Process-Driven Integrated Product Design Platform

Bo Zhao, Jun Luo, Rongmei Nie, Xiaojun Hu, and Hui Zhao

Abstract. In order to realize effective process management and application integration used in the whole product development process, some advanced techniques or methods have been proposed and successfully applied in product design. But in distributed environment, how these engineering tools can effectively access right data in right format at the right time inevitably becomes a very important factor in product development. So a flexible integrated design platform for process integration has been established, which consists of representation layer, process control layer, services layer and repository layer. Based on analyzing three patterns of organizing data, context data model is introduced, which organizes data based on process. The mechanism of interaction between context data model and process unit is described in detail. The model-view-controller (MVC) design pattern is adopted in our design platform which separates data model from its various views. Web services are developed to realize the dynamic data integration between applications. The design platform based on the key techniques is developed as a prototype to prove the concept.

1 Introduction

The development of information technology and the global economy strongly drive the transformation of enterprise of design and manufacture. Enterprise should have a rapid response to the needs from market. Nowadays the advance methods and techniques, such as: CAx(computer-aided technology), PDM (Product Data Management) and DFX (Design For Assemble: DFA, Design For Manufacturing: DFM and etc.), have been successfully applied in product design. But plenty of heterogeneous data which generated by different applications or systems exists in product lifecycle throughout the enterprise. Moreover, the product

Bo Zhao
Beijing Institute of Astronautical Systems Engineering, Beijing, China
e-mail: zzzbbb163@163.com

F.L. Gaol (Ed.): Recent Progress in DEIT, Vol. 1, LNEE 156, pp. 275–280.
springerlink.com

development process involves multidisciplinary knowledge, multi-designers, multi-parts of company. So the simple use of CAx applications ceases to be an important factor in different competitors. Therefore how multi-participant collaboratively finish design or analyse and how applications can access right data in right format at the right time determine whether a specific project can be successful or not. So it is necessary to establish a flexible integrated design platform which can integrate different applications and support designers' work.

Some of valuable works have been researched by different person or groups, For example, integrated web-based PDM system [1], collaborative product development environment [2], process integration framework for product design [3], process integration and resource sharing of different design software [4], the model for application integration [5], and etc.

2 Architecture

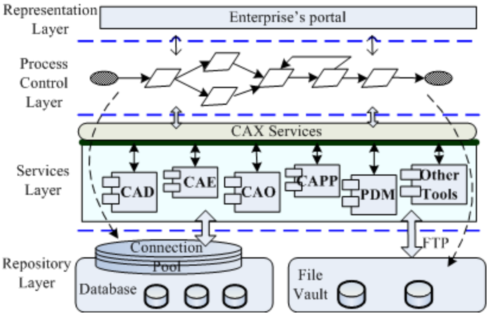

Fig. 1. For optimizing the design process and smoothly exchange data, document and knowledge, we establish a extensible process-driven integrated design platform[3], which consists of four tiers, namely, representation layer, process control layer, services layer and repository layer. Every layer will be ex-plained in more detail below.

(1) Representation layer

Web-based interface for users is created, which allows every authorized user to visit the enterprise's portal in everywhere. The model-view-controller (MVC) design pattern is used to design modular and interactive user interfaces, through which the data model can be separated from its various representations. This pattern will be introduced in the following key techniques.

(2) Process control layer

This layer is the functional core of this architecture, which realizes most functions of process integration, namely, mapping enterprise's business flow, regulating the logic sequence of actions in process, optimizing every phase of process, encapsuling business logic, managing users and their roles, and maintaining security of system. In this layer, as data organization is based on process, the services provided by applications are invoked by process engine, resources are configured by activities from process and human are organized by process, all elements involved in product design are process-driven. So this architecture is process-centralized.

(3) Services layer

In distributed environment, all applications and engineering tools can be regards as service providers and service applicants. Some operations and functions can be developed as web services. There is a service registration centre which records the information of services provided by applications. All the requestors can find the required services which have been registered.

(4) Repository layer

It stores all data in process, which has the capability of retaining the data's integrity and security while being concurrently accessed. Meta-data are read and written in database through connection pool, while files are stored in file vault. Knowledge of context is also stored in database in a regular format.

For establishment of the four-tier architecture, we research four key techniques, which will be introduced as follows.

3 Key Techniques

A Patterns of Organizing Data

In order to effectively manage and utilize various data in development process, there are some following patterns for organizing data [6]:

(1)Organizing data based on product structure

In this pattern, Product structure is the framework of data organization and management. Due to the dynamic change of product structure tree in development process, the operation that adding, deleting or modifying a node would influence the data on this node and the relevant nodes. Moreover, there are many different views in product development process, such as design view, assembly view, and etc., so the organization may change for the special view, and express the static relations about product information in a certain view. In other words, product data simply stack on structure tree.

(2)Organizing data based on folder

In this pattern, file folder is regarded as the container which reflects the product structure. The structure of folder is decided by product structure, which can differentiate and manage different information about different parts. The consecution and rationality of the folder directly influence the convenience of data's access.

(3)Organizing data based on product lifecycle

A.McKay[7] proposed a framework for product data, in which data is divided into three kinds, namely, specification, definition and actual data. The three kinds of data correspond with the different phases of product development. Organizing data based on product lifecycle express generation order of data in different phases, which can be regarded as process-oriented in part. However, this pattern also classifies data from product structure, which groups information in different folders at different phases.

(4) Organizing data based on process

The three pattern mentioned above only orient a certain view of product data. Due to the complexity and dynamic of product and its development, organizing data based on product structure or lifecycle is a static method comparatively, which weaken the relations of different information generated in process. So for the dynamic process, we present context data model. Context data model should not only describe many kinds of data type, but also reflect complex relations in process, through which the steps of process can exchange information more fluently and engineering applications will collaboratively work.

