Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Systems and Software 76 (2005) 3–13
The Journal of Systems and Software
www.elsevier.com/locate/jss

Reliability assessment and sensitivity analysis of software reliability growth modeling based on software module structure

Jung-Hua Lo [a], Chin-Yu Huang [b], Ing-Yi Chen [c], Sy-Yen Kuo [d,*], Michael R. Lyu [e]

[a] Department of Information Management, Lan-Yang Institute of Technology, I-Lan, Taiwan
[b] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[c] Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan
[d] Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[e] Department of Computer Science and Information Engineering, The Chinese University of Hong Kong, Hong Kong

Received 15 December 2003; received in revised form 15 April 2004; accepted 15 June 2004
Available online 11 September 2004

Abstract

Software reliability is an important characteristic for most systems. A number of reliability models have been developed to evaluate the reliability of a software system. The parameters in these software reliability models are usually directly obtained from the field failure data. Due to the dynamic properties of the system and the insufficiency of the failure data, the accurate values of the parameters are hard to determine. Therefore, the sensitivity analysis is often used in this stage to deal with this problem. Sensitivity analysis provides a way to analyzing the impact of the different parameters. In order to assess the reliability of a component-based software, we propose a new approach to analyzing the reliability of the system, based on the reliabilities of the individual components and the architecture of the system. Furthermore, we present the sensitivity analysis on the reliability of a component-based software in order to determine which of the components affects the reliability of the system most. Finally, three general examples are evaluated to validate and show the effectiveness of the proposed approach.

Keywords: Non-homogeneous Poisson process; Sensitivity analysis; Software reliability; Software reliability growth models

1. Introduction

Due to the recent rapid developments of computer and network technologies, the Internet and World Wide Web make it possible for users to access a variety of resources and applications distributed over the world. Since software is embedded in computer technologies and permeates our daily life, the correct performance of software systems becomes an important issue of many critical systems. Software designers are motivated to integrate commercial off-the-shelf (COTS) software components for rapid software development. To ensure high reliability for such applications using software components as their building blocks to construct a software system, dependable components have to be deployed to meet the reliability requirements. Therefore, it is necessary to assess the reliabilities of such systems by investigating the architectures, the testing strategies, and the component reliabilities (Lyu, 1996; Musa et al., 1987; Musa, 1998). For instance, Everett (1999) uses the extended execution time (EET) reliability growth model and several test cases to model the reliability growth of each component. Gokhale et al. (1998) and Gokhale and Trivedi (2002) assumes the failure behavior of each component is described by a

* Corresponding author. Tel.: +886 2 2368 9172/2363 525; fax: +889 2 2367 1909/886 2 2363 8247.
E-mail address: sykuo@cc.ee.ntu.edu.tw (S.-Y. Kuo).

0164-1212/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2004.06.025

time-dependent failure intensity. The total number of failures is obtained and the reliability is estimated via the enhanced non-homogeneous Poisson process (ENHPP). Krishnamurthy and Mathur (1999) evaluated a method, known as component based reliability estimation (CBRE), to estimate the reliability of a component-based software system using reliabilities of its components. Yacoub et al. (1999) proposed a reliability analysis technique called the scenario-based reliability analysis (SBRA), which is based on execution scenarios to derive a probabilistic model for the analysis of a component-based system. In the field of software reliability modeling, Musa (1994, 1998) and Musa et al. (1987) first investigated the validity of the execution time theory and operational profile. Recently, we (Huang et al., 2002; Lo et al., 2002, 2003) adopted the concept of testing-effort within an NHPP model to get a better description of the software fault phenomenon. The failure behavior of each component can be described by failure intensity. Thus, an appropriate software reliability growth model can be applied for estimating the component reliability from the failure data. However, due to the insufficiencyof failure data, sensitivity analysis of the reliability estimation has also been investigated in the context of black-box models (Musa, 1994; Chen et al., 1994; Pasquini et al., 1986). The parameters in these software reliability models are usually obtained from the failure data. Sensitivity analysis provides an approach to analyzing the impact of the parameters (Musa, 1993, 1994). In general, one difficulty in estimating the reliability of a system in the testing stage is the insufficiency of the failure data and therefore the accurate values of the parameters are hard to get. Sensitivity analysis is often used in this stage to deal with this problem. In this paper, we investigate the sensitivity analysis of the reliability for a component-based software system. The method is very useful in practice. For example, if we use the approach to determine that a parameter or a component in a system is the most sensitive, it is critical for the software testing-team to have this parameter estimated as accurately as possible or allocate more resources for this component. The organization of this paper is as follows. Section 2 presents an analytical approach to estimating the reliability of a system. Three different architecture styles are conducted in Section 3. Sensitivity analysis is discussed in Section 4. Experimental results of sensitivity analysis of system reliability are depicted in Section 5. Conclusions are presented in Section 6.

2. Reliability prediction for module software systems

2.1. Notations

Pr	probability function
$\mathrm{pow}(x,y)$	power function, i.e. $\mathrm{pow}(x,y)=x^y$
X_n	discrete-time Markov chain with some absorbing and some transient states
P_{ij}	probability of X_{n+1} being in state j given X_n is in state i, i.e. $Pr(X_{n+1}=j\|X_n=i)$
N_{ij}	number of visits to state j before entering an absorbing state given $X_0=i$
μ_{ij}	expected value of N_{ij}, $E(N_{ij})$
η_k	probability of absorption when a process terminates at an absorbing state k
θ_i	expected value of the number of visits to Component i
R_i	reliability of Component i
R_s	reliability of the system
T_{p,θ_i}	relative change of the system reliability as θ_i is changed by $100p\%$
S_{p,θ_i}	sensitivity of the relative change of the system reliability to θ_i when θ_i is changed by $100p\%$
T_{p,R_i}	relative change of the system reliability as R_i is changed by $100p\%$
S_{p,R_i}	sensitivity of the relative change of the system reliability to R_i as R_i is changed by $100p\%$
p_{ij}^{E}	erroneous transition probability with respect to the estimated software usage used in the test
p_{ij}^{T}	true transition probability regarding the true software usage
ε_{ij}	error corresponding to p_{ij}, i.e. $\varepsilon_{ij}=p_{ij}^{\mathrm{E}}-p_{ij}^{\mathrm{T}}$
ρ_{ij}	relative error, i.e. $\rho_{ij}=\varepsilon_{ij}/p_{ij}^{\mathrm{T}}$
ρ_i	relative error of transition probability in Component i, i.e. $\rho_i=-\rho_{ij}p_{ij}^{\mathrm{T}}/(1-p_{ij}^{\mathrm{T}})$
$T_{p,P_{ij}}$	relative change of the system reliability as p_{ij}^{T} is changed by $100p\%$
$S_{p,P_{ij}}$	sensitivity of the relative change of the system reliability to p_{ij}^{T} as p_{ij}^{T} is changed by $100p\%$
T_{p,ρ_i}	relative change of the system reliability as ρ_i is changed by $100p\%$
S_{p,ρ_i}	sensitivity of the relative change of the system reliability to ρ_i as ρ_i is changed by $100p\%$

2.2. Reliability assessment

To construct a component-based software, it involves assembling components together, allowing plug-and-play in a collection of self-contained and loosely coupled components, and interactions among these integrated components (Lyu, 1996; Musa, 1998). Such an approach facilitates software construction and has been increasingly adopted for software development. Due to the complex characteristics of a component-based software, the assessment of the reliability of a software system contains two major tasks: the estimation of the reliability of a component-based system and the estimation of the reliabilities of its individual components. The novelty of this integrated approach presented here lies in the fact that not only the failure behavior of each component

can be specified but also the architecture of a component-based software system can be modeled as a Markov process. For this approach, we have the following assumptions:

- The failures of the software components are independent: A software system can be viewed as composed of logically independent components, which can be implemented and tested individually (Gokhale and Trivedi, 2002; Krishnamurthy and Mathur, 1997; Lo et al., 2002). We assume in this paper that a software system contains a collection of software components, which may be developed or executed concurrently.
- The transfer of control among software components follows a Markov process: We assume that the exchanges of controls among these components are characterized according to the rules of a Markov process. The probability of transition form one component to another is determined from the operational profile of a system. This indicates that the next transfer of control to be executed is independent of the past history and depends only on the present component.

