Research on Modeling of Graduate Training System Based on UML

Yan Li* and Jin Xu

College of Intelligent Science and Information Engineering, Xi’an Peihua University, Xi’an, China

*79132181@qq.com

Keywords: Unified Modeling Language; Graduate training system; Object-oriented; Modeling.

Abstract. In order to realize the remote submission of basic data of degree applicants, the examination of defense qualification of departments and postgraduate departments is networked, so as to improve work efficiency and adopt unified construction. Modeling language UML describes the whole process from function analysis, system design to physical model design. The complex postgraduate training system is simple and clear. Visual graphics representation provides favorable conditions for the development of the whole system and has practical significance.

Introduction

With the continuous development and improvement of China's education level, postgraduate training is also gradually expanding. At present, the application for postgraduate degree and qualification examination mainly adopt manual management mode, which cannot meet the current situation. It is undoubtedly an effective way to improve management by using scientific and technological means. Based on the Unified Modeling Language (UML), this paper elaborates the whole modeling process of the graduate training system based on UML, and analyses the relationship between the models, which provides the necessary guarantee for the realization of the system. By using object-oriented modeling technology based on UML to develop postgraduate training system, this system not only has good reliability, maintainability and scalability, but also can lay a good foundation for the reuse of the whole system, which has practical significance.

UML Overview

UML (Unified Modeling Language) is a general-purpose visual modeling language for software applications. It describes, visualizes, constructs and establishes the documents of the software system. It enables analysts to describe the processed products in a standardized graphical way. It records the decision and understanding of the system that must be constructed, and can be used to understand, design, browse, configure and control information about the system. UML includes semantics, representation and description of concepts. It provides models of static, dynamic, system environment and organizational structure. It is supported by interactive visual modeling tools. These tools provide code generators and report generators. They can use code generator tools to convert UML models into multiple programming language codes, or use reverse generation tools to convert program sources into codes. UML includes three kinds of views: functional view, static view and dynamic view.

1) Functional View
   (1) Use case diagram: describing system functions;
   (2) Activity diagram: Describe the activities involved in the workflow of executing the algorithm;
2) Static View
   (1) Class diagrams: describing entities in systems or domains and relationships between entities;
   (2) Object diagram: Describes the static structure of the system at a certain time, mostly for the design of specific examples;
   (3) Deployment diagram: Describes the configuration of environment elements, and maps the elements of the implementation system to the configuration.
3) Dynamic View
   (1) Cooperative Diagram (Communication Diagram): Describe the interaction between system elements and their relationship in time and space order;
   (2) State diagram: describe the interaction between system elements and the relationship between them in the order of time and space;
   (3) Activity diagram: Activity diagram is a deformation of state machine, in which the state is activity, indicating that the excitation and completion of operations are triggered by the completion of operations;
   (4) Sequence diagram: Represents the time sequence in which messages are transmitted between objects

From the application point of view, when using object-oriented technology to design a system, the first step is to describe the requirements, and the second step is to build the system according to the requirements. The third step is to describe the behavior of the system. The models established in the first and second steps are static, including use case diagrams, class diagrams, object diagrams, package diagrams, component diagrams and configuration diagrams, which are the static modeling mechanism of UML. Therefore, the main contents of UML can also be summarized into two categories: static modeling mechanism and dynamic modeling mechanism.

System Function Analysis and Modeling

This system uses RUP (Rational Unified Process) modeling method. RUP is proposed by the originator of UML, and can be well integrated with UML. RUP is architecture-centric, use-case-driven, iterative and incremental development-based. It can be static Structures and dynamic structures are described. Rational Rose98 is used as the modeling tool to analyze and design the system. The system is divided into several subsystems. Different subsystems are modeled with different models. The following is a detailed analysis of the modeling process.

Static Model of System

The static modeling mechanism of UML describes the static structure of the system through use case diagrams, class diagrams, object diagrams, package diagrams, component diagrams and configuration diagrams, and constructs the static model of the system.

Use Case Diagram. Use case diagrams focus on considering what functions the system needs to provide from the user's point of view. In use case diagrams, there are two entities: Actors and Use Cases. Therefore, the system use case diagram using UML should first identify the system users, divide them into different Actors according to their identities, and secondly divide the system use cases according to the system functions, and establish the system use case view.

Class Diagram. The system static model aggregates model elements into classes according to the requirement use case model diagram. Class diagram is the most widely used graph to describe the static structure of the proposed system at all levels. It is mainly used to describe the relationship among classes, interfaces and subsystems.

Class diagrams contain a set of classes, interfaces and their relationships, dependencies and generalizations. Class diagram not only establishes a visual and documented structure model for the system, but also establishes a forward and reverse engineering execution system.

Dynamic Model of System

On the basis of establishing the static model of the system, it is necessary to analyze and design the dynamic structure of the system, and establish the corresponding dynamic model. The dynamic model describes the time-varying behavior, which is described by the change of the instantaneous value of the system extracted from the static view.

The dynamic behavior model of the system is expressed by sequence diagrams, collaboration diagrams, state diagrams and activity diagrams, which are used to analyze the dynamic characteristics and behavior of the system. Sequence diagrams emphasize the time order of interaction, while collaboration diagrams emphasize the static link relationship between interactive objects.
Physical Model of System

**Component Diagram.** The construction and implementation of the system is also the stage of programming. First, the component view is established. Component is the realization of concepts and functions defined in logical architecture in physical architecture, which can include code base and running files. In component view, the actual structure of the system is emphasized. Using this block diagram, the person responsible for compiling and deploying the system knows which code base exists and which execution files are generated when compiling the code. Developers know which codebases exist and what relationships they have with each other. Component dependencies make the compiler aware of the compilation order required by the component.

**Deployment Chart.** The postgraduate training system should ensure that postgraduate managers can handle normal affairs, and also provide Internet services. Considering the problem of data security, the database server is divided into C/S server and B/S server. C/S servers are for internal managers and B/S servers are for browsers. When operating, both C/S server and B/S server request access to the database server.

Conclusions

Based on the application background of graduate training system development, this paper uses object-oriented modeling language UML to model its use cases. The analysis and establishment of model C, static model and dynamic model. The complex postgraduate training system is represented by simple and clear visual graphics, which provides a flexible, consistent and readable expression for the development of the whole system. It can also improve the reusability and maintainability of the system, and has a broad application prospect.

Acknowledgement

This research was financially supported by the The First Applied Demonstration Courses of Xi'an Peihua University.

References


