A Description and Transformation Approach for Model Driven Architecture

Lei Wang
School of Computer Engineering
Weifang University
Weifang 261061, China
40337143@qq.com

***Abstract*—By referring to the widely used web application frameworks on JavaEE, a description and transformation approach is proposed to resolve the problems of the transformation from platform independent models to platform specific models. To achieve model transformation, mapping rules can be defined according to the structure of the target model. Specific technologies of target application domains have less influence on the use of this approach. The experiment shows that the approach captures the essence, process and requirements of model-driven architecture, and thus can be used widely as a new guidance for model transformation.**

***Keywords**- model driven architecture; platform specific model; model description; model transformation*

I. INTRODUCTION

The transformation form Platform Independent Model (PIM) to Platform Specific Model (PSM)is a key technology of MDA (Model Driven Architecture) proposed by OMG [1]. The meta-model of the source modeling system and the target application are built formally through the abstract analysis, and then the abstract elements and their semantics are compared to define the mapping rules between them, and finally automatic transformation is achieved through auxiliary tools.

A number of model transformation approaches has been proposed by researchers at home and abroad. Graph theory is used to analyze the process graph in the transformation approach based on process graph [2], which decomposes the graph into the structure information flow to realize model transformation. It comes with rich semantic information to describe the process structure, but most of the business model does not provide enough precise semantic information to support this approach. MIDAS [3] uses a PIM hierarchical-based (conceptual layer, navigation layer, presentation layer) structure mapping approach for the development of Web information systems using XML and object-relational technology. This approach does a strong limit with the realization technology of the target system. A meta-tag based transformation approach is used in [4], in which PIM is marked according software architecture and then the labeled PIM will be transformed to PSM based on a specific platform. This approach is the lack of appropriate tools for model storage and exchange, and the establishing process of marks is also limited to XML-based architecture description, which makes it difficultly to be widely used. Other approaches, such as ATL-based model transformation approach, pattern-based model transformation approach and ODAC approach, all have only proposed a transformation framework with a lower degree of automatic model transformation, and the target applications are mostly limited to specific applications.

In this paper, a JavaEE platform specific modeling approach is proposed, which is composed of three parts: static model, action model and presentation model. It can be taken as the target and define mapping rules according to its structure to realize model transformation. This solution can be used widely for suffering little from the domain and some special techniques of target applications.

II. FMP: THE SOURCE OF TRANSFORMATION

The FMP (Function, Model, Presentation) [5] modeling approach is on the basis of the traditional application modeling approach by adding a description view of user interface. The model of this approach is not a description of the specific forms of interface elements and their attributes, but the description of the abstract data and behavioral elements of user interface. At the same time, the corresponding relations between interface elements and presentation objects are also shown. So that the interface elements is independent on any specific application platform, and data elements and behavioral elements are separated from the specific interface elements. As the source of model mapping, the FMP modeling approach can be used to build platform-independent Web application model.

A model of FMP-PIM includes the following five parts: architecture model defines the upper structure of the application which provides the macro components for the system partition; functional model provides the functional description of components of architecture model, which includes the system functions for the uses and relations of function decomposition of the system; object model describes the class required in the application system, which is a model of domain concepts; interaction model provides the detailed description for interactions between the application system and users, i.e. the antecedent factor and consequence action of system behavior caused by the user. It is also an abstract description of behavior of the user interface, which provides internal associations between user interface behaviors with the system functions; interface template is the presentation of interaction model, which provides the binding relations between abstract interface elements and interaction model elements.

Published by Atlantis Press, Paris, France.
© the authors

III. ANALYSIS OF TARGET APPLICATION FRAMEWORK

Java EE is used as a target platform to help interpreting the process of using the approaches proposed in this paper. Most of Web applications on J2EE platform are based on a certain kind of framework, such as Struts, Spring, JSF, Cocoon, Axis and Xalan et al, which are all widely used. By analyzing and abstracting the common features of these frameworks, we provided a JavaEE-based architecture model named SAP-PSM. Application models based on it can be transformed into target codes based on many frameworks easily for it contains the specific information on Java EE platform.

A. Static Model

As shown in Figure 1, static model of SAP-PSM adds detail information of Java EE platform to PIM, which mainly contains elements about Web application programs, such as Package, Java-Interface, Java-Class, Attribute, Method and the relationships between Java-Classes.

