Design and implementation of the whole process collaborative system for software development

Wang Kai1,*

1Jiangsu Automation Research Institute, Lianyungang, China

wangkai@jari.cn

Abstract. With the increasingly urgent and arduous task of software development, software engineering development tools that can efficiently respond to the continuous changes in software requirements, rationally plan the software development process, and control the development cost will be widely used in software engineering. In response to actual needs, the supporting tool is designed to meet the system requirements, software requirements and software design and development. First, we use visualization modeling optimization technology of structured analysis and design method to build demand analysis models and system design models; Secondly, the dynamic verification technology of data model logical consistency is adopted to verify the logical consistency of data model in evolution; finally, the verification of the system model is achieved through hierarchical flow diagrams, fully expanded flow diagrams and statistical views. The vehicle cruise control system is selected for application verification, and the tool meets the requirements of software development and improves the software quality.

1. Introduction
In the existing software development process, the software requirements are issued by the overall unit to the software development department, and then the software requirements analysts prepare the software requirements specifications, and carry out the analysis, design and coding realization of the software. With the increasingly pressing and onerous tasks of software development, this simple development process based on manual methods can no longer cope with the constantly changing situation of software requirements in a standardized and efficient manner, nor can it reasonably plan the software development process to meet the control requirements of time and cost [1]. Specific performance is as follows:

1) Lack of standardized and unified requirements description. At present, the core of requirements analysis is document compilation, while the focus of requirements document compilation is often on document format and uniformity, and no more attention is paid to requirements themselves. In the analysis and description of requirements, due to the differences in the descriptions of different demand analysts, different people often have different expressions for the same demand [2].

2) The workload of compiling and maintaining software requirements is huge. In the face of requirements change, it is a huge workload to maintain the logical consistency of requirements and the correctness of documents.

3) Lack of supporting platform for demand analysis. At present, under the labor-based working mode, the cost of team communication is high, and a lot of time and energy are spent on discussing the
function division and interface definition between modules, and there is no supporting platform for requirement establishment and maintenance.

To sum up, it is urgent to develop a kind of software engineering development tool which can satisfy domestic independent control, follow industry or software design standards, meet system requirements, software requirements, and integrate software design and development.

2. Requirements analysis and design technology based on system software engineering method

2.1. Visualization modeling optimization technology of structured analysis and design method

In this paper, "Data Flow Diagram/Control Flow Diagram", "Processing Specification" and "Control Specification" are used to describe the software structure. Use the input and output data of the system to build a data dictionary, so as to build a structured system diagram [3]. As shown in Figure 1.

![Figure 1. System structured architecture diagram.](image)

1) Between processing model and Data Flow Diagram (DFD), the data flow composed of fixed components can be processed and transmitted interactively, showing the data flow direction.

2) Between control model and Control Flow Diagram (CFD), the control flow of program execution order can be transformed.

3) The data information formed by Data Flow Diagram (DFD) is transmitted to the data dictionary in the form of data flow. Each processing of DFD can accurately describe its function by processing description, which can include the input/output index of related data flow graph, and check the consistency of the information flow between processing description and data flow diagram.

4) Data dictionary can be used to save the definition information of data flow, control flow, data elements and data storage in the model. Each data item in the data dictionary must be defined and decomposed into a series of basic data items, which can be expressed as plain text description or structural description, and the structural description uses simplified BNF format.

5) The control structure of the system disclosed in Control Flow Diagram (CFD) corresponds to the data flow formed by Data Flow Diagram (DFD) one by one, and has the same naming, layering and balance attributes.

6) Control specification (CSPEC) describes that the control signal is converted into a process start signal, which has consistency verification with the control rod and also has consistency verification with the control process.

7) Process specification (PSPEC) describes a process. A process can only have one process description, and its contents can include input/output indexes, characters, tables and graphs in related flow charts.