B Mechanism of Interaction between Context Data Model and Process Unit

Organizing data based on process reflects the close relation between data and process. Generally speaking, the process of product development can span a widely period, from customers' needs, conceptual design, and detailed design to manufacturing. The development process can be divided into three levels according to different granularity from macroscopy to microcosm: the project level, the workflow level and the design process level, which could be called as project flow, workflow and design process. No matter how size of the granularity of the nodes, there are some cells which hold the independent functions in process. These cells can be called process unit. Process unit specifies how context data model transfer. The logic permutation and combination of units decides the flow direction of data. Context data model materializes the input and output of unit, and decide which unit will be executed in branch, who will run it, when and how it runs. The network constructed by process units defines the framework of data's flow and transformation, while the speed and direction of data flow are decided by itself.

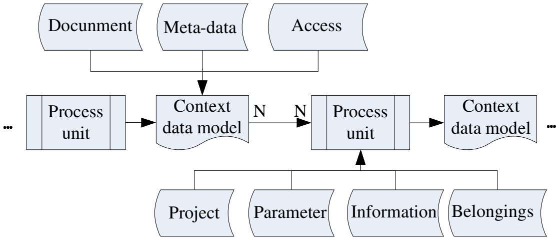

Fig. 2. The Mechanism of interaction between context data model and process unit is depicted in Fig. 6. The task of process integration is defining the topology based on process units and constructing the container for context data model. So the product development process is the instantiation process for data model, in which data experience a course from generation, transformation, and enrichment to per-fectness.

C Model-View-Controller (MVC) Design Pattern

Models are those components of application systems that hold specific information, in which data stores as a certain format. Views deal with everything graphical, which request data from a Model and display data. Views are closely associated with a Controller. Each Controller-View pair has one model, whereas each model may have many Controller-View pairs. Controllers are the bridges which connect Models and Views.

For example, XML file which represents a class of Context model could be presented to requestors as the particular format. Therefore, if the interface needs to change, the View-Controller pairs can be rapidly modified without changing the Model.

D Dynamic Data Integration Based on Web Services

In the design process, applications need to exchange data with design platform or other applications at run time. So, an effective method should be presented to satisfy the requirement. A web service is a software module performing a discrete task or set of tasks that can be found and invoked over a network including and especially the World Wide Web. The developer can create a client application that invokes a series of web services through remote procedure calls (RPC) or a messaging service to provide some or most of the application's logic. Web services can be written in any language and run on any platform. Therefore, it can be used to realize the dynamic data integration between applications with the following steps:

Step 1: the development of web services. Some application’s logic is developed as Web services. Take PDM system for example, we write the program by Microsoft Visual C#. And users can upload or obtain product data through web pages. But it is impossible for an application to exchange data with PDM through web pages. So some interfaces should be developed to facilitate the data availability between applications and PDM. We encapsulate some functions which can be only achieved through web pages before as web services by invoking or combining the methods which exist in the business logic layer of PDM. These functions include checking in or out documents, searching, obtaining attribute data of a part or a component, etc.

Step 2: the registration of web services. All developed web services should be registered so that they can be found.

We use Web Services Description Language (WSDL) to describe the entrance, the interface, the input and the output of a web service. And then, we register it in UDDI （Universal Description, Discovery & Integration） registration centre which contains basic information of all developed web services.

Step 3: finding and invoking web services. After service provider register its services, consumers can lookup the required service by service information in registration centre which will provide the location and the WSDL file of service provider. A service broker which is created for each application can remotely invoke the methods of web services as the prescriptive format. They communicate with the SOAP (Simple Object Access Protocol) messages which are actually XML documents. SOAP messages are transported by HTTP between a web service and the calling application. Thus, web services can normally work through different platforms.

4 Conclusion

Based on the flexible architecture and the key techniques mentioned above, we have developed a web-based prototype for an institute which supports the design of air filter for engine. The system consists of ten steps for design, and integrates some applications, such as: MatLab, Pro/ENGINEER, ANSYS and other tools we developed. So the key techniques and architecture are confirmed by the prototype. And we will develop another system for a specific product; meanwhile, the theory about process integration, process modeling and architecture will be improved.

References

[1] Wang, L., Xu, Z., Ruxin, N.: Key Techniques for the Development of Web-Based PDM System. Journal of Beijing Institute of Technology 15(3), 269–272 (2006)

[2] Jianshou, K., Zhang, Y., Wang, H.: Study on Integration of Distributed PM and Workflow in Collaborative Product Development Environment. China Mechanical Engineering 14(13), 1122–1125 (2003)

[3] Bo, Z., Yan, Y., Ruxin, N., et al.: Key techniques about the process integration for product design. In: IEEM 2007: IEEE International Conference on Industrial Engineering and Engineering Management, pp. 568–572 (2007)

[4] Hu, Y., Fan, Y., Tong, X.: Process integration & resource sharing based on XML & process ontology for MDO. Computer Integrated Manufacturing Systems 13(6), 1076–1082 (2007) (in Chinese)

[5] Harikumar, A.K., Lee, R., Yang, H.S., Kang, B.: A model for application integration using Web services. In: Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS 2005), pp. 468–475 (2005)

[6] Wang, B.: Research on Theory, Technology and Application of Digital Product Development Process Management. Ph. D. dissertation of Beijing Institute of Technology, pp. 32–34 (2004) (in Chinese)

[7] McKay, A., Bloor, M.S., de Pennington, A.: A Framework for Product Data. IEEE Transaction on Knowledge and Data Engineering 8(5), 825–838 (1996)