THE ENGINE TO SUPPORT BUILDING CPC SOLUTIONS

Jinmi Jung, Hyun Kim, Hyungsun Kim, Joohaeng Lee
Electronics and Telecommnucations Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon, Korea

Keywords: Collaborative Product Commerce, Virtual Enterprise, Product Information, Web Services, XML.

Abstract: In modern business environment, inter-organizational collaborative product development is very important. CPC (Collaborative Product Commerce) is a new category of software, now emerging, to support inter-enterprise collaboration through the product life-cycle. But, so far there are not striking solutions to support collaboration among collaborative engineering groups. In this paper, we describe the engine to support building a CPC solution which is being developed by the Electronics and Telecommunications Research Institute (ETRI) as a part of a CPC project. It makes it possible to collaborate among geographically dispersed enterprises by sharing product information. We look over problems to have to be solved for designing the engine and propose solutions. In addition, we mention the developed CPC solution using the engine.

1 INTRODUCTION

Many of today's enterprises need new business environment due to the increase of keen competition among enterprises day by day and pressure from customers. Many customers are no longer satisfied with mass-produced goods. They are demanding customization and rapid delivery of innovative products (Aberdeen, 1997). The sooner manufacturers can bring a new product to market, the more dominant position they hold in their industries. Consequently many enterprises try to reduce development time and cost as well as improve the quality of product by developing a diverse and complex product through collaboration with partners having prominent capability rather than doing independently. That is, multinational companies that possess expertise in a specific field form a temporary union called 'virtual enterprise' to jointly develop, produce and market products (D.Tony, 2001). The most important requirement for a successful virtual enterprise is the inter-enterprise collaboration. But, it is not easy to realize because collaborating organizations are geographically dispersed and use different computer technology (e.g. CAD/CAM systems, PDM systems) as shown in Figure 1.

Formerly, PDM systems which are very sophisticated IT products with many users in industry are used to support collaborative engineering groups. But, they do not provide sharing of application data at a flexible, user-requested data granularity level and data files use application specific formats and may be modified only by their native applications (Dragan, 2000).

CPC is a newly emerging technology to support inter-enterprise collaboration. Aberdeen Group defines that CPC is a class of software and services that use internet technologies to permit individuals to collaboratively develop, build, and manage products throughout their entire life cycle. And it predicts that manufacturing executives will demand CPC tools to enable their organizations to respond to these commercial market pressures (Aberdeen, 1997). But, currently there are not striking CPC solutions.

In this paper, we propose the engine that support building the Web-centric CPC solution which is based on open standards, such as STEP, Web services, XML, and so on. Section 2 addressed the problems to be solved for collaboration. Section 3 discusses the framework of the proposed engine. Section 4 and section 5 describes system design and implementation, respectively. Section 6 concludes the paper.

2 PROBLEM DEFINITION

The engine aims at supporting to share and manage the diverse information and document among virtual enterprises through the product life cycle for

Jung J., Kim H., Kim H. and Lee J. (2005).
THE ENGINE TO SUPPORT BUILDING CPC SOLUTIONS.
In *Proceedings of the Second International Conference on e-Business and Telecommunication Networks*, pages 117-121
DOI: 10.5220/0001417801170121

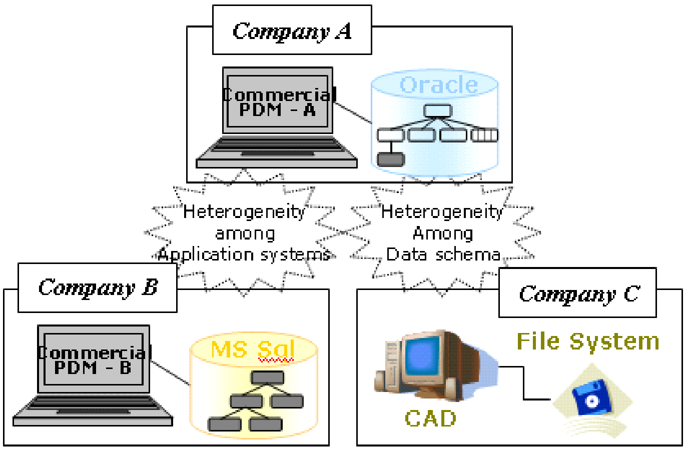


Figure 1: Interoperability problems in product data and application systems.

collaboration. For this, the following issues should be addressed:

- Product data interoperability: product structures and contents in each collaborating enterprise may be different. How can business partners share product data for collaboration when they cannot understand each other's product data?
- System application interoperability: though the problem of the heterogeneity of product data has been solved, business-to-business application integration is an important problem. Business partners manage product data in heterogeneous applications and platforms. In such a situation, how can they seamlessly integrate distributed applications for inter-enterprise collaboration?

