Selecting Optimal COTS Products Considering Cost and Failure Rate

Ho-Won Jung
College of Business Administration
Korea University
hwjung@kuccnx.korea.ac.kr

Chang-Shin Chung and Kyu Ouk Lee
K-R Collaborative Development Team
ETRI
cschung@nice.etri.re.kr Kolee@etri.re.kr

1. Introduction

A reliable and cost effective system is crucial in satisfying the requirements of users and managers alike. One way to develop such a system is to use existing COTS (Commercial Off-The-Shelf) software products. As an appropriate process model for a software development project, Boehm [2] recommended the use of COTS products in all possible applications. In the US, COTS products are mandated by several large contractors such as the Department of Defence's Message System and NASA's X2000 technology for deep-space mission [12]. The use of COTS products is also viewed as a way to reduce schedule delay and cost overrun in software development projects.

This study addresses an optimization model to develop a modular software system. The system is comprised of a set of serially-executed modules, where each module is configured with only one COTS product among several alternatives subject to an overall system failure rate. The objective function of the model minimizes a system development cost (initial cost).

The software system configuration of our model is the same as based on the first reliability model of Berman et. al [1], where the initial cost and reliability were only considered. In contrast, our model includes the system development cost and a system failure rate. Kubat [1] presented a stochastic model to minimize a cost subject to an overall system failure rate. The overall system failure rate of our model is derived with a similar approach in Kubat. Our model does not considered the inter-module failure rate addressed by Littlewood [10], where he stated the inter-module failures as ``Unfortunately, there seems to be little information available about interfacing failures...''. Helander et. al [5] introduced two models (a reliability constraint cost-minimization and a budget constraint reliability maximization) for distributing development effort among software components. Jung [7] presented an optimization model to optimize the life-time costs, i.e. the sum of initial and risk cost. His risk cost is similar to the risk cost introduced by Pham and Zhang [11].

Our model is an integer problem and can be utilized in various phases of system development to manage cost and system failure rate. The following section derives an optimization model and final remarks are presented at the end of this paper.

2. Optimization Model

The following denotes our notations, where $j = 1, \ldots, n_i$ for $i = 1, \ldots, m$.

m	Number of modules within the software system
n_i	Number of products available for module i
c_{ij}	Purchase cost of product j for module i
l_{ij}	Failure rate of product j for module i
f	Call arrival rate to the system
e_f	Maximum-allowable overall system failure rate
s_{ij}	Sojourn time of a call in module i with product j

$$x_{ij} = \begin{cases} 1 & \text{if product } j \text{ of module } i \text{ is selected;} \\ 0 & \text{otherwise.} \end{cases}$$

The failure rate and the sojourn time given by COTS's developers would be estimated during the system prototyping, module testing, and the early stage of integration testing [8]. The call arrival rate, maximum-allowable system failure rate and delay will be estimated and then described in a specification.

In the underlying model, the initial cost for the modular system development includes the procurement cost of COTS products, where each module should be configured by a single COTS product. These concepts can be represented by:

$$\sum_{i=1}^{m} \sum_{j=1}^{n_i} c_{ij} x_{ij} \ , \qquad (1)$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad i = 1, \ldots, m, \qquad (2)$$

where x_{ij} has a value of 0 or 1.

System failure rate is defined in the same way as hardware failure rate as 1/MTBF, where MTBF (Mean Time-Between-Failures). This study makes the same assumption as [10] that the MTBR (Mean Time-To-Recovery) of software system is very small compared with the MTBF. In order to find an overall system failure rate of the underlying system configuration, the failure rate l_{ij} is assumed to be constant (note that the distribution between failures follows an exponential). In addition, the sojourn time s_{ij} is assumed to be constant.

With the above assumptions, let p_i define the probability of at least one failure occurrence during the module i execution of a call. Then, the probability of no failure occurrences during an execution of module i is:

$$1-p_i=\exp[-\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}],\quad \text{for all } i,$$

which represents the probability of no failures occurring in a Poisson distribution with parameter $\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}$. Note the Poisson probability law that the probability of a failure occurrence is proportional to the length of the sojourn time.

