The Analysis and Design of the Object-oriented System

Li Xin [1, a]

[1] Shijiazhuang Vocational Technology Institute,Department of Electrical & Electronic Engineering, Hebei Shijiazhuang China 050081

[a] lixinsjzpt@163.com

Keywords: Object oriented; System analysis; Design.

Abstract. The object-oriented system analysis and design (OOA and OOD) is an effective technology widely used by the modern software enterprise, the OOAD method requires specified in the design to map the real world problem domain objects and entities. The object-oriented analysis is found in problem areas and describes objects (or concept). The object-oriented design emphasis is to define the software objects and how they collaborate to achieve requirements. The paper is after learning the object-oriented analysis and design, and according to own understanding to summarize this course. First of all this course is summarized; Then, in relation to the instructions of the object-oriented analysis and design, respectively. The summary is given.

Introduction

The database system (DBS) is systematically and dynamically stored a large number of related structured data, the convenient user use the database system of computer software and hardware resources. Since the 1950 s to the file system for the prototype to now has experienced six stages of development; Namely, the stage of file system of the late 1950 s; In the late 1960 s hierarchy and network database system; In the late 1970 s of the relational database system; Phase after the 1980 s of the various forms of database system (such as the distributed database system appeared in the 80 s and 80 s of the object-oriented database system), and based on the relational database system developed into the stage of object relational database system. The emergence of the object-oriented database system overcomes the traditional database system's flexibility is poor, poor expression ability and disadvantages poor scalability and become a journey in the database system development an important milestone.

The Overview of the Object-oriented Database

(1) The object-oriented data model

Just like other data model, the object-oriented data model also includes the data structure, data operation and integrity constraints three parts, but its performance in the three aspects of form and implementation approach and the traditional relation model has great difference. Object-oriented data structure through the object and class these two fundamental elements, with the aid of encapsulation, inheritance and combination mechanism to build a complex structure, and the data operation is done by method and message, object-oriented database systems support object-oriented data model (OODM), that is, object-oriented database system is a durable, can be Shared object library storage and management. OODM is to use object-oriented point of view to describe the real world entities (objects) logical organization, such as the restrictions, relationship between object model and object and the object id, class, class hierarchy, attributes, behavior, message, persistence, formed the basis of OODM such core concepts.

(2) The Characteristics and Advantage of the Object-oriented Database

Object oriented programming language to manipulate objects, so the object-oriented database (OODB) is one advantage of object-oriented program language programmers doing, can store data in the form of objects directly. Object data model has the following characteristics: 1) Using the object data model to organize by individual from the objective world by semantic complex system composed of interrelated object unit, object can be defined as attributes of the object and the object of behavior description, the relationship between objects can be divided into direct and indirect relations. 2) Semantic similar objects are organized into classes, the class is a collection of objects, the object is an instance of a class, by creating instances of the class implementation to access and manipulate objects. 3) The object data model with a "packaging", "inheritance", the basic concepts such as "polymorphic". 4) Method is similar to a relational database to realize the storage process, but a stored procedure is not associated with a particular object, methods are part of the class. 5) The practical applications, the object-oriented database can implement some describe the application of the system with complex data, such as temporal and spatial transaction, multimedia data management, etc. In addition, the above characteristics lead to object-oriented database has a wealth of data types, thus can express complex nested objects, and obtain good performance on complex object. Database and the programming language's type system and operation mode of unified, eliminate the impedance mismatch problem in the database, expand the application fields of database system, and improve the quality of the application system and developer productivity.

The Object-oriented Design Principles

When we develop software, often because of the design rule are not clear maintenance cost increases, so will the object-oriented design rule standardized, scientific, is a very important thing. Usually the object-oriented design criteria are:

(1) The Modular

Modularity is an important criterion of software design. In the object-oriented development method, the object is defined as a module. Object data structure and function in the operation on data encapsulate composition module. The object is the basic module of system.

(2) The abstract

Class is a kind of abstract data types, on the data type; you can create objects (a member of the class). Class contains the common properties of similar objects and services; it defines a common interface to make up the class specifications (agreement), for legitimate access to the outside world. Through this interface is to access data in a class instance. Often call this kind of abstract specifications abstraction.

(3) The information hiding

In the object-oriented method, the object is the properties and service encapsulation body, it's a way to achieve information hiding. Class structure separates the interface and implementation, the representation method of attributes of a class and operation, the implementation of the algorithm for a class of users, should be hidden, users can only through a common interface to access attributes in the class.

(4) The weak coupling

Coupling refers to the interconnection between a software structures in different modules, dependency between the stronger the coupling, the more the less dependency on the coupling. In the object-oriented method, the object is the most basic module, correlated dependencies between different objects expressed coupling. Is an important measure of well-designed weak coupling and weak coupling of an object in the design of the change will never or rarely affect other objects. So to understand, test, or modify bring great convenience. On the other hand, the strong coupling will bring understanding, test or modify a lot of difficulty, and also reduces the reusability and portability of the class. Two objects must contact each other; each other should be accomplished by agreement (i.e., public interface) of class two objects depend on each other (coupling), rather than through the class described in the specific implementation details.

