Architecture-based Reliability Analysis With Uncertain Parameters

Derek Doran, Matthew Tran, Lance Fiondella, and Swapna S. Gokhale
Dept. of Computer Science & Engineering
University of Connecticut, Storrs, CT, 06269
{derek.doran,lfiondella,matthew.tran,ssg}@engr.uconn.edu

Abstract

Architecture-based reliability analysis has gained prominence in the recent years as a way to predict the reliability of a software application during the design phase, before an investment is made in any implementation. To apply this analysis, the parameters comprising the architectural model must be estimated using the limited data and knowledge available during the design phase. These estimates, as a result, are inherently uncertain. Contemporary approaches, however, do not consider these uncertainties, and hence, may produce inaccurate reliability results. This paper presents a Bayesian approach to systematically consider parametric uncertainties in architecture-based analysis. The novelty of this approach lies in determining credible intervals for the model parameters as a function of their posterior distributions. By leveraging these intervals, we illustrate how to: (i) quantify the impact of uncertainty in a specific parameter on the system reliability estimate; (ii) evaluate when a sufficient amount of data has been collected to reduce the uncertainty to an acceptable level; and (iii) assess the impact of prior knowledge regarding the parameters in improving the system reliability estimate.

1 Introduction

Architecture-based analysis [8, 9] is a powerful approach to assess the reliability of a software application[1] earlier in its lifecycle, and to offer guidance for cost-effective reliability improvement. Such guidance often consists of identifying critical components from the perspective of application reliability, so that additional resources can be allocated to these components to improve their reliability. Architecture-based analysis is thus especially useful in the context of modern, complex software systems which are composed from a variety of components; picked off-the-shelf, built contractually, and developed in-house [13].

In the architecture-based approach, application reliability is expressed in terms of the failure characteristics of the components and the application architecture. The parameters comprising the architectural model thus include the transition probabilities among the components and the reliabilities of the components comprising the application. Thus, the number of parameters (transition probabilities and component reliabilities) that need to be estimated increases with the size of the application. Additionally, these parameters must be estimated based on the limited data or scant knowledge that may be available earlier in the lifecycle before the application is implemented and can be tested extensively. Because of such limited available information, architecture-based analysis must be performed with uncertain parameters. These parametric uncertainties then propagate to the system-level reliability results, which include the reliability estimate and the ranks of the components based on their criticality. Most of the contemporary architecture-based analysis approaches, however, do not consider these parametric uncertainties [2]. Thus, in the best case, the system reliability estimate that these approaches produce underestimate the true reliability, which may result in dedicating more than necessary resources for the system to reach its reliability target. In the worst case, however, the system will be deployed with an optimistic expectation of its reliability [3]. To avoid the pitfalls associated with over and under-estimation, it is necessary to systematically and quantitatively incorporate the impact of uncertain parameters into architecture-based reliability analysis.

In this paper, we present a Bayesian approach to incorporate parametric uncertainties into architecture-based software reliability analysis. The proposed approach constructs joint credible intervals using the posterior distributions of the model parameters as a function of limited information. We illustrate how these credible intervals could be used to explore the influence of uncertain parameters on application reliability. We also demonstrate how the Bayesian approach could be used to quantify the impact of additional data or prior knowledge in reducing the uncertainty in the system-level reliability results.

The paper is organized as follows: Section 2 presents the Bayesian approach. Section 3 illustrates the approach. Section 4 summarizes related research. Section 5 offers concluding remarks and directions for future research.

[1]The terms application, software, and system are used interchangeably throughout this paper.

2 Bayesian analysis methodology

In this section, we introduce a preliminary model for architecture-based reliability analysis that ignores uncertainties in the model parameters. We then describe how this preliminary model can be extended to incorporate parametric uncertainties using the Bayesian approach.

2.1 Preliminary model

We consider a terminating application with n components. We assume that the transfer of control between components is independent and that the failure of a single component causes the entire application to fail. The architecture of the application is represented as a probabilistic control flow graph. This graph is mapped to a discrete time Markov chain (DTMC) which is represented by an $n \times n$ matrix $\mathbf{P}$. The $(i,j)^{th}$ element of $\mathbf{P}$ is the expected value of the random variable $\mathcal{P}_{i,j}$, denoted $p_{i,j}$, and represents the expected probability that control is transferred from component i to component j. Thus, there exists a one-to-one mapping between the application components and the states of the DTMC. Without loss of generality, we assume that the application starts execution with component one and terminates after component n executes, so the n^{th} row of $\mathbf{P}$ contains all zeros.

