Research on Heterogeneous Component Assembly Problems Based on Software Product Line

Jianli DONG [1, a]

[1]HuaiHai Institute of Technology, Lianyungang, 222005, P.R.China

[a]dongjl1019@sina.com

Keywords: Heterogeneous component, Component wrapper, Component assembly, Component based software engineering, Software product line

Abstract. With the research and development of software product line and component-based software engineering methodology, it has become key technology how to build component assembly environment and realize heterogeneous component assembly. Thus, a new industrialized product line based integrated software engineering environment (PL-ISEE) model is firstly proposed, and the heterogeneity problems of developed components are analyzed and discussed. For removing component's heterogeneity and realizing heterogeneous components' assembly, the wrapper wrapping component is studied further. These researches and ideas will play significant role in promoting the formations of component-based software engineering methodology and future industrialized software production technology.

Introduction

In recent years, with the research and development of the component based software engineering, component assembly technology has become the core technology of component-based software development and software product line, and it has been very popular in the field of software engineering and industrial circles. Only when the components are assembled to form the application system, and can the values of the components are realized. From the current study, component assembly it is still to study the hot topic and difficult point in the software component, software architecture, software product lines, software reuse and so on related technology. The main difficulty of component assembly is the heterogeneity of the components, namely, the differences among the components developed by different manufacturers have used different component models and different programming languages and different operating environment. It is main reason that the different components can not be assembled together to form an application software system running under the same operation environment. There are two methods to solve the heterogeneous components assembly problems. One is overthrow all heterogeneous components, and reproduce new standardized components according to a unified component model and environment; the other is standardizing reform to the developed heterogeneous components in accordance with the requirements of the assembly and application, making them as available components.

Obviously, the first method is not feasible because it requires companies to destroy all of the component products that they have developed and redevelop them with standard component model, which will cause too much costs and losses. The second method is a feasible way. Although the way can not fundamentally solve and remove the heterogeneity of the components and a lot of assembling troubles, it is still a compromised method under the premise of maintaining the common interests of the enterprises. Actually it is also a popular method to use and assemble the developed components today. Thus, the component assemblies produce a powerful constraints and limitations to component technology development in the opposite direction from component production to component services.

This paper will discuss how to realize heterogeneous components' assembly by removing components' heterogeneity in product line based integrated software engineering environment (PL-ISEE) [1-2].

Published by Atlantis Press, Paris, France.

An Industrialized PL-ISEE Architecture Model

For realizing heterogeneous components' assembly, the component assembly environment (or called as assembly line) should be firstly researched and created.

The component-based industrialized software engineering paradigm emphasizes producing software products by the method assembling components rather than the traditional programming method from scratch. And it also emphasizes that its software production process and life cycle model is completely different from the traditional software engineering one. Its life cycle is defined as: 1) Requirements analysis; 2) The design and evaluation of software architecture; 3) Components selection and customizing; 4) Components assembly and system configuration; 5) System testing; 6) Software application and maintenance. The components assembly takes the system architecture and framework as design blueprint, and assembles and configures the selected and customized components to generate the application system. It shows that component-based software development method is another new software engineering methodology aimed at realizing the industrialization production of software products after the traditional structured software engineering methodology and modern object-oriented software engineering one. However, we still face many problems to translate the new methodology into reality and produce software products in the COTS (Commercial Off-The-Shelf) component assembly method. One of them is how to build an ideal industrialized component assembly environment (assembly line). This environment is also called as the product line based integrated software engineering environment in the software (PL-ISEE).

The research and creation of PL-ISEE model must be on the basis of correct and complete product line based software engineering process models and life-cycle model. This process and life-cycle model is adopted to define and describe all the production process and requirements, such as activity sequences, task frameworks, the technology methods, standard specifications, quality controls, asset components structures and so on in product line software development. It is consider as the behavior guideline and activities norms of realizing software product line engineering and its products production, and has been the pre-conditions and important foundation of PL-ISEE research. Withal, we have proposed an opened “N-Life Cycle Model” in article [3] with modern manufacturing industry features. There, based on the “N-Life Cycle Model”, a new industrialized PL-ISEE architecture model is proposed by authors in the paper [4].

The new industrialized PL-ISEE model is proposed based on the unified product line engineering concept model, large granularity reusable asset data model, component assembly behavior model, core assets development and application software production iterated evolution model. The framework of this new environment model is essentially a three-layer (interface, tool and data layers) architecture model with core asset and agent components as the software bus. On the bus or data layer, it is a PL-ISEE (product line based integrated software engineering environment) similar to an industrialized product line of the modern manufacturing industry, which will realize the assembling production of software product. All the components needed on software assembly line can be provided by core asset and agent components bus as a component conveyor belt in the data layer; under the bus or data layer, it is the traditional ISEE based on traditional software engineering methods, which support the developments of source program and documents of the core asset components. Obviously, the new framework model is a real PL-ISEE model with the software industrialized producing (assembling) ability and control mechanism which is very similar to the automated assembly line and management mechanism of the modern manufacturing industry [5-6].

Assembling problems of heterogeneous components in the PL-ISEE

To realize the industrialized assembling production of application software products in the component-based software engineering (CBSE) and software product line methods, it has two basic conditions: Unified component model and integrated component assembly environment (also called assembly model). The component model defines the internal structure and external features of a component. External features provide the component interfaces and how interactive mode with other components; assembly environment is an integrated software development environment and infrastructure supported components assembly process and application system assembly production, such as the PL-ISEE is just a new component assembly environment proposed in paper [4].