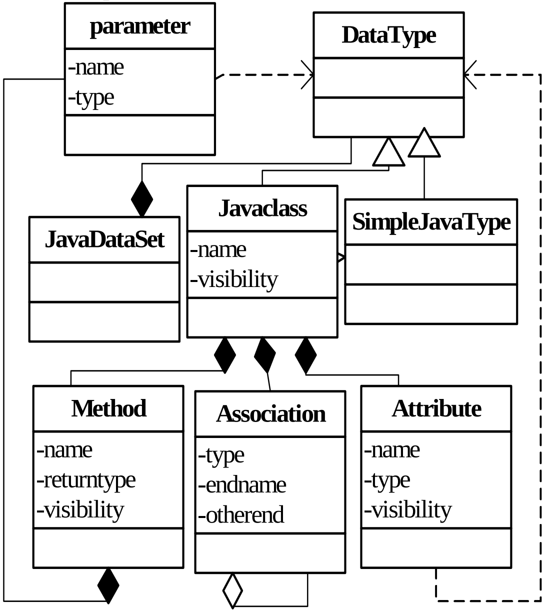


Figure 1. Definition of static model of SAP-PSM

B. Action Model

Action model of SAP-PSM is used to describe a system from dynamic aspect, and its elements are organized surrounding the solution of system tasks. According to the solution process of request of users, action component model is constructed with reference to the interaction relations between user and the system. The definition of action model is shown in Figure 2.

There are two kinds of action elements (*WebAction*) in action model. The first kind represents the entry-points for interactions between users and the system, and it can be touched off directly. The second kind of *WebAction* represents action elements within the system and can be touched off via the first kind. Navigation is the target of next step after the request is resolved. *ActionPara* represents the parameter object applied in the solution process of a system action. *EntityClass* represents the kind of object that is the target of an operating. Invoke is an invoked relation from *WebAction* to *DataObjec*t.

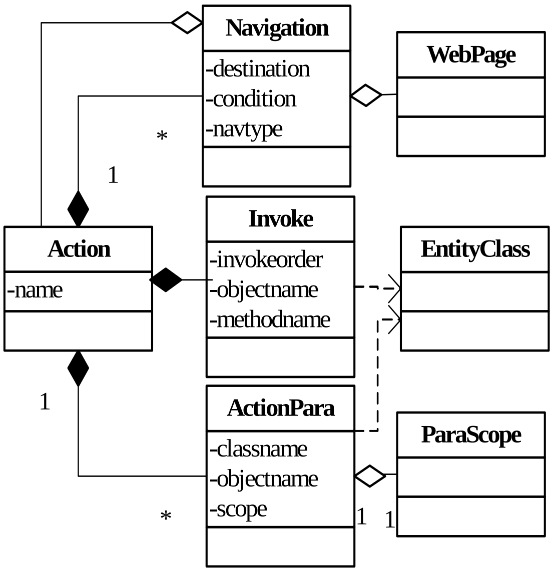


Figure 2. Definition of action model of SAP-PSM

The association of navigations represents the navigation relationship from the source to the target. Invoke association is call relations of the methods of the entity objects in the process of action management. Parameter association implies the participation relations between parameters and behavior management.

The essential semantic information brought by Action model is as follows: users touch a *WebAction*, and the application system receives user's request that maybe including *ActionParas*, then it analyses and dispatches the request to the corresponding actions. Next, *WebAction* invokes the method of *Invoke-Objects* to resolve the task. After completing, actions forward the request to next page according to the result, which is determined by the conditions and the target of Navigations.

C. Presentation Model

Presentation model of SAP-PSM is organized as a hierarchical tree-like form to represent the specific relationships of UI elements for Web applications, which is shown in Figure 3. Tree structure is used to present the division of a web pages or a page region, which has the attribute of distribution strategy (*Layout*). The user can nest in the region by sub-regions. Interface widgets (*WebUIComponen*t) area components included in the display region, such as data tables, forms, graphics, data trees, behavioral trigger points, and so on. Interface component should also specify the binding relationship between itself and data elements or behavior triggers. Interface elements (*WebUIElement*) are controller widgets which compose some page components. For example, the form can be composed of text, radio, combox, etc.