Two problems above define how to share product data and integrate applications across different enterprises to support dynamic and agile aspects of virtual enterprises.

Early works investigating sharing product data and integrating its applications dealt with data exchange between CAD systems using STEP which is the standard for the exchange of product model data and CORBA. Hardwick et al. (Hardwick, 1996) proposed an information infrastructure that enhances collaboration in a virtual manufacturing enterprise using STEP and CORBA. There has been similar research (Bliznakov, 1996. Urban, 1996. Shah, 1996) to integrate product data and support collaboration using STEP-based schemas, commercial databases, and CORBA. Wallace et al. (Senin, 1997) introduced object-based modelling and evaluation of design problems in a network-oriented design environment. Kim et al. (Kim, 1999) proposed a framework for process-centric collaborative design in a distributed environment. There have also been studies on product data management (PDM). Liu et al. (Liu, 2001) discussed the integration of PDM methodology with Web architecture to enhance a traditional PDM system. Yeh et al. (Yeh, 2002) proposed an integrated data model and an implementation approach for a PDM system that uses the STEP as a standard for the exchange of product data. Several research efforts have addressed how inter-enterprise applications should be integrated to support business information sharing and collaboration. However, their systems have been mainly built on the assumption that information and systems to be integrated were known at the time of software design because collaboration partners had been previously organized. Therefore, they dealt with information-sharing under tightly-coupled system architecture. But if we take dynamic and agile aspects of virtual enterprises, we need to consider more flexible and open architecture in which the autonomy of each enterprise is more guaranteed. It means information-sharing under loosely-coupled structure should be considered.

3 FRAMEWORK OF THE ENGINE

In order to solve the problems discussed in prior section, we propose product metadata for product data interoperability and CPC adaptor for system application interoperability.

3.1 Product metadata

Because different enterprises manage product information in different ways, semantic and schematic conflicts may be occurred in sharing information. We propose a standard data schema of

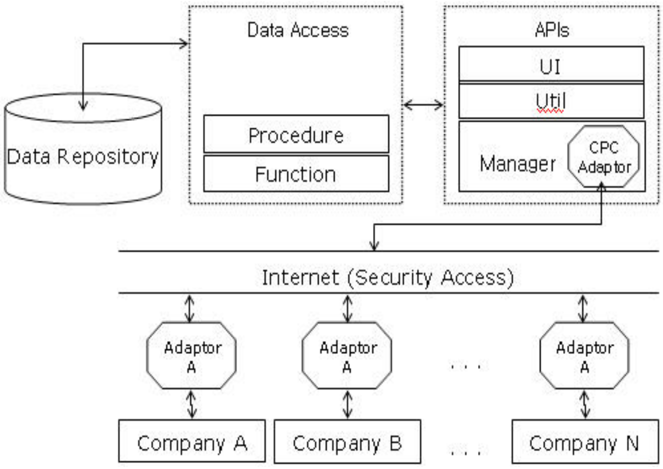


Figure 2: System Architecture.

common use to support the interoperability of product data and call it 'product metadata.' Different data schema managed in different enterprises is mapped into it. Product metadata includes data such as product model, product configuration, product item, product structure, product properties, engineering change, document, project, organization, and alias that are minimal information shared among enterprises to collaborate one another.

For product metadata to serve efficiently as a common data schema, it is desirable that product metadata comply with standards to be supported by commercial systems. So, we made it based on ISO STEP PDM Schema (Jim, 2000) which is a reference information model for the exchange of a central, common subset of the data being managed within a PDM system. In STEP, expression is represented with EXPRESS language. But, it is not appropriate for Web-centric system. So we translate product metadata into XML schema which provides a means for defining the structure, content, and semantics of XML documents.

