

Software Development Method Based on Structured Management of Code

Xia Chen and Junsuo Zhao

Abstract Code management based on files has a lot of problems and shortages. This paper puts forward the software development method based on structured management of code. Structured management of code reduces the source code unit from files to minimum unit code, and achieves the fine granular and structuration. The software development based on code structured management has shown obvious advantages in many aspects, such as improving personal and collaborative working efficiency, statistical analysis, roles, and authority management etc.

Keywords Software development · Structured management of code · File-based management of code · Database

1 Introduction

The scale and complexity of software has increased dramatically along with the increasing demand for software [1]. Software development gradually changes from stand-alone to network, centralized to distributed, exclusive to shared, and last but not least, isolation to collaboration. This phenomenon is referred to as Global Software Development, Distributed Software Development, or Multi-Site Software Development [2]. Nowadays, Software development is no longer a short-term

X. Chen (✉) · J. Zhao
Science and Technology on Integrated Information System Laboratory,
Institute of Software, Chinese Academy of Sciences, Beijing, China
e-mail: chen918xia@sina.com

J. Zhao
e-mail: junsuo@iscas.ac.cn

X. Chen
University of Chinese Academy of Sciences, Beijing, China

individual behavior, but a long-term group collaboration process. More and more software development activities are facing severe problems including the rapid increase of software development tasks and number of developers in one project, the extension of software development cycle, as well as dramatic decrease in average efficiency. Many factors will lead to the decrease in average efficiency. On one hand, personal ability differs from each other, on the other hand, the cost of communication and coordination between groups cannot be ignored [3, 4]. DeMarco and Jones pointed out that during large-scale system development, about 70–85 % of the development cost is spend on teamwork activities [5, 6].

Until now, there have been many theories and methods focusing on improving software engineering, such as role-based collaboration modeling method [7], component-based software development [8] paradigm of software development for mass customization [9], etc. However, these methods ignore the important roles that code management and software development tools played in improving personal and collaborative work efficiency. For so long software development proceeds based on file-based code management which means the source code space consists of multiple source code files which contain partial codes, respectively. There are many disadvantages for file-based code management such as uneasy to read, unobvious relevance, etc. These shortages will result in the increasing personal and collaborative work costs.

In this paper, we put forward software development method based on structured management of code. We reduce the source code management unit from files to minimum unit code, and utilize database to realize the fine granular and structuration. It should be noted that code management is the foundation of software engineering. Therefore, structured management of code which based on database will change the way of former code management (file-based code management) fundamentally, cause changes in many steps of software engineering, reduce collaboration costs, improve personal efficiency, and bring more convenience to project management.

2 Software Development Based on Files and Documentations

2.1 Environment of Software Development Based on Files and Documentations

Structure of Software Development based on files and documentations is shown in Fig. 1. Source code files and documentations are the core of software product. The activity of software development is proceeded under source code files and related documentations. The essential is the process of increasing, deleting, and updating of source code files and documentations.

Each source code file contains an independent section code under the form of code management based on files. Many source code files consist of the code space

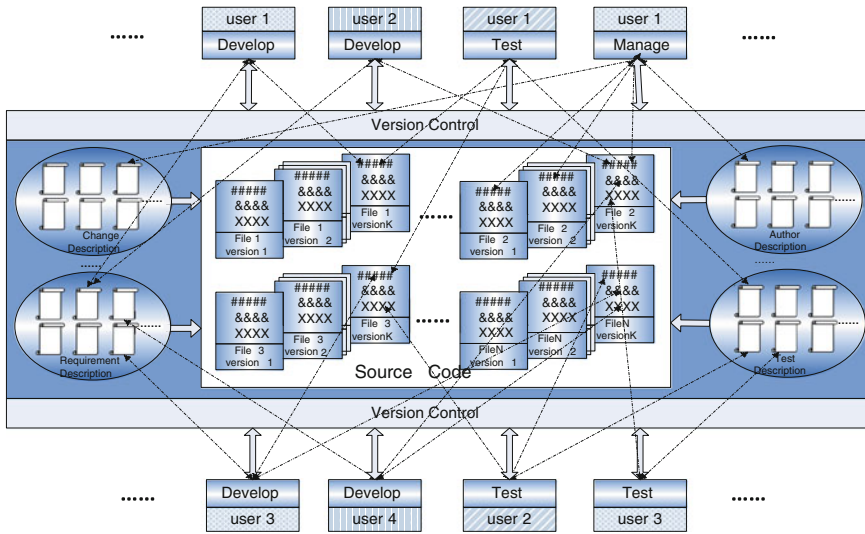


Fig. 1 Environment of software development

of software product, and also as the documentations. In the life cycle of the software product, different roles, such as development people and testing people, operate the source code and related documentations via development environment, testing environment and management environment. Source code files and documentations are evolutionary. As the increasing of software products and functional modules, the number of file and documentation is increasing. The content of each source code file or documentation is becoming perfect along with the advance of development. The increasing of files and documentations in space and time forms the process of software product.