Under the assumption of independence among the successive executions of the components, the reliability of a system can be predicted by the path-based approaches (Gokhale and Trivedi, 2002; Pasquini et al., 1986; Xie and Shen, 1998). That is, if the reliability of an individual component is assumed to be independent of others, the path reliability is the product of component reliabilities. Therefore, in this section, we propose an approach to estimating the reliability of a component-based system by taking the architecture of the software system and the reliabilities of the components into consideration. For example, if a system consists of n components with reliabilities denoted by $R_1,\ldots,R_n$ respectively, the reliability of an execution path, 1, 3, 2, 3, 2, 3, 4, 3, n, is given by $R_1 \times {R_2}^2 \times {R_3}^4 \times R_4 \times R_n$. The architecture of a software system gives a description of how the individual components are expected to interact during system operations. In addition, we obtain an estimate of the reliability of the overall system as well as describe the corresponding mathematical properties explicitly in detail. In particular, we consider systems with different architecture styles and utilize the Markov process to model the failure behaviors of the applications. Three general input–output cases were employed. Finally, we develop three methodologies to estimating the reliability of a software system. The proofs of the following theorems can be found in our previous results (Lo et al., 2002).

Theorem 1. (Single-input/single-output system) *Consider a single-input and single-output system consisting of N components with reliabilities* $R_1,\ldots,R_N$. *Let* $\{X_n\}$ *be the Markov process where state N is an absorbing state, i.e. an output node, while states* $\{1,2,\ldots,N-1\}$ *are transient states. In particular, assume state* 1 *is the input node. Therefore, we have the system reliability*: $R_s = R_1 \times R_N \times \prod_{i=2}^{N-1}\{\mathrm{pow}(R_i,\mu_{1i})\}$,

Theorem 2. (Single-input/multiple-output system) *Consider a single-input and r-output system consisting of N components with individual reliabilities denoted by* $R_1,\ldots,R_N$. *Let* $\{X_n\}$ *be the Markov process where* $\{N,N-1,\ldots,N-r+1\}$ *are absorbing states (i.e. r output nodes) and* $\{1,2,\ldots,N-r\}$ *are transient states. In particular, assume state* 1 *is the input node. Therefore, we have the reliability of the system*:

$$R_s = R_1 \times \prod_{i=2}^{N-r}\mathrm{pow}(R_i,\mu_{1i}) \times \prod_{k=N-r+1}^{N}\mathrm{pow}(R_j,\eta_j).$$

Theorem 3. (Multiple-input/multiple-output system) *Consider an s-input and r-output system consisting of N components with reliabilities* $R_1,\ldots,R_N$. *Let* $\{X_n\}$ *be a Markov process where* $\{N,N-1,\ldots,N-r+1\}$ *are absorbing states (i.e. r output nodes) and* $\{1,2,\ldots,N-r\}$ *are transient states. In particular, assume states* $\{1,2,\ldots,s\}$ *are the input nodes with probability* $p_1,p_2,\ldots,p_s$ *respectively. Therefore, the system reliability,* R_s *equals*

$$\prod_{i=1}^{s}\mathrm{pow}(R_i,p_i) \times \prod_{j=s+1}^{N-r}\mathrm{pow}\left(R_j,\sum_{l=1}^{s}p_l\mu_{lj}\right) \times \prod_{k=N-r+1}^{N}\mathrm{pow}(R_k,\eta_k).$$

3. Illustrations for system reliability

The following examples adapted from Cheung (1980), Gokhale and Trivedi (2002) are used to illustrate the three architecture cases discussed in Section 2. Without loss of generality, we use the terminating application reported in Gokhale and Trivedi (2002) as a running example and let the estimated reliabilities of the components be regarded as unchanged throughout the following three subsections and listed in Table 1.

3.1. Example 1: a single-input/single-output type

The first example is a single-input/single-output system. It consists of 10 components where Component 1 is the input component and Component 10 the output component. Fig. 1 depicts the control-flow graph of the example, and the transition probabilities among the components are given as follows: $P_{1,2}=0.6$, $P_{1,3}=0.2$, $P_{1,4}=0.2$, $P_{2,3}=0.7$, $P_{2,5}=0.3$, $P_{3,5}=1.0$, $P_{4,5}=0.4$, $P_{4,6}=0.6$, $P_{5,7}=0.4$, $P_{5,8}=0.6$, $P_{6,3}=0.3$,

Table 1
The estimated component reliabilities (R_i) and the expected values of the number of visits

	1	2	3	4	5	6	7	8	9	10
R_i	0.99	0.98	0.99	0.96	0.98	0.95	0.98	0.96	0.97	0.99
θ_i in Example 1	1	1.4717	1.3254	0.5289	1.9784	0.3173	1.7433	1.3155	0.9669	1
θ_i in Example 2	1	0.6	0.6845	0.3581	1.0077	0.2149	0.6326	0.0645	0.4676	0.5324
θ_i in Example 3	0.5	0.5	0.673	0.4057	0.9853	0.2434	0.4672	0.6228	0.073	0.5327

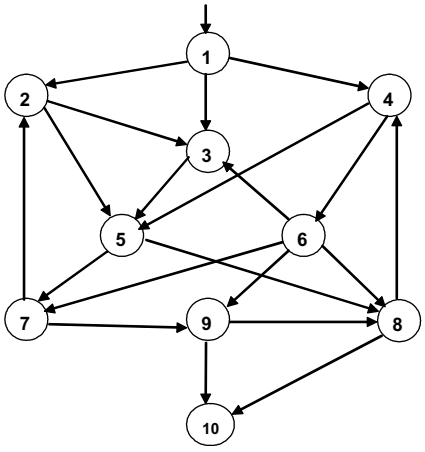

Fig. 1. A single-input/single-output system.

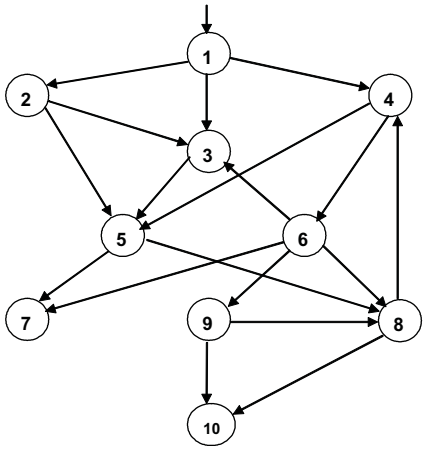

Fig. 2. A single-input/multiple-output system.

$P_{6,7} = 0.3$, $P_{6,8} = 0.1$, $P_{6,9} = 0.3$, $P_{7,2} = 0.5$, $P_{7,9} = 0.5$, $P_{8,4} = 0.25$, $P_{8,10} = 0.75$, $P_{9,8} = 0.1$, $P_{9,10} = 0.9$. Therefore, according to Theorem 1, the expected number of visits on each transient state before absorption from the input node (Component 1) and the probability of absorption can be derived as listed in Table 1. Thus, the system reliability is estimated as $R_1 = 0.8482$.