3.2 CPC adaptor

To get product information from geographically dispersed organizations which are using different system applications (e.g. CAD/CAM/CAE, PDM system, File system) and map it into product metadata, the method how to acquire and distribute product information from/to organizations is needed.

Distributed computing middleware such as CORBA, RMI, or DCOM has been proposed to solve the application interoperability problem, but it is not suitable for enterprise-to-enterprise applications. CORBA, RMI, and COM are all designed for tightly-coupled applications. They are based on an RPC model and assume a great deal of similarity between the calling and called systems. They may depend on the conventions of particular programming languages (such as Java) or platforms (such as Windows). In addition, they require a special network protocol such as IIOP. Thus, firewall problems may occur in the applications of enterprise-to-enterprise communication.

Considering the features of virtual enterprise, more loosely-coupled system integration is required. Web services (Web) is targeted specifically at providing a loosely-coupled architecture designed for exchanging information over the Internet. One of the goals of Web services is to realize a distributed architecture across the Web in a platform-independent manner.

So, we use CPC adaptor which is a set of Web services to solve system application interoperability. CPC adaptor extracts product information from a legacy application such as PDM, CAD, or ERP, translates it into product metadata and puts it to the repository of the engine. There is also an adaptor implemented using Web services in legacy application to communicate with CPC adaptor. These adaptors are managed in a CPC service registry, which is based on a UDDI registry. From the WSDL of the adaptor, the information such as operations, I/O data type, service URL, and so on, may be acquired.

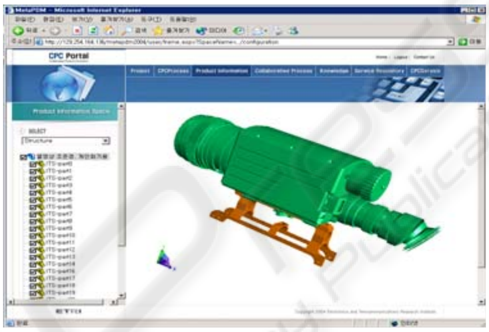
4 SYSTEM ARCHITECTURE

The architecture of the proposed engine in this paper is like Figure 2. It is composed of the data repository

```
Mgr_OrganizationalProject projectMgr = new Mgr_OrganizationalProjectMgr(sessionId command);
UI_ArrayOfProjects allProjects = projectMgr.GetAll(loginId);
int projectNum = allProjects.Getorganizational_projectCount();
Response.Write(" ◈ Projects belong to "+ loginId+ " ◈<br><br>");
Response.Write("Project ID<br>------------------");
for(int i=0;i<projectNum;i++ ){
  Project project = allProjects.Getorganizational_projectAt(i);
  Response.Write("<br>");
  Response.Write(" ·"+ project.Getid().Value);
}
```

Figure 3: Example of the Engine API usage.

(a) The first page



(b) Product Information

Figure 4: CPC portal site

for storing product metadata, the APIs used to program user interfaces, and the procedures for accessing and manipulating data in repository. Figure 2 also shows the phase of sharing information among enterprises through CPC Adaptor.

- Data Repository: we use a relational database as a data repository and its schema is based on product metadata. Product metadata and any other information to be shared across enterprises for collaboration are stored into this repository. The process to map product metadata represented as XML schema into relational database schema has been progressed.
- APIs: APIs provided by the engine are a type of C# class and classified into three packages. Manager package is used to make programs to transfer user input to stored procedure for inserting, modifying and deleting data in the repository. By using the APIs in Manager package, the CPC solution developer can implement the functions such as project management, engineering change order, user management, organization management, data access control. And there are also APIs to make CPC adaptor for sharing product metadata. UI package provide the interface to display the data retrieved from repository on a web page. For example, hierarchical data may be displayed in the shape of tree. Util package provide methods for connecting to the data repository.
- Stored Procedure: it makes SQL queries, accesses to the data repository, executes queries and returns results. These stored procedures are called by the programs made with APIs in Manager package. The programs transfer user input to the procedure as parameters and receive the query results. Result type is not a set of rows but XML. SQL Query which is simple or used repeatedly is made as function for reusability. In case query is complex or many queries have to be executed in turn, it is made as procedure.

