37

A PDM-BASED VE-ORIENTED INFRASTRUCTURE FOR DISTRIBUTED COLLABORATIVE DESIGN

Yinsheng Li[1♦], Weiming Shen[2]
[1]*Tsinghua University, Beijing, P.R. CHINA*
[2]*National Research Council - IMTI, CANADA, weiming.shen@nrc.ca*

This paper presents a Product Data Management (PDM) based infrastructure with a coordination agent for distributed collaborative design. The proposed approach exploits applications and resources from PDM systems as well as web platforms, and provides a solution for both heterogeneous PDM systems integration and VE-oriented collaborative product development. The proposed coordination agent (CA) helps ensure the data consistency and secure collaboration among remote sites. Such an approach shows some advantages over the peer-peer interaction in feasibility, efficiency and reconfigurability. Because of the introduction of CA, the resulting collaborative workspaces are prospective to be further developed to interact with other agent-oriented systems for VE.

1. INTRODUCTION

As cross-platform applications, Product Data Management (PDM) Systems are becoming efficient facilities for collaboration in more and more organizations. PDM systems take product data as basic information objects and product structures (generally bills of materials) as core organizing backbone, combine a series of engineering data and documents with powerful consistency and concurrency control, and support data exchanges between distributed engineering applications with their I/O data uniformly managed (Interleaf, 1998; Enovia, 1998). All these features bring about an efficient workspace for product data and workflows management, facilitate distributed collaborative activities on product design, process planning and manufacturing scheduling, provide data support for higher-level enterprise management and market systems, make PDM systems a promising candidate for integrating product development platforms in a distributed ad hoc virtual organization, and make enterprises with PDM systems implemented more competent for VE participation.

However, it is usually rather by adapting existing PDM systems than installing brand-new systems that a VE development environment is constructed. Considering partners' independency, there unfortunately always exist so many PDM systems, which are usually distinct and proprietary, even with heterogeneous product data and

[♦] Currently at: National Research Council - IMTI, London, ON, Canada, e-mail: yinsheng.li@nrc.ca

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35585-6_68

L. M. Camarinha-Matos (ed.), *Collaborative Business Ecosystems and Virtual Enterprises*

processes. It is imperative to find out an optimal solution to interlink these PDM systems, make their data accessible to each other, extend their workflows across the distributed collaborative platforms, and come into being integrated, efficient and unified product development workspaces for multidisciplinary collaborations (Li, 2000; Dreer and Koonce, 1995).

For integration methodologies, distributed computation architectures are being adopted extensively for collaborative engineering systems (de Graaf and Kornelius, 1996; Jasnoch and Haas, 1996). However, considering the fact that collaborative developments on these ad hoc non-arbitrated systems have not been carried forward in practice, we can conclude that there must be some unreasonable factors behind. In a virtual enterprise, despite partners' information sharing and collaborative design, there usually are intense competitions among themselves, in that all of them are independent and profit-earning organizations. On the other hand, considering the non-arbitrated distributed architectures' point-point communication, it is necessary for all involved subsystems or agents to know a lot about product data and systems of their partners. This is not only so incompetent in business security that collaborations cannot go forward in-depth, but also leads to enormous overheads in communication, with their dynamic descriptors for the entire system having to be updated whenever a subsystem joins, quits or changes.

More and more facts are showing that, it is inoffensive at all for partners' proprietary, independent, and distributed operations to take a coordination management over autonomous participated subsystems in an information system. In fact, we can see in human community that intermediating activities/services are fairly essential and universal in almost all communicating sessions between either companies or people. In the case of VE-oriented collaboration, since there is always an advocator/originator, it is viable to build up a powerful and robust coordination system with their solvency and cooperative motivations (Li, 2000).

As a human society-imitated result, it is practical to take an intelligent component as a global coordinator in a VE-oriented distributed collaborative PDM-based system. A paradigm catering to these arguments has been proposed in (Li, 2000). Herein, the paradigm gets improved with some agent-compliant features added into its coordinating facilitator, and will be discussed as "PDM-based Infrastructure with a Coordination Agent (PICA)". Its architecture, coordination agent (CA) and mechanisms are outlined in the following sections.

