Nuclear Engineering and Design 328 (2018) 345–352

Contents lists available at ScienceDirect

Nuclear Engineering and Design

journal homepage: www.elsevier.com/locate/nucengdes

Reliability analysis and allocation: Development of a hierarchical structure modeling platform in I&C system Software Life Cycle

Bowen Zou[a], Ming Yang[a], Jun Yang[a], Jia Guo[b], Yanqin Su[c], Chao Zhang[c], Wenlin Wang[d,*]

[a] School of Electric Power, South China University of Technology, Guangzhou 510641, China
[b] Fundamental Science on Nuclear Safety and Simulation Technology Laboratory, Harbin Engineering University, Harbin 150001, China
[c] Engineering Center, State Nuclear Power Automation System Engineering Company, Shanghai 200241, China
[d] School of Automation, Wuhan University of Technology, Wuhan 430070, China

ARTICLE INFO

Keywords:
Software reliability analysis
Software reliability allocation
Software reliability hierarchical structure modeling

ABSTRACT

A Software Life Cycle (SLC) is used to describe the phases of software cycle and ensure the good quality software is built. A Software Reliability Hierarchical Structure Modeling (SRHSM) is proposed for the analysis and allocation of the software reliability before it is used. A software system can be regard as hierarchical and composed of a set of interacting system elements which implement to fulfill its respective specified requirements. Based on the previous work, the concepts of SRSHM method is optimized and increased some basic elements. The SRHSM method comprises two processes in SLC, including Partition and Composition process and divides the system into four levels: System level, Subsystem level, Unit level, Code level. A software partition process is to allocate the software reliability to each module and composition process is applied for the analysis of software reliability.

1. Introduction

In software engineering, Software Life Cycle (SLC) means the project-specific sequence of activities that is created by mapping the activities of this standard (IEEE Standard for Developing Software Life Cycle Processes) onto a selected Software Life Cycle Model (SLCM) (Schultz, 1997). In other words, a project-specific mapping of sequence tasks (any large complex system is divided into tasks or groups of related tasks) in progressive evolution from concept stage to retirement stage will be used for engineers to produce software and perform their tasks (Horie et al., 2009).

IEEE Std 24,748-1TM-2011 (2011) divided SLC into six stages, a) Concept Stage, to develop preliminary system requirements include system analysis, estimations, trade-off study, etc., and conclude a feasible design solution; b) Development Stage, to develop a system-of-interest that meets stakeholder requirements, and the system-of-interest may be a prototype in this stage; c) Production Stage, to produce the system-of-interest, which may undergo enhancements or redesign; d) Utilization Stage, to operate the product and provide sustainable operational services; e) Support Stage, to provide logistics, maintains, and services to support a sustainable utilization of the product; f) Retirement Stage, to provide some services about the removal of the system-of-interest and related operational when the system is taken out of service. This paper only considers three stages (that is a, b, and c stages) to conduct the hierarchical structure modeling, allocate and access the reliability, and assess the confidence coefficient of software reliability in the SLC.

International standard (ISO/IEC/IEEE: 15288, 2015) demonstrates that a software system can be hierarchical and composed of a set of interacting system elements. A system element is implemented to fulfill its respective specified requirements. As a complicated system, to implement the fast analysis, a system element can be regarded as a system.

Hierarchical modeling is almost used in all fields to promote the analysis progress of a complex system. Delahaye et al. (2005) proposed a functional model based a hierarchical modeling approach in Software Defined Radio systems (SDR systems). They distinguished two approaches to design the system, which are a top-down approach that called the common function approach and a bottom-up approach that called common operator approach. SDR systems can be composed of 3 levels of hierarchy. Level 1 is as a host for the 3 clusters, Level 2 dedicates several configuration managers to each hardware components, and Level 3 means each configuration manager is responsible for a processing component.

Also, in the software reliability analysis field, *USES* relationship was presented by Leblanc et al. (2002) to demonstrate a method about how to estimate the reliability of hierarchy software system. The

* Corresponding author.
E-mail address: wlwang0618@126.com (W. Wang).

https://doi.org/10.1016/j.nucengdes.2017.12.020
Received 29 April 2017; Received in revised form 5 December 2017; Accepted 11 December 2017
0029-5493/

relationship of *USES* was defined by formula $USES(M_i,M_j)$. The conditions of this equation are established by Module M_i calls Module M_j and M_j determines the success and failure of M_i. The method of *USES* assigns modules to the level of a hierarchy by the following rules:

a) Level 0 is the bottom layer of system hierarchical model;
b) Level i ($i \geqslant 1$) is the higher layer than Level $i-1$ and contain more than one Level $i-1$.

Analytical Hierarchy Process (AHP) is applied as software reliability hierarchical method and considers all factors in each system. On the basis of AHP method, Zahedi and Ashrafi (1991) presented a method used Structure, Utility, Price, and Cost for the allocation of reliability. AHP method can analyze the software structure, while incapable to design and illustrate software hierarchical structure (Vaidya and Kumar, 2006).