5 SYSTEM IMPLEMENTATION AND BUILDING CPC SOLUTION

The APIs are implemented using C#. Currently, there are 21 classes in Manager package, 26 in UI package, 3 classes in Util package. The code of Figure 3 shows the usage of APIs. It retrieves projects which is managed by the current log in user from the repository and displays project IDs on a web page. GetAll() method of

Mgr_OrganizationalProject class in Manager package need one parameter, user's loginId, and returns all projects according to that. Getorganizational_projectCount() method of UI_ArrayofProjects class in UI package returns the counts of returned projects by GetAll() method.

To demonstrate the functionality of the engine, we built the CPC solution with the engine. It is a kind of Web portal site that provides GUIs for user. In Figure 4, (a) is the first page of CPC portal site and (b) is the snapshot of displaying the product structure acquired from other application system through CPC adaptor. Collaborating enterprises can manage and share a lot of information during the entire product life cycle at this portal site. It has been developed using Microsoft's ASP.Net and Microsoft SQL Server is used as a data repository.

Especially, to test for sharing product information across enterprises, an adaptor for SmarTeam (SmarTeam), a commercial PDM system, and an adaptor for SolidWorks (SolidWorks), a commercial CAD system, are developed together. For security communication between adaptors, each adaptor encrypts data and includes digital signature using X509SecurityToken of WSE 2.0.

5 CONCLUSIONS

The engine proposed in this paper supports building CPC solutions. We solved product data interoperability and system application interoperability that is required to collaborate across enterprises with common data schema following ISO STEMP PDM Schema and CPC adaptor which is a set of Web services, respectively. The CPC solution built using the engine can provide project management, engineering change order, user and organization management, data access control, data search, and so on. Specially, it also provides the functions of sharing information that is deficiency of existing PDM systems. We constructed CPC solution, a kind of Web portal site, with the engine ourselves.

REFERENCES

Aberdeen group., 1997. Collaborative Product Commerce: Delivering Product Innovations at Internet Speed. *MARKET VIEWOPINT Volume 12/Number 9.*

D.Tony L, X.William X., 2001. A review of web-based product data management systems. *Computers in Industry 44, pp 251-262.*

Dragan S. D., Miao C. Y., Chee F., Yew C.,2000. An Infrastructure for Inter-Organizational Collaborative Product Development. *Proceedings of the 33rd Hawaii International Conference on System Sciences.*

Jim, k., Markus, H., Rogerio, B., 2000. Usage Guide for the STEP PDM Schema. *PDM Implementor Forum.*

Web Services Activity. *http://www.w3.org/2002/ws*

Hardwick M, Spooner DL, Rando T, Morris, 1996. Sharing manufacturing information in virtual enterprises. *Commun ACM 39(2):46–54*.

Bliznakov PI, Shah JJ, Urban SD, 1996. Integration infrastructure to support concurrence and collaboration in engineering design. *Proceedings of the ASME DETC and CIE Conference*.

Urban SD, Shah JJ, Liu H, Rogers M, 1996. The shared design manager, interoperability in engineering design. *Integr Comput Aided Eng 3(3):158–177*.

Shah JJ, Jeon DK, Urban SD, Bliznakov P, Rogers M, 1996. Database infrastructure for supporting engineering design histories. *Comput Aided Des 28(5):347–360*.

Senin N, Borland N, Wallace DR, 1997. Distributed modeling of product design problems in a ollaborative design environment. *CIRP International Design Seminar Proceedings*.

Kim H, Lee JY, Han S-B, 1999. Process-centric istributed collaborative design based on the Web. *Proceedings of ASME DETC and CIE Conference*.

Liu DT, Xu XW, 2001. A review of web-based product data management systems. *Comput Ind 44(3):251–262*.

Yeh SC, You CF, 2002. STEP-based data schema for implementing product data management system. *Int J Comput Integr Manuf 15(1):1–17*.

SmarTeam. *http://www.smarteam.com/homepage.asp*

SolidWorks. *http://www.solidworks.com*