3.2. Example 2: a single-input/multiple-output type

In this example, we delete two links of the original program control graph in Example 1, and obtain the modified graph as shown in Fig. 2. The modification is a simple transformation from a single-output system to a multiple-output system and the corresponding transition probabilities are similar to Example 1. Again, following the same approach in Theorem 2, we can have following results as listed in Table 1. To go a further step, the reliability of the application, R_2 is 0.8890.

3.3. Example 3: a multiple-input/multiple-output type

In this example, the process will start from one of the two input components (Components 1 and 2) with equal probability and terminates at the output components (Components 7 and 10). That is, the modification is a transformation from a single-input system to a multiple-input system. Fig. 3 depicts the control-flow graph of the example and the transition probabilities are similar to Example 2 except $P_{1,3} = 0.5$ and $P_{1,4} = 0.5$. As a result, according to Theorem 3, portion of the vector of weights can be obtained by $\theta_k = \sum_{l=1}^{2} p_l \mu_{lk}$. Therefore, we have $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_8, \theta_9) = (0.5, 0.5, 0.673, 0.4057, 0.9853, 0.2434, 0.6228, 0.073)$. On the other hand, with the aim to computing the probability of absorption at each absorbing state, the following information about the two absorbing states is obtained based on Theorem 3: $\theta_7 = 0.4672$, $\theta_{10} = 0.5327$. Thus, we have the reliability of the system is $R_3 = 0.8929$.

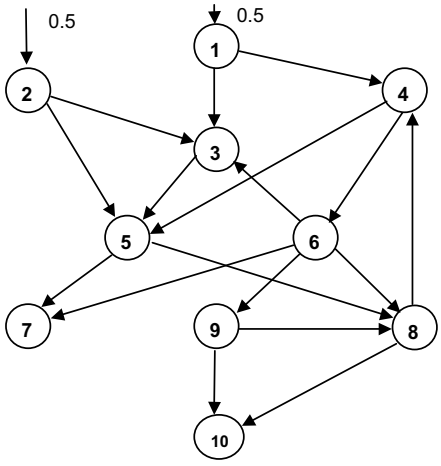

Fig. 3. A multiple-input/multiple-output system.

4. Sensitivity analysis

The reliability of a component-based software system is often higher through the improvement of some components in the system. Therefore, considering such a system we are often interested to know which component is more important than others. Thus, the improvement of that important component will increase the system reliability more than others. Sensitivity analysis gives an approach to analyze the relative importance of input model parameters in determining the value of an assigned output value (Xie and Shen, 1989). That is this method can help make a reasonable decision for this problem. Furthermore, a number of software reliability models have been developed to evaluate the reliability of a system. The parameters in these models are usually obtained from the field failure data. In general, one difficulty in estimating the reliability of a system in the testing stage is the insufficiency of failure data that makes the exact values of the parameters hard to get. Sensitivity analysis is often used in this stage due to the deviations of parameters (Musa, 1993, 1994; Xie and Shen, 1998). That is, sensitivity analysis can help in investigating the effect of the uncertainty in parameters on the reliability estimated from model. In this paper, we will study the sensitivity analysis of the reliability of the component-based software applications in order to know which of the components affects the reliability of the system most. Consequently, from Theorems 1–3 in Section 2, the reliability of this system can be expressed as the general form:

$$R_s = \prod_{i=1}^{N} \mathrm{pow}(R_i, \theta_i), \tag{1}$$

where R_i is the estimated reliability of Component i and θ_i is the expected value of the number of visits to Component i. Here R_s can be regarded as a function of parameters R_i and θ_i, $i = 1, \ldots, N$. From the above discussion, in order to estimate the parameters in Eq. (1) by using the hierarchical approach, it is necessary for software testers to know the information regarding a particular application: architecture of the application (structure of component interactions), software usage profile (the exchange of controls among components determined by transition probabilities), and component failure behaviors (component reliabilities or failure intensity). However, the estimates may not always be accurate, especially in the early stage of the testing phase when a limited amount of information is available. Therefore, it is essential to know the sensitivity of required knowledge regarding the estimated parameters.

4.1. The most sensitive parameter

Considering the parameter θ_i of Eq. (1), it would be helpful to know which of the parameters affects the reliability of the system most, so that more accurate measurements can be made for the most important one (Gokhale and Trivedi, 2002). That is, we are concerned whether the condition of the following formula is sufficed:

$$\left|\frac{\partial R_s}{\partial \theta_i}\right| \geqslant \left|\frac{\partial R_s}{\partial \theta_j}\right|, \quad \text{for all } j = 1, 2, \ldots, N. \tag{2}$$

In practice, the frequency of a component being executed affects the overall system reliability. A higher frequency indicates a greater effect of that component on the performance of the system. This fact shows that the components should have distinct weights according to the architecture of the software system. In other words, the change in the intercomponent transition probabilities of the software architecture manifests the change in the parameter θ_i of Eq. (1). On the other hand, define T_{p,θ_i} as the relative change of the system reliability, R_s, when θ_i is changed by $100p\%$. That is

$$T_{p,\theta_i} = \frac{|R_s(\theta_1, \ldots, \theta_i + p\theta_i, \ldots, \theta_N) - R_s(\theta_1, \ldots, \theta_N)|}{R_s(\theta_1, \ldots, \theta_N)} \times 100\%. \tag{3}$$

Comparatively, let S_{p,θ_i} be the sensitivity of the relative change of the system reliability to θ_i when θ_i is changed by $100p\%$ as the ratio of relative change for the two quantities. That is

$$S_{p,\theta_i} = \frac{\Delta R_s / R_s}{\Delta \theta_i / \theta_i} = \frac{T_{p,\theta_i}}{p} \times 100\%. \tag{4}$$

Then, from Eqs. (2)–(4), the sensitivity analysis can be conducted and the most sensitive parameter, θ_i, in discrete situation can also be found. That is

$$S_{p,\theta_i} \geqslant S_{p,\theta_j}, \quad \text{for all } j = 1, 2, \ldots, N. \tag{5}$$

Therefore, we have the desired results: if power $(R_j, \theta_j) \geqslant$ power(R_i, θ_i), for all $j = 1,2,\ldots,N.$, then θ_i is the most sensitive parameter.

4.2. The most sensitive component reliability

Similar to the reasoning in Section 4.1, for comparing the estimated component reliability R_i in Eq. (1), it would be useful to know which of the components affects the reliability of the system most. That is, we are concerned whether the condition of the following formula is sufficed:

$$\left|\frac{\partial R_s}{\partial R_i}\right| \leqslant \left|\frac{\partial R_s}{\partial R_j}\right|, \quad \text{for all } j = 1, 2, \ldots, N. \tag{6}$$

Again, define T_{p,R_i} as the relative change of the system reliability, R_s, when R_i is changed by $100p\%$, and also let S_{p,R_i} be the sensitivity of the relative change of

the system reliability to R_i when R_i is changed by $100p\%$, that is,

$$T_{p,R_i} = \frac{|R_s(R_1,\ldots,R_i+pR_i,\ldots,R_N) - R_s(R_1,\ldots,R_N)|}{R_s(R_1,\ldots,R_N)} \times 100\%, \tag{7}$$

$$S_{p,R_i} = T_{p,R_i}/p \times 100\%. \tag{8}$$

Thus, the sensitivity analysis with respect to the relative change of component reliability can be performed and the most sensitive component, R_i, in discrete situation can also be found. That is

$$S_{p,R_i} \geqslant S_{p,R_j}, \quad \text{for all } j = 1,2,\ldots,N. \tag{9}$$

According to Eq. (9), we have the result that the one with the maximum parameter value is the most sensitive.

