Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2020.010101
ISSN 2790-0916 Vol. 1, Issue 1: 1-8

UML Language in Distributed System Modeling

Tanyi William*

Griffith Base Hosp, Murrumbidgee Local Hlth Dist, 1 Noorebar Ave, Griffith, NSW 2680, Australia

**corresponding author*

Keywords: UML Language, Distributed System, System Modeling, Examination System

Abstract: System modeling is widely used in all aspects of modern life. In today's Internet era, complex distributed systems play an important role in modern production. The purpose of this paper is to study the application of UML language in distributed system modeling. Firstly, it discusses the importance of distributed system software model from a scientific point of view, considers the definition of distributed system, discusses the theoretical basis and development requirements of distributed test system development, and puts forward a scientific model. Comprehensive application of unified modeling language UML, combined with CORBA distributed computing technology to build a distributed test system modeling. Analyze system components and extended views, and implement a grade management module. It can be concluded that the modeling method in this paper is an effective method, which provides a reference for the development of the distributed examination system.

1. Introduction

With the development of object-oriented technology, many object-based modeling languages appeared when analyzing and designing with object-based methods. Users of object-based methods do not know the advantages and disadvantages of different modeling languages, so it is difficult to choose a suitable modeling language according to the characteristics of the application [1-2]. In addition, the differences between different modeling languages also hinder the communication between users who use different modeling languages. So far, with the integration of systems represented by engineering software masters and the further development of modeling languages, UML (Unified Modeling Language) has become the technical standard on which object-based technologies depend [3-4]. In this case, the traditional operation mode of colleges and universities relies on operation. In this case, developing the corresponding college information management system according to the fixed financial process obviously cannot meet the needs of modern college information [5].

Using modeling languages requires proper application support, and modeling tools are immature.

With the release of UML, tool makers are developing high-quality modern modeling tools that support UML [6]. Amel M proposed the formal verification of embedded systems. The approach relies on automatic modeling and code generation based on system behavior. The key concept is the combined use of a subset of UML behavioral diagrams extended with timing annotations (real-time state diagrams and real-time collaboration diagrams) for system modeling and verification using the Maude language. First, UML modeling tools were developed. Then, perform automatic generation of the equivalent Maude specification. The method is based on code generation. This is why certain timing properties represented in linear temporal logic (LTL) can be verified using the available model checking tools [7]. Puspita D used Unified Modeling Language (UML) as a web design tool to produce a web-based presentation system for small and medium enterprises (SMEs) in Pagar Alam City. For SMEs in Baganan City, the introduction is still manual, i.e. tourists have to go directly to the Ministry of Industry, Trade, Cooperatives and Small and Medium Enterprises in Baga Anan City to find information and view the products sold by SME enterprises [8]. Combined with the new aspect positioning technology, a new modeling method is proposed, which has important practical application value.

The following work is done in this paper: The basic concepts and basic concepts of model-oriented language technology are systematically introduced, and the model-oriented research scheme is comprehensively introduced. The characteristics, characteristics and modeling requirements of distributed systems are analyzed and discussed. Learn UML and its dynamic characteristics; describe the modeling method that combines the concepts and methods of UML language with CORBA distributed computing technology, and give a typical application example.

2. Research on UML Language in Distributed System Modeling

2.1. UML Language

UML is a general-purpose modeling language rather than a highly structured language that can be used for theoretical modeling [9-10]. UML can automatically add this change to the system model, keeping the model and code in sync [11-12].

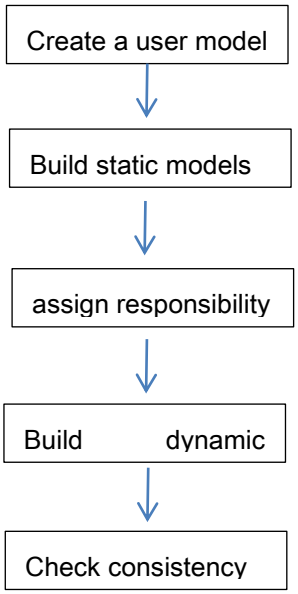

Figure 1. UML modeling process

(1) Functional model: represented by an example diagram.