2. ARCHITECTURE AND DATA FACILITIES OF PICA

The objective of PICA is to build up a collaborative development platform for Virtual Enterprises. As illustrated in Figure 1, unlike most other integrated paradigms which usually develop their proprietary integrated infrastructures (Jasnoch and Haas, 1996; Dreer and Koonce, 1995), PICA is based on applications and resources from PDM systems. It takes these PDM-based applications and resources as essential components of the integrated framework for distributed collaboration. The strategy has a twofold result: excellent at implementation efficiency and work performance while limiting applicable subjects to the enterprises with existing or prospective PDM systems. The proposed integrated infrastructure is therefore very suitable for the enterprises using PDM systems.

By customizing and improving existing PDM systems in terms of distributed collaborations and interactions between them (Interleaf, 1998; Enovia, 1998), PICA deploys local web-based services in partner enterprises. As mentioned, one of PICA's prominent features is its web-based autonomous coordination agent (CA). There are three kinds of essential components in total (CA, local web services and PDM systems, refer to Figure 2), including their interactive resources (services), and all of them make up an integrated infrastructure, which stands over the existing PDM systems and allow for web-based distributed collaborative developments among remote sites (Li, 2000).

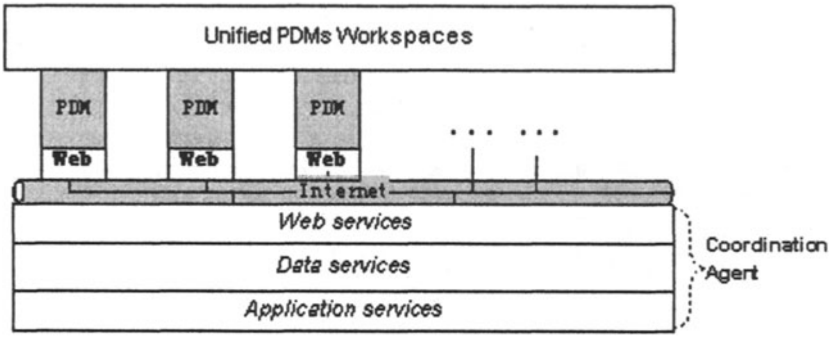

Figure 1 - The PDM- and web-based architecture of PICA

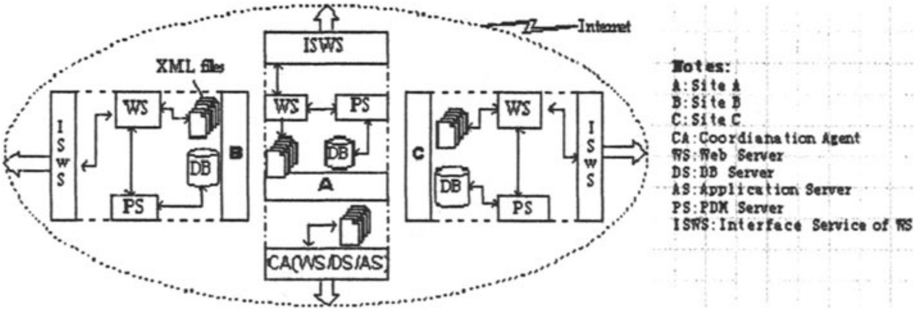

Figure 2 - Deployment of PICA (CA locates at Site A)

From the viewpoint of data facilities, PICA has a three-layer data facility (Figure 2). Firstly, there are proprietary product databases (DBs in PDM systems) in all partners. Next, by filtering, translating, abstracting and improving local product data in PDM systems, some DBs (represented by XML files) for sharing, collaborating and interacting are built up dynamically in their respective web servers. As formulated and mapped subsets of DBs in PDM systems, these data are non-persistent, and only when interacting with the CA, services in the local web server retrieve data in real-time from PDM systems, and translate them into XML files for communicating with the CA and external systems (Goldfarb and Prescod, 1998). Finally, a global DB (GDB) of the CA plays as a primary facility for product data sharing and interaction between distributed subsystems. The GDB consists of two types of sub-DBs: one is about structured metadata representations of the general and integrated models for subject products, cooperative enterprises, users/designers, and capabilities, features and paths of all applications in the current distributed systems; and the other is about rational knowledge, including the representations of actions, rules, history, privileges, and so on.