We let R_i denote the expected reliability of component i. Component reliability is binary, that is, when control transitions to component i, the probability that it executes successfully is represented by the random variable Λ_i whose expected value is R_i. If a component does not execute successfully, the entire system fails. The expected application reliability can be computed using the following algorithm:

1. Define $\mathbf{D}$ as a diagonal matrix where $\mathbf{D}_{i,i} = R_i$
2. Let $\mathbf{Q} = \mathbf{D} \times \mathbf{P}$
3. Solve $\mathbf{S} = \sum_{k=0}^{\infty} \mathbf{Q}^k = (\mathbf{I} - \mathbf{Q})^{-1}$
4. System reliability is given as $\mathbf{S}_{1,n} \cdot R_n$

The matrix $\mathbf{Q}$ is a sub-stochastic matrix where $q_{i,j}$ is the transition probability among components i and j, weighted by the reliability of component i. This captures the notion that a transition from component i to component j occurs conditional to the successful execution of component i, else component i fails causing the application to fail. Using $\mathbf{S}$ to obtain the expected number of times the application reaches the end state n starting from state 1, the system reliability is thus given by $\mathbf{S}_{1,n} \cdot R_n$.

2.2 Incorporating parametric uncertainties

In this section, we discuss our approach to incorporate parametric uncertainties into the preliminary model presented above. Towards this end, we use Bayesian techniques to construct credible intervals for each parameter, such that the true value of the parameter lies within that credible interval at a specified confidence level. The credible interval will provide the range over which each parameter must be varied in order to systematically explore the impact of its uncertainty on the application-level reliability results.

2.2.1 Posterior distribution

The parameters of the architecture-based model include the intercomponent transition probabilities, and the component reliabilities. To construct a credible interval for each parameter, it is first necessary to determine its posterior distribution. A point estimate for the expectation of Λ_i is given by $R_i = x_i/n_i$, where x_i is the number of successful executions of the component across n_i trials. Each x_i then follows a Binomial distribution:

$$Bin(x_i, R_i) = \binom{n_i}{x_i} R_i^{x_i}(1-R_i)^{n_i-x_i}.$$

We can define a *posterior* distribution $G(\Lambda_i|x_i)$ of the random variable representing component reliability, incorporating any prior knowledge we have about this distribution with additional data we have obtained since then, as follows. Let $g(\Lambda_i)$ be the *prior* distribution of Λ_i, incorporating any *a priori* knowledge about Λ_i, and $f(x_i|\Lambda_i)$ be the likelihood function that provides the probability of observing x_i given Λ_i. From Bayes theorem [1], it follows that:

$$G(\Lambda_i|X) = \frac{g(\Lambda_i)f(X|\Lambda_i)}{\int_0^1 g(\Lambda_i)f(X|R_i)d\Lambda_i}.$$

Since the expected value of Λ_i is used as the parameter of a Binomial distribution, we know that Λ_i has a prior and a posterior that follows a Beta distribution, given as:

$$Beta(\Lambda_i|a_i,b_i) = \frac{\Gamma(a_i+b_i)}{\Gamma(a_i)\Gamma(b_i)}\Lambda_i^{a_i-1}(1-\Lambda_i)^{b_i-1}.$$

The parameters of the posterior can be expressed in terms of the parameters of the prior. If $g(\Lambda_i) \sim Beta(\Lambda_i|\alpha,\gamma)$, then $G(\Lambda_i|x_i) \sim Beta(\alpha + x_i, \gamma + (n_i - x_i))$ [4].

