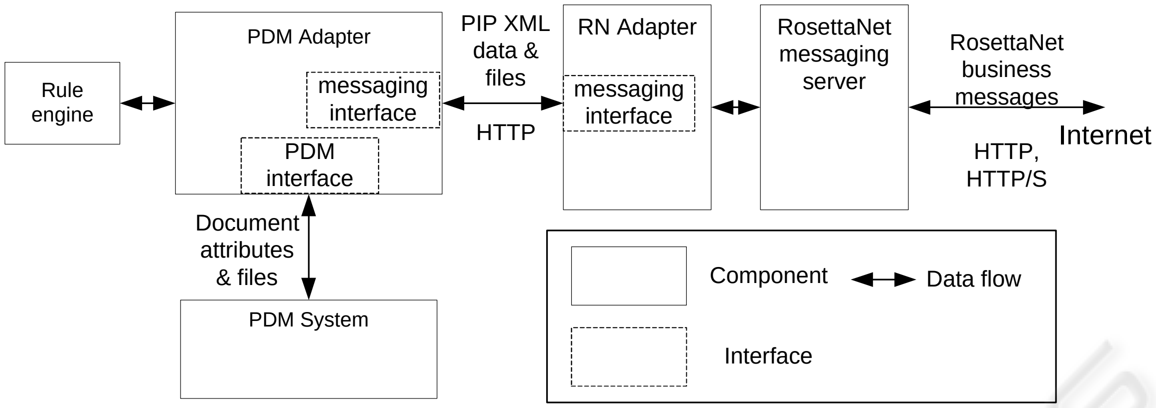

Figure 3: System Architecture

The first use case is automated delivery of a design document update in the PDM system to a trading partner using RosettaNet. This guarantees that the trading partners always have the latest version of the design document in their PDM system. A prerequisite for this is that the prototype system has been configured with information about trading partners, design documents in the PDM system, and document sending rules.

Information about changes in the design documents in the PDM system is notified through the PDM adapter to the rule engine, which evaluates the event against the defined rules. If design document update must be sent to a trading partner, the PDM adapter takes control. It retrieves the design document and its metadata from the PDM system, constructs the PIP service content from the metadata, and sends them and the name of the trading partner to the RN adapter.

The RN adapter instructs the RosettaNet messaging server to initiate a new PIP with the trading partner. The design document is added as an attachment to the business message and the metadata is added to the PIP service content.

The second use case is receiving design document updates from trading partners, which is essentially the reverse of the first use case. Upon receiving a business message that contains the design document the prototype system adds this new document version as such to the PDM system.

3.3 Data model of design documents

The prototype system uses an internal data model for the exchanged design documents. The data model defines how the design documents should be stored in the partner's PDM system. As the current PIPs do not include sufficient design document metadata to accomplish this, we had to define our own internal data model for the integration.

The internal data model describes the structure of the document, its life cycle status, its relations to other documents, projects and products, and other relevant information, such as its creator and creation time. The data model is based on an analysis of three companies' internal data models for design documents in their PDM systems (Jokinen, 2003).

3.4 Implementation

3.4.1 PDM System

The PDM system manages the design documents and their metadata within a company. It notifies the PDM adapter of changes in its contents, e.g. creation of a new document version.

Our PDM system is called EDMS (Engineering Data Management System). It was built in our research group in a project with KONE Elevators, where it is in production use. (Peltonen, 2000)

EDMS includes commit procedures that are executed each time an object is modified, e.g. new documents are created. These procedures have been configured to send an event notification to the PDM adapter. The notification includes an identifier of the object concerned. We have defined partner specific access rights to EDMS and a way to define the capabilities to define what PIPs they can receive.

3.4.2 PDM Adapter

The PDM adapter connects three other components: the rule engine, the PDM system, and the RN adapter. It forwards event notifications from the PDM system to the rule engine, retrieves design document files and metadata from the PDM system, and sends them as PIP service content and attachment to the RN adapter.

The PIP service content is formed in the PDM adapter. If the evaluation of the delivery rules results in sending a document to partners, the PDM adapter retrieves the document file from the PDM system and sends it to the RN adapter. It also retrieves the document metadata defined in the internal data model and forms an XML representation of it. The PDM adapter decides which PIP should be used, depending on the event, the type of document, and the receiver. The rules for deciding the PIP have been hardcoded, as there are only four of them in the prototype. The PDM adapter then transforms the metadata with Extensible Stylesheet Language Transformation (XSLT) to the service content of the corresponding PIP.

The same procedure is conducted in reverse order when receiving a design document from a trading partner. The service content is transformed to the data model format using the XSLT file corresponding to the PIP used. The information is then stored in the PDM system.

The PDM adapter is implemented using Java programming language and it runs under Apache Tomcat application server on Linux platform.

