Research and Application of Software Multi-state Customization Management

Jianping Zhao[1, a], Xiaoyang Liu[1, b], Huihua Wu[1, c], Yong Zhou[1, d], XiaoChun Liu [1, e], Liya Xu [1, f]

[1]Chinese Satellite Marine Tracking and Control Department, Jiangyin, 214431, China

[a]email: niuniu-jiangying @163.com, [b]email: liuxiaoyang120@163.com,
[c]email:wuhuihua.xjtu@126.com, [d]email: ZZZ_781029 @hotmail.com,
[e]email:lxc439.student@sina.com, [f]email:xlych2003@163.com

Keywords: Software Multi-state; Customization Management; Reflex mechanism; Basic Framework

Abstract. It put forward software multi-state customize preservation technology, which accomplished software multi-task mode quick switch and recovery, and actualized the goal of executing dissimilarity satellite model task alternatively for experiment information surveillance display software, by applied reflex technology, stored and managed one or more task environments at the same time. The foundation frame of software multi-state customized management based on reflex mechanism enormously enhanced the elasticity of application system exploitation for software multi-state customized management. Through configured the parameters for the application program by XML, and loaded it into the allocation table when system start-up, it attained the reflex infusion similar Web exploitation.

Introduction

It designed software multi-state customized management subsystem. In accordance with the dissimilarity and complex situation of the technology condition between G-satellite, I-satellite and M-satellite task for BEIDOU-satellite project, it put forward software multi-state customize preservation technology, which accomplished software multi-task mode quick switch and recovery, and actualized the goal of executing dissimilarity satellite model task alternatively for experiment information surveillance display software, by applied reflex technology, stored and managed N species of task environment at the same time. This method could adapt to future operation platform and the change of task mode.

The foundation frame of software multi-state customized management based on reflex mechanism enormously enhanced the elasticity of application system exploitation for software multi-state customized management.Through configured the parameters for the application program by XML, and loaded it into the allocation table when system start-up, it attained the reflex infusion similar Web exploitation. In addition, it actualized relax combination for software structure while it had no interlink compile between each function module and system frame, and the function relativity was reduced greatly. The system frame offered basal software multi-state customize preservation function, while the display of the function could be controlled through modifying the common set-up interface. When it required to exploit the software multi-state customized management application for command display on the basis frame of software multi-state customized management, it only needed to inherit basis interface, exploit tool and produce the corresponding application program collection, i.e., executable file or dynamic interlink library. By means of the allocation of the application program collection, it could control the status in basis frame thereby achieved the purpose which was to allocate the system dynamically in operation stage, not in compile stage. Adopting the reflex mechanism, the basis frame of software multi-state customized management didn't need the compile and link of the concrete functional application program collection, it captured concrete functional example by reflex visit module in operation on reflex mode, and transferred by the system [1].

Correlation Principle

Reflex mechanism and principle. In 1982 the concept reflex was put forward by Smith for the first time. It mainly means a sort of ability of the program that could visit, examine and modify itself's status or behavior. In computer science, reflex means a kind of application which could be described and controlled by itself. This kind of application enforced self-representation and examination by adopting some species of mechanism. According to itself's behavior status and consequence it could adjust or modify the behavior status and correlative semanteme described by the application. In .NET, reflex was a sort of powerful tool. It could build agile code while these codes could be assembled in operation, and didn't need to conduct source code interlink between the modules. During writing and executing program reflex permitted the written program code to be joined and loaded into the interior information of the class in executable program, not the synergic code of the class adopted in source code. This made reflex to be turned into the main tool of construct agile application[2]. The relation between application program and exterior program collection is shown as figure1.

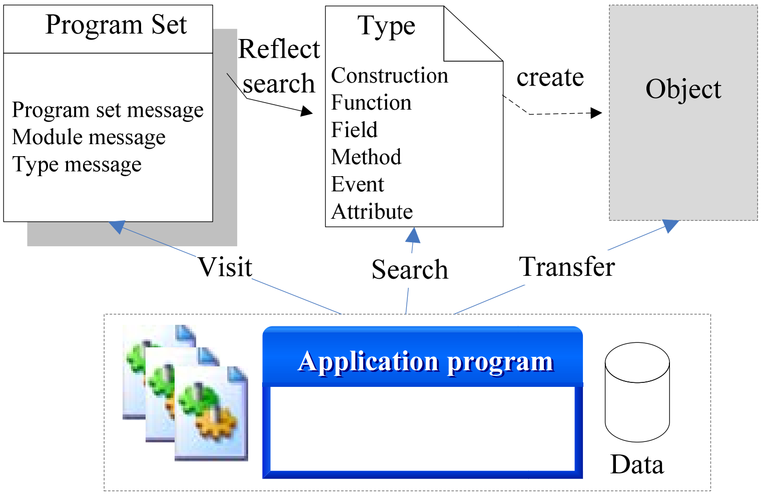

Fig.1. The relationship between application program and exterior program collection

By way of the reflex characteristic it could dynamically load the types in the application program collection, parse its construction, extract the method in the types and transfer it during the system operation. Through interface agreement the application program didn't need to know the concrete actualize detail of the types, but need to change the types into the interface example. At the same time, as the detail information of the interface was known by the application program it was equivalent to fulfill the transparency transfer of the types.