2.2. Dynamic verification technology of data model logic consistency

In order to ensure the behavior consistency of the data model in evolution and avoid illegal model evolution, this study carried out model verification based on process algebra, introduced external active request interface and internal connection, formally described the model and its external
interaction protocol, and specified verification rules to verify interface compatibility and external interaction behavior consistency[4][5].

In the data model, the external interface and the internal interface are described by data flow and control flow. Decompose the corresponding data flow control flow into the minimum elements-data items, and then compare them. Inconsistent basic data items can be obtained by comparison, but this prompt can't clearly point out the specific data flow or control flow, and it needs to be merged until it can't be merged.

For the rule check of the model, firstly, configure the inspection rules of the analysis model; secondly, execute the rule check, and generate the inspection results and output them to relevant windows. In case of violation of the rules, the font of the displayed information is bold (or red), and the process of rule check is shown in Figure 2.

![Figure 2. Rule checking process.](image)

In the technical implementation of this paper, the inspection rules shown in Table 1 are designed. And the customization of inspection rules is supported, as shown in Figure 3.

**Table 1. Procuratorial rules.**

| Serial number | Procuratorial rules                                      |
|---------------|--------------------------------------------------------|
| 1             | Consistency of requirement analysis model               |
| 2             | Uniqueness of requirement analysis model                |
| 3             | The unused condition of requirement analysis model     |
| 4             | The omission condition of requirement analysis model   |
| 5             | Consistency of system design model                      |
| 6             | Uniqueness of system design model                       |
| 7             | The unused condition of system design model            |
| 8             | The undefined nature of system design model            |
| 9             | Grammar check of software design model                 |

2.3. Automatic generation technology of software requirements and design documents based on GJB438B

According to the requirements of GJB438B, the requirements and design documents conforming to the standard specifications are automatically generated from the aspects of data storage and representation of requirements and design models, document generation framework, document standard template design, document filling element extraction method, document element validity and integrity, and document generation fault tolerance and control, as shown in Figure 4.

Automatic generation technology of software requirement specification and design document based
on GJB438B includes dynamic connection of Word, automatic loading of graphics, and automatic generation of words and automatic generation of tables. Dynamic connection of Word is the basis of automatic generation of documents, and Word interface library is introduced and initialized. Automatic loading of graphics refers to automatically loading various graphics in the set document area; automatically generate characters by using fields and keywords; automatically generate tables by using the function of appending tables [6].

3. System design and development

3.1. System modules and hierarchical division

The main contents of the system implementation include: model representation, model editing, version management, collaborative development, and report generation.

Model representation: using models to represent various software data objects, including CSCI, CSC, CSU, Type, Data, etc., as well as analysis diagrams such as flowcharts, call structure diagrams, etc.; the data structure obtained from the framework analysis can be converted into a software model and recorded, and generate corresponding flowcharts and call structure diagrams based on the relationship.

Model editing: organizational division of CSU (construction of CSC, CSCI), and level changes (reconstruction).

Version management: use the server to record each version data of the model.

Collaborative development: Collaborative development can be realized through the version management server.

Report generation: Customized report templates can be used to generate Word reports in a format that meets user requirements.

3.2. Software architecture

The software architecture is shown in Figure 5, including server, client, and module. The server is responsible for storing version data, the client is responsible for collaborative development, and the module is responsible for function realization [7].

![Figure 4. Template-based report generation.](image)

![Figure 5. Software architecture.](image)
3.3. Architectural analysis and optimization steps

Architectural analysis and optimization are completed in 3 steps, as shown in Figure 6.

Step 1: Architecture analysis. Realize function, type and variable analysis, and get function call relationship and variable reference relationship.

Step 2: Software model optimization. Generate CSU, Type, and Data model objects. Generate flowcharts and call graphs.

Step 3: Architecture optimization. Classify organizations and create CSCI and CSU. Perform module level adjustment and reconstruction.