4.3. The most sensitive transition flow

In Section 4.1, we have found the way to deal with the sensitive parameter problem in Eq. (1). Here, we will work on the sensitivity analysis of system reliability resulting from the relative change of transition probability. For a component-based software, different users will have different reliability performances, because they use the system in various ways or use different parts of the system. This dynamic knowledge about the probabilities for different uses in a component-based software is determined by the transition probabilities and apparently depends on the software usage, i.e. operational profile. In general, the operational profile is an estimated description of how the system will be used. One can characterize the usage by the operational profile, the set of operations available on the system and their associated probabilities of occurrences (Musa, 1993, 1994). In order to study the sensitivity of the system reliability to an error in one of the transition probability in the software usage, the method is carried out and the following definitions and symbols are used as follows. Suppose the transition probability, p_{ij}, is incorrect, and let ε_{ij} be the error. Therefore, we have

$$\varepsilon_{ij} = p_{ij}^{\mathrm{E}} - p_{ij}^{\mathrm{T}}, \tag{10}$$

where p_{ij}^{E} is the erroneous transition probability with respect to the estimated software usage used in the test, and p_{ij}^{T} indicates the true transition probability regarding the true software usage. And let ρ_{ij} be the relative error, $\rho_{ij} = \varepsilon_{ij}/p_{ij}^{\mathrm{T}}$. To go a step further, from the property of Markov process, we have $\sum_{j=1}^{N} p_{ij}^{\mathrm{T}} = 1$ and $\sum_{j=1}^{N} p_{ij}^{\mathrm{E}} = 1$ for Component i. In particular, we assume that ρ_{ij} does not have an effect on other ρ_{ik} so that they all have the same relative error ρ_i. Therefore, from Eq. (10), $\sum_{j=1}^{N} p_{ij}^{\mathrm{T}} = 1$, and $\sum_{j=1}^{N} p_{ij}^{\mathrm{E}} = 1$ we have

$$\sum_{j=1}^{N} \varepsilon_{ij} = \rho_{ij}p_{ij}^{\mathrm{T}} + \sum_{\substack{k=1\\k\neq j}}^{N} \rho_{ik}p_{ik}^{\mathrm{T}} = \rho_{ij}p_{ij}^{\mathrm{T}} + \rho_i \sum_{\substack{k=1\\k\neq j}}^{N} p_{ij}^{\mathrm{T}} = 0. \tag{11}$$

Finally, we obtain $\rho_i = -\rho_{ij}p_{ij}^{\mathrm{T}}/(1-p_{ij}^{\mathrm{T}})$

4.3.1. The most sensitive interaction

Afterward, we can define $T_{p,P_{ij}}$ as the relative change of the system reliability when the transition probability p_{ij}^{T} is changed by $100p\%$, and also let $S_{p,P_{ij}}$ be the sensitivity of the relative change of the system reliability to p_{ij}^{T} when p_{ij}^{T} is changed by $100p\%$, that is,

$$T_{p,P_{ij}} = \frac{|R_s(P_{11}^{\mathrm{T}},\ldots,P_{ij}^{\mathrm{T}}+pP_{ij}^{\mathrm{T}},\ldots,P_{NN}^{\mathrm{T}}) - R_s(P_{11}^{\mathrm{T}},\ldots,P_{NN}^{\mathrm{T}})|}{R_s(P_{ij}^{\mathrm{T}},\ldots,P_{ij}^{\mathrm{T}})} \times 100\%, \tag{12}$$

$$S_{p,P_{ij}} = T_{p,P_{ij}}/p \times 100\%. \tag{13}$$

Thus, the sensitivity analysis with respect to the relative change of transition probability can be conducted and the most sensitive interaction between components can be found.

4.3.2. The most sensitive relative error component

On the other hand, we define T_{p,ρ_i} as the relative change of the system reliability when the relative error of transition probability in Component i, ρ_i, is changed by $100p\%$, and also let S_{p,ρ_i} be the sensitivity of the relative change of the system reliability to ρ_i when ρ_i is changed by $100p\%$, that is,

$$T_{p,\rho_i} = \frac{|R_s(\rho_1,\ldots,\rho_i+p\rho_i,\ldots,\rho_N) - R_s(\rho_1,\ldots,\rho_N)|}{R_s(\rho_1,\ldots,\rho_N)} \times 100\%, \tag{14}$$

$$S_{p,\rho_i} = T_{p,\rho_i}/p \times 100\%. \tag{15}$$

Thus, the sensitivity analysis with respect to the relative change of the relative error of transition probability in one component can be conducted and the most sensitive relative error in component can be found.

5. Experimental results for sensitivity analysis

5.1. The most sensitive parameter

As for the example of the software architecture with single-input/single-output in Section 3.1, we apply the results listed in Table 1 (the component reliability and the estimated expected visits for each component, i.e. μ_{1i}) to Eq. (5). After some computations, we can figure out which of the parameters affects the system reliability

more than the other so that more accurate estimates can be obtained for the most important one. In this case, the parameter of Component 8 (μ_{18}) is the most sensitive parameter because it has the minimum value ($\mathrm{pow}(0.96, 1.3155) = 0.9477$) than others. Furthermore, the relationship between the sensitivity, S_{p,θ_i}, and the relative change of component parameter θ_i is depicted in Fig. 4. In order to present the importance of each parameter, the curves in the figures are ordered by its sensitivity decreasingly. We also apply the same approach to Example 2 and Example 3 and conclude that the parameter of Component 8 is the most sensitive parameter as shown in Figs. 5 and 6.

5.2. The most sensitive component reliability

Similarly, we use the estimated vector, μ_{1i} for Eq. (9). Because Component 5 has the maximum parameter

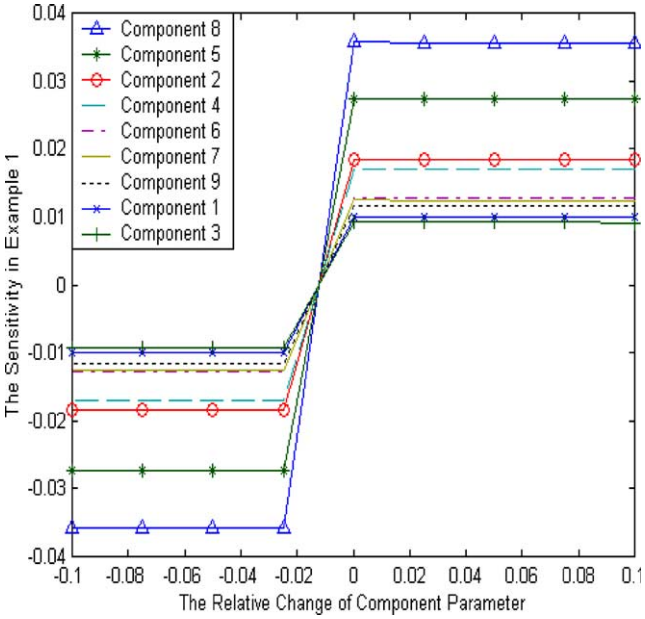

Fig. 4. The most sensitive parameter in Example 1.