3.4.3 Rule engine

The rule engine has a user interface for defining document delivery rules, e.g. a rule saying that updates to specified design documents must be sent to a specified trading partner. The rule engine evaluates the rules each time there are changes in the PDM system, such as new documents created.

The document delivery rules are defined with a web user interface. They have inputs, conditions and consequences. The rules take PDM objects (e.g. documents, users, projects) as inputs. The conditions can be related to, e.g., document version state or document type. The consequence defines the action to be taken if the rule applies: normally the consequence is to send the document to one or more trading partners. The rule engine evaluates the rules after receiving event notifications from the PDM adapter.

The user interface also includes functions for sending documents manually and simulating events from the PDM system to see which documents would be sent to trading partners, and according to which rules. The rules can be project specific and have time constraints. There is also a delivery manager, which shows the delivery information of all the documents sent.

The rule engine is implemented with the same technologies as the PDM adapter. The user interface is implemented using Java servlets and HTML templates. The evaluation is based on an algorithmic engine, Drools, for implementing custom object-related business rules.

3.4.4 RN Adapter

The RN adapter facilitates communication between the PDM adapter and the RosettaNet messaging server. The design documents and their metadata that the RN adapter receives from and sends to the PDM adapter are information used by PDM systems. The RosettaNet messaging server, however, sees everything as RosettaNet business messages belonging to a PIP. The RN adapter has the logic to do transformations, such as replacing organization names used as identifiers in the PDM system with DUNS codes required by RosettaNet.

We have implemented most of the RN adapter using Microsoft .NET architecture and C# programming language. The interface to the PDM adapter is implemented using the Web Service (WS) technologies Web Services Description Language (WSDL) and Simple Object Access Protocol (SOAP). The integration with the RosettaNet messaging server is implemented using native methods of the RosettaNet messaging server.

3.4.5 RosettaNet messaging server

The RosettaNet messaging server provides the means to extend the PDM integration to trading partners. It implements the functionality specified in RNIF, e.g. it creates RosettaNet business messages and provides their secure exchange over the Internet with trading partners. The RosettaNet messaging server cannot initiate new PIPs or respond to PIPs on its own. The RN adapter does this.

Our RosettaNet messaging server is Microsoft BizTalk server 2002 extended with Microsoft Accelerator for RosettaNet 2.0. For our prototype system to work, it needs to be configured with necessary partner information such as the used PIP, DUNS numbers of the trading partners, IP addresses etc. The configuration requires manual insertion of data to a database. Typically, the configuration work for a new trading partner takes a few minutes.

3.4.6 Test set-up

The end-to-end solution has been tested in an environment that simulates two partner companies. We had two instances of EDMS servers with different database schemas to represent two different companies. A design document update in one EDMS server led to the construction of a complete RosettaNet business messages according to the four PIPs used. The business message was sent to the other EDMS server, where the design document was saved correctly.

4 EVALUATION & DISCUSSION

Integrating design document management systems using the RosettaNet e-business framework seems to be feasible. However, the integration took a significant effort although tools specifically designed for this purpose were used. Furthermore, RosettaNet did not support all the aspects of the integration needs adequately, which may lead to problems in wide-scale e-business use of RosettaNet. In this section, we motivate these findings through discussing the system architecture and implementation and by evaluating the support RosettaNet offered. We also discuss our solution with respect to the relatively scarce earlier research reporting on the practical aspects of integration in PD and the usage of e-business frameworks.

4.1 Architecture and design

The architecture of the prototype system allowed successful implementation of the requirements posed by the case. The goal of our prototype system was a minimalist yet functional implementation. Our design choices were to leverage code reuse by using commercially available applications when possible and to aim for modularity to accommodate possible future extensions. The choices for Microsoft RosettaNet Accelerator and EDMS over competing products were made primarily because of their availability to us, but also because of the relative ease of modifying them.

Based on the requirements and the chosen applications, we identified three separate units of logic in our prototype that we needed to implement ourselves: the PDM adapter, the RN adapter and the rule engine. The integration of the PDM system with the RosettaNet messaging server needs functionality that is specific to both of them. As the PDM adapter and the RN adapter are implemented in two separate units, either the PDM system or the RN messaging server can be replaced with some other product while still retaining the implementation for the other adapter. We also needed to implement the rule engine, as we were unable to find any existing product suitable for the task. In addition, we had to define the internal data model to provide flexibility to adopting new PIPs, while the available PIPs as such were not adequate.

Overall, we consider our architecture modular. Changing the PDM system or the RosettaNet messaging server would only require modifying the associated adapter, and it is also possible to replace the rule engine. The use of interoperable communication protocols and an internal data model throughout the prototype system should increase the number of potential candidates for a new PDM system or RosettaNet messaging server. The internal data model enables easy transition to new PIPs, as it can be transformed to a specific PIP with XSLT. Otherwise, the addition of a new PIP or modifying an existing one would require a lot more effort.