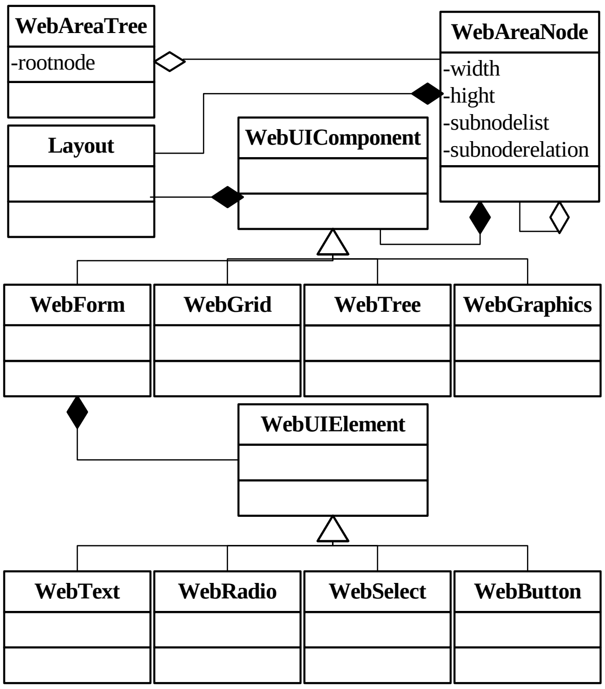


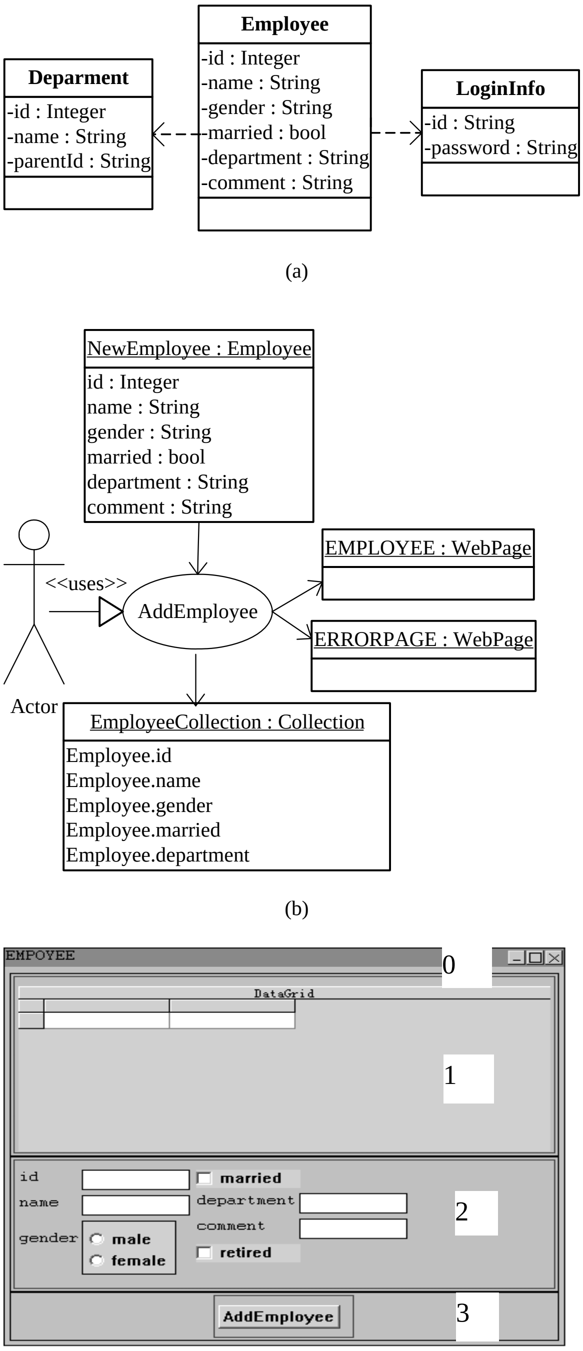
Figure 3. Definition of presentation model of SAP-PSM

IV. Mapping Scheme

Using modeling descriptions of FMP and SAP-PSM as source and target respectively, the mapping rules are defined according to the structure of SAP-PSM, which is shown in Figure 6. Detailed description information of JavaEE platform is added into the corresponding elements of FMP-PIM which will be converted to target model elements. The formal approach in [6] is extended and revised in this paper, and the transformation rules are omitted here for the limited space

V. Case Study

Figure 7 depicts a constitutive part of a PIM instance in the development of a human resources management system. Figure 4 (a) is the object model, which defines three classes: department, staff, and user login information. Figure 4 (b) displays the interaction information to add staff to the system. A NewEmpolyee parameter object is required when the user requests to add staff (*AddEmplyee*). If the operation is successful, the web page is forward to the page of *EMPLOYEE* to display the results, otherwise forward to the page of *ERRORPAGE*, which display the error messages. Interface template is shown in Figure 4 (c), which shows the interface for adding employee information template. Display area is composed of 3 sub-regions.