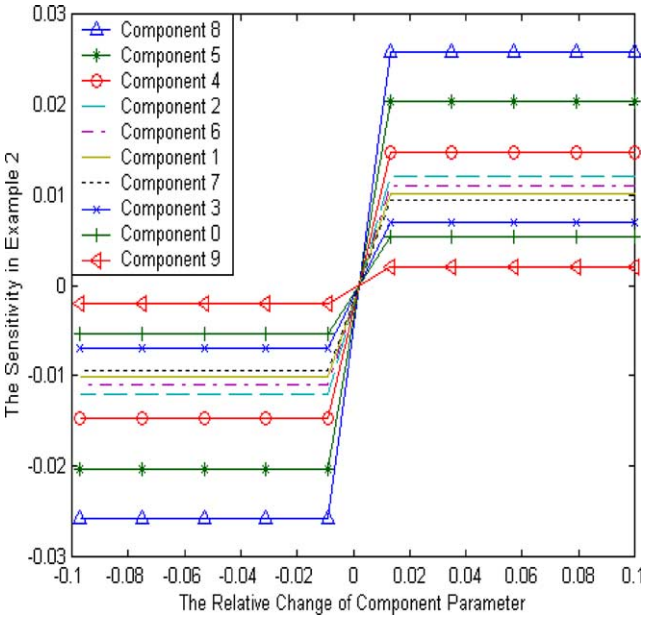

Fig. 5. The most sensitive parameter in Example 2.

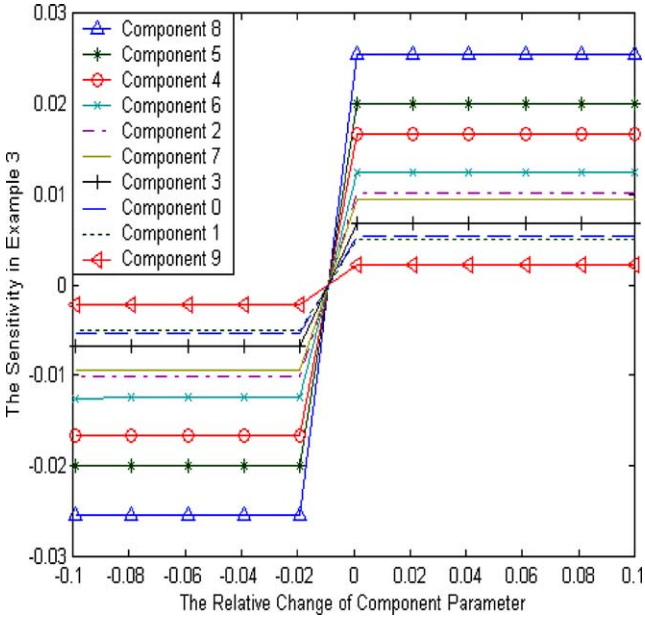

Fig. 6. The most sensitive parameter in Example 3.

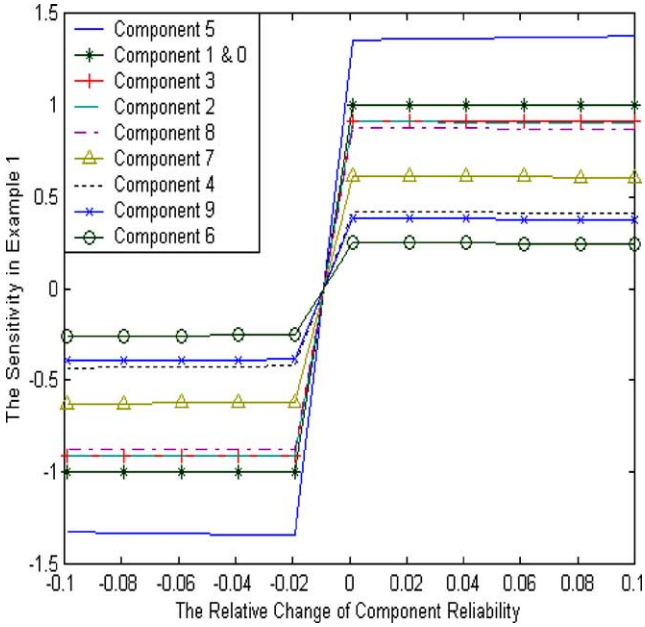

Fig. 7. The most sensitive component reliability in Example 1.

value ($\mu_{15} = 1.9784$), thus from the result in Section 4.2 we know Component 5 is the most sensitive. As for Example 2, the parameter value of Component 5 is 1.0077 and is larger than the others. That is Component 5 is the most sensitive in Example 2. For Example 3, the most sensitive component is also Component 5 because its parameter value is 0.9853 and is the largest. Figs. 7–9 illustrates the relationships between the sensitivity, S_{p,R_i}, and the relative change of component reliability in Examples 1–3.

5.3. The most sensitive interaction

Fig. 10 depict the relationships between the sensitivity, $S_{p,P_{ij}}$, and the relative change of P_{ij} for Example 1. This figure presents the first six sensitive interactions and the first one is the most sensitive. For example, the transition from Component 8 to Component 4 is the most sensitive in these three examples. In particular, a 10% change of P_{84} will imply a 0.39% change of the system reliability in Example 1. This means that it is

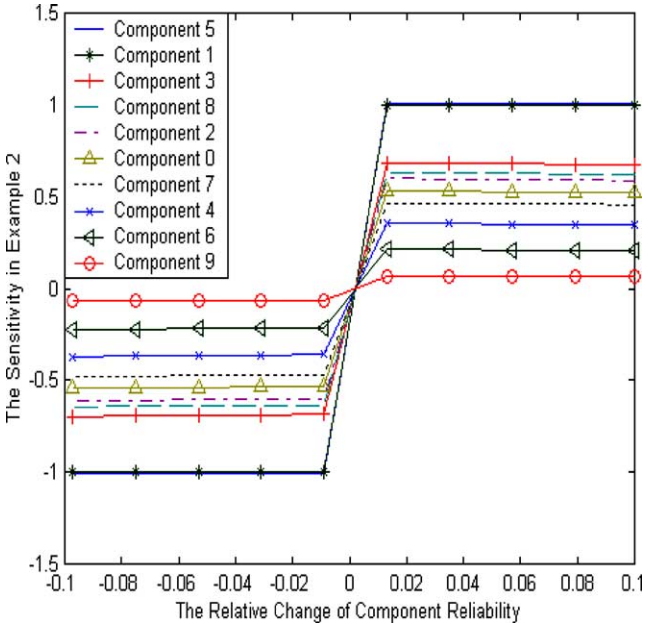

Fig. 8. The most sensitive component reliability in Example 2.

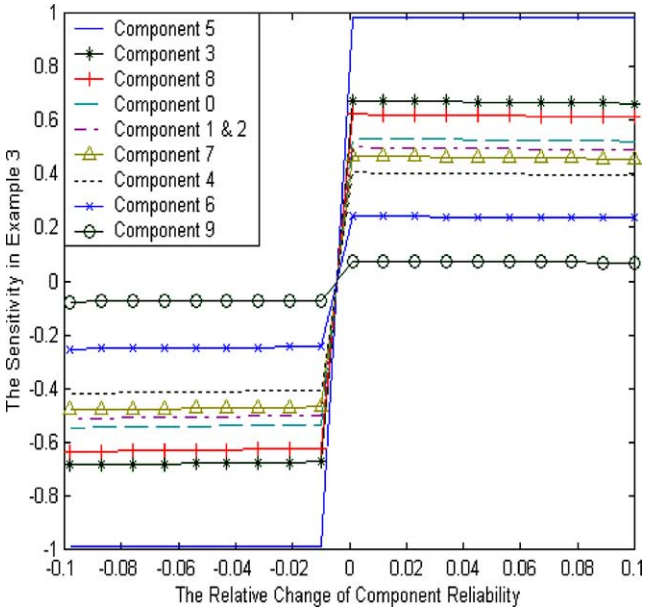

Fig. 9. The most sensitive component reliability in Example 3.