Next, we determine the posterior distribution for each random variable $\mathcal{P}_{i,j}$ representing the transition probabilities in the architecture. The expectation $p_{i,j}$ of this random variable represents the average probability of transitioning from component i to component j. This average transition probability can be estimated as $p_{i,j} = x_{i,j}/n_i$ where $x_{i,j}$ is the number of times control is transferred from component i to component j and $n_i = \sum_j x_{i,j}$ is the total number of times control is transferred out of component i. The number of times control is transferred to various components from component i thus follows a Multinomial distribution [12]:

$$Multinomial(X_i, k_i) = \binom{k_i}{x_{i,1}x_{i,2}...x_{i,k}} p_{i,1}^{x_{i,1}} p_{i,2}^{x_{i,2}} ... p_{i,k}^{x_{i,k}}$$

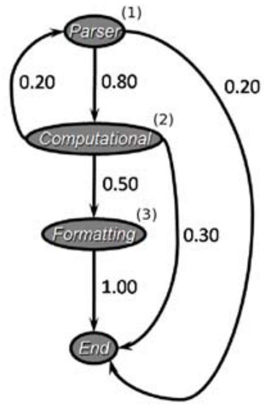

Figure 1. Architecture of the ESA application

Param.	Estimate	True Value	% Difference
R_1	0.932	0.840	9.87%
R_2	0.800	0.830	3.75%
R_3	0.933	0.950	1.82%
$p_{1,2}$			

where k_i is the number downstream components that control can be transferred to, $X_i = (x_{i,1}, x_{i,2}, ...x_{i,k_i})$, and $\sum_{j=1}^{k_i} x_{i,j} = n_i$. Let $P_i = (\mathcal{P}_{i,1}, \mathcal{P}_{i,2}, ...\mathcal{P}_{i,k_i})$ be the vector whose elements are the random variables that represent the transition probabilities associated with state i. These random variables have an expectation of $p_{i,1}, p_{1,2}, ..., p_{i,k_i}$. It is important to note that $k_i \leq n$, that is, the number of downstream components that component i can transfer control to, in general, may be less than the total number of components in the application. When the draws from the Multinomial distribution are independent, the posterior and prior distribution for P_i follows a Dirichlet distribution, given by:

$$Dir(P_i|\Psi_i) = \frac{\Gamma(\psi_{i,1} + \psi_{i,2} + ... + \psi_{i,k_i})}{\Gamma(\psi_{i,1})\Gamma(\psi_{i,2})...\Gamma(\psi_{i,k_i})} \prod_{j=1}^{k_i} p_{i,j}^{\psi_{i,j}-1}.$$

Here, $\Psi_i = (\psi_{i,1}, \psi_{i,2}, ..., \psi_{i,k_i}), \psi_{i,j} > 0 \; \forall j$ is the set of parameters representing the number of times each transition was followed, $p_{i,j} \geq 0$, and $\sum_{j=1}^{k_i} p_{i,j} = 1$. Since the marginals of the $Dir(P_i|\Psi_i)$ are $Beta(\psi_{i,j}, \psi - \psi_{i,j})$ distributed, where $\psi = \sum_{j=1}^{k_i} \psi_{i,j}$, it follows that the prior and posterior of each $\mathcal{P}_{i,j}$ are Beta distributed. Letting $\sum_{j=1}^{k_i} x_{i,j} = n_i$, if $g(\mathcal{P}_{i,j}) \sim Beta(\alpha, \gamma)$, then its posterior follows a $Beta(\alpha + x_{i,j}, \gamma + n_i - x_{i,j})$ distribution [4].

2.2.2 Bayesian credible intervals

With the posterior distribution derived for each model parameter, a Bayesian credible interval can be established that provides a confidence level within which the true value of the parameter lies. For model parameter π_i (either a component reliability or a transition probability) that has a posterior distribution $G(\pi_i|X)$, the bounds of the $(1-\alpha)\%$ Bayesian credible interval is the value y_i for which:

$$\int_0^{y_i} G(\pi_i|X) d\pi_i = \epsilon$$

where $\epsilon = \alpha/2$ defines the lower and $\epsilon = 1 - \alpha/2$ defines the upper bound of the interval. These bounds for all the pa-

Parameter	IP (Rank), Est.	IP (Rank), Actual
$Parser$	.0626 (2)	.1484 (1)
$Computational$	.1701 (1)	.1201 (2)
$Formatting$	.0200 (3)	.0157 (3)

Table 2. Component ranks with and without uncertainty

These estimates of the model parameters lead to a system reliability estimate of 0.7200, which is off by 6.17% from its true value. Comparing the reliability estimate obtained from limited data with its true value highlights how the uncertainties in the parameters can propagate to system reliability.