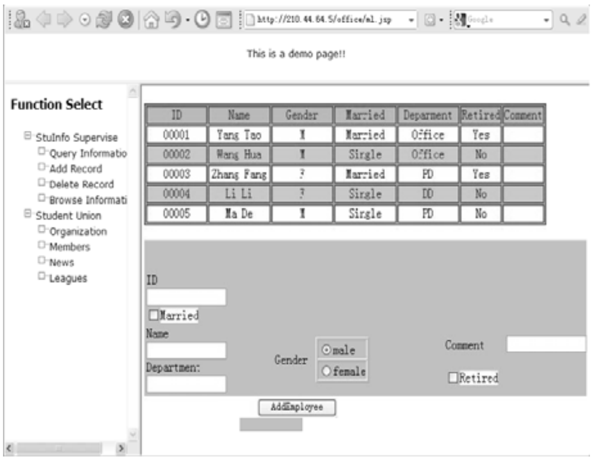
(a)

(b)

(c)

Figure 4. An example of FMP-PIM

Struts is used as the target application framework in the experiment, and the corresponding code generator generate target codes for entity class on the basis of static model of SAP-PSM. JSP codes are generated according to presentation model, and system configuration files and controller classes are generated based on dynamic models. The application system generated according to above model instance is shown in Figure 5, which shows a running page for adding staffs.



This is a demo page!!

ID	Name	Gender	Married	Deparment	Retired	Comment
00001	Yang Tao	I	Married	Office	Yes	
00002	Wang Hua	I	Single	Office	No	
00003	Zhang Fang	?	Married	PD	Yes	
00004	Li Li	?	Single	DD	No	
00005	Ma De	I	Single	PD	No	

ID

□Married

Name

Departmen:

Gender ●male ○female

Comment

□Retired

Figure 5. A running example

Through this example, it can be seen that SAP-PSM model and related approach for model mapping can make an effective support for automatic code generation for Web applications based on J2EE platform.

VI. CONCLUSION

In this paper, a description and transformation approach is proposed to resolve the problems of the transformation from platform independent models to platform specific models. The resolution support automatic model transformation, and mapping rules can be defined according to the structure of the target model. Specific technologies of target application domains have less influence on the use of this approach. The experiment shows that the approach captures the essence, process and requirements of model-driven architecture, and thus can be used widely as a new guidance for model transformation.

ACKNOWLEDGMENT

The author is most grateful to the anonymous referees for their constructive and helpful comments on the earlier version of the manuscript that helped to improve the presentation of the paper considerably. This research was supported by the foundation of science-technology development project of Weifang City of China under Grant No. 201001126.

REFERENCES

[1] J. Miller, and J. Mukerji. MDA Guide Version. http://www.omg.com/mda, 2010.

[2] J. Koehler, R. Hauser, S. Kapoor, F.Y. Wu, S. Kumaran, “A model-driven transformation method”. Proceedings of Seventh IEEE International conference on Enterprise Distributed Object Computing, IEEE Computer Society, 2007, pp. 186-197

[3] P. Cáceres, E. Marcos, B. Vela, “A MDA-Based Approach for Web Information System Development”. Proceedings of Software Model Engineering Workshop in UML'2010, IEEE Computer Society, 2010, pp. 86-97

[4] L. P. Fuentes, M. Vallecillo, “How MDA can help designing component and aspect-based applications”. Proceedings of Seventh IEEE International Conference on Enterprise Distributed Object Computing, IEEE Computer Society, 2009. pp. 124-135.

[5] J. Wan, B. SUN, “Interface Model to Support Automated Generation of User Interface”, Computer Engineering and Applications, Vol 39, No. 18, 2011, pp. 730-736.

[6] A. Kleppe, J. Warmer, W. Bast, MDA Explained, The Model Driven Architecture: Practice and Promise. Boston: Addison-Wesley, 2008

© the authors