(2) Static model: The UML modeling process is shown in Figure 1, which describes the function of the system from the user's point of view, that is, the system "should do", and is used to record the static structure of the system, that is, the work described in the functional model, mainly Class descriptions (or relationships, properties, and functions between objects) are represented by class diagrams and objects [13].

(3) Dynamic models: describe the temporal, behavioral, and management characteristics of a problem, that is, describe how and when the activities described in the above models are accomplished, including the sequence, location, and functional things of events [14].

2.2. Platforms that Currently Support Server-side Middleware Technology

CORBA distributed computing technology is a common speech representation system specification based on shared speech interoperability content submitted by many open system platform manufacturers. It has the characteristics of complete model, progress, programming platform and development language independence, and extensive support [15-16].

The technical characteristics of the CORBA specification: (1) The concept of Broke is introduced. (2) The complete separation of the client system and the server system is realized. (3) Combine distributed computing and object-based concepts to improve software reusability and redundancy management. (4) It provides a software bus, that is, defines a set of specifications, any application, software program or application can be easily integrated into the CORBA system, as long as it has an interface definition specification corresponding to watch, its The interface specification is independent of any language and implementation environment [17].

2.3. System Requirements

Data analysis, manipulation, and processing (other than select SQL) in data warehouse applications is often complex and time-consuming. In the C/S mode, the computation is usually performed on the client or server database, which requires high hardware investment [18]. Also, as the client processing load increases. Limit the number of client connections the server can support. At the same time, increasing the number of client connections also increases the system's response time to client requests. In the three-tier architecture, data analysis and processing are performed on dedicated application servers, reducing the load on data servers. Application servers can balance load, improve resource utilization, and reduce server load.

3. Investigation and Research of UML Language in Distributed System Modeling

3.1. Describe System Functions in UML Language

The whole system is designed as a CORBA application system, and the modeling process of the system from design, realization to deployment is completed by using various views of UML. During the implementation of the system, the software modeling method of the distributed examination system is adopted, and RationalRose2003 Enterprise Edition is selected as the UML modeling tool. Since the development system has the general functions and characteristics of the distributed examination system, the models of each stage of the distributed examination system can basically be reused. Only the demand model for the university's distributed examination system is given here, as shown in Figure 2.

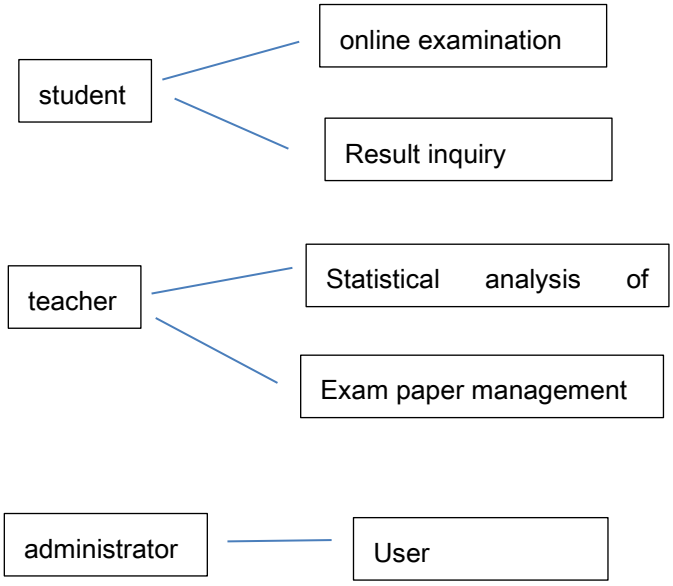

Figure 2. Requirement model of distributed examination system

3.2. Network Question Bank Construction

Classical measurement theory is based on the following three assumptions:
(1) The measurement error obeys a normal distribution, and its mean is zero.
(2) The error and the true score are independent of each other, and the correlation is zero.
(3) Within the range in question, the individual true score remains unchanged.