Watson et al. (1996) adopted hierarchical theory to modularize the software to have the complexity analysis. This method puts the test results of the same level into the upper level to assess the software complexity, via modular test.

Bansiya and Davis (2002) proposed a hierarchical model for object-oriented design quality assessment and used hierarchies to represent different generalization-specialization concepts in a design. This methodology used in the development of the hierarchical Quality Model for Object-Oriented Design (QMOOD) assessment, and followed the four basic levels (L1 through L4) and three mappings (like: L_{12}, L_{23}, and L_{34}).

For the software structural decomposition, Alkhalid et al. (2013) adopted clustering techniques, including Hierarchical Agglomerative Clustering (HAC), Adaptive K-Nearest Neighbor (A-KNN) Clustering, and Fuzzy C-Means Clustering (FCM), to realize the reasonable decomposition of software. The decomposed software module can be independent developed. The reasonable decomposition of software structure and functions in the early stage of software development and the traceability of software decision-making in the design stage can develop high-quality software system. (Chang and Lu, 2009).

A software design is updated with the progress of software development. For the analysis of software reliability in the different stages of SLC, many methods have been have presented. Also, there have many software reliability allocation methods, while mismatch the software reliability analysis method. The authors have a previous study on the reliability analysis of the digital instrumentation and control software system (Zou et al., 2017).

In this paper, a method based on the Flow Network Model (FNM) approach to design and analyze software by a software reliability hierarchical structure modeling (SRHSM) platform in SLC. In the previous work, the authors have introduced the initial preliminary study of SRHSM (Yang et al., 2014). This paper optimizes the SRHSM and supplements some basic elements for the better modeling and analysis of software. By the SRHSM platform, analyst can accomplish the allocation and analysis of the software. The results of software analysis can verify and rectify the reasonability of software allocation at some stage due to the same theory (like FNM) both used.

2. Software reliability hierarchical structure modeling

2.1. The concepts of hierarchical modeling method

A large and complex system can be modeled and analyzed quickly with the help of the hierarchical modeling. Batory and O'malley (1992) indicated that a hierarchical system is a modeling tool featured as acyclic and graphical expression, where node represents module and edge represents the relationship of two modules. Kahraman and Çebı (2009) illustrated that more than three levels comprise a system, including top level which covers total problems and targets, middle level which defines standard alternative elements, and bottom level which computes alternatives. As shown in Fig. 1, module A contains module B and C, and the module B and C call system X together which is a series system comprised by module D and E.

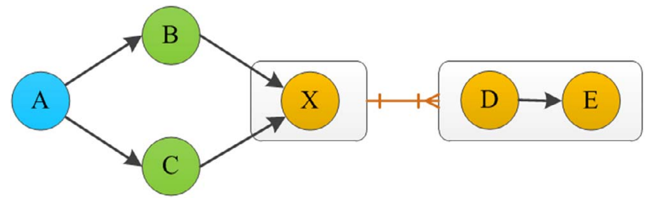

Fig. 1. A case for a common hierarchical software system.

It is a common way to analyze complex software system (for example, a digital I&C system in nuclear power plant) by hierarchical structure modeling. For example, assuming a code block has about 100 lines of code, 14 condition statements or more. If the test paths of this code block developed by this condition statements, the number of path will be increased exponentially, that is, the number of path can be reached more than 10,000 ($2^{14} = 16384$). Even if the most accepted method to analyze a code block which has 10,000 paths, the analysis data would be easy to deviate from the actual results, the reasons are that

a) It is difficult to implement full coverage test and analyze for the above code block;
b) It is difficult to find the sensitive paths, while easy to test the same path which will result in incomplete reliability analysis;
c) It is a time consuming project to test and analyze this code block completely.

Thus, SRHSM approach can solve the above problems and refine the model to locate the defects directly to the corresponding module in a better way. Based on the software structure, SLC and other hierarchical modeling, some SRHSM regulations are proposed in this paper.

a) Each level has its explicit information flow;
b) The sub-module of each level should be partitioned rational. The standards of partition rational can accord the module functions, module complexity, etc.;
c) The relationship between levels should be clear and rational. The number of levels should be moderate. Too many levels, it will increases the computational complexity; too few levels, it cannot be simplified software.

In this paper, SRHSM is divided into four levels based on the SRHSM regulations.

a) System level: a holonomic system comprised by subsystems can represent the entire characteristics and be judged whether met the requirements or not. The decision makers can make effective decisions in this level.
b) Subsystem level: between system level and unit level. The number of subsystem level due to the software complexity.
c) Unit level: indicate the information flow of each code block, in other words, every unit module can be used as an independent test module. It is worth nothing that a defect found in unit level can be considered as a programmer coding bugs and located the problem specifically.
d) Code level: it refer to underlying code for analyzing code block, including complexity, the number of rows, test time, etc.

As shown in Fig. 2, A is system level, B is subsystem level, and C is unit level. Detailed code can be found in Code level, software detailed structure can be found in unit level. Some decisions-making about reliability allocation and the test domain can be accomplished in