# System Design and Development Based on Object Oriented

Deng Tian

NanChang Institute of Science & Technology

**Abstract-The general concept of object oriented does not guarantee the correctness, completeness and consistency of system design. It is more important to apply the object oriented method. The software system analysis and design based on computer software is a technical difficulty, which is the intersection of system engineering and software engineering, which has a decisive influence on the whole system development. An under design system will cause the system to develop a lack of sufficient basis for each subsystem, need a large number of supplementary design and change design, waste a lot of manpower and material resources and other serious problems. In this paper, we will propose a method of object oriented method for system analysis and design.**

***Keywords-Software Engineering, System Design, Object Oriented***

## I. INTRODUCTION

With the wide application of computer technology, the complexity of the application software is becoming more and more complex. How to design developing software more reasonably and manage the process of developing software more scientifically have already become a hot topic with wide concern. Object oriented software development technology came into being.

Traditional structured design is the focus of the software design on the function of the realization of the process and the process of control, once the needs of the software changes or system improvement, software maintenance becomes difficult. As the name implies, the core idea of object-oriented software design is to design the software as a collection of abstract concepts in the objective world. These abstract concepts define the attributes and behaviors of the entities, and the operation of the software is the simulation of the objective business operations. As a result, the software of the transplant, modification and maintenance have become easy to a lot of software engineering mainly about the principle of software development. The goal of software engineering is to improve the quality and productivity of software, and to realize the industrialization production of software. There is no fundamental contradiction between quality and productivity, and good software engineering method can improve the quality and productivity.

In order to understand the structure of object oriented software, the paper puts forward the idea of structure of object oriented software and the method of software understanding based on above structure. In addition, according to the requirements of research, this paper establishes a network model of object oriented software and designs and implements a software analysis tool based on object oriented software structure.

## II. OBJECT ORIENTED PROGRAMMING TECHNIQUE

In simple terms, the object-oriented programming technology is the use of object-oriented technology for programming technology, and object-oriented technology is a system modeling technology, it provides a lot of the concept of system modeling. Based on object-oriented technology, we build a system through many objects and their relationships. Regardless of the type of system, we all believe that they are composed of multiple objects and their relations, and the system contains what type of object and the relationship between them is determined by the system to solve the problem. Due to the reduction of the semantic difference between the system model and the real environment, it is easier to understand and apply the system constructed by the object oriented technology.

Object oriented technology can directly correspond to the model and the problem to be solved, making it easier to understand the model constructed. An object represents an entity with a specific behavior and property, and any access to this object can only be executed by its public interface. Each object is an instance of a class, which describes the behavior and properties of the instance. The introduction of polymorphism adds flexibility to the definition of class, since an object does not need to know the exact class when it is in contact with another object, so it can add new classes without changing other classes. By inheritance, we can define a new class based on a class that has already existed.

## III. IDEOLOGY OF SOFTWARE ENGINEERING

### A. *Concept of Software Engineering*

Software engineering is the application of computer science, mathematics and management science, and other principles, learn from the traditional engineering principles, methods, development of software engineering. A definition of the current comparison is that software engineering is a research and application of a systematic, standardized, quantitative approach to the development and maintenance of software, and how to combine the test of time to prove that the right management techniques and the best technical methods currently available.

### B. *Contents of Research into Software Engineering*

The content of software engineering research includes software development method, software development model, software support process and software management process. The content of the software development method covers the whole process of market research, demand analysis, outline design, detailed design, programming, testing and quality control. Mainstream of the process oriented method, object oriented method, data oriented method, formal method, each is suitable for different occasions, each have advantages and disadvantages, constitute diversified development theory. Software development model refers to the structure of the whole process, activity and task of the software development. At present, the development model has the waterfall model, the iterative model, the incremental model and the prototype model. Software support process is mainly composed of the production tools, such as requirement analysis tools, design tools, tools, tools, maintenance tools, configuration tools, development environment and so on. Common CASE tools such as Designer Rational, Rose Power. Software management process is to develop software development, operation and maintenance of the relevant process activities and the process of supervision, the current more mature software management process mainly CMM, ISO9000, Microsoft corporate culture and the rapid cultural phenomenon.

### *C. Process of Designing and Developing Software Engineering*

The principle of software engineering refers to the principles that must be followed in the process of engineering design, engineering support and project management. The following four main items are as follows: select the appropriate development model, adopt appropriate design method, provide high quality engineering support, and pay attention to the management of the development process.

Software development model based on incremental model of incremental construction model, the software life cycle is divided six of basic activities: planning, needs analysis, system design, programming, software testing, operation and maintenance, in demand analysis and software design stage application of organ distribution model for the overall development, behind the stage with the teaching plan of the first, according to the constructing model of incremental development, as shown in Figure 1.