The first conclusion that can be drawn from the above assumptions is that the average of several tests is close to the true score. Basic assumptions, reliability and validity are known as the three pillars of classical measurement theory, and factor analysis and scaling are its methods. The main research object of project analysis is the parameters of the test instrument, such as difficulty, discrimination, etc. On this basis, the classical metrology theory developed a set of quality index calculation formulas, improved the metrology calibration process, made the whole metrology process more effective and scientific, and had great practical value.

Paper difficulty refers to the level of difficulty of the paper. One of the most important aspects of written test quality control is the degree of difficulty, which is also an important basis for selecting test questions. The calculation formula is shown below.

$$P = 1 - \frac{\overline{X}}{N} \tag{1}$$

P is the difficulty coefficient of the test paper, x is the average score of all test takers, and N is the total score of the test paper. In the above formula, the value range of P is between [0-1]. When P continues to increase, the difficulty of the test paper will become more and more difficult; on the contrary, if P becomes smaller and smaller, the difficulty of the test paper will become more and more difficult. getting smaller.

Validity is used to express the degree of consistency between the results of the test and the expected objectives of the test, usually using the person correlation coefficient to calculate the

validity of the test paper. Its calculation formula is as follows.

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{y})}{ns_x s_y} \quad (2)$$

Among them, R represents validity, X; represents the score obtained by candidate i in the x test paper, Y; represents the score obtained by the i candidate in the y test paper, Sx and Sy represent the standard deviation of the two scores respectively, and n represents The total number of students taking the exam.

4. Research and Application of UML Language in Distributed System Modeling

4.1. Components

UML models can describe logical analysis and physical implementation, and physical views provide a way to separate classes into elements. UML defines a component as a system element that conforms to a set of interfaces and provides a physical, physical replacement system. The component diagram of the implementation phase of the distributed examination system is a representation of the physical implementation of the system. Take the GoodsCarORB object as an example, it is invisible to the user, and all methods called by the client are implemented by the CORBA object. Therefore, there is an integration between CORBA objects and GoodsCarORB objects. In this system, GoodsCar and GoodsCarORB are encapsulated into GoodsCar components, which are components at the realization level under the UML concept.

4.2. Expand the View

Most of the computing tasks and system data in the CORBA system exist on the server side in the form of CORBA components. The expanded view of the CORBA system is the expanded view of the client-server approach. The expanded view of the client and server in the examination system is shown in Figure 3.

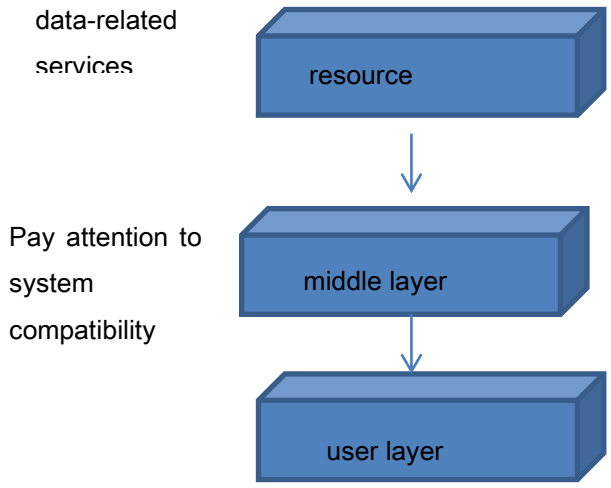

Figure 3. Expanded view of the client, server in the exam system

4.3. Implementation of the Achievement Management Module

Candidates can simply inquire about personal test subjects and scores at the front desk, but candidates only have the right to inquire, and do not have the right to do other operations on the test scores, such as modifying, adding, deleting, etc. Teachers can manage candidates' scores by logging into the background. While adding and modifying students' scores, they can also make more detailed queries on candidates' scores according to different conditions. As shown in Table 1, some candidates' scores in the background are ranked. As shown in Figure 4.

