Model-Based Whole Process Software Development

Jiazhen Hu

Jilin University, Jilin, Changchun

Jiazhenhu@tom.com

Keyword: Model-driven, service-oriented architecture, business modeling, automated code

Abstract: This paper uses the idea of model-driven development, taking a digital library as an example to model the whole process of the library management business process, and establishes a library system based on SOA architecture, which is automatically generated by code. The architecture is a typical the information system that includes the data layer, the functional service layer, and the business layer illustrates the model-driven software development process.

1. The main idea of model-driven development

Model-driven development is part of the Model-Driven Architecture (MDA). MDA represents a conceptual framework for model-driven development methods. It was proposed by OMG in 2001. Its core idea is abstraction and implementation technology. A core platform-independent model (PIM, Platform Independent Model) that completely describes business functions, and then develops multiple transformation rules for different implementation technologies. These transformation rules and auxiliary tools are used to transform PIM into platform-dependent models related to specific implementation technologies. (PSM, Platform Specific Model), and finally convert the enriched PSM into code. Through PIM and PSM, MDA aims to separate business modeling and underlying platform technologies to protect the results of modeling from technological changes.

2. The development process of digital library system based on MDD technology

2.1 System Challenges and Issues

The rapid development of information technology has led to fundamental changes in the library system. This change is reflected not only in the library management of the library, but also in the service for readers:

In the management of library management, the application and deepening of information technology in the library will glue together many of the original business links, the integration and integration of the library management, the extension of various management business processes, and the participation of business process processes. The increase in personnel, the classification of business is more complicated, and the requirements for library management are constantly improving. In the library management, the reader is not only the main body of the library service, but also the participant of the library management. For example, the subscription activity is purely input into the library management. Now the reader is also involved in the subscription activity, and the reader can directly decide the subscription information. kind of.

In terms of reader service, the development of knowledge management management technology has promoted the development of library services towards integration and knowledge. Previous borrowing services can only provide a single type of borrowing service for original materials. In the present, this simple service model has been met. The readers' expectations for library services, the application of automatic aggregation, automatic classification, and comprehensive analysis functions in libraries, thematic knowledge services, knowledge system services, and knowledge navigation services have become the main service modes of the library at this stage. Because of the data silos and application silos in the current library, the traditional technology makes the data

aggregation and mapping process time-consuming and error-prone, and creates tightly coupled dependencies on the infrastructure. In addition, applications and multiple data sources The point-to-point connection between them will increase management overhead. These problems will eventually lead to the library system facing great difficulties in implementing comprehensive knowledge services, affecting the reader's library knowledge acquisition experience.

2.2 Technical options

Combining the challenges and problems faced by the above-mentioned libraries, in this example we choose to use a combination of model-driven development models and SOA technologies to meet these two requirements. The general principles and ideas for system development include several key points:

Using model-driven development methods to model the full business process of library management, the starting point is to consider that a library management system has complex workflows and many stakeholders, and the needs of various readers change rapidly, suitable for model-driven development. Application scenario.

Follow the SOA development principles and strictly follow the SOA development process in the software service layer construction, including service identification, service modeling, interface design, contract design and SOA architecture design.

Unified modeling language, system modeling using SoaML (Service Oriented Architecture Modeling Language) modeling language;

Integrate data and data services under the guidance of SOA thinking to achieve layer-by-layer isolation, separate encapsulation and loose coupling of data, services and applications;

The code is automatically generated by the model, combined with manual coding, and the model and code are synchronized in both directions.

2.3 System function logic architecture

The functional layers are briefly described as follows:

The data layer, the main form is the relational database and the various data stored in the file system. The data is divided into three types of data, one is to guarantee the business process and its supporting data, and the other is the data, including documents, books, etc. Structured data, the last category is full-text electronic documents such as literature and e-books, which support the vast majority of digital library functions;

The aggregation layer is mainly used to aggregate data from databases, data from historical legacy systems, and data from other outsourced platforms to provide data services such as data aggregation, data analysis, technical topic clustering, and knowledge-based information provision. The core of the service. This layer isolates the difference between the data and each system, provides a consistent data access interface to the upper layer, and the upper layer Web Service calls the data provider of this layer.

The service layer (Web Services layer), which satisfies various data requests at the front end, and supports front-end library management and library service activities.

Presentation layer (Web layer), which provides access interfaces for various users of the system, that is, the system front-end, which can include the portal system, mobile phone application or as a proxy for other system call data. In this example, it is the interface of the library management. It is also the interface of library services.

3. System implementation process

3.1 Procurement Management Process Modeling

The main goal of establishing a business model is to provide some symbols that are easily understood by all business users. In this paper, the “procurement management” business model is the basic model of the software development method, which is the business model-service model-SOA engineering model development chain. Business model, so we first need to establish a

business process modeling annotation map to model the procurement process, the "purchasing management" business model can be seen as a pool, where the lane represents the role of the various functional departments in the process, and The task of each role. The process involves four roles, the administrator - responsible for the overall procurement of data, the supplier (bookSeller) - responsible for providing procurement lists and delivery of procurement materials, librarian - responsible for receiving information and logging in information.

