UML-based modeling and development for CAPP system

Zhang Shijie, Song Laigang
Shenyang Institute of Automation, Chinese Academy of Sciences,
114 Nanta Street, Shenyang, 110016, P.R. China

ABSTRACT UML (Unified Modeling Language) leads the trend of OO system modeling; it can be used in the whole system developing life cycle. A CAPP system modeled by the UML is introduced including the modeling process, some basic concepts. At the end, a brief comparison is made between UML and IDEF.

INTRODUCTION

With the increasing complexity of software system, the traditional software development model does not meet the current demand. The software developing has not been one cycle process from demand analysis, software design to the final software system. An iterated incremental developing model is being widely used. The UML is a language for specifying, visualizing, constructing, and documenting the archives of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems. It has many advantages such as good definition, easy expressing, good adaptability and etc. The new thought, new method and new technology in software engineering have been integrated into the UML. The UML can be used not only in the object-oriented analysis and design, but also in the whole process of developing software system. In China, several application systems using UML have been reported recently. For example, Applications UML to MIS system (Wang et al. 2002), Medical Insurance Management Information System (Lin et al. 2002) and Sales Information system(Hu et al. 2002). But most of them just adopt the basic concepts and the comparison between UML and other modeling language is seldom reported. This paper presents a CAPP system, which is modeled by IDEF and UML separately, and the further introduction to UML and applying UML to system modeling and utilization. The content of the paper mainly includes the basic concept of UML, the modeling process of UML, an instance of CAPP system modeling. In the end, a comparison between UML and IDEF (a widely used modeling method in Chinese CIMS projects) is presented.

ELEMENTARY KNOWLEDGE OF UML

As a modeling language, the definition of UML includes two parts: UML semantics and UML Notation.

1) UML semantics defines the abstract syntax and the meta model of object modeling. UML semantics: describe the UML-based precise meta model. The meta model provides all the elements in UML with simple, unified and general definition, thus make the developer work on the unanimous basis, eliminates the manmade differences between the different people. Furthermore, UML also support the extent definition of UML meta model. The UML grammar is an abstract grammar that is symbols-independent. It can be mapped to different symbol systems.

2) UML notation defines the legend set that is used to describe the semantic of UML. Using the same legend and text grammar, the developer and the developing tool have the unified basis to work on. These graphic symbols and text illustrate the model in application layer. It is the instance of UML meta model in semantics. UML is mainly composed by the following five kinds of diagram:

1) Use Case diagram. Describing the system function from the customer point of view and pointing out the operator of every function.

2) Static Diagram. This kind of diagram is dedicated to show the static relations in the system. These relations exist and validate through the life cycle of system. Class diagram and Package diagram belong to
static diagram. Class diagram shows the interaction between the classes in system and provides the static diagram among system components. Package diagram includes package diagram and class diagram. Using package diagram can put the classes with some common features together. It facilitates the system description.

3) Behavior Diagram. This diagram describes the dynamical relation inside system. It is mainly consists of State chart diagram and Activity diagram. State chart diagram provides every status of modeling object and the condition that make the status shift. But usually it is not needed to generate the state chart diagram for every class, only these complex classes. Activity diagrams show the system function flow. It is the powerful method to understand the system.

4) Interaction Diagram. This diagram describes the interaction among the objects. It is mainly composed by sequence diagram and collaboration diagram. Sequence diagram shows the interaction of object in the function flow of use case. It emphasizes the sequence of message delivering. Collaboration diagram shows the collaborative relation between the objects. They both show the dynamical relation between objects, but emphasize different aspects. For emphasizing time and sequence, sequence diagram is the best; for emphasizing rational distribution of objects, collaboration diagram is the best.

5) Realization Diagram. This diagram describes the physical realization. It is mainly composed by component diagram, deployment diagram. Component diagram show the relations of component in software, it can help to analysis and understand the inter influence between the components. Deployment diagram shows the physical position of all components.