4.2 Implementation

The resources used for implementation were significant. The prototype system allows design documents to be transferred to the partner's PDM system without human intervention. However, transferring large files may introduce a problem.

The prototype implementation took a lot of resources although existing tools were used. Two student groups used 700 hours implementing the PDM adapter and the rule engine. The EDMS configuration took 100 hours. The RN adapter was implemented in 80 hours, and the RosettaNet messaging server configuration took 40 hours. Integration of the different components, testing, and other additional work took about 80 hours, so altogether the implementation of the prototype system was done in about 1000 hours. This only includes the actual implementation time, excluding e.g. getting to know the systems used. An industrial implementation would naturally require more time, as our system is only a prototype. As this implementation is for only design document delivery, it is a considerable effort.

In order for the system to be flexible, adding new PIPs or trading partners should be easy. To add support for a new PIP in the prototype system takes less than two days. Building a new XSLT takes about a day. The modification needed for the system components takes altogether less than one day. To add a new partner or PIP to the prototype system requires minor modifications to the PDM system, the PDM adapter, the RN adapter and the RosettaNet messaging server.

The test set-up was only a laboratory simulation using two different PDM database schemas and system instances of the same PDM systems representing two companies. It showed that the system basically works, but further testing in more realistic settings is obviously needed.

One issue that the tests revealed was that transfer of large files as attachments is a potential problem, because RNIF recommends BASE-64 encoding. This increases their size by one third, which is significant, as design documents can be larger than 100 MB. PIPs allow typically only two hours to acknowledge the RosettaNet business messages, which is challenging, as the network delivery and server encodings may take a lot of time.

4.3 The use of RosettaNet

There were several problems with the use of RosettaNet. The existing PIPs are not defined with PD in mind. As a result, the definitions available in current PIPs service contents did not provide sufficient support for most of the document metadata we wished to exchange in business documents. To carry the document metadata identified in the internal data model, we had to misuse certain PIPs in order not to lose information. Misusing the standard obviously affects the interoperability of our prototype system. The RNIF seems well thought out although those big attachments can cause problems.

So far, we have used PIP2C5 "Notify of Engineering Change Order" with change request documents, initially PIP2A1 "Product Information Notification", and later the newer versions of PIP 2A1 "Distribute Product Catalog Information" for new document delivery. Later the 2A1 PIPs were replaced by 2A10 "Distribute Design Engineering Information", which was released after we had implemented the solution using PIP 2A1. The first PIPs enabled meaningful carrying of only roughly 30% of the internal data model information. PIP 2A10 increased this to 85%, but for some of the attributes of our internal data model this meant only a close but not precise match to the term. For example with the PIP service content fragment in figure 2, we did not follow strictly the MGs for the contents of "objectName" and "Supplier". We used them to carry the name of the exchanged document and the document creator information. In the remaining 15%, there were few trivial attributes for which we did not find a close match in the PIP MG.

In addition, some of the information in the PIP service contents needs to be better defined to avoid misinterpretations. An example of needed extensions to RNBD and from there to MGs is *life cycle statuses* for documents. As companies define the life cycle statuses, e.g. "pending, ready, approved, obsolete" in different ways, RosettaNet should provide clear definitions for these kinds of enumerated lists to avoid misinterpretations, e.g. whether a document should be ready before it can be approved or vice versa. In addition, it is currently not possible to specify that document A has subdocuments A1 and A2. In the future, these issues might be rectified by the forthcoming standards for collaborative design and engineering (PIP 2D).

Technically the use of both the DTDs and MGs for business document validation is problematic. If the MGs are ignored and just DTDs used, the trading partners need to agree each time on how to use certain elements in the TPA. In future, at least some of these issues are rectified as RosettaNet plans to introduce W3C XML Schema for business document validation, as it can carry the information currently defined by DTDs and MGs.

We have noted that the more recent PIP specifications tend to be a lot better than the older ones, such as PIPs 2C5 and the 2A1 used in the beginning. The resource usage for implementation could have been also a lot less with implementation guidelines, such as the ones available in RosettaNet for collaborative forecasting.

4.4 Related work

There are few experience papers on e-business integration implementations (Nurmilaakso, 2003). We do not know of any other research on applying e-business frameworks for PD collaboration.

Liu and Xu (2001) state similar requirements for web-based PDM systems supporting collaboration. They propose web browser PDM interface as a solution, instead of the unique interfaces that were typical at the time. The modern PDM systems have web interfaces, but they still differ in usage logic.