3.2 Entity Relationship (E-R) Modeling

Based on the procurement management business model, you can identify the roles (various types of personnel) involved in the process, the entities such as process objects (orders, materials), and the activities involved in the process. These elements can be used to create ER models and create The ER diagram contains only some of the main entities and their relationships, as shown in Figure 1.

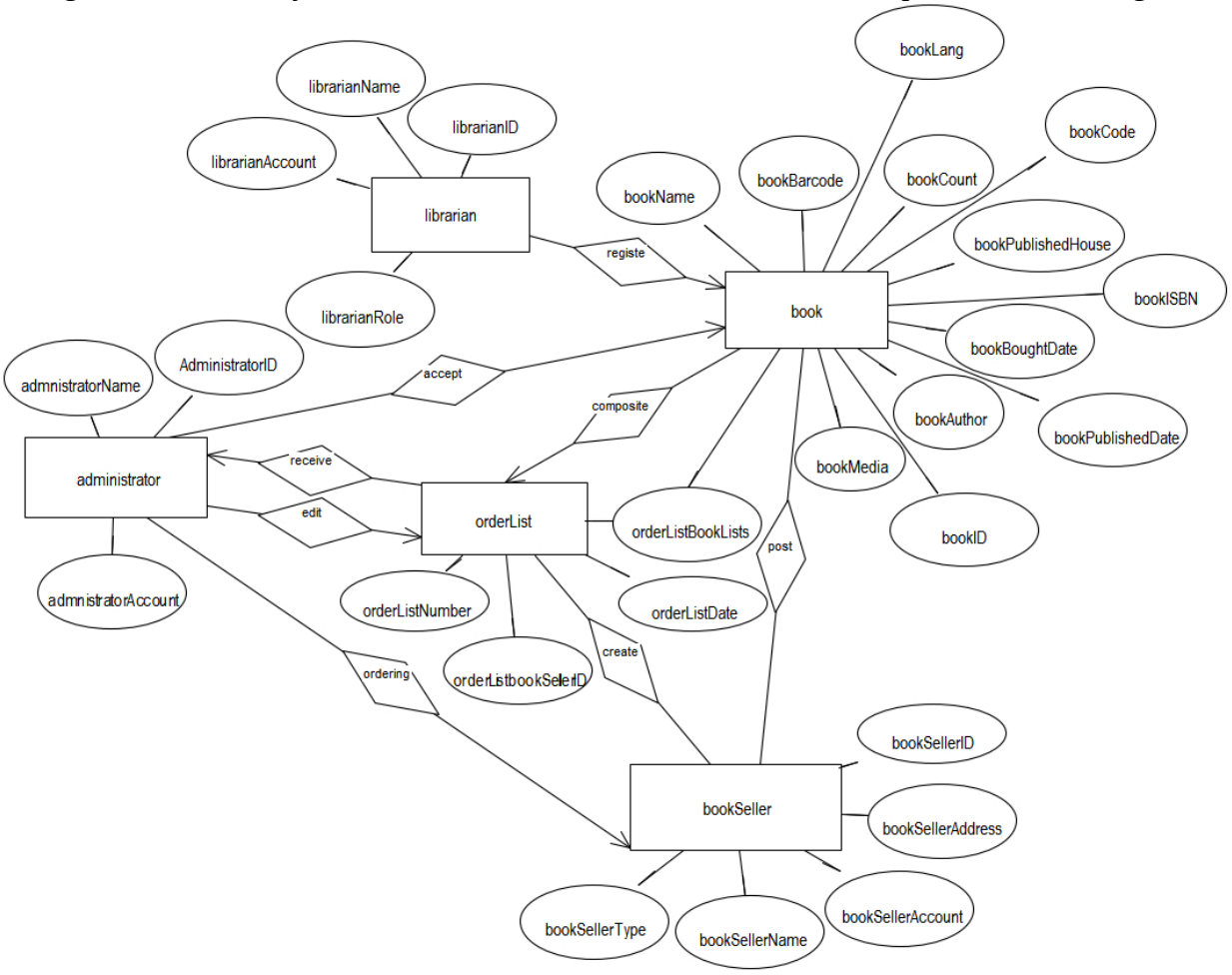

Figure 1 procurement management E-R diagram

3.3. Generating the database

After completing the procurement management E-R diagram, the system's underlying database creation script can be generated based on this diagram. This requires two steps. The first step is to convert the E-R diagram into a database model (as shown in Figure 2).