3 THE COMMON STEPS OF DEVELOPING SOFTWARE USING UML

Traditional developing method is one cycle and one direction method from the demand analysis, system design, system developing, system test and system deployment. The main problem of this method is to trace back step by step. If the demands changed, system will be redesigned. In practice it is impossible to know all the demands in the early times of project. The figure 1 shows the basic steps of repeated developing method.

The main steps of using UML to system model and develop is as follows.

(1) Demand analysis. The Use Case diagram can express the customer’s demands. After Use Case mod-

eling, the developer can know the external actors and the function they need. Every Use Case represents the customer’s demand.

![Diagram: Repeated Developing Method]

(2) System analysis. In this phase, static diagram and dynamical diagram will be used to describe the system. The class diagram describes the static structure of system. Collaboration diagram, state chart diagram, sequence diagram and activity diagram describe the dynamical features of system.

(3) System design. In this phase, the results of analysis phase will be converted to technical solutions. Some new classes will be added into the model to provide the technical infrastructure, such as user interface, database. The detailed specification will be generated.

(4) System Construct. In this phase, the classes obtained in the design phase will the converted to program codes. The rest part of system will be coded according to the system architecture in the design phase.

(5) System Test. The system test might be divided to different levels. Such as unit test, integrates test and system test.

4 A CASE OF CAPP SYSTEM MODELING

According to the real situation of process planning in machine tools enterprise, we developed a CAPP system, which integrates interactive, variant and generative mechanisms into one system. The CAPP system consists of five parts: Information Input of parts, Process decision-making, Process data and knowledge database management, User Interface and Process file output and edit. The architecture of CAPP system is shown in figure 2.

The basic functions of CAPP system consists of part and process information input, query existing process files, retrieving the part information and process knowledge from the database, and interactive process planning. While the variant and the generative mechanisms are also provided. After
carefully investigation to Shenyang No.1 Machine Tools, we finished the CAPP system modeling using UML, and developed a product oriented CAPP system. The detail process of CAPP system modeling is introduced in the following.

4.1 Use case diagram

Use Case represents a full function of system. The use case in UML is a collection of several actions. Action is an execution of system and can present results to some actors (Jiang et al. 2001). In the CAPP system modeling, the following Use Cases are used. As Figure 3 is shown.

(1) Part Information Input. An interface is provided to facilitate the interactive part information input and reduce the amount of input information as little as possible.
(2) Process Planning. Three ways are provided. The interactive way is based on the standard working-step, typical working-operation and combines with the function of Use Case for process file management. The Variant way is to find the appropriate typical process based on the part information and categorized code. Then the customer can modify the typical process file according to the specific object. The generative way is to use knowledge-based methods to generate the process file.
(3) Process file management. Providing a convenient process file management mechanism, thus it can help the customer to manage the process files including the version control and typical process.
(4) Process data and knowledge management. Providing convenient process data and knowledge management, helping the customers to find any information they need in the system.

Because the process planning Use Case is the most important module in the CAPP system, it will be introduced in the following parts of the papers.

4.2 Interactive diagram

In the process of system modeling, the sequence diagram is drawn. It shows the dynamical relation between the objects in the system, analysis the sequence of message and the interaction between the object and the system execution process. The sequence diagram of CAPP system is shown in Figure 4.

The collaboration diagram is not presented in the CAPP system. It can be generated by means of the sequence diagram. Both of them show the interaction between the objects, so one of them can be selected to present the required information.

4.3 Class diagram and package diagram

The system adopts the Browser/Server architecture. The HTTP protocol is adopted to transmit data. HTTP is a stateless protocol, it means that the client cannot hold any connection with the business logic
object when sever is handling requests. For this reason, HTTP protocol can not be used to design a system with states, so all the classes in the system are stateless, that is, all the classes system provided have no attributes. The class diagram is shown in Figure 5.

