A New Method for Estimating the Reliability of Software System Based on Components

Zhao Hui-Qun   Sun Jing   Gao Yuan
(Northeastern University Qin-Huang-Dao Campus 066004 P.R.China.Zhaohq-mail@263.net)

Abstract: For a typical software system, it is generally considered infeasible to calculate system reliability from the reliabilities of its constituent components because software systems, unlike hardware systems, tend to violate the underlying independence assumptions inherent in the usual reliability calculations. In this paper, we present new method for calculating the system reliability from its constituents, components and connectors, reliability.

Keywords: software reliability, sime-Markov, components, connectors

1 Introduction

In the area of software reliability engineering, much attention is focused on means by which the reliability of software systems can be estimated through testing. Reliability models typically consider the software in question to be autonomous. Thus, reliability estimations for a software system are normally employed at the system level; testing must be performed at this level in order to generate the underlying data for the estimation [1]. However, there is growing interest in techniques by which the system reliability can be estimated from the reliabilities of its constituent components. This technique is pragmatically appealing. Because of approaches such as code reuse, COTS use, modularity, and information hiding, reliability estimates may be available for components of a system; in such situations, it may be more efficient to calculate system reliability from these existing estimates, rather than expend further resources at the system level, for the purpose of reliability estimation. Paper [2] proposed a modular approach to reliability estimation for software system. It suppose that a software system can be divided into independent components and the reliability of each components was provided by vendor so that the overall system reliability can be calculated using well known Markov analysis techniques in software system. However, it does not take the reliability of connector between a pair of components into account. From the architecture point of view, software system is typically composed of components and connectors, so the reliability of connector should be added into the reliability of overall software system. In this paper, we present a refined approach for estimating software system reliability from modular reliability.

2 Reliability Estimation

Different with [2], We suppose the software system is composed of components and connectors. In order to estimate the reliability of a modular system from the reliabilities of its constituents, we require (1) an estimate of the reliability of each constituent and (2) some description of how the constituents are expected to interact during system operation. Individual component and connector reliability is estimated using standard methods, such as those described in [3]. Early study has shown that an underlying Simi-Markov model can be derived from software architecture based on components and connector [4], so that a description of constituent interaction is typically stipulated using a first order Markov chain (1OMC). Without loss of generality, let us name each component $C_1,C_2,C_3,\ldots C_n$ as state of 1OMC, their reliability represent as p_i $i\in[1; n]$. Each transition probability,

$p_{i,j}$; i,j∈[1; n] is a real number∈[0; 1] such that $p_{i,j}$ is the probability that the system will next execute component C_j, given that it is currently executing component Ci. Similar with $p_{i,j,}$ the probability of each connector between components C1 and C2 represent as $q_{i,j}$ i,j∈[1; n] is a real number∈[0; 1], so the connectivity between i and j can note as $l_{i,j}= p_{i,j} * q_{i,j}$. Based on the definition above, the sime-Markov reliability model is as follow:

$$\begin{bmatrix} C_{1n} \\ C_{2n} \\ . \\ C_{nn} \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12}\ldots & L_{1n} \\ L_{21} & L_{22}\ldots & L_{2n} \\ \ldots & \ldots & \\ L_{n1} & L_{n2\ldots} & L_{nn} \end{bmatrix} \begin{bmatrix} C_{1n} \\ C_{2n} \\ \ldots \\ C_{nn} \end{bmatrix} \quad (2.1)$$

Meanwhile $L_{ij}(t)=l_{ij}*p_i(t)$, here $p_i(t)=P\{T_{n+1}-T_n\leq t|\ Z_n=i\ ,Z_{n+1}=j\ \}$is reliability of state C_i. In (2.1), we suppose that C_{1n} is start state and C_{nn} is terminal state for successful.

For example: there five component software architecture,$C_1;C_2;C_3;C_4;C_5$ and their use single connector L with probability 0.99 for connectivity from C_1 to C_5. The probability of each constituent is as follow:

$p_1=0.99,p_2=0.99,p_3=0.99$ and $p_4=0.98$, p5=1;

$p_{1,2}=0.2$, $p_{1,3}=0.5$, $p_{1,4}=0.3$, $p_{2,1}=0.7$, $p_{2,2}=0.3$, $p_{3,4}=0.5,p_{3,5}=0.5$ and $p_{5,2}=0.1$. (2.2)

$l_{1,2}=0.189,l_{1,3}=0.459,l_{1,4}=0.297,l_{2,1}=0.693,l_{2,2}=0.297,l_{3,4}=0.495,l_{3,5}=0.495,l_{5,2}=0.099$.

Replacing (2.1) with (2.2) can get (2.3).

$$\begin{aligned} R_{1,5}&=0.264+0.264R_{2,5}+0.441R_{3,5} \\ R_{2,5}&=0.97R_{1,5} \qquad (2.3) \\ R_{3,5}&=0.49+0.47R_{4,5} \\ R_{4,5}&=0.95R_{2,5} \end{aligned}$$

By computing (2.3), we can gain the overall system reliability undergoing from state C_1 to C5 $R_{1,5}=0.868$.

3 Conclusion

We outlined ways in which the reliability of software systems based on components and connectors can be estimated by reliability of constituents. The preconditions for model (2.1) are as follow:

(1)Design the system using architecture in which the constituents are logic independent.

(2)Code in a programming that constrains pointer and automates memory management, such as Java.

To make the Model usable in practice, we plan to provide tool support by exploiting PETOOL. The ideal is to allow the user to specify software architectures with the graphical or the textual notation, implement routines for architectural reliability computing, compatibility, conformity checking.

References

[1] MUSA,J.Applying operational profiles in testing. In Proc. 10th Intl. Software Quality Week (San Francisco,CA, May 27 30 1997).

[2] WOIT,D.Specifying component interactions for modular reliability estimation. In Proc. 1st International Software Quality Week Europe(QWE 97).

[3] Lyu,M.,Ed. Handbook of Software Reliability Engineering. McGraw-Hill, New York,1996.

[4] Zhao H.Q, Gao Y. Reliability Analyzing for Software Architecture Based on Components. MINI-MICRO SYSTEMS, pp100~103,3, 2001.