Towards Estimation of Reliability for Embedded Software Systems

Aleksandar Dimov[1]

Abstract – **Software reliability is a critical concern to be taken into account when reasoning about embedded software systems. As in many other areas in software engineering, architectural approach towards estimation of reliability has many advantages, especially when applied in early development phases. This paper presents a review of state of the art in architectural based reliability models applicable for embedded software systems and points out the unresolved issues with respect to modelling of such systems.**

Keywords – **Software reliability, Architecture-based software reliability models, Uncertainty**

I. Introduction

In recent years there is a trend towards increasing usage of embedded software systems in all areas of human life. Additionally, some of these systems are responsible for the control of different safety-critical processes, for instance in space, nuclear or transportation systems, which have high requirement toward their non-functional characteristics. Non-functional characteristics are also known as quality parameters and define additional constraints and requirements on how software should perform its functionality. In such conditions of paramount importance is to provide methods modeling and reasoning about non-functional properties in order to be able to adequately design safety-critical software systems.

One significant quality parameter is *dependability* [[2]], which is defined as the ability of a computing system is to deliver services that can justifiably be trusted. Dependability is characterized by several attributes, such as reliability, availability, safety[1], confidentiality, integrity and maintainability. In this paper we are going to look at reliability only and the methods for estimation of software reliability. Reliability is defined as the continuity of correct service, i.e. the belief that a software system will behave as per specification over a given period of time and is usually modeled as a stochastic value. It may have different measures like: probability of failure; mean time between system failures or failure rate.

Basically, there exist two broad categories of reliability assessment models: black-box and white-box models. Black-box models are used to reason about reliability of software systems, without taking into account their internal processes or structure. On the contrary, white box-models consider some internal information about architecture of the system. White box models are also called Architecture-Based Reliability Models (ABRMs). Usually architecture-based software reliability estimation takes the following main steps [[8]]: (1) Identification of computational modules (components)[2] within software architecture; (2) Description of the actual architectural model – this includes how components are interconnected and interact with each other (3) Definition of components failure behaviour – at this step the reliability parameters of components and their measures are identified and (4) Combination of the failure behaviour with the architectural model.

Application of white box models has a lot of advantages, among them are: ability to reuse information about reliability parameters of both the system and the components that constitute it; ability to find these modules that influence systems reliability the most, i.e; possibility to isolate and remove reliability "bottlenecks" within the system and etc. For these reasons we focus our research work on white box models.

Although significant amount of research has been undertaken in recent years in the area of architectural software reliability models, there exists a lot of work to be done in the area [[4]]. One very important such issue which exists in the field is *uncertainty* and more than 30 years of research failed to find universally accepted solution on how to model uncertainty.

Our previous research work [[3]] assessed white-box software reliability models upon several important issues that should be taken into account when analyzing embedded and safety-critical software system. In this paper we are continuing this research by adding one more issue – the uncertainty, inherent in reliability parameters. Uncertainty may exist either on individual components reliabilities or in their operational profile and this way it propagates to the calculated reliability estimation of the overall system. Uncertainties are very common mainly due to limitations and assumptions taken by different techniques adopted for reliability estimation.

A reliability model for embedded software systems should be able to take into account the following issues:

- Modeling of dependency between component failures – most of the models assume that a failure in one component never result from failures in another components.

[1] Aleksandar Dimov, PhD is an assistant professor at the Department of Software Technologies, Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski"; 125 Tsarigradsko Shosse Blvd., bl. 2, room 222, 1113, Sofia, Bulgaria e-mail: aldi@