Multi-state performance application. The definition of multi-state performance was: if the same operation acted on example of different class it would generate different outcome, namely when the object of different class received the identical message it would get different consequence. Multi-state performance included the compiling multi-state performance and the multi-state performance of operation[3].

a. compiling multi-state performance: compiling multi-state performance was accomplished passage reload. To the non-virtual member, on the basis of the transferred parameter the information such as the returned types decided which operation would fulfill in system compiling.

b. multi-state performance of operation: the multi-state performance of operation just indicated that it would decide fulfill which operation by reality situation until system operation. The multi-state performance of C# was actualized by covering write virtual member.

The experiment information surveillance display software executed dissimilarity satellite model task, as the software operation status had different operation environment on the basis of the task dissimilarity. By combined the multi-state performance of experiment information surveillance display software with reflex mechanism, constructed multi-state customized management system of command display software, and deposited N sorts of task environment, it would be easy to execute different task alternately for experiment information surveillance display software.

Basic Framework Design of Software Multi-state Customized Management by Reflex Mechanism

The core content of multi-state customized management system was data. Data visit and data representation was the important component of the system. In the application program related with data the most in common use construction was "model-view-controller" triple construction mode. The basis frame of software multi-state customized management subsystem design adopted this mode. The form of data source was lucid to application system in model end. But as a application frame the system actualized basic data visit interface. In view end it was composed with Toolbar, planar data representation window, status and control column, chart layer control window and so on by adopted user interface founded Windows forms. In control end, which corresponded with business logic layer of the frame, it offered the input and output of data, and the actualization of business logic[4]. The frame actualized the basal software multi-state customized management function, included align, selection demand, and others. The frame construction is shown as figure 2.

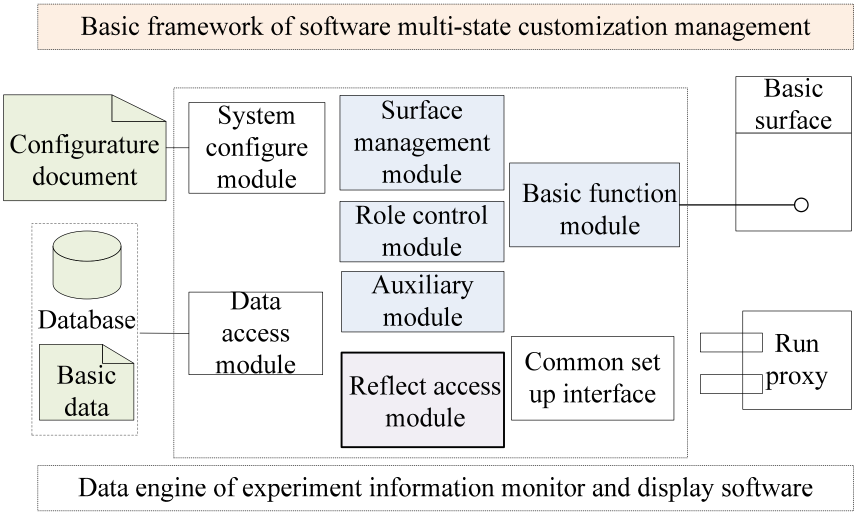

Fig.2. Basic framework composition

It built an incompact construction on the experiment information surveillance display software Data Engine for the software multi-state customized management basis frame which further administrated surface and offered interface. It was composed with below several parts:

a. Experiment information surveillance display software Data Engine: frame basis, as the core of software multi-state performance data expression;

b. System configure module: control system start-up flow, which loaded and preserved configure information, set up system surface and all sorts of parameter on the basis of configure information , in charge of changing and saving system configure information;

c. Data access module: offered basic data access, included XML, INI and others, and designed a series of access interface by factory mode for subsystem to actualized;

d. Surface administrate module: which actualized surface administrate on the basis of system initialize parameter and all, included window status, scene control, etc, administrated system element such as menu, child window and that;

e. Role control module: offered function-level role (guest, administrator) control, which conducted code matching for system load function so as to confirm the function usability, as the visibility of function menu was controlled by surface administrate module;

f. Assist module: offered basis function, document visit, mathematics function, log output and others;

g. Basis interface: defined the interface of ITool, ICommand and so forth, as the function restrain for basis frame. Child system function should be inherited from these interfaces, so as to be identified by basis frame and use operation proxy class as the function initialize parameter;

h. Reflex visit module: the core of the frame and the basis of relax structure. It offered types information of application program collection outside visit frame. If types was actualized from basis interface, it could found an example as function to load to program, and found the function menu term for user transferring;

i. Operation proxy module: offered one operation proxy class, which saved the current information such as system status, user status, function status, and as the intermediary for each module to transfer parameter. In addition, as frame was reposed as a sort of relax construction, each application module was unable to visit mutually without interlink, therefore their correspondence also were actualized by event message offered from operation proxy modular;

j. Common set up interface: when it constructed software multi-state customized management application on basis frame, it needed to change some basis set up for the basis frame, such as applied title, icon, about information and so forth[5]. The start-up flow is shown as figure 3.