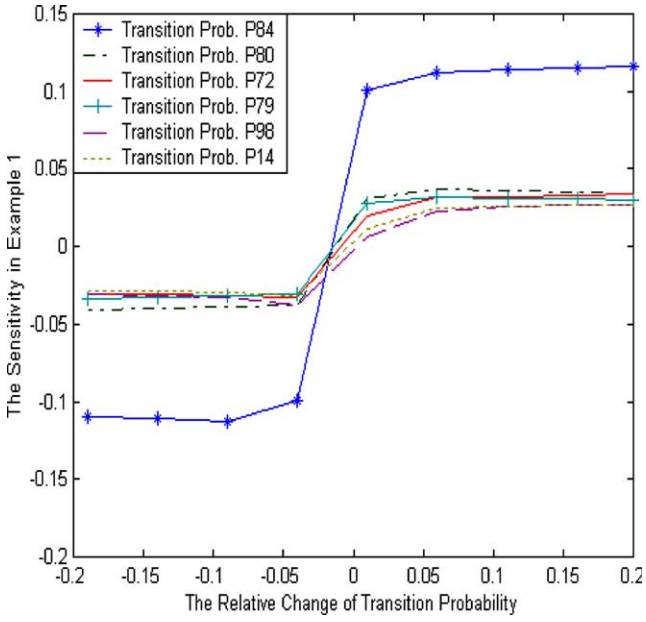

Fig. 10. The most sensitive transition in Example 1.

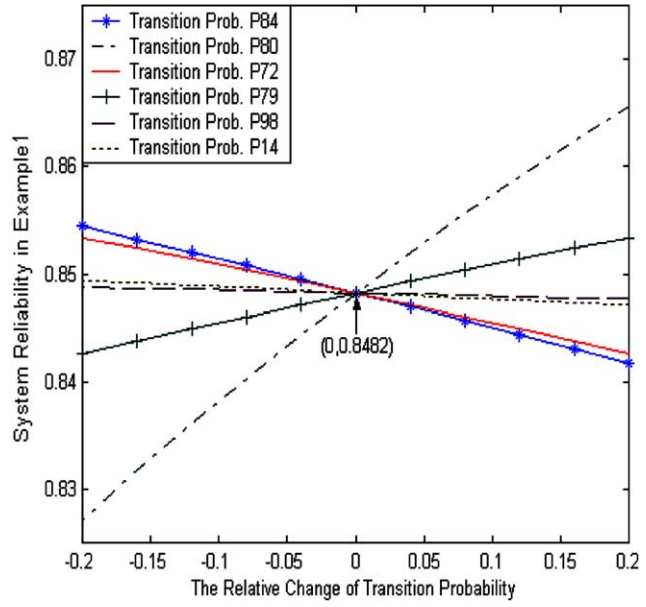

Fig. 11. Reliability vs a change of transition prob. in Example 1.

much more important to obtain an accurate estimate of P_{84} than others. Moreover, Figs. 11–13 also give a description about the relative change of system reliability as the change of the first six sensitive transition probability in Examples 1–3.

5.4. The most sensitive relative error component

Fig. 14 shows the result about the relationships between the sensitivity, S_{p,ρ_i}, and the relative change of the relative error, ρ_i in Example 1. Besides, the most sensitive relative error of transition probability in Examples 1–3 is Component 8. For example, a 10% change of ρ_8 will imply a 0.63% change of the system reliability in Example 2 as shown in Table 4 and Fig. 15. Finally, we also list the results for all of the most sensitivity case in Tables 2–4.

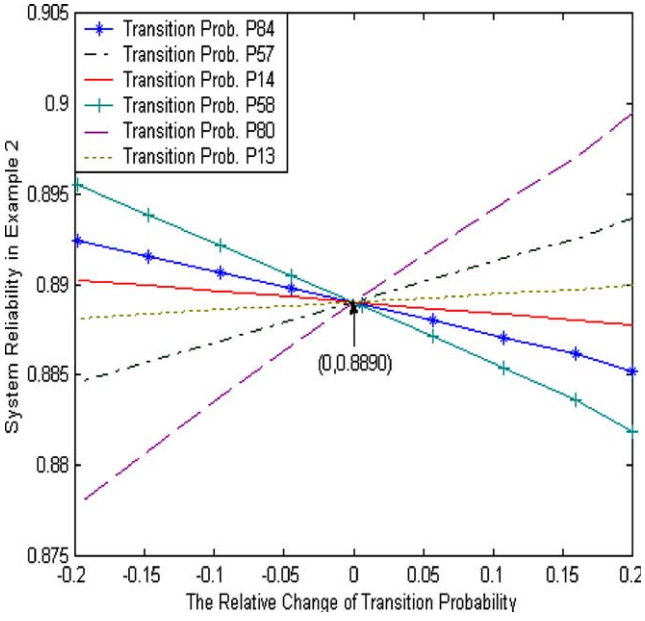

Fig. 12. Reliability vs a change of transition probability in Example 2.

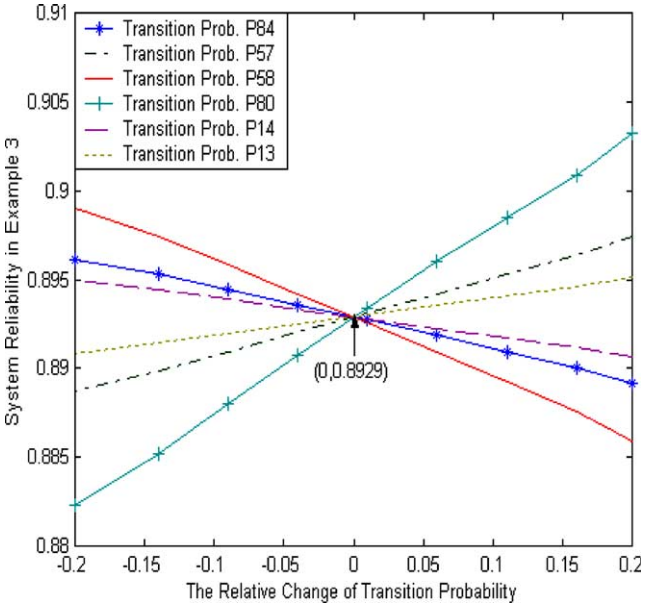

Fig. 13. Reliability vs a change of transition probability in Example 3.

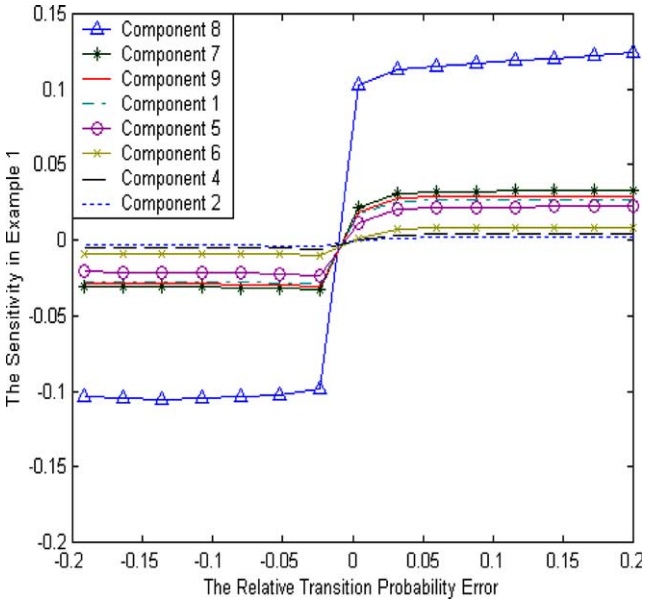

Fig. 14. The most sensitive relative error in Example 1.