Under the assumption of s-independent operation of each module, the probability that no failures happen to the software system is as follows:

$$\Pi_{i=1}^{m}(1-p_i)=\exp[-\sum_{i=1}^{m}\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}].$$

Accordingly, the probability that at least one failure will occur in the system during the execution of a run for a call becomes:

$$1-\Pi_{i=1}^{m}(1-p_i)=1-\exp[-\sum_{i=1}^{m}\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}].$$

The overall system failure rate q with the call arrival rate f becomes as follows:

$$q=f(1-\exp[-\sum_{i=1}^{m}\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}]).$$

For a mathematically tractable model, the use of the first two terms of Maclaurin series expansion to the exponential term gives an approximation value of the overall system failure rate as follows:

$$q=f\sum_{i=1}^{m}\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}. \quad (3)$$

Since the COTS failure rate and the sojourn time have small values, the remaining term of the Maclaurin series can be considered negligible.

Using (1)–(3), an optimization model for selecting a COTS product for each module becomes:

$$\text{Min} \quad \sum_{i=1}^{m}\sum_{j=1}^{n_i} c_{ij}x_{ij}$$

subject to

$$f\sum_{i=1}^{m}\sum_{j=1}^{n_i} l_{ij}s_{ij}x_{ij}\le e_f,$$

$$\sum_{j=1}^{n_i} x_{ij}=1 \quad i=1,\ldots,m,$$

$$x_{ij}=0/1 \quad \text{for all } i \text{ and j}.$$

The underlying optimization model is a binary integer program. It can easily be solved by mathematical codes such as Lingo [9]. Since the constraint of the system failure rate is tight, a sensitivity analysis can be carried out to see the changes of optimal solution by the changes of e_f.

Final Remarks

Our optimization model selects a set of optimal COTS products of a modular software system, while minimizing the system development cost subject to an overall system failure rate. Our model assumes statistically independent operation for COTS products in a serially-connected modules. This assumption is not correct in the operation for COTS products of fault-tolerance software systems configured as the second and third models of Berman et. al [1]. See [3] and [4] in details.

The underlying model assumed that all COTS products could be purchased individually. However, at times a COTS product needs to be purchased together with other COTS products due to problems such as implementation technology, interfaces, and licensing. This condition can be easily represented by contingent decision in the model.

References

[1] O. Berman and N. Ashrafi, "Optimization models for reliability of modular software systems", *IEEE Trans. on Software Eng.,* vol. 19, no. 11, 1993, pp. 1119–1123.

[2] B.W. Boehm, *Software Risk Management* (Tutorial), New York: IEEE Computer Society Press, 1989.

[3] D.E. Eckhardt, Jr. and L.D. Lee, "A theoretical basis for the analysis of multiversion software subject to coincident errors", *IEEE Trans. on Software Eng.,* vol. SE-11, no. 12, 1985, pp. 1511–1517.

[4] D.E. Eckhardt, Jr., A.K. Caglayan, J.C. Knight, and L.D. Lee, "An experimental evaluation of software redundancy as a strategy for improving reliability"', *IEEE Trans. on Software Eng.,* vol. 17, no. 7, 1991, pp. 692–702.

[5] M.E. Helander, M. Zhao, and N. Ohlsson, "Planning models for software reliability and cost", *IEEE Trans. on Software Eng.*, vol. 24, no. 6, 1998, pp. 420–434.

[6] F. Hiller and G. Lieberman, *Introduction to Operations Research*, 6th Ed, McGraw-Hill, New York, 1995.

[7] H-W Jung, "Optimizing the life time cost of a modular software system", under review in a journal.

[8] P. Kubat, "Assessing reliability of modular software", *Operations Research Letters*, vol. 8, 1989, pp. 35–41.

[9] LINDO Systems Inc., LINGO Optimization Modeling Language, Chicago, 1993.

[10] B. Littlewood, "Software reliability model for modular program structure", *IEEE Trans. on Reliability*, vol. 28, no.3, 1979, pp. 241–246.

[11] H. Pham and X. Zhang, "A software cost model with warranty and risk cost", *IEEE Trans. on Computers*, vol. 48, no. 1, 1999, pp.71–75.

[12] N. Talbert, "The cost of COTS", *Computer*, 1998 June, pp. 46–52.