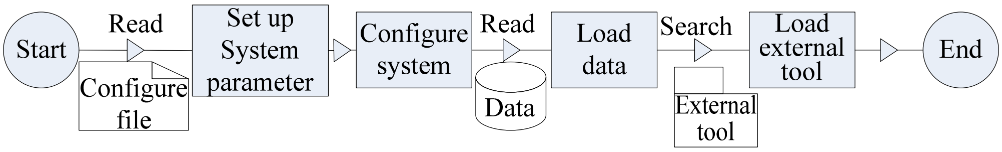

Fig.3. Basic framework start-up flow

Prototype Actualization for Command Display Software Multi-state Customized Management System Based on Basic Framework

As a system prototype, it actualized the function in common use for multi-state customized management: display chart layer administrate, data types statistics and trajectory data trend analysis, and all that. In basis frame it is appointed that all the function of software multi-state customized management subsystem should be inherited from basis interface. In it, software chart layer administration was inherited from ICommand, to control the on-off of display chart layer; trajectory data trend analysis function was inherited from ITool, to offer the function of mutual operation for user[6]. As the mode of plug-in components, after being discovered by the search mechanism of basis frame the function was dynamically configured into system menu for user call. Regard as one part of which the role controlled, the function required to be configured before entering into the system to offer the correlate information such as menu path, jurisdiction grade. The construction of software multi-state customized management subsystem is shown as figure 4.

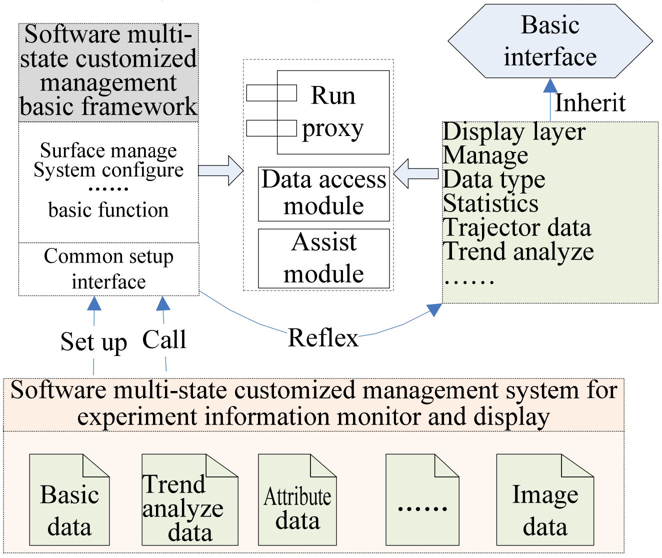

Fig.4. Construction of multi-state customized management system

Conclusion

The software multi-state customized management subsystem was constructed on the basic frame of software multi-state customized management, while the function class was inherited from basis interface, each parameter offered from basis frame was obtained by operation proxy class, as well as the correspondence with the child module of basis frame used event message mechanism. As software multi-state customized management subsystem itself required to visit its professional data, if basis frame could not offer the actualization of given types data visit, it needed to inherit the factory class of data visit module, to achieved its data access. It required carrying on some set up to basis frame after the actualization of function module, as it could be fulfilled by common set up interface. The parameter needed by software multi-state customized management subsystem could be offered by configured the configure document of basis frame. In the document except the system parameter was saved, a parameter table on account of key-value fit was also saved. Finally, the multi-state customized management subsystem for experiment information surveillance display software transferred the start-up function of basis frame and start-up system.

References

[1] Wang Xuefeng. Creating Object Browser Through .NET Reflection Technology [J]. Computer Programming Skills & Maintenance, 2010: 16 (2) 34-35,63.

[2] Zhan Zedong,Xie Shuang,Zang Wenya1,Lu Hua1,Li Zhengmin. Using Reflection Technology to Implement a New Kind of Software System Maintenance [J]. Microcomputer & Its Applications. 2012:31 (1):70-72.

[3] Wu Fei. Loosely Coupled of Data and System Interface Design Base on Net Reflection Technology [J]. Geospatial Information. 2011:8 (4):118-119.

[4] Zhong Guangrui. Research on the Plug-in Design for Application Framework of GIS by.NET Reflection Technology [J]. Geospatial Information. 2011:8 (9):4-7.

[5] Li Jing. Application on Web Development Based on .NET Layered Architecture and Abstract Factory Design Pattern[J]. Software Guide. 2015: 4 (14):105-107.

[6] Liang Fenglan , Yu Qihong. Using Reflection Technology to Implement the Design of Dynamic Menu [J]. Journal of Changzhou Institute of Technology. 2010: 6 (23):56-59.