Domazet et al. (2000) present an infrastructure for collaboration based on an event-driven software component framework using the Common Object Request Broker Architecture (CORBA) and STEP. The modelling capabilities distributed over the Internet using CORBA facilitate collaboration between product designers, and STEP provides the common terminology. This infrastructure is partially based on standards that are not used in our case network and would be hard to introduce into it.

Sundaram and Shim (2001) present an infrastructure for B2B exchanges with RosettaNet. They have a three-tier client-server prototype that allows customers to send RosettaNet PIPs using a browser. Their prototype constructs RosettaNet PIP service contents.

Sayal et al. (2002) present a tool, HP Process Manager, that supports RosettaNet PIPs and allows generating complete processes from PIPs by taking also internal integration needs into account. The functionality of this tool is similar to what our RN adapter and RosettaNet messaging server does by providing support for RNIF and the common elements used in all RosettaNet PIPs.

Both experience articles with RosettaNet contrast with our solution, as they do not present integration to any enterprise information systems (IS). To integrate the IS with the RosettaNet PIPs was the most time demanding part according to our experiences as there were the conflicts between the terms used in IS and the ones specified in the PIP service contents. Similar difficulties may well exist with also with other e-business processes.

5 CONCLUSIONS AND FUTURE WORK

We have presented a design and implementation of a prototype system for supporting design data management in product development (PD) networks. The prototype system integrates design document management systems (DMS) using the RosettaNet e-business framework. It can automatically deliver and receive e.g. the latest version of design documents.

Our experiences with the implementation and use indicate that using RosettaNet e-business framework for this purpose seems basically feasible. However, there are also major potential problems regarding the effort to implement the integration and the level of support offered by RosettaNet. The resources used for implementation were significant as this relatively simple implementation took about 1000 hours, including only the actual implementation time. The secure delivery of documents according to RosettaNet Implementation Framework was easy to implement, but attachments handling is ineffective.

The RosettaNet specifications for the common processes and the related business document definitions should support design document information exchange better, e.g. by ensuring common understanding of important terms and providing better guidelines for the usage. As business document definitions currently enable misusing the standard, two implementations of the same RosettaNet process are not necessarily compatible. Hence, the aim of industry-wide cost-effective B2B integration through the use of the e-business frameworks may be compromised, particularly in the case of PD. This lack of support by the e-business framework may also affect other business processes besides PD.

Future work includes further testing of our prototype system and integrating it with different DMS and messaging servers. In addition, the utility of the prototype system and the effect of this type of DMS integration on the whole PD process effectiveness should be evaluated. As the existing RosettaNet specifications lacked the support for design document information exchange, the relevant future specifications should be tried. Furthermore, more empirical data on integration efforts is needed to better understand what is the impact of e-business frameworks in practical systems integration efforts.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support from the National Technology Agency of Finland (Tekes). We thank Jukka Borgman and Simo Jokela for their advice on the requirements, Hannu Peltonen for support in the implementation, and the students who participated in the implementation.

REFERENCES

Borgman, J. and Sulonen, R. 2003. A Case Study of the Impacts of Preliminary Design Data Exchange on Networked Product Development Project Controllability, in *Proc. of 14th Int. Conf on Engineering Design.*

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

Domazet, D.R., Miao Chun Yan, Chee Fook Yew, Kong, H.P.H. and Goh, A. 2000. An Infrastructure for Inter-Organizational Collaborative Product Development", in *Proc. of the 33rd Hawaii Int. Conf. on System Sciences*, pp. 2176 –2185.

Jokinen, K. 2003. *A Standard Data Model for Product Development Documents*. Master's Thesis. Helsinki School of Economics.

Kotinurmi, P., Borgman, J. and Soininen T. 2003. Design Document Management in Networked Product Development Using Standard Frameworks, in *Proc. of 14th Int. Conf. on Engineering Design.*

Lee, H and Whang, S. 2001. E-Business and Supply Chain Integration, *Stanford Global Supply Chain Management Forum* SGSCMF- W2-2001.

Liu, D.T., Xu and X. W. P. 2001. A review of web-based product data management systems, in *Computers in Industry*, v.44, n.3, p.251-262.

Nurmilaakso, J.M. 2003. *XML-based Supply Chain Integration: A Review and a Case Study*. Licentiate theses, Helsinki University of Technology.

Peltonen, H. 2000. *Concepts and an Implementation for Product Data Management.* In Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 105. The Finnish Academies of Technology, Espoo.

RosettaNet, 2003, http://www.rosettanet.org/.

Sayal, M., Casati, F., Dayal, U. and Shan, M-C. 2002. Integrating Workflow Management Systems with Business-to-Business Interaction Standards, in *Proc. of 18th Int. Conf. on Data Engineering*, pp. 287 –296.

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

Sundaram, M. and Shim, S. 2001. Infrastructure for B2B Exchanges with RosettaNet, in *Proc. of the 3rd Int. Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pp. 110-119.