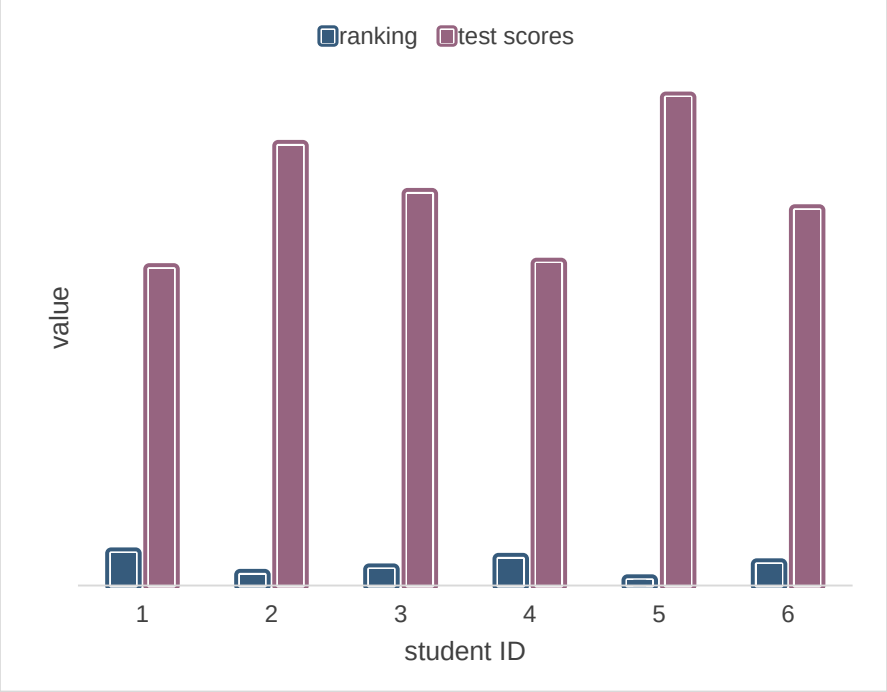

Figure 4. Ranking of some grades

Table 1. Comparison of scores of some candidates

Student ID	Ranking	Test scores	Grade type
001	6	59	Failed
002	2	82	Good
003	3	73	Good
004	5	60	Pass
005	1	91	Excellent
006	4	70	Good

Other functional modules in the system adopt similar implementation methods, such as test paper management module, test question bank management module, article online reading module and so on.

By applying the software modeling method of the distributed examination system and the constructed software model to the development process of the distributed examination system for universities, it can be concluded that the software modeling method of the distributed examination system is basically correct, and the model built in the software development process is basically effective., can play a certain guiding role in the development of distributed examination system.

5. Conclusion

Concepts and methods for distributed exam systems have been created but not finalized, and some programming languages are not yet mature, so corresponding modeling methods should evolve as various aspects of the programming language section evolve. Therefore, the use of UML and CORBA frameworks with real-time system modeling requires more in-depth study and discussion. In this paper, different aspects of the techniques are covered and a general UML modeling approach is presented. The actual distributed system is very complex, many fields are not involved, and there are interactions between various fields. The design method still needs to be improved, and there is still a lot of work to be done in the next step: improving the compatibility of UML and CORBA. Allows for multidimensional separation, modeling and programming.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Setiaji S, Sastra R. Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Penggajian. Jurnal Teknik Komputer, 2020, 7(1):106-111. https://doi.org/10.31294/jtk.v7i1.9773

[2] Pakaya R, Tapate A R, Suleman S. Perancangan Aplikasi Penjualan Hewan Ternak Untuk Qurban Dan Aqiqah Dengan Metode Unified Modeling Language (UML). Jurnal Technopreneur (JTech), 2020, 8(1):31-40. https://doi.org/10.30869/jtech.v8i1.531