2.2 The Problems of Code Management Based on files

Uneasy to Read. Source code files are transparent before people read them. And people can only capture the functions and interfaces of the code but not the amount of code, content, structure of the code and other details.

Every time the smallest unit people can get is the whole codes of a source code file if they manage codes based on file. But they actually only need to modify a piece of codes (Fig. 1). As the source code files are transparent, only after reading the whole codes, can developers modify relevant part. This kind of repetition is time consuming.

Unobvious Relevance. Source code includes lots of related information, such as compiling, testing, author, functions, version, modules, references etc. Code has strong correlation with its related information logically. Different codes also have

correlation. The correlation is hidden because the different information is in different source code files or documentations.

The operation of code often involves many source code files or documentations which have correlation. It should be analyzed and extracted artificially because it is hidden. People must read many source code files or documentations if they just want to get some information of a piece of code. The updating of code will cause many changes of different files and documentations. The changes would be completed by different people related. It should be noted that the relevance hidden will bring bad effect on maintaining the synchronization of source code and information.

3 Structured Management of Code

In order to overcome the problem induced by file-based management, this article puts forward the structured management of code method. Structured management of code refers to the saving of all the minimum unit code in a software product source code space in relevance database, and realizing the easy, accurate retrieval, up-to-date operation of any minimum unit code. The structured management of source code brings about the fine-grained control of source code space. In the mean time, it also makes the combination of source code space and various information described space come true.

3.1 Features of Structured Management of Code

Completeness. Source code and related information are saved in database. The information is complete therefore we can retrieve any source code or related information in a very short time.

Fine Granular. Fine-grained means abandoning the concept of files, and adopting the management method which combine minimum unit code and modules.

Minimum unit code means the minimum code part which has complete and accurate physical meaning. Take C language as an example, function definition, function declaration, type definition, global variable definition, variable declaration, macro definition are all Minimum unit codes.

Fine-grained management of code has two meanings: first, source code space consists of minimum unit code instead of file, and plenty of minimum unit code together forms the source code space; second, minimum unit codes in source codes space are not saved disorderly but managed by detailed multi-level modules. The properties of minimum unit code determine which modules level it should belong to.

Therefore, adding, deleting, updating, and testing minimum unit codes or modules can be operated based on either the modulus or minimum unit code.

Obvious Relevance. Relevance is unobvious in file-based code management method. If adopting the relevance database, the code and its relevant information will be one or several entries of the relevance database table. The relationships between code and its relevant information, code and code could be easily presented through the connection in relevance tables.

Fast and Accurate Retrieval. All the content saved in database is accurate, complete, and clearly correlated because of the completeness, fine granular, and obvious relevance of database management. All of these characters will ensure us to easily, fast, and accurately retrieve the source codes as well as its related information by simply operating on database such as selecting, connecting, and projection.

3.2 Implementation of Structured Management of Code Based on Database

Figure 2 illustrates the design of database for the project in C programming language. The project information, module information, and code information related to a project are all stored in database. Each information record is comprised of several relationship tables and each relationship table including several table items saves the source code or related description information. In function information table's case, the main content of table item including function ID, function name, return values, function description, function declaration statement, status(have been written or have been tested), module ID, public or private, programmer's ID, version, whether is the latest, creation time, the latest modification time, lines of source code, compiler-generated file name, etc.

Function is the most frequently minimum unit code appearing in source code. For example, by using function information table and function body table, source code can be obtained. The code information such as type, parameters, return values, local variables, and lines of source code can be obtained from function information table, function parameters table, local variables table, and function return values table without function code analysis, respectively. Functional description, versions (such as the version number and whether it is the latest), status (writing or testing), and history (creation time and the latest modification time) can be obtained from function information table. Through the combination of selecting operation of module information table, function reference table and function information table, the module information which belongs to function, visibility, attribute of public or private, and the relationship of function calls can also be obtained. The author information about the function in different versions can be obtained from function information table and user information table.