(5) The strong cohesion

So-called cohesion, it is a module within each element combined with the close degree of each other. The more the more closely the cohesion, in combination with the more don't close the cohesion and the weaker. Strong cohesion is an important measure of excellent design. In object-oriented design, cohesion can be divided into the following three categories: 1) service cohesion. A service should be a single, that is, only to complete a task. 2) the class cohesion. Class cohesion requirements class attributes and services should be high cohesion, and they should be necessary for system task. A class should have only one function, if a class has multiple functions, usually should be more than breaking it down into special classes. 3) general - special cohesion. General - special cohesive said: general - special structure accord with domain knowledge representation, that is to say, the special classes should be as far as possible to inherit the generic attributes and services. The general - special structure are highly cohesive.

(6) The reusable

In object-oriented design, the design of a class should have generality, for the development of similar systems may provide software reuse. Software reuse can improve software development productivity; ensure the quality of the target system. Reuse is an important feature of object-oriented development method, that is, using the object-oriented concept and method is easier to reuse. Therefore, in the process of software development, in order to achieve the reuse, try to reuse existing classes, and to create a reusable class.

The Object-oriented System Analysis

The key to object-oriented analysis is to identify a problem in the domain object, and analyzes the relationship between them; finally establish the problem domain is concise, accurate and understandable model correctly. Object-oriented analysis is needed by modeling of software system user domain model, need to understand is the demand of the software system concepts and terms, its analysis of the content is in the real world entity objects, and the relationship between each object, does not involve programming concepts.

(1) use case diagram (model) of UML modeling language, use case diagram is used to describe the relationship between the interaction between the user and the system, system of business capabilities and business processes, can facilitate the developers understand the user in the field of specialized terms and business content. Participation (Actor): system related to the system of class, interacts with the system of people, machines, or other system, used to reflect the system relationship with the surrounding environment. Divided into communication relationship, the relationship between the uses and extend. Communication Relationship (Communicates Relationship): describes the Relationship between participation and cases, which involve a single person can contact with multiple use cases, on the contrary, a use case can also contact involved in multiple, clearly describes the "who uses which use cases", the Relationship between the use case diagram cases. Extension Relationship (Extends Relationship): when a basic use cases by the need to attach a use case to expand or extending its original function, the expansion of the additional use cases and the Relationship between the original use cases is to extension. Extended use case can inherit the original cases some of the basic functions, at the same time it can have some new special function.

(3) The auxiliary model is set up

Sequence diagram: a kind of interaction diagram, describe the dynamic relationship between the object and the process of cooperation. Sequence diagram of the order, commonly used to describe the behavior of a use case.

Communication: an emphasis on sending and receiving message object structure diagram, to show around the object and the connector between them and interaction of the organization.

Activity diagrams: description of the operation implementation completed work, and use case instance or object in the activity, the activity diagram is a variant on the state diagram.

State diagram: describe the life cycle of an object, subsystem, system.

Package diagram: a grouping of model elements and diagram of dependencies between teams of package is required for the mechanism of model elements into groups.

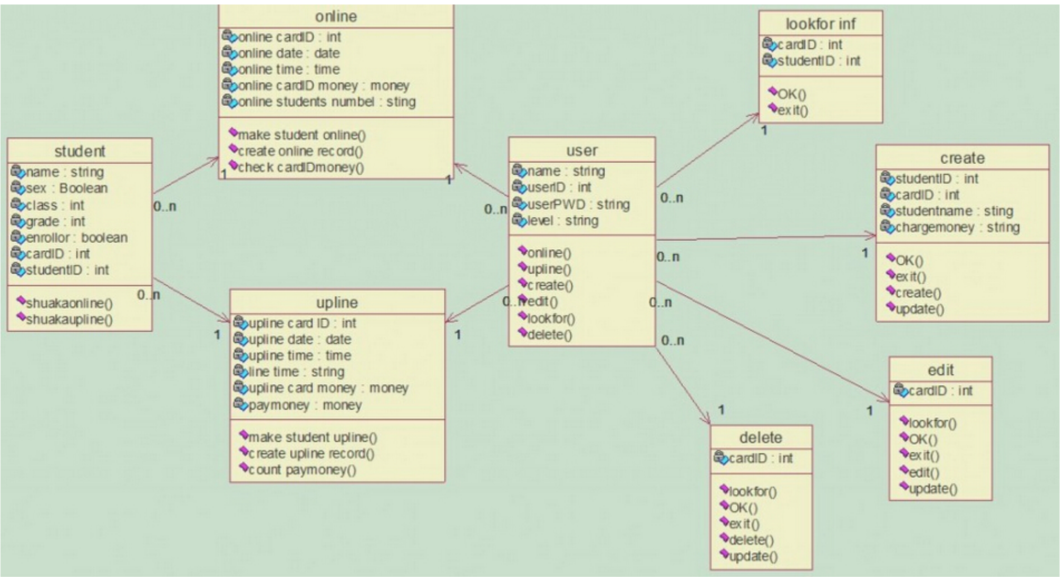

Figure 1 the Analysis of the Object-oriented System

The Object-oriented System Designed

Design modeling need to expand the results of the analysis phase into a technical solution, need is the technology of software system structural model is established. OOA and OOD relations: from OOA to OOD not transformation, but the adjustment and supplement, supplementation of human-computer interaction, building and subordinate part, control drive parts, data management.