[3] Mubarak A. Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek. Jiko (Jurnal Informatika dan Komputer), 2019, 2(1):19-25. https://doi.org/10.33387/jiko.v2i1.1052

[4] Sunitha E V, Samuel P. Object constraint language for code generation from activity models. Information and software technology, 2018, 103(NOV.):págs. 92-111. https://doi.org/10.1016/j.infsof.2018.06.010

[5] Tedyyana A, Ratnawati F, Kurniati R. Rancangan Sistem Informasi Penelitian Dan Pengabdian Masyarakat Politeknik Negeri Bengkalis Menggunakan Metode Uml (Unified Modeling Language). Sistemasi, 2019, 8(3):413. https://doi.org/10.32520/stmsi.v8i3.535

[6] Kalnins A, Barzdins J. Metamodel specialization for graphical language support. Software & Systems Modeling, 2019, 18(3):1699-1735. https://doi.org/10.1007/s10270-018-0668-3

[7] Amel M, Allaoua C. Embedded System Verification Using Formal Model an Approach Based on the Combined Use of UML and Maude Language. International Journal of Conceptual

Structures and Smart Applications, 2018, 6(2):42-58.

[8] Puspita D, Anggita M. Penerapan Unified Modeling Language (Uml) Dalam Membangun Sistem Pengenalan Usaha Kecil Menengah (Ukm) Kota Pagaralam. Jusim (Jurnal Sistem Informasi Musirawas), 2020, 5(2):103-110. https://doi.org/10.32767/jusim.v5i02.805

[9] Farias K, Oliveira T, Gonales L J, et al. UML2Merge: a UML extension for model merging. IET Software, 2019, 13(6):575-586. https://doi.org/10.32767/jusim.v5i02.805

[10] Ahmad T, Iqbal J, Ashraf A, et al. Model-based testing using UML activity diagrams: A systematic mapping study. Computer Science Review, 2019, 33(AUG.):98-112. https://doi.org/10.1016/j.cosrev.2019.07.001

[11] Liantoni F, Yusincha A. Pemodelan UML Pada Sistem Pengajuan Dana Anggaran Untuk Peningkatan Produktivitas Perusahaaan. Digital Zone Jurnal Teknologi Informasi dan Komunikasi, 2018, 9(2):94-105. https://doi.org/10.31849/digitalzone.v9i2.1763

[12] Bhutto A, Hussain D. Validate UML model and OCL expressions using USE Tool. Pertanika Journal of Science and Technology, 2018, 26(3):1465-1480.

[13] Skersys T, Danenas P, Butleris R. Extracting SBVR Business Vocabularies and Business Rules from UML Use Case Diagrams. Journal of Systems and Software, 2018, 141(JUL.):111–130. https://doi.org/10.1016/j.jss.2018.03.061

[14] H Gábor, M István, P András, et al. Quantitative Analysis of UML Statechart Models of Dependable Systems. The Computer Journal, 2018, 45(3):260-277. https://doi.org/10.1093/comjnl/45.3.260

[15] Fadahunsi I, Sodiya A, Olajuwon B. Towards a UMLsec-Based Proctored Examination Model. International journal of secure software engineering, 2019, 10(2):44-68. https://doi.org/10.4018/IJSSSP.2019070103

[16] Kebande V R, Karie N M. A UML-Based Approach for Analysing Potential Digital Forensic Evidence. International Journal of Electronic Security and Digital Forensics, 2018, 7(4):354-362.

[17] Signori A. Optimal treatment for a phase field system of Cahn-Hilliard type modeling tumor growth by asymptotic scheme. Mathematical Control, Related Fields, 2020, 10(2):305-331.

[18] Stepin Y P, Leonov D G, Papilina T M, et al. System modeling, risks evaluation and optimization of a distributed computer system. Computer Research and Modeling, 2020, 12(6):1349-1359.