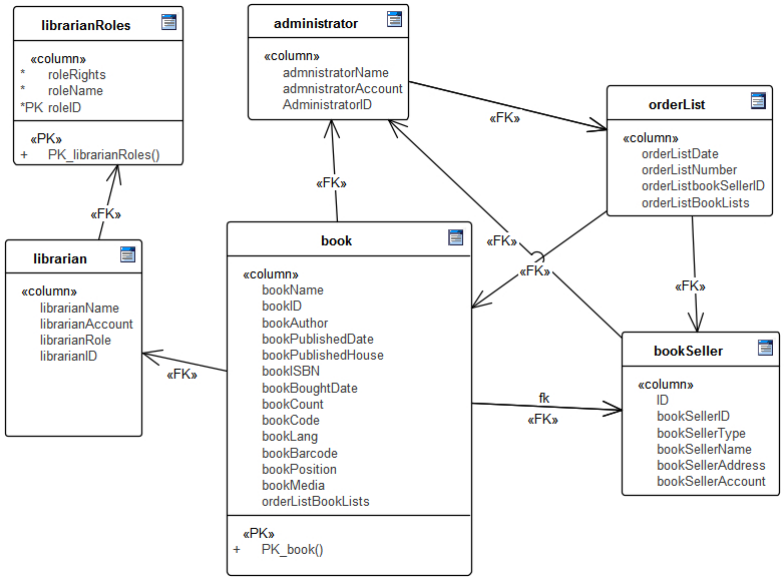

Figure 2 System Data Sheet

Secondly, based on this model, for the final database platform, the model is converted into a database operation script described by the data description language (DDL). Take SQL Server 2008 as an example, the script is as follows.

```
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('[book]') AND OBJECTPROPERTY(id, 'IsUserTable') = 1)
DROP TABLE [book]
CREATE TABLE [book]
(
[bookName],
[bookID] IDENTITY (1, 1),
[bookAuthor],
[bookPublishedDate],
[bookPublishedHouse],
[bookISBN],
[bookBoughtDate],
[bookCount],
[bookCode],
[bookLang],
[bookBarcode],
[bookPosition],
[bookMedia],
[orderListBookLists]
)
ALTER TABLE [book]
 ADD CONSTRAINT [PK_book]
PRIMARY KEY CLUSTERED ([bookID])
```

Then operate in the database management platform or terminal, run the generated SQL script, and generate the procurement management database in the database platform. After completing all the entity transformations in the E-R diagram, the corresponding script can be executed to create the main work of the data layer in the system architecture.

3.4 Front-end programming

In recent years, the development of front-end technology has been changing with each passing day. The functions that can only be completed by C/S can now be completed in B/S mode. The B/S

front-end has quickly become the mainstream platform of business systems. This trend has led to a sharp increase in front-end development workload. The expansion, coupled with the increasing complexity of the business, in this case the traditional manual coding method can not meet the actual needs, front-end developers urgent new means to improve work efficiency, and strengthen the code quality control of the entire project.

In response to this situation, there are many ways to improve coding efficiency in front-end development, such as web page templates, JavaScript package libraries for function packaging, front-end development frameworks, etc. In the design concept, MVC, MVVM, web components, etc. appear. These methods or design concepts are more or less helpful in improving coding efficiency. Manual coding still occupies a large workload, which leads to an increasing demand for automatic coding.

However, from the perspective of the development trend of technology, componentization is the general trend of front-end development. In large-scale software, componentization is a consensus that everyone has reached. On the one hand, componentization can greatly improve development efficiency. On the other hand, The consistency of component coding also reduces the maintenance cost to a large extent. On this basis, the front-end design is viewed from the perspective of MVC. The driver model of the front-end design should consider several aspects: First, the model should adapt to different front-end frameworks. For example, angular, React, and Vue, when designing the model, the configuration parameters of the model can satisfy the mapping of the model to the components of the front-end framework, and fully implement all the functions of the target component, such as asynchronous data update, inter-component communication, and application state management. Etc., and can be automatically converted to the code of the target framework; secondly, the model needs to adapt to different page layouts, such as bootstrap, easyUI, etc., through the configuration model can be converted to different layout modes, while considering the diversity of terminal devices, need to adopt responsive Front-end page and automatically generate object code;

4. Conclusion

Model-driven development technology provides a rapid application development model that overcomes the problems of traditional software development cycle, slow response to requirements, and large deviation between design goals and expectations. MDD technology can effectively improve software development efficiency. Guarantee the quality of development.

References

[1] Li Hongying. Design and implementation of model-driven Web form development tools [D]. Shandong University, 2015.

[2] Kang Jie. Research and design of procurement management system based on SoaML [D]. Fudan University, 2010.

[3] Zhao Xiaofeng. Research on Model-driven Software Development Model[J]. Information Technology and Informatization, 2015(07): 190-191.

[4] Zhang Yong. The application of rapid development platform in MDA and NET environment [J]. Electronic test, 2018 (20): 62-63+41.

[5] Peng Jun. Research on architecture metamodel and model transformation method in domain software development [D]. Jilin University, 2010.

[6] Teng Jianfa. Research and design of model-driven rapid business construction platform [D]. Shanghai Jiaotong University, 2015.