![CAPP class diagram](image)

**Figure 5. CAPP class diagram**

### 4.4 Component diagram and deployment diagram

Component Diagram is used to describe the relations between the components of software, show the structure of codes. Usually, it is the realization file of the developing environment. It can be binary component or executive component. In our system, the part information input component, process planning component, process file management component and process data and knowledge management component are provided. Beside these components, the relevant components provided by the operating system of computer are also parts of the CAPP system. Deployment Diagram is used for representing the final physical description of system topology. The figure 6 shows the deployment diagram.

![CAPP deployment diagram](image)

**Figure 6. CAPP deployment diagram**

5 THE COMPARISON BETWEEN UML AND IDEF

IDEF (ICAM DEFinition Method) is a system modeling method that is widely used in modeling complex system. It was based on the structure analysis promoted in 1970's. Now the KBSI Company is dedicating to developing this method. IDEF is composed by several sub-methods: IDEF0 describes the system function activity and relations. IDEF1 describes the system information structure and semantics. IDEF2 describes the system dynamical model. IDEF3 focuses on the system process. IDEF4 is mainly used to provide model to object oriented language. IDEF5 is a method using ontology. Now, IDEF2 had been abandoned, and IDEF5 is not mature.

The differences between IDEF and UML are as following.

1. UML combines the current software development experiences and focus on the system architecture and the Use Case. It emphasizes more on the view of customers and pays more attention to the system function, thus make the model more extensible. IDEF emphasizes the whole system, and analyzes from the view of system. Thus it can show the internal modules and the relations between the modules. But its extensibility is not so good as UML.
2. UML is already used in various application domains. Although currently IDEF is used in CIMS and lots of information system analysis and modeling, it is not good compared to the scope of UML (including financial system, mechanical system, enterprise process, real-time system, distributed system and so on).
3. The sequence diagram in UML can help the developer understand the system process more clearly and reduce the chances of change in the later phase. The collaboration diagram can help the developer understand the system load and avoid the performance bottleneck. Such kind of functions is not enough in IDEF; it emphasizes more on the global modeling and lack in the detailed analysis.
4. From the very beginning, UML adopts the Object Oriented thought. The Object Oriented method is the most prevailing software developing method and supported by most of advanced developing tools. Besides, the UML modeling process can help the popularization of OO thought and help the developer analyze the problems in the way of OO thought. IDEF is earlier than UML. It is developed on the basis of structured analysis method and cannot full embody the OO thought.
5. UML provides a set of consistent, easy understanding graphic legends. Using these legends can show the information in the model exactly and di-
rectly to the user, thus the whole architecture can be easily grasped. Although IDEF provides a set of legends too, these legends are not so direct as those of UML and need review or refer to the annotation regularly.

(6) UML can be used in the whole process of system development (from the demand analysis to deployment). IDEF need to change in the several sub-methods. For example, function modeling needs IDEF0, information modeling needs IDEF1, process modeling needs IDEF3, and Object oriented developing needs IDEF4. Many changes are need in the whole process.

(7) The standard software development emphasizes integrity of the document very much. In this point, both UML and IDEF can provide the need documents. UML can provide more detailed documents compared with IDEF.

Through the above analysis we can safely come to a conclusion: UML had many advantages in several aspects. These advantages make UML the international standard of modeling language.

6 CONCLUSIONS

After modeling the CAPP system using IDEF and UML separately, we deeply feel that the conveniences brought by the integrated process of UML from the demand analysis to deployment. Recently, Authors had finished the conversion from system modeling to code framework using the Rational Rose (a UML developing Tool). The system meets the demands of Object Oriented thought and possesses good extensibility. With the prevalence of Object Oriented thought, UML will be the first choice among all modeling languages.

ACKNOWLEDGEMENTS

The authors wish to thank Engineer Chen Dongsheng and Wang Qingwei for their rich process knowledge contributions.

REFERENCES

Lin, Z. & Deng, J.H. 2002, Apply UML to Design Domain Model of Medical Insurance MIS.