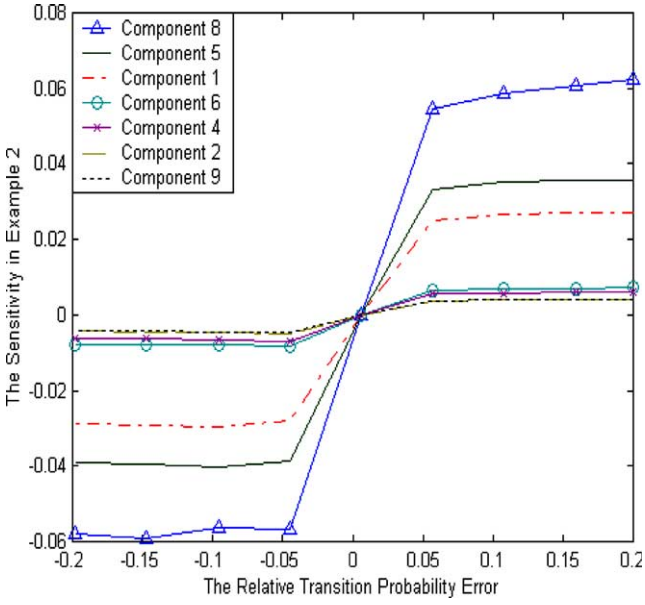

Fig. 15. The most sensitive relative error in Example 2.

6. Conclusions

In this paper we have presented an approach for assessing the reliability of a component-based software.

Table 2
The most sensitive cases in Example 1

	The most sensitive parameter case			The most sensitive component reliability case			The most sensitive interaction case			The most sensitive relative error case		
$p\%$	R_s	$T_{p/2,\theta_8}$	$S_{p/2,\theta_8}$	R_s	$T_{p/2,R_5}$	$S_{p/2,\ R_5}$	R_s	$T_{p,P_{8,4}}$	$S_{p,P_{8,4}}$	R_s	T_{p,ρ_8}	S_{p,ρ_8}
20	0.8452	0.0036	0.0356	0.9659	0.2010	1.3693	0.8413	0.0081	0.1160	0.8271	0.0249	0.1244
15	0.8460	0.0027	0.0356	0.9423	0.1371	1.3647	0.8430	0.0061	0.1148	0.8335	0.0174	0.1206
10	0.8467	0.0018	0.0356	0.9188	0.0920	1.3600	0.8447	0.0039	0.1136	0.8395	0.0103	0.1171
5	0.8475	0.009	0.0357	0.8955	0.0213	1.3554	0.8463	0.0022	0.1117	0.8451	0.0036	0.1131
−5	0.8488	0.0009	−0.0357	0.8265	0.0123	−1.3459	0.8495	0.0015	−0.0999	0.8531	0.0024	−0.0980
−10	0.8490	0.0018	−0.0357	0.8039	0.0609	−1.3411	0.8511	0.0035	−0.1123	0.8580	0.0053	−0.1026
−15	0.8503	0.0027	−0.0357	0.7814	0.1302	−1.3363	0.8526	0.0056	−0.1110	0.8628	0.0084	−0.1040
−20	0.8513	0.0036	−0.0357	0.7590	0.1762	−1.3314	0.8541	0.0078	−0.1099	0.8650	0.0144	−0.1060

Table 3
The most sensitive cases in Example 2

	The most sensitive parameter case		The most sensitive component reliability case				The most sensitive interaction case		The most sensitive relative error case			
$p\%$	R_s	$T_{p/2,\theta_8}$	$S_{p/2,\theta_8}$	R_s	$T_{p/2,R_5}$	$S_{p/2,R_5}$	R_s	$T_{p,P_{8,4}}$	$S_{p,P_{8,4}}$	R_s	T_{p,ρ_8}	S_{p,ρ_8}
20	0.8867	0.0026	0.0258	0.9598	0.0796	1.0080	0.8994	0.0124	0.0593	0.8774	0.0131	0.0623
15	0.8873	0.0019	0.0258	0.9401	0.0575	1.0079	0.8970	0.0093	0.0584	0.8804	0.0096	0.0606
10	0.8879	0.0013	0.0258	0.9204	0.0353	1.0078	0.8945	0.0062	0.0570	0.8834	0.0063	0.0585
5	0.8884	0.0006	0.0258	0.9006	0.0131	1.0078	0.8920	0.0031	0.0537	0.8862	0.0031	0.0544
−5	0.8896	0.0006	−0.0258	0.8612	0.0312	−1.0077	0.8866	0.0026	−0.0572	0.8913	0.0026	−0.0567
−10	0.8901	0.0013	−0.0258	0.8415	0.0534	−1.0076	0.8838	0.0060	−0.0627	0.8938	0.0054	−0.0565
−15	0.8907	0.0019	−0.0258	0.8218	0.0756	−1.0075	0.8808	0.0090	−0.0611	0.8967	0.0087	−0.0593
−20	0.8913	0.0026	−0.0259	0.8021	0.0977	−1.0074	0.8778	0.0119	−0.0601	0.8992	0.0114	−0.0578

Table 4
The most sensitive cases in Example 3

	The most sensitive parameter case		The most sensitive component reliability case				The most sensitive interaction case		The most sensitive relative error case			
$p\%$	R_s	$T_{p/2,\theta_8}$	$S_{p/2,\theta_8}$	R_s	$T_{p/2,R_5}$	$S_{p/2,R_5}$	R_s	$T_{p,P_{8,4}}$	$S_{p,P_{8,4}}$	R_s	T_{p,ρ_8}	S_{p,ρ_8}
20	0.8906	0.0025	0.0254	0.9808	0.0985	0.9846	0.8891	0.0120	0.0572	0.8816	0.0126	0.1244
15	0.8912	0.0019	0.0254	0.9588	0.0739	0.9848	0.8901	0.0090	0.0560	0.8846	0.0093	0.1206
10	0.8918	0.0013	0.0254	0.9369	0.0492	0.9850	0.8910	0.0059	0.0540	0.8874	0.0061	0.1171
5	0.8923	0.0006	0.0254	0.9149	0.0246	0.9851	0.8918	0.0030	0.0493	0.8902	0.0030	0.1131
−5	0.8935	0.0006	−0.0254	0.8709	0.0246	−0.9855	0.8936	0.0024	−0.0611	0.8950	0.0023	−0.103
−10	0.8940	0.0013	−0.0254	0.8489	0.0493	−0.9857	0.8944	0.0058	−0.0646	0.8976	0.0053	−0.104
−15	0.8946	0.0019	−0.0254	0.8269	0.0739	−0.9859	0.8953	0.0087	−0.0620	0.9004	0.0084	−0.106
−20	0.8952	0.0025	−0.0255	0.8049	0.0986	−0.9861	0.8961	0.0115	−0.0606	0.9028	0.0111	−0.105

This methodology assumes that the software components are heterogeneous and the transfers of control between components follow a discrete time Markov process. Besides, we also present the sensitivity analysis on the reliability of a component-based software in order to determine which of the components affects the reliability of the system most. Sensitivity analysis provides a way to analyzing the impact of the parameters. In particular, we define several metrics on how to assess the most sensitive parameter in a system and derive some useful mathematical properties for the sensitivity analysis of system reliability. Finally, three different architecture styles are utilized to validate the proposed approach. For the future works, we will focus on topics including comparisons with different approaches, sensitivity analysis of resource allocation problems, and other sensitivity of software attributes.