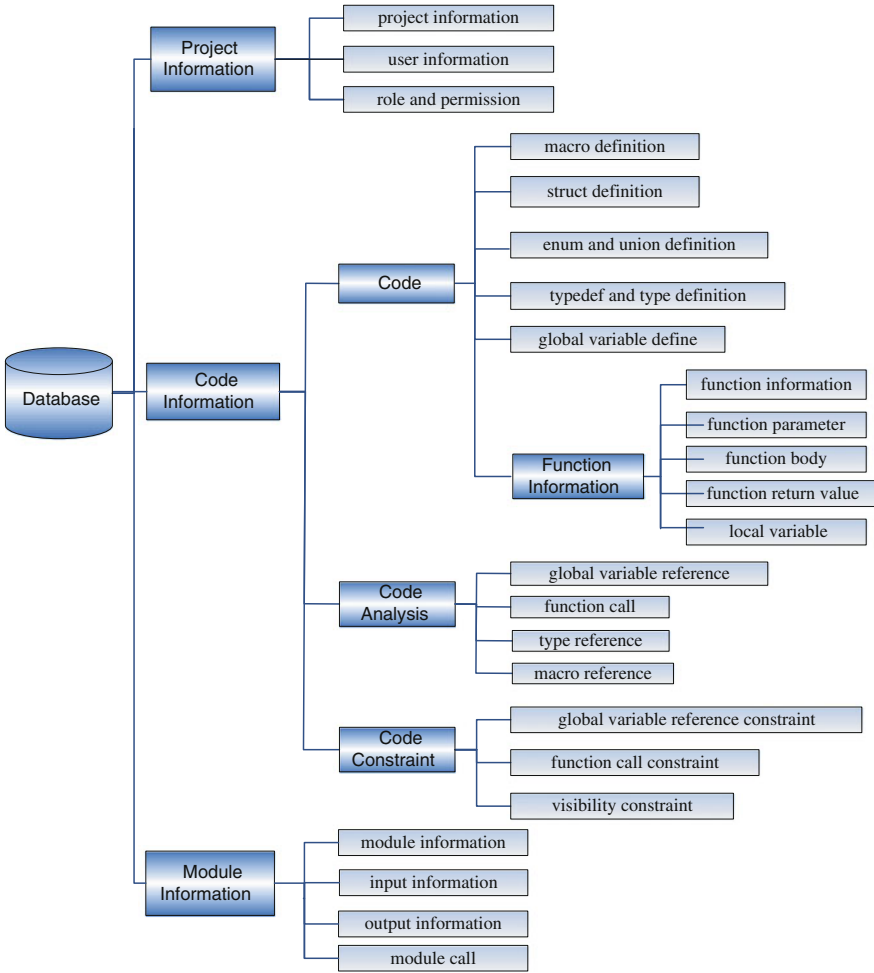


Fig. 2 The realization of C language code structured storage via database

4 Software Development Environment Based on the Structured Management of Code

4.1 System Design

The software development environment and the server architecture based on the code structured management are shown in Figs. 3 and 4, respectively. We use C/S structure to establish the connection between software development client and software development server. The client sends data to the server and requests services. According to the service request, the server combining with user roles

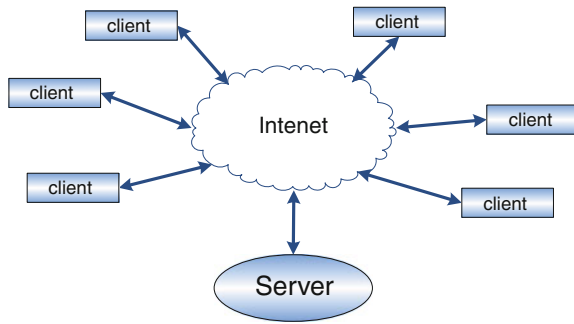


Fig. 3 The software development environment based on internet

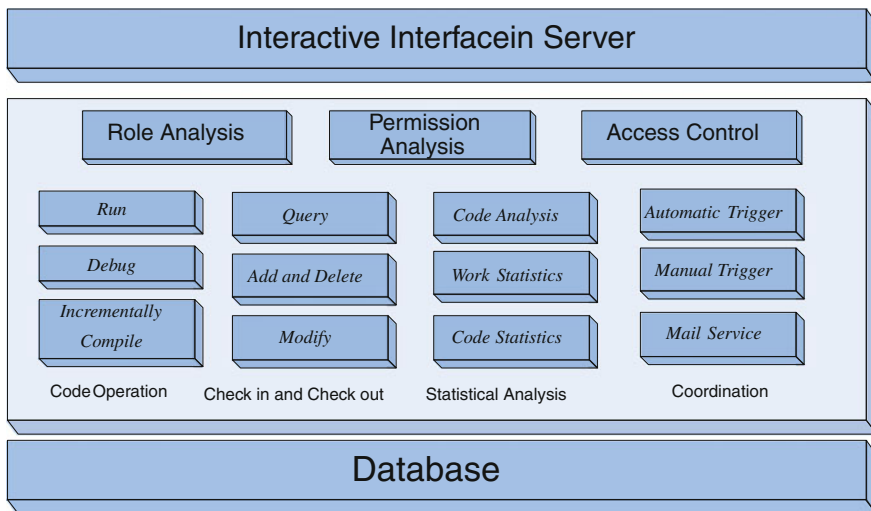


Fig. 4 The architecture diagram of the software development environment on the server side

and access control returns the results to the user through the database and service mechanism. All the source code and related information are stored on the server side in the database. Users log in software development from terminal side and work cooperatively in a unified, comprehensive and up-to-date space.