![Figure 6. Architectural analysis and optimization steps.](image_url)

3.4. Collaboration between software modules

This system mainly consists of several parts such as interaction layer with other software, UI layer/user operation layer, function module layer, version management and module reuse layer. The relationship between the components is shown in Figure 7.

![Figure 7. Collaboration between software modules.](image_url)

After the requirements are imported successfully, the software automatically generates the root node marking the entire system design according to the requirement identification results. The root node attribute items correspond to the identified requirements. In principle, each attribute item corresponds to an overall requirement item, and the integrity of the requirements is guaranteed. Nodes can be used to build system block diagrams. Nodes are generated by users according to the basic configuration of primitives provided by the software, and support graphical display and drag-and-drop nodes. Each node in the system has an attribute item consistent with the root node, and the attribute item entries can be optimized on this basis.

The software platform supports automatic decomposition of requirements according to the design block diagram and specified connection relationship, according to the requirements decomposition method of the characteristics selected by the user, and the adjustment and optimization of the decomposition results.

3.5. Software prototype

The software platform runs through the entire system development process. It has functions such as system and software requirements analysis, system architecture modeling, full support for concurrent model analysis and design in the system, code framework generation, and comprehensive report...
customization. It supports a team of at least 20 people for collaborative design and object-oriented and process-oriented software design mode. Provide a construction model (or physical model) in the system and software design stage to describe how the system is built. Provide system architecture model for system design, realize the description of the system environment, system composition, information interaction between the system and the external environment, and information interaction between various subsystems. The software interface is shown in Figure 8.

Figure 8. The whole process collaborative system for software development.

4. Cruise control and monitoring system design example
This paper combines the design of cruise control and monitoring system, adopts a top-down, layer-by-layer decomposition method, and abstracts the system into a series of logical processing units according to the relationship of data transfer and transformation within the system. Each unit is related by data flow to establish the logical model of the system and obtain the final system model.

4.1. Structured demand analysis
Starting from the analysis of the external environment, define the boundary of the system, that is, what is the external interface of the system, the basic functions that the system must complete, describe the user and the user of the system, and the information exchange between the system user and the user and the system (Including data information and control information). Through structured demand analysis, the cruise control and monitoring system analyzes a total of 5 control requirements such as engine status, braking, and maintenance command, and 4 data requirements such as shaft revolution, oil volume, and speed control value. Figure 9 shows the structured demand analysis results of the cruise control and monitoring system.

Figure 9. Structured demand analysis results of the cruise control and monitoring system.

4.2. Data flow diagram and control flow diagram
The software platform supports displaying data flow graph and control flow graph in Figure 10. The data flow diagram and control flow diagram of the cruise control and monitoring system are shown in the figure. For the cruise control and monitoring system, it is decomposed into 4 first-level processing,
which are cruise control processing, calibration processing, status monitoring processing, distance and speed calculation processing.

![Data flow diagram and control flow diagram.](image)

**Figure 10.** Data flow diagram and control flow diagram.

4.3. Report generation

Using GJB438B-based automatic document generation technology to automatically generate software requirements documents and design documents, greatly reducing the workload of document writing and maintenance. The result of automatic document generation is shown in Figure 11.

![Automatic generation of software requirement specifications and design documents.](image)

**Figure 11.** Automatic generation of software requirement specifications and design documents.

4.4. Auxiliary analysis

The software platform verifies the system model through hierarchical flow diagrams, fully expanded flow diagrams and statistical views, analyzes the composition and hierarchy of the system, displays the relationship between data flow and control flow, and stats model data information.

The hierarchical flow diagram is used to display the tree structure between the requirements in the project and the data flow or control flow interaction analysis between the requirements. The statistical information of the hierarchical flow chart of the cruise control and monitoring system is shown in Figure 12. The cruise control and monitoring system has 4 first-level processing and 9 second-level processing. The processing relationship of each level is shown in Table 2.