Acknowledgments

This research was supported by the National Science Council, Taiwan, ROC, under Grant NSC 91-2213-E-002-042 and 93-2213-E-267-001, and also substantially

supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project no. CUHK4360/02E).

References

Chen, M.H., Mathur, A.P., Rego, V.J., 1994. A case study to investigate sensitivity of reliability estimates to errors in operational profile. In: Proceedings of the 5th International Symposium on Software Reliability Engineering, October, pp. 276–281.

Cheung, R.C., 1980. A user-oriented software reliability model. IEEE Trans. Softw. Eng. 6, 118–125.

Everett, W., 1999, Software component reliability analysis. In: Proceedings of the International Symposium on Application-Specific Systems and Software Engineering &Technology, March, pp. 204–211.

Gokhale, S.S., Trivedi, K.S., 2002. Reliability prediction and sensitivity analysis based on software architecture. In: Proceedings of the 13th International Symposium on Software Reliability Engineering, November, pp. 64–75.

Gokhale, S.S., Wong, W.E., Trivedi, K.S., Horgan, J.R., 1998. An analytic approach to architecture-based software reliability prediction. In: Proceedings of the International Symposium on Performance and Dependability, September, pp. 13–22.

Krishnamurthy, S., Mathur, A.P., 1997. On the estimation of reliability of a software system using reliabilities of its component. In: Proceedings of the 8th International Symposium on Software Reliability Engineering, November, pp. 146–155.

Lo, J.H., Kuo, S.Y., Lyu, M.R., Huang, C.Y., 2002. Optimal resource allocation and reliability analysis for component-based software applications. In: The 26th Annual International Computer Software and Applications Conference, August, pp. 7–12.

Lo, J.H., Huang, C.Y., Kuo, S.Y., Lyu, M.R., 2003. Sensitivity analysis of software reliability for component-based software applications. In: The 27th Annual International Computer Software and Applications Conference, November, pp. 500–505.

Lyu, M.R., 1996. Handbook of Software Reliability Engineering. McGraw-Hill.

Musa, J.D., 1993. Operational profiles in software reliability engineering. IEEE Trans. Softw. 10, 14–32.

Musa, J.D., 1994. Sensitivity of field failure intensity to operational profile errors. In: Proceedings of the 5th International Symposium on Software Reliability Engineering, October, pp. 334–337.

Musa, J.D., 1998. Software Reliability Engineering: More Reliable Software, Faster Development and Testing. McGraw-Hill.

Musa, J.D., Iannino, A., Okumoto, K., 1987. Software Reliability, Measurement, Prediction and Application. McGraw-Hill.

Pasquini, A., Crespo, A.N., Matrella, P., 1986. Sensitivity of reliability-growth model to operational profile errors vs. testing accuracy. IEEE Trans. Reliab. 45, 531–540.

Xie, M., Shen, K., 1989. On ranking of system components with respect to different improvement actions. Microelectron. Reliab. 29, 159–164.

Xie, M., Shen, G.Y., 1998. A study of the sensitivity of software release time. J. Syst. Softw. 44, 163–168.

Yacoub, S.M.B., Cukic, Ammar, H.H., 1999. A component-based approach to reliability analysis of distributed systems. In: Proceedings of the 18th International Symposium on Reliable Distributed Systems, pp. 158–167.

Jung-Hua Lo is currently an Assistant Professor in the Department of Information Management, LanYang Institute of Technology, I-Lan, Taiwan. Dr. Lo received the BS (1993) in Mathematics from National Taiwan University and the MS (1995) and the PhD (2003) in Electrical Engineering from National Taiwan University. His research interests are software reliability and testing.

Chin-Yu Huang is currently an Assistant Professor in the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Dr. Huang received the MS (1994) and the PhD (2000) in Electrical Engineering from National Taiwan University, Taipei. He was with the Bank of Taiwan from 1994 to 1999 and was a senior software engineer at Taiwan Semiconductor Manufacturing Company from 1999 to 2000. Prior to joining Tsing Hua University, he was a division chief in the department of planning and design, information systems office, the Central Bank of China, Taipei, from 2000 to 2003. His research interests are software engineering, software reliability and testing, and software fault-tolerance, etc.

Ing-Yi Chen received a BS in physic from National Central University, Taiwan, in 1984, and MS and PhD degrees in Electrical and Computer Engineering from the University of Arizona, USA, in 1989 and 1992, respectively. Dr. Chen is currently an associate professor in Computer Science and Information Engineering Department, National Taipei University of Technology, Taipei, Taiwan. Prior to joining NTUT, he served as chief technology officer for Chinatimes from 2000 to 2001 with responsibility for the corporation's strategic technology planning and external technical affiliations with universities. Previously, he was a faculty member in the Department of Electronic Engineering at Chung Yuan Christian University, Taiwan between 1992 and 2000. His research interests include various topics in integration of web applications and services, corporate evolving web uses, internet application layer protocols, database systems, web site security, and web site performance management. A member of the Phi Tau Phi scholastic honor society, Dr. Chen was a recipient of the IEEE/ACM Design Automation Scholarship Award in 1991, and the Distinguished Teaching Award at CYCU in 1996.

Sy-Yen Kuo received the BS (1979) in Electrical Engineering from National Taiwan University, the MS (1982) in Electrical and Computer Engineering from the University of California at Santa Barbara, and the PhD (1987) in Computer Science from the University of Illinois at Urbana-Champaign. Since 1991 he has been with National Taiwan University, where he is currently a professor and the Chairman of Department of Electrical Engineering. He spent his sabbatical year as a visiting researcher at AT&T Labs-Research, New Jersey from 1999 to 2000. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering at the University of Arizona from 1988 to 1991, and an engineer at Fairchild Semiconductor and Silvar-Lisco, both in California, from 1982 to 1984. In 1989, he also worked as a summer faculty fellow at Jet Propulsion Laboratory of California Institute of Technology. His current research interests include mobile computing and networks, dependable distributed systems, software reliability, and optical WDM networks.

Professor Kuo is an IEEE Fellow. He has published more than 200 papers in journals and conferences. He received the distinguished research award (1997–2005) from the National Science Council, Taiwan. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.

Michael Rung-Tsong Lyu received the BS (1981) in Electrical Engineering from National Taiwan University, the MS (1985) in Computer Engineering from University of California, Santa Barbara, and the PhD (1988) in Computer Science from University of California, Los Angeles. He is a Professor in the Computer Science and Engineering Department of the Chinese University of Hong Kong. He worked at the Jet Propulsion Laboratory, Bellcore, and Bell Labs, and taught at the University of Iowa. His research interests include software reliability engineering, software fault tolerance, distributed systems, image and video processing, web technologies, multimedia systems, and wireless communications. He has published over 150 papers in these areas. He initiated International Symposium on Software Reliability Engineering (ISSRE), and was Program Chair for ISSRE'1996, Program Co-Chair for WWW10, and General Chair for ISSRE'2001. He also received Best Paper Awards in ISSRE'98 and in ISSRE'2002. He is the editor for two book volumes: Software Fault Tolerance, published by Wiley in 1995, and the Handbook of Software Reliability Engineering, published by IEEE and McGraw-Hill in 1996. He has been an associated editor of IEEE Transactions on Reliability, IEEE Transactions on Knowledge and Data Engineering, and Journal of Information Science and Engineering. Dr. Lyu is a fellow of IEEE.