3.2 Component ranks

The architecture-based approach allows us to rank the application components according to their importance measures [3] or significance from the point of view of system reliability. In this section, we demonstrate how uncertainties in the parameter estimates can lead to incorrect ranks. For the sake of illustration, we rank each component i based on its Improvement Potential (IP) measure [3], which is defined as the difference between the system reliability estimate when R_i is 1 and the estimate when R_i is assigned its true or estimated value. This expresses the maximum gain in system reliability that can be attained by devoting resources toward perfecting component i. Table 2 shows the IP of each component when using its true and estimated values. While the true IP ranking reveals the $Parser$ as having the highest improvement potential, using estimated parameters causes the $Computational$ component to be ranked higher than the $Parser$. Parametric uncertainties can thus result in a mis-allocation of resources towards prioritizing the $Computational$ component, when in reality improvements to the $Parser$ component will deliver the largest gain in application reliability.

In order to alleviate the impact of making such wrong decisions, it is necessary to perform a sensitivity analysis for each component across a range of parameter values, and use the insights gained from that analysis to evaluate the validity of the component ranks. The more sensitive the system-level reliability is to the reliability of component i, it is more likely that its estimated component rank is incorrect.

3.3 Sensitivity analysis

In order to explore the sensitivity of the reliability estimate to the individual model parameters, we present a study where we set a single model parameter to each bound of its 95% credible interval and then estimate the application reliability. The reliability value evaluated at each bound represents the 95% credible interval for system reliability with respect to the uncertainty in that single parameter. We define the *reliability uncertainty* (RU) metric as the effect of a parameter's uncertainty on application reliability. This metric is given as the size of the credible interval of the application reliability, found by computing the system reliability at the two extreme values of the parameter's credible interval. If the RU for a parameter is very small, we will have a high level of certainty in the reliability estimate, but as RU increases we can only be certain that the system reliability estimate is off by an increasingly higher percentage. The study makes a worse-case assessment by assuming no prior knowledge about any of the parameters, and hence, uses the uniform priors [4] $g(\Lambda_i) \sim Beta(1,1)$ and $g(\mathcal{P}_i) \sim Dir(1,1,...1)$ for all of the parameters.

Param	Credible Interval	RU Value	Rank
R_1	(0.8380 , 0.9724)	0.1212	7
R_2	(0.6609 , 0.8905)	0.1846	4
R_3	(0.6977 , 0.9845)	0.0860	8
$p_{1,2}$	(0.6960 , 0.8977)	0.1587	6
$p_{1,4}$	(0.1023 , 0.3040)	0.2218	3
$p_{2,1}$	(0.1295 , 0.4259)	0.1619	5
$p_{2,3}$	(0.2595 , 0.5923)	0.2237	2
$p_{2,4}$	(0.1920 , 0.5116)	0.2301	1
$p_{3,4}$	(1.0000 , 1.0000)	0.0000	9

Table 3. Credible intervals of the parameters and their impact on application reliability

Table 3 presents the credible interval for each parameter and its resulting RU. The table illustrates how uncertainties in the parameters reduce the confidence in the application reliability. For example, when parameter $p_{2,3}$ is off by approximately by 20%, there is a 95% chance that it causes an uncertainty of up to 24% in the application reliability. Similarly, uncertainty in the parameters $p_{2,4}, p_{2,3}$, and R_2 results in up to 25.57% uncertainty in the application reliability. Furthermore, R_2 has the highest RU value among all components, and corresponds to the component that was incorrectly ranked as having the highest IP. This illustrates how sensitivity analysis can be used to decide on the accuracy of the derived importance ranks of the components.

We also observe that the level of uncertainty in a parameter seems unrelated to the resulting degree of uncertainty in the application-level reliability. For example, parameter $p_{1,4}$ has the smallest non-zero credible interval among all the transition probabilities, but is ranked as having the 3^{rd} most significant impact on the system reliability. This means that the level of uncertainty in a parameter is not the only factor that determines its impact on the level of uncertainty in the application reliability. Another factor may include the frequency with which a transition is taken or a component is executed during a typical run of the application. This frequency indicates the significance of a given transition or a component to the application reliability. Although these significant transitions and components will have smaller credible intervals (because these will be estimated using a larger number of observations), the uncertainties in them will still have a disproportionate influence on the uncertainty in application-level reliability.