Server provides services mainly include the following issues.

Incremental Compiling, Debugging, and Running. The server provides code compiling, debugging, and operating environment. When a minimum unit code has been written, it can be submitted to a server to build, run, and view the results in a real time rather than only after writing a lot of code can they compile and error check.

Check in and Check out. Combining with roles and permissions, the user can get the latest and most complete view about code space and related information on their own authority. Writing operation on code and related information within the scope of user’s own authority will be saved to the database in the first time.

The database stores all the history of the minimum unit of code and its associated information change process.

Statistical Analysis. Miscellaneous statistical functions can be finished easily by using database, such as the number of functions in code space, the number of global variable, the number of structure, enumerations and macros, the rows of function body, the amount of code in each modules etc. The size of static data area, the size of code segment can be exactly acquired by using these statistics. In embedded software development, due to the relatively limited hardware resources, the statistics and assessments in the development process can avoid the huge cost of late readjustment.

Each programmer's workload, working efficiency, daily online time can be counted by using database and client terminal. It is easier to complete the assessments of capacity for programmer. By using representation reference relational table, the call graph and global variable reference graph can be easily constructed, which can be used for function dependence analysis, global variable reference analysis, etc.

Communication and Coordination. The traditional cooperative mode is mainly by mail or documentation [10]. According to the cooperative problems, people usually write corresponding documentations, analysis the code segment or module of the problems, find the responsible person involved, and send documentations to them [11].

The cooperative work can save a lot of manpower by using code structured management. For example, if there is any change of the interface in development process, people that can directly retrieve the minimum unit code citing this interface as well as the responsible user involved in the interface changing. What's more, the interface changes specification can be automatic generated. As long as the responsible user logs in development client side, they can immediately receive an interface changing notice. On the other hand they can also get the interface changing notice email through the automatic mail service.

The whole process of the cooperative work can be triggered automatically or manually. The completeness of the database information and powerful search function can reduce the manpower cost on communication and cooperation, improve response time, and ensure the efficiency and accuracy of the cooperation.

4.2 Improving the Efficiency of Development

This method improves the efficiency and accuracy of personal development perform in many aspects, such as detecting an error as soon as code incremental compiling, doing named conflict detection by using statistical analysis of the database, preventing renaming error finding at link stage with high cost, and reducing duplication development and improving the code reusability. Integration testing can be on as soon as possible which means that in the file-based code management mode before, waiting for the completing developing all the files

involved in integrated test function is needed, now as long as all minimum unit codes involved in integration testing complete developing, process can continue. Minimum compiling can be affected by local changes. By using minimum compiling strategy, minimum unit code as basic element can be used as much as possible. If local changes appear, people only need to recompile the minimum amount of local codes.

5 Conclusion

By analyzing the problems of the file-based code management, this article puts forward the concept of structured management, gives the designs and implements of a software development environment based on this method. Structured management of code minimizes the management unit of source code and provides a detailed, organically organized view of the space of source code. The development of software based on the structured management of code changes the way of code's organization and management fundamentally. It shows advancement in the following four aspects: (1) overcoming problems in the file-based code management, (2) conveniently providing functions such as the control of roles and permission, the statistical analysis, and the latest systematic view etc. (3) implementing the communication and teamwork accurately and efficiently, (4) accelerating individuals' efficiency and accuracy etc.

References

1. Source lines of code. <http://en.wikipedia.org/wiki/>
2. Ultra-Large-Scale-Systems The software challenge of the future. software engineering institute, Carnegie Mellon (2006)
3. PJim W (2007) Collaboration in software engineering: a roadmap. *Future Softw Eng* 214–225
4. Kraut RE, Streeter LA (1995) Coordination in software development. *communications of the ACM*. ACM Press, New York, pp 69–81
5. DeMarco T, Lister T (1987) *Peopleware: productive projects and teams*. DorsetHouse, New York
6. Jones TC (1986) *Programming productivity*. McGraw-Hill, New York
7. Ge S, Sun YL, Du ZX (2003) A role based collaboration modeling method. *Comput Eng Appl* pp 14–18
8. Wei HY, Zhan JF Wang Q (2004) Overview of distributed component technologies. *Appl Res Comput* 12–15
9. Cao YF, Zhao JW, Han YS, Dai GZ (2002) A paradigm of software development for mass customization. *J Comput Res Dev* 593–598
10. Jeff K (1994) Distributed software engineering. In: *Proceedings of the 16th international conference on software engineering*, pp 253–263 1994
11. Maryam P, Martin P, Bastin TRS (2006) Facilitating collaboration in a distributed software development environment using P2P architecture. *The information science discussion paper series*