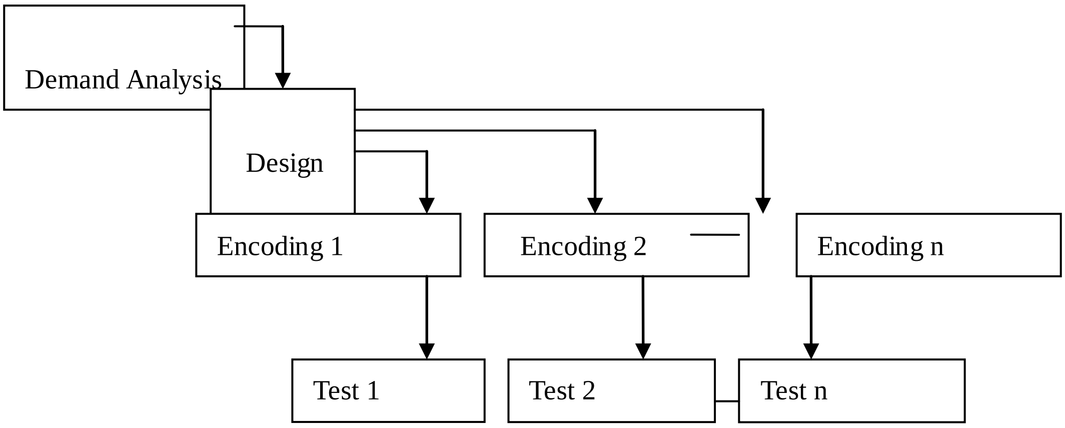

Fig 1. Incremental Structure Model

## IV. Analysis of Object-Oriented System

From the perspective of system analysis, regard system elements in the “computer-based system” as the objects, and the system elements can be divided as follows:

| System Element | Classification of System Elements and Objects | Notes |
|---|---|---|
| Hardware | Single Equipment (HWCI) | Internal or system-related equipment |
| | Sub-system (SUBSYS) | System internal or system directly related to the analysis of the design stage of the internal system analysis |
| Software | Independent Software (CSCI) | A separate installation of software configuration items that are directly related to the system or system |
| Database | DBASE）<br>Database (DBASE) | A database containing a data table that is directly related to the system or system |

|  |  |  |
|---|---|---|
|  | Database Table (TABLE) | Data tables in a database that are directly related to the system's internal or system |
| People | Operator (HUMAN) | The personnel or personnel involved in the operation of the system or system |

To facilitate the reading recognition and automatic extraction tools to identify, establish corresponding to these elements in the ROSE, the object's icon and version.

## V. Object Oriented System Design

Interface design and database design are two main contents in system design. Interface design corresponds with the interface demands while interface design also contains interface design and man-machine interface.

### *A. Interface Design*

The device interface design includes a physical interface, interface protocol, and interface data format for each interface message. In order to facilitate the naming, management and sharing of interface data, we use the "interface data object", an interface data object contains all data items and data format required for a message. In the design, try to make more interface information contain the same interface data object.

### *B. Man-Machine Interface Design*

The main task of the design of human computer interface is to give the interface of human machine interface or the requirements of the interface, which need to be clear about the type of the output data, the display mode and the validity of the interface.

Every man machine interface message can only be related to the design of a hardware element or a software element, so the design of the man-machine interface can be performed in the subsystem design or not.

Database design according to the system model of all system elements object interaction diagram for each DBASE version and TABLE version of the data access requirements, the design of the database objects and data access rules. Using ROSE tools to support the database modeling, we can directly generate the actual database in ORACLE and other database management system.

## VI. Conclusion

To sum up and through practice, we feel object-oriented method in the system analysis design and application can reflect the irreplaceable superiority. As a whole application software system, I carefully analyzed from user needs analysis to outline design, detailed design, program encoding, as well as post user training, on-site commissioning work and a series of issues. In this paper, some key issues in the design are discussed and the solutions of the system are given, which are helpful to solve the problems of the same or similar problems in other systems.

Of course, the design of the system, its design is not static, but with the development of various technologies, the use of a complete object-oriented database will be the future direction of development. So it is needed to make further research on the information system based on object oriented database. Similarly, with the development of network technology, all kinds of multimedia data will need to appear in the system design, which need to be further studied.

## References

[1] Wohlin C, Runeson P, Höst M, et al. Experimentation in software engineering : an introduction[J]. Kluwer International, 2000, 9(2):171-180.

[2] Chung L, Nixon B A, Yu E, et al. Non-functional requirements in software engineering[J]. Springer Berlin, 1999:1634 - 1638.

[3] Boehm B W. Software Engineering Economics[J]. Pioneers & Their Contributions to Software Engineering, 1984, se-10(1):4 - 21.

[4] Brooks F P. No Silver Bullet Essence and Accidents of Software Engineering[J]. Computer, 1987, 20(4):10-19.

[5] Humphrey W S. A Discipline for Software Engineering[J]. Sei, 1995, 30(3):1-3.

[6] Turski W M. No Silver Bullet - Essence and Accidents of Software Engineering - Response.[C]// World Computer Congress. 1986:1077-1080.

[7] Briand L C, Morasca S, Basili V R. Property-based software engineering measurement[J]. Software Engineering IEEE Transactions on, 1996, 22(1):68-86.

[8] Beyer D, Henzinger T A, Jhala R, et al. The software model checker B last : Applications to software engineering[J]. International Journal on Software Tools for Technology Transfer, 2007, 9(5):505-525.

[9] Chung L. On Non-Functional Requirements in Software Engineering[J]. Lecture Notes in Computer Science, 2009, 5600:363-379.

[10] Pressman R S. Software Engineering: a Praclitioner's Approach[J]. McGraw-Hill computer science series, 1992, 8(2):45-55.