The basis implementing component assembly must conform to a unified component model and standard. Unfortunately, component's development and application are still far from reaching this requirement. Currently, the component implementation models which have been universally acknowledged by this area include three types: OMG organization's CORBA model, Microsoft's COM / DCOM model and SUN Company's JavaBeans/EJB model. But each company goes his own way, and leads to the different component models and standards. This brought huge challenges to the components assembly [7].

Different component models and their corresponding component products will lead to the heterogeneity and non-interoperability issues between the components. Heterogeneity mainly represents two aspects: the first is component model's heterogeneity, which limits that application developers can only use the components from the same component model. While others model components having the same functions are used, these components must be redeveloped in same model. But this redeveloping work certainly increases the cost of software development. The second is the heterogeneity of component running platform or environment, which makes that the application system assembled by using one model's components can not run under the platform and environment supported another heterogeneous component model. It can also make the component assembly suffered a serious setback. Precisely because of the emergence of heterogeneous components, the components assembly will be confronted with many difficulties which are as follows [8]:

1) The same component model can be implemented by different programming languages. This will likely lead to inaccessible components belonging to the same one component model. What is more serious is that the data types and data structures defined and used in a component in a language are not directly supported by other programming languages.

2) For a similar concept and behavior, the different component model provides different accessing modes. For example, in the JavaBean system, access to the component attribute is achieved by declaring a public method member's name. In the COM, however, it is not dependent on its method member's name, but is realized by method sequence in component's binary interfaces. Additionally, different component models provide different component instantiation processes.

3) Different component model provide the different persistence mechanism and method of the component. This will imply that an application system assembled with the heterogeneous components may produce inconsistent data storage structures or database schemata during operation, Such as RDB and OODB schema. It is absolutely not allowed in the same application system.

4) The graphical identities of different components depend on the different operation platforms or running environments. Such as the JavaBean components' visualization needs a virtual Java machine and graphics container, but COM ones need the window handles of the Windows Operation System.

5) Heterogeneous component assembly requires developers to be very familiar with all component models. In fact, is very difficult that the different types of application systems are developed by the different component technologies.

Component Wrapper and Its Realization Mechanism

The heterogeneous component models have respective mechanisms and methods to access to the component features (such as the properties, methods and events of component). In order to shield the heterogeneous characteristics of the components, the component assembly environment must use the component wrapper as a united component model to generate a consistent component view and provide a unified external interface and accessing mechanism for all the components of the different models.

In the component assembly environment, all the component wrappers corresponding to any original component models should provide a united component interface CInterface. The method, property and event members of the wrapped component can be accessed by using the CInterface and a consistent way. Although the existed component models can support the three types of the feature information with the method, property and event, these feature information have not been restricted and normalized so far. This requires that the wrapping capacities of the component wrapper and abilities generating corresponding component instance should have better inclusiveness and expansibility. There are also many different implementation versions even for one same component model. A component wrapper base-class model is abstracted from the various wrapper models. The bass-class model can provide a mapping mechanism from component wrapper model to original component model, and then the mapping from a wrapper model instance to an original component model instance can be realized by the mechanism. Due to the limit of space, the functions and internal structure about component wrapper are discussed in relational paper [9].

Conclusions

The biggest difference between component-based software engineering methodology and traditional structured or object-oriented software engineering ones is the industrialized production of the software products. Thus, component-based software engineering methodology should include the studies of component assembly technology, software architecture, and product line based integrated software engineering environment. Although the paper has proposed a new industrialized PL-ISEE and wrapping and assembling heterogeneous components' ideas, but we have a long way to go for exploring how to form the mature industrialized software engineering environment and product line.

References

[1] YANG FuQing, Thinking on the Development of Software Engineering Technology, JOURNAL OF SOFTWARE / RanJian XueBao, Science China Press, Vol.16, No.1, pp.1-7, 2005.

[2] Zhang Xinyu, Zheng Li, Sun Cheng, The research of the component-based software engineering, In Proceedings of the 6th International Conference on Information Technology: New Generations, pp.1590-1591, 2009.

[3] Dong Jianli, Research on software engineering process model based on software product line architecture, Computer Engineering and Design, Vol.29, No.12, pp.3016-3018, 2008.

[4] Jainli DONG, Ningguo SHI, Research on the CORBA Implementation Mechanism of a New Industrialized PL-ISEE, Journal of Software, Academy Publisher, Vol.7, No.5, pp.1171-1176, 2012.

[5] SHEN Li-wei, PENG Xin, ZHAO Wen-yun, Software Product Line Architecture Modeling and Component Composition Implementation with Extension of Aspectual Mechanism, Chinese Journal of Electronics, Beijing: Chinese Institute of Electronic, Vol.37, No.4A, pp140-145, 2009.

[6] XIE Wu-ping, XUE Jin-yun, WAN Song-song, Aspect-Based Component Model and Its Assembly and Implementation, Computer Technology and Development, Vol.19, No.4, pp.160-160, 2009.

[7] Zhijian WANG, Yukui FEI, Yuanqing LOU, Software Component technology and Application. Beijing: Science China Press, 2005.

[8] HE Tian-zhang,WANG Pan-qing, LI Xiao-hui, Research and Application on CORBA Component Assembly, Science Technology and Engineering, Vol.8, No.1, pp.84-86, 2008.

[9] MAO Yingchi, LIANG Yi, WANG Zhijian, Design and Implementation of Model for Heterogeneous Software Component Composition, Computer Engineering, Vol.31, No.4, pp.56-57, 2005.