3.4 Reducing parametric uncertainties

The previous sections highlighted the impact of parametric uncertainties on the system-level reliability results. The uncertainties in the parameters arise because of the limited data used in the estimation process. Thus, to reduce these uncertainties, it is necessary to either use a large amount of data or to incorporate prior knowledge while estimating the parameters. In the following illustrations, we show the impact of either approach on reducing the uncertainty.

3.4.1 Increasing data

Increasing the amount of data used in the estimation process can improve the estimates of the parameters. In the limit, however, for these parameters to approach their true values, an infinite amount of data is required. This is not feasible, and hence, a trade-off between the level of uncertainty that can be tolerated and the amount of data used in the estimation process must be reached.

In Figure 2, the level of uncertainty in the application reliability versus the number of simulation trials used to collect data for parameter estimation is plotted. The figure shows that the degree of uncertainty in system reliability decreases at an exponential rate as the number of simulation runs increases. In this illustration, all of the data used to estimate the parameters is collected from the simulation. In practice, however, this data may come from a variety of sources. Thus, the figure more generally illustrates that devoting a small cost towards improving our pre-existing knowledge can lead to a measurable increase in the accuracy of the system reliability estimate. Devoting too much cost towards gathering this knowledge, however, may be wasteful because the level of uncertainty in the system reliability will not appreciably decrease beyond a certain threshold.

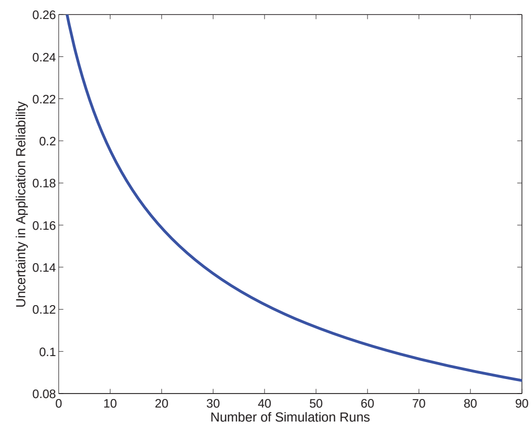

Figure 2. Relationship between reliability uncertainty and available data

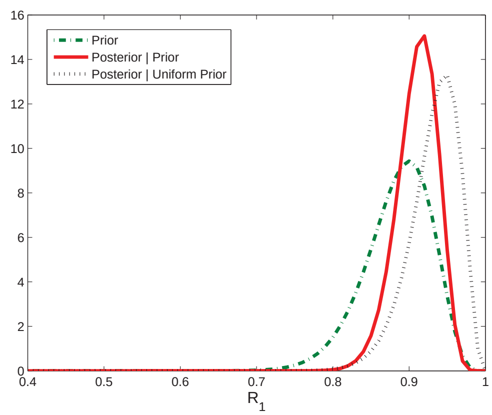

Figure 3. Prior and posterior distributions for the reliability of the $Parser$ component

3.4.2 Incorporating prior knowledge

An alternative way to reduce uncertainty is to incorporate prior knowledge in the parameter estimation process. We illustrate this by introducing a non-uniform prior for the reliability of the *Parser* component. This non-uniform prior may be derived based on: (i) how this component operated in a different system; (ii) data provided by the manufacturer; or (iii) our prior experience in its use. We say that the *Parser* component is being reused from a previous system, where it exhibited a reliability of 0.9 across 50 uses.

Figure 3 shows the shape of the resulting prior distribution, the posterior distribution derived with a uniform prior, and the posterior distribution that incorporates this non-uniform prior for the $Parser$ component. Under a uniform prior, the posterior distribution has a wide body whose mean value is below its true reliability. The prior distribution, however, overstates the true reliability of the component and exhibits a similarly wide body as the posterior under the uniform prior.

In incorporating this prior knowledge, we get a new posterior distribution with a mean value of 0.9099. This mean is closer to the true component reliability of 0.84 when compared to the estimate found under a uniform prior (0.932). This new component reliability estimate results in a new system reliability estimate of 0.6998, which is inaccurate by only 3.52%, compared to 6.57% in the case of using a uniform prior for the $Parser$ reliability. Furthermore, the sharper peak of this new posterior reduces the size of the 95% credible interval for the component from 0.1212 to 0.1051, resulting in a reduction in the impact of its uncertainty on the application reliability from 0.1846 to 0.0947.

