Transactions of the Korean Nuclear Society Autumn Meeting
PyeongChang, Korea, October 25-26, 2007

Dependability Engineering in Software Life Cycle Process

Y. M. Kim and C. H. Jeong
Korea Institute of Nuclear Safety
Ymkim@kins.re.kr

1. Introduction

Recently, with the rapid development of digital computer and information processing technologies, nuclear I&C (Instrument & Control) system which needs safety-critical function has adopted digital technologies. Software used in safety-critical system must have high dependability.

Dependability of the software may have several different attributes such as reliability, safety, confidentiality, integrity, availability, and real-time response. Also, such attributes need different levels of adherence. For shaping of the dependability, there are several dependability processes: fault prevention, fault tolerance, fault removal and fault forecasting.

In this paper, we present an integrated model of dependability processes and software life cycle processes and dependability task.

This paper is organized as follows. Section 2 describes related research by surveying dependability of the software and dependability processes. Section 3 describes our integrated dependability model and Section 4 shows the dependability task by the development phase. Section 5 concludes the paper.

2. Related Research and Background

2.1 Dependability of the Software

Dependability is commonly recognized as an integrative concept that encompasses different attributes. It is defined as the trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers [1]. Dependability may be viewed according to different, but complementary properties, which enable the attributes of dependability to be defined such as reliability, safety, security, availability, survivability, maintainability, etc. The choice of the attributes depends on the applications and the stakeholder's needs. The definition of the attribute is also various. Some of the attribute's definitions are as follows.

- Reliability is the continuity of correct service (a service is correct when it implements its specification) [2].
- Safety is the nonoccurrence of catastrophic consequences on the environment (this means freedom from those conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment) [2].
- Security is the ability of the system to deliver its required service without unauthorized disclosure or alteration of sensitive information, or denial of service to legitimate users by unauthorized persons [3].

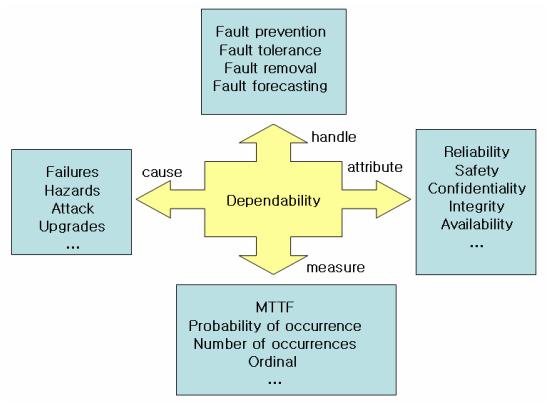

Figure 1 Overall characteristics of the S/W Dependability

2.2 Basic Processes of the Software Dependability

Figure 1 shows the overall characteristics of the S/W dependability. Dependability can be handled by fault prevention process, fault tolerance process, fault removal process and fault forecasting process. Fault prevention refers to the way to prevent the occurrence or introduction of faults and fault tolerance means how to deliver correct service in the presence of faults. Fault removal refers to the method how to reduce the number or severity of faults and fault forecasting means how to estimate the present number, the future incidence, and the likely consequences of faults [4, 5].

3. Dependability Process in Software Life Cycle

ISO/IEC 12207 and IEEE Std. 1074 are representative software engineering standards which are related to the software life cycle. ISO/IEC 12207 presents the software life cycle model and IEEE Std. 1074 presents the specified method for implementing software life cycle. The primary life cycle process includes acquisition, supply, development, operation and maintenance [6]. Dependability processes such as fault prevention, fault tolerance, fault removal and fault estimation must be executed over a software life cycle. In this paper, we focus on only the development process.

4. Dependability Processes for the Software Development

The phases of software development include concept phase, requirement phase, implementation phase, test phase and install and checkout phase. In this paper, we present the list of the dependability tasks for the dependability attributes: reliability, security and safety by each phase. The dependability processes must be performed from the early stage of the development phase. Especially, for fault prevention process, many tasks are executed in requirement phase. Also, the result of the fault prevention tasks provides the necessary information for other dependability processes. In this paper, we mention only the fault prevention process during concept phase and requirement phase.

4.1 Fault Prevention Process and Software Life Cycle Process

Fault prevention process aims at preventing fault previously and is related to the development plan, organization, standard and tool. These activities are determined in the early phase and affect the product continuously during software life cycle. Table 1 shows the primary fault prevention task during the concept phase and the requirement phase.

In the concept phase, for the improvement of the reliability, fault prevention process has to identify the user's needs and concept documentation. For the safety, it must determine software integrity level (SIL) of each software and must identify the potential hazard and technical and managerial risk. For the security, it must identify the allowable level of the user for the security risk.

In the requirement phase, for the reliability, the traceability of the software requirement must be analyzed in the aspect of correctness, consistency and completeness. Also, for the safety, risk analysis must be performed every phase. For the security, fault prevention process must identify the system security requirement which was identified in the concept phase.

4.2 Fault Removal Process and Software Life Cycle Process

Most of the verification and validation activities aim at the prevention and removal of the software fault. Fault prevention tasks which are executed during software life cycle are mostly considered during fault removal process as a view point. In the concept phase, for reliability, the requirement of the system must be analyzed and software concept must be identified. For safety, user's requirement for fault detection, fault isolation, fault diagnosis and error recovery must be identified. In the requirement phase, for reliability and safety, system test plan and acceptance test plan must be prepared. Every test plan, test design, test procedure, test case, test execution must be performed properly by SIL of the software.

Table 1 Example of the Dependability Process Tasks

Phase	Fault Prevention Task
Concept	*Reliability* ● Identification of the concept document and user's need ● System requirement analysis *Safety* ● Determination of the software safety integrity level(SIL) ● Identification of the potential hazard from the conceptual system ● Identification of the technical, managerial risk ● Risk analysis *Security* ● Identification the allowable level of the users for the security risk
Require-ment	*Reliability* ● Correctness, consistency, completeness analysis of the requirement *Safety* ● Review the predetermined SIL ● Identification of the requirements which affect system hazard ● Review and update the risk analysis *Security* ● Identification of the system security requirement which identified in concept phase

5. Summary

In nuclear field, they have strived to guarantee absence of failures of safety-critical applications development. In this research, we present dependability process model and dependability tasks which must be executed during product development. Also, we classified them into several dependability attributes. The dependability tasks can be customized appropriately for their application. In the future, we will improve the dependability process model with dependability measurement and assessment.

REFERENCES

[1] Laprie, J.-C., Dependability: basic concepts and terminology, dependable computing and fault-tolerant systems. Springer Verlag, Wien-New York.
[2] Michael R. Lyu, Handbook of Software Reliability Engineering, 1996.
[3] US Department of Defense, 2000. Standard practice for system safety, MIL-STD-882D.
[4] Paolo Donzelli, Victor Basili, A practical framework for elicting and modeling system dependability requirements: Experience from the NASA high dependability computing project, The Journal of Systems and Software, 2005
[5] Jean-Claude Laprie, A framework for dependability engineering of critical computing systems, Mohamed Kaaniche, Safety Science, Volume 40, December 2002
[6] ISO/IEC 12207, 1995