(1) The problem domain part of the design

The problem domain part of the design is to the results of OOA supplement and adjustment according to the conditions. You want to continue using the method of OOA, including concepts, notation and part of the strategy. Not only to realize condition based on OOD design, and due to demand change or newly discovered the mistake, also want to modify the results of OOA.

(2) The human-computer interaction part of the design

The human-computer interaction part as an independent component in the system, to carry on the analysis and design, is advantageous to the isolation interface support system change on the part of the problem domain.

(3) The control drive part of the design

Concurrent behavior, in order to describe the problem domain inherent expression to achieve the required design decisions that need to be in OOD part to control the drive part modeling. Control flow driven part, is used to define and represent each control flow of the concurrent systems. Using active object represents each control flow (process, thread) all active classes constitute control flow driving part.

(4) The data management part of the design

Data management section is responsible for the specific data part of the management system to store and retrieve the object. Its purpose is, persistent object storage problem domain, encapsulate these objects searching and storage mechanism, and in order to isolate the influence of data management solutions. In the traditional design method of structured and is easy to map the entities a diagram to relation database. In object-oriented design, we can see a UML class diagram as the concept of database model, but in the UML class diagram in addition to the relationship between classes, and inheritance relationships.

(5) The component and design of the deployment

The current a mainstream practice is in the later stages of the object-oriented system design stage, consider how to carry on the description of system components, structure and organization, and how components on the node distribution.

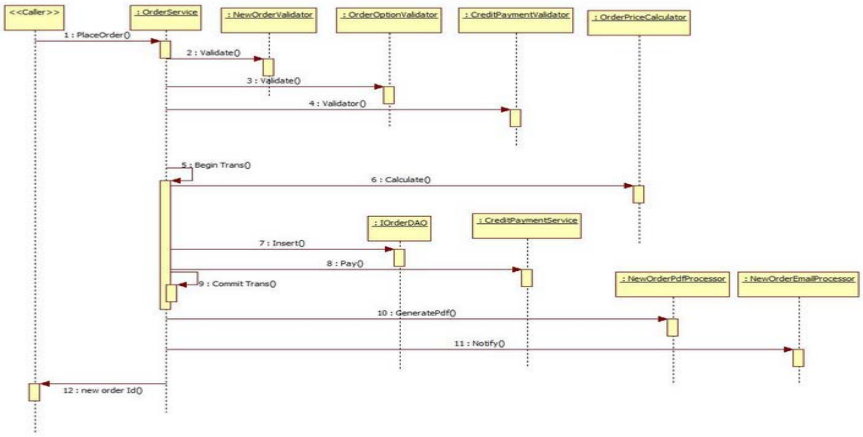

Figure 2 The Design of the Object-oriented System

Summary

Throughout the development process of the database system, the emergence of the object-oriented database system undoubtedly injected a new vigor for the database research and application fields. Although the emergence of the object-oriented database system overcomes the traditional database system's flexibility is poor, poor expression ability and poor scalability of many shortcomings, also have a relational database for the use of more convenient usability and broader community, but it is in terms of performance, model modification and standardization still exists some problems that don’t allow to ignore. Object-oriented database development, meanwhile, also does not mean to completely replace the relational database, relational database, after all, already has been widely used in the field of global each big, skilled master by most database users, therefore, the database developer in the continuous improvement of object-oriented database at the same time, will also be a relational database with object-oriented database, in order to achieve complementary advantages so as to realize the win-win purpose.

References

[1] Dipti Prasad Mukherjee,Yury Potapovich,Ilya Levner,Hong Zhang. Ore image segmentation by learning image and shape features[J]. Pattern Recognition Letters. 2011(6) 19-28.

[2] Geoffrey J. Hay,Thomas Blaschke, Danielle J. Marceau,André Bouchard. A comparison of three image-object methods for the multiscale analysis of landscape structure[J]. ISPRS Journal of Photogrammetry and Remote Sensing. 2012(5)12-18.

[3] Hisao Ishibuchi,Tomoharu Nakashima,Tadahiko Murata. Three-objective genetics-based machine learning for linguistic rule extraction[J]. Information Sciences. 2013 (1) 2-8.

[4] A. B. Miller,E. S. Bryant,R. W. Birnie. An analysis of land cover changes in the Northern Forest of New England using multitemporal Landsat MSS data[J]. International Journal of Remote Sensing. 2012 (2) 21-24.

[5] C. Clark,A. Canas. Spectral identification by artificial neural network and genetic algorithm[J]. International Journal of Remote Sensing. 2013(12) 87-88.

[6] Azriel Rosenfeld. Picture Processing by Computer[J]. ACM Computing Surveys (CSUR). 2012(3) 89-92.