4 Related research

In this section, we summarize and compare our work to related efforts. Goseva *et al.* [6] model component reliabilities and transitions with Binomial and Multinomial distributions respectively. However, they only estimate the model parameters based on the Beta and Dirichlet distributions. A more recent study [5] applies their Bayesian approach to an empirical case study of the GCC compiler. Uncertainty analysis based on perturbation theory has also been proposed in the context of architecture-based analysis [10, 7].

Unlike previous approaches, this research employs Bayesian techniques to compute credible intervals for all the model parameters and proposes an approach to quantify the impact of uncertainty in the parameters on the uncertainty in the system reliability. This approach thus provides a practical tool for test planners who must identify where to target limited resources to improve the confidence in the reliability of their software application.

5 Conclusions and future research

This paper introduced a Bayesian approach to incorporate parametric uncertainties in the assessment of software application reliability based on its architecture. We illustrated how the approach could be used to systematically assess the impact of uncertainties in the parameters on system-level reliability in a systematic, quantitative manner. The case study suggests that there exist additional factors besides uncertainty in the parameter itself that can affect its impact on system-level reliability. We also explored how the level of uncertainty in the application reliability reduces as a function of increasing data or prior knowledge used in the estimation process.

Our future research seeks to extend the uncertainty quantification procedures to importance assessment and optimization. In addition, understanding how uncertainty across multiple model parameters affects the uncertainty in the application reliability needs to be investigated.

Acknowledgements

This research was supported by a CAREER award from NSF (#CNS-0643971).

References

[1] W. M. Bolstad. *Introduction to Bayesian Statistics*. John Wiley & Sons, Inc., 2nd edition, 2007.

[2] R. Cheung. A User-Oriented Software Reliability Model. *IEEE Trans. Software Eng.*, 6(2):118–125, March 1980.

[3] L. Fiondella and S. Gokhale. Importance Measures for Modular Software with Uncertain Parameters. *Software Testing, Verification, and Reliability*, 20(1):63–85, 2010.

[4] A. Gelman. *Bayesian Data Analysis*. Taylor & Francis, 2003.

[5] K. Goševa-Popstojanova, M. Hamill, and X. Wang. Adequacy, Accuracy, Scalability, and Uncertainty of Architecture-based Software Reliability: Lessons Learned from Large Empirical Case Studies. In *Proc. Intl. Symposium on Software Reliability Engineering*, pages 197–203, Raleigh, NC, November 2006.

[6] K. Goševa-Popstojanova and S. Kamavaram. Assessing Uncertainty in Reliability of Component-Based Software Systems. In *Proc. of Intl. Symposium on Software Reliability Engineering*, pages 307–320, Denver, CO, nov 2003.

[7] K. Goševa-Popstojanova and S. Kamavaram. Software Reliability Estimation under Uncertainty: Generalization of the Method of Moments. In *HASE'04*, pages 209–218, Tampa, FL, mar 2004.

[8] K. Goševa-Popstojanova and K. Trivedi. Architecture Based Approach to Quantitative Assessment of Software Systems. *Performance Evaluation*, 45(2-3):179–204, June 2001.

[9] S. Gokhale and K. Trivedi. Analytical Models for Architecture-Based Software Reliability Prediction: A Unification Framework. *IEEE Trans. Rel.*, 55(4):578–590, December 2006.

[10] S. Kamavaram and K. Goševa-Popstojanova. Sensitivity of Software Usage to Changes in the Operational Profile. In *NASA/IEEE Software Engineering Workshop*, pages 157–164, Greenbelt, MD, dec 2003.

[11] S. Karris. *Numerical Analysis: Using MATLAB and Excel*. Orchard Publications, 3rd edition, 2007.

[12] K. S. Trivedi. *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. John Wiley & Sons, Inc., 2nd edition, 2002.

[13] S. Wadekar and S. Gokhale. Exploring Cost and Reliability Tradeoffs in Architectual Alternatives Using a Genetic Algorithm. In *Proc. of Intl. Symposium on Software Reliability Engineering (ISSRE 99)*, pages 104–113, Boca Raton, FL, nov 1999.