![Hierarchical flow diagram.](image)

**Figure 12.** Hierarchical flow diagram.
Table 2. Dealing with relationships at all levels.

| level               | 1                               | 2                               |
|---------------------|---------------------------------|---------------------------------|
| Cruise Control and  | Calculate Distance and Speed    | Calculate Distance              |
| Monitoring System   | Calculate Speed                 | Calculate Speed                 |
| Calibration         | Calculate Calibration Number    | Calculate Calibration Constant  |
| Car Monitoring      | -                               |                                 |
| Cruise Control      | Select Speed                    | Clearing Speed                  |
|                     |                                 | Maintain Speed                  |
|                     |                                 | Restore Cruising Speed          |
|                     |                                 | Maintain Accelerated            |

The fully expanded flow chart is used to analyze the data flow or control flow interaction between all the functional processes of the project, and you can intuitively see the relationship between all the processes in the project. As shown in Figure 13, the fully expanded flow chart of the cruise control and monitoring system shows the relationship between the upper processing and the lower processing, as well as the data flow and control flow information between the terminal and the processing, and between the processing and the processing.

The statistical view provides the usage number of model objects such as requirements, data flow, control flow, and basic processing in the entire project. Based on these statistics, the scale of the cruise control and monitoring system can be analyzed. The statistical view of the cruise control and monitoring system is shown in Figure 14.

Figure 13. The fully expanded flow diagram.

Figure 14. Statistical view.

Model data statistics of the cruise control and monitoring system is shown in Table 3. Through model data statistics, we can intuitively see the statistical value of each statistical item, provide data support for system demand model creation and software design, and help optimize the demand model.

Table 3. Model data statistics of the cruise control and monitoring system.

| Serial number | statistical item                                 | statistical value |
|---------------|-------------------------------------------------|-------------------|
| 1             | Terminal number                                 | 9                 |
| 2             | Data flow graph decomposition layer number     | 2                 |
| 3             | Basic processing number                         | 18                |
| 4             | Data flow number                                | 22                |
5. Conclusion
This paper adopts a model-based system requirement analysis method to effectively solve the problem of low software development efficiency caused by non-standard requirement establishment. The developed tool meet system requirements, software requirements, and software design and development integration of software development requirements, meet domestic independent controllability, follow software design standards, and provide support for the improvement of software quality.

References
[1] RONG GuoPing, ZHANG He, SHAO Dong, WANG Qing. Survey of Process and Management Approaches for Software are Development [J]. Journal of Software, 2019, 30(01):62-79.
[2] Jinzhi Lu, Yuejie Wen, Qi Liu, Didem Gürdür, Martin Törngren. MBSE Applicability Analysis in Chinese Industry [J]. INCOSE International Symposium, 2018, 28(1).
[3] Huang Ping, Du De-hui. A co-modelling approach for cyber-physical systems with SysML/MARTE/pCCSL [J]. Journal of East China Normal University (Natural Science), 2019(01):48-57.
[4] YU Ming-gang, QUAN Ji-chuan, DONG Jing-wei. Requirement Model Consistency Verification Method for C4ISR System [J]. Fire Control & Command Control, 2019, 44(11):143-150.
[5] Yu Shuangshuang, Zeng Yi, Liu Huijun, Yang Yanning. The Consistency Detection Method of Polymorphism and Java Interface Code Information Based on UML Model [J]. Computer Applications and Software, 2017, 34(02):8-13+47.
[6] Luo Rong, Huang Jun, Li Maofeng, Liu Zhiqin. A FAST GENERATION METHOD OF COMPLEX DOCUMENTS BASED ON WORD TEMPLATE [J]. Computer Applications and Software, 2020, 37(10):57-63.
[7] Shi Jiaoli, Huang Chuanhe, He Kai, Shen Xiaying, Hua Chao. An Access Control Method Supporting Multi-User Collaborative Edit in Cloud Storage [J]. Journal of Computer Research and Development, 2017, 54(07):1603-1616.