3. CONFIGURATION AND MECHANISMS OF PICA

As shown in Figure 3, there are three essential configurable components in PICA: local PDM systems, local web servers and the CA. At their server, a plenty of interacting services have been built up on respective DBs. These services could be applets, servlets, daemons, DLLs, customizing APIs, and other applications - depending on their underlying platforms, and with them being constructed by Java, JSP, and XML. They are apt to standardized middleware and plug-ins, which in return enhance the generalization, modularization and reconfigurability of PICA.

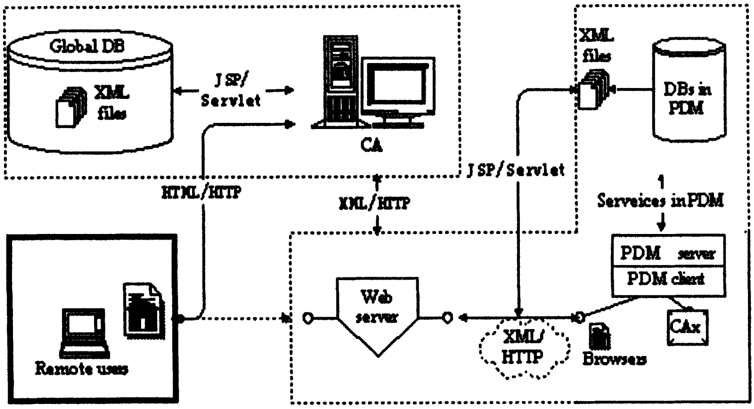

Figure 3 - Configuration and work mechanism of PICA

As an independent facilitator, the CA provides translating, searching, intermediating and interacting services for product data sharing and interacting transactions during collaboration. When an HTTP request arrives, the CA calls pertinent servlets to look up the Global DB, validate the identity of the request client and determine specific responding schemes against security/privilege rules and users descriptions, and return individual pages for the client's further accesses.

At the local web servers in collaborative enterprises, there are specialized routines, which are built up with APIs of PDM systems and web resources, as media to access DBs of the local PDM systems and interact with the CA, i.e., interlink specific product data with the items in individual pages to be sent to the client.

As PICA's essential components, PDM systems not only manage product structures, workflows and applications in their local development workspaces, but also analyze and execute extended data updating and workflows distribution with their customized services and client applications, and put the collaboration into practice.

Ideally, with its CA, local websites and PDM systems cooperating with each other, the data flows and interactive activities in PICA-based distributed environments can expect to be executed as they were in a single enterprise.

4. THE COORDINATION AGENT (CA)

It is necessary for a good integration coordinator to provide diverse intelligent interacting and mediating services, i.e., the CA must be smart and rational. For example, it should guide a request to a provider that can not only serve what the request wants, but also perform best in that case. In light of data security and traffics, the CA should make differences between specific requests in terms of their user identities, application types and data objects.

Specifically in the current VE case, it also has to be fair and impartial so as to get all partners' commitments to play a mediating role, which requires it to be autonomous and neutral with the collectively appointed institute to maintain it. All these features have been encouraging us to design it as an intelligent agent. Considering integrated methodologies prevailing in PDM community, however, it is difficult and hardly feasible to design the CA as a strict agent from scratch. As a result, an acceptable solution is to adapt it based on its similar features to agents, and then improve its intelligent performance by agent technologies (Shen et al., 2001).

Essentially, the CA provides dynamic global metadata views of subject products, partners and collaborative information, so as to enable perfect interactions between partners. Furthermore, as an integration coordinator, it is necessary for it to provide enough global applications and services, such as alias service, querying service, reserving service, authorizing service, workflow service, version management and data updating service, object lock and concurrent control, partners joining and quitting service, and so on. Contrasting to commercial PDM systems, which generally provide most of similar services (Interleaf, 1998; Enovia, 1998), the CA's mission is to extend them in space and types.

In order to enhance collaboration performance, a number of professional applications and facilities on the CA can be introduced, e.g., a friendly administration console for the CA and an engineering data management system for collaborative design have been developed based on the CA in the prototype implementation (Li, 2000).

Considering the scalability and reconfigurability of the CA, there are a separate HTTP server, a DB server and an application server, which construct a multi-layer transaction model (see Figure 4). The HTTP server responds to the requests from browsers and returns with "individual pages". The application server works on all the coordinating and mediating transactions. The DB server maintains the global DBs and provides knowledge for its associates. Most services mentioned above fall under the application server. It can take these services as groupware, combine its own functions with applications from similar servers/agents in other platforms, with application and content interacting protocols, e.g., EJB/CORBA/XML, and carry out together the CA's essential tasks.

Both the interaction between applications/agents and the Global DB model are interesting subjects in this paradigm, but it is difficult to describe these in details hcrc duc to the space limitation. More details can be found in (Li, 2000).

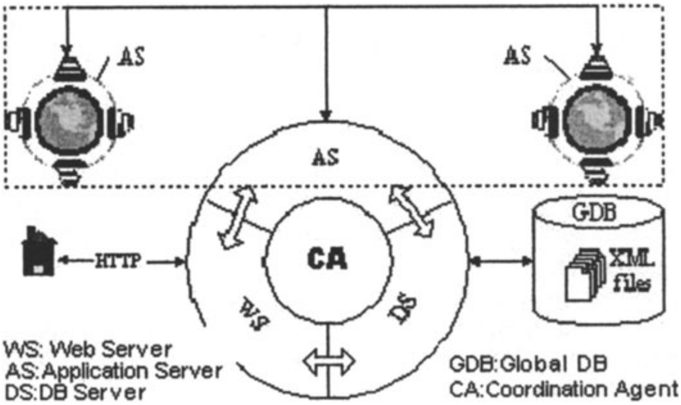

Figure 4 - Multi-layer transaction model in the CA

5. PICA-BASED DISTRIBUTED TRANSACTION SCENARIOS

With coordination services at the CA and local web servers, and customized PDM routines (daemons), the subject product data can be kept consistent, and distributed workflows can be executed over the VE development workspaces.

5.1 Data consistency in distributed collaborative engineering environment

In PICA, it is by "updating services", which consist of applets in the CA and local web servers that keep data consistent in a distributed environment. Firstly, together with involved PDM services, a retrieving/distributing service in the web servers of partner enterprises dynamically retrieves the selected product data from databases of local PDM systems according to defined mapping lists, translates and supplements them into XML files following conventional semantic definition, and puts them into the specific local web directories. Next, a local updating service pushes the files to the CA as a routine or in real-time when a change happens, and the updating service in the CA validates and provides them with a conventional version, and stores into corresponding directories. As a rule, only the metadata such as locations, status, and some other management properties of product data, are submitted to the CA, and the data objects are still kept where they are.

On the CA side, a lookup service scans product data regularly, and as soon as a data change is found. It determines the related data objects by consulting product structure trees, bills of materials, and inter-organization relation lists, and works out an updating program along with data consistency rules. Then, it accesses the GDB and all the web servers, sends them notifying messages for changing the correlated data in the GDB and PDM databases to keep them consistent with each other.

Another service in the local web servers, which usually is established by customized PDM applications with APIs that vendors provide, is to check notifying messages periodically, and as soon as an updating suggestion is found. It deals with the versions of destined product structures and their data with suggcstcd opcrations.

5.2 Data accessing and manipulating

In PICA, when an involved user navigates or manipulates product data in other enterprises, he has to visit the web server of the CA first. There are product lists, bills of materials and other product information in the "individual pages" on the server. In case that he passes the validation and is entitled to go farther into the product data after intelligently determined and assigned, he can activate applets under the data links, which subsequently send requests to the destined sites with a corresponding parameter, guide them to build a session and leave them interacting with each other. A publishing/interacting service in the destined web server interacts with its PDM system, returns appropriate encrypted data to the browser of the request user, and translates between user and PDM system.

5.3 Distributed workflows

For a process/workflow, which is executed in predefined procedures in a PDM system, if one of its procedures is involved in a remote site, the PDM system in which the workflow is initialized has to activate a monitoring service by an event message, which in turn sends a workflow notification in XML into the local workflow directory. With a workflow service in the CA finding the notification, it analyzes and translates the workflow into a conventional distributed workflow by referring to common workflow templates in the GDBs, and intelligently determines its executive details of destinations, subjects, transactions (check, review, correct or approve, etc.), as well as its elements in the destined PDM systems. With the intervention of a specific global workflow controller, workflow notification files are generated and sent to the destined sites with its data objects. The workflow service in the destined site transfers its data and properties into a mailbox in the receiving PDM system. After it is transferred successfully, a status message is returned along the reverse way through the controller. As such, as soon as a procedure is finished, the message is returned to the controller and involved PDM systems respectively.

5.4 CA-based communication security between sites

In current PDM-based distributed collaborative designs and their data sharing, there are few professional safeguarding considerations for site-site communications. Take Product Manager (Version 3.2.0B) of ENOVIA Corp. for an example (Enovia, 1998), when communicating between its remote servers, product metadata are translated into a simple PDIF (Product Data Interface Format) file, which is text-based, defined by themselves and liable to be recognized by external users, and then transfers it, along with data files, to another site by common FTP for its ACM service (daemon) to revert and put into PDM DBs and file directories over there. In this way, the PDIF file is transferred online by ASCII texts, and any crackers or servers connecting with Internet/Intranet can seize it as long as they can find any account/password for it. As a result, communications between organizations can be only engaged in idle theorizing, it is hardly practical to interchange data between different PDM systems. Many other systems are at a similar situation at large.

PICA provides an efficient way to secure and maintain remote communications between PDM systems. With its CA and GDBs, an Internet-like certificate authority can be built up by developing a special authenticating service within a VE. Thereafter, the certificate authority can authenticate communicators, and with a public-key encryption solution (e.g., a symmetrical method, an unsymmetrical method, or their integration), offer them a certificate, a public key and digital signature. Therefore, a secure infrastructure comes into being, and the data security and integrity are subsequently guaranteed across all over the VE.

6. CONCLUSIONS

With its keeping partners' independency and autonomy, PICA enhances global and dynamic development collaboration across all over a VE, and allows remote designers for collaborating smartly with each other in a similar way as they were in a single studio. Contrasting with those usual and well-known distributed integrated frameworks, PICA has some promising features: 1) a neutral CA abating distrusts of potential opponents in a VE and enhancing data sharing viability; 2) inheriting the resources and legacy data of PDM systems; 3) obvious configuring/reconfiguring efficiency; 4) enabling partners' quickly joining or quitting; 5) standardized services and the CA's agent features; 6) dramatically reduced overloads by avoiding excessive peer-peer communications; 7) an enhanced communication security with its CA providing a security solution between remote development platforms.

In summary, PICA is an open, integrated and practical infrastructure for ad hoc VE-oriented collaborative design, which not only resolves the issue of integrating heterogeneous PDM systems, but also provides a web-based distributed development platform for VE-oriented collaboration. Because of the introduction of CA, the resulting collaborative workspaces are prospective to be further developed to interact with other agent-oriented systems for VE.

7. REFERENCES

1. de Graaf, R. and Kornelius, L., Inter-organizational concurrent engineering: A case study in PCB manufacturing, Computers in Industry, 1996, 30:37-47.
2. Dreer, P. and Koonce, D.A. Development of an Integrated Information Model for Computer Integrated Manufacturing, Computers ind. Engng, 1995, 29(1~4):109-112.
3. ENOVIA Corp., ProductManager: Advanced Customization Environment Script Reference and Operations Manual, Version 3.2.0, IBM Corp., 1998.
4. Goh, A., Hui, S.C. and Song, B. An integrated environment for product development using STEP/EXPRESS, Computers in Industry, 1996, (31):303-313.
5. Goldfarb, C.F. and Prescod, P., The XML Handbook, Prentice Hall PTR, 1998.
6. Interleaf Inc., IMAN Online Help 2.1.0/sunos5, Interleaf Inc, 1998.
7. Jasnoch, U. and Haas, S., A collaborative environment based on distributed Object-oriented databases, Computers in Industry, 1996, 29:51-61.
8. Li, Y., Integrated Infrastructures for Collaborative Development in a Virtual Enterprise. PhD Thesis, Tsinghua University, Beijing, P.R. China, 2000.
9. Munch, B.P., Conradi, R., etc., Integrated Product and Process Management in EPOS, Integrated Computer-Aided Engineering, 1996, 3(1):5-19.
10. Shen, W., Norrie, D.H. and Barthes, J.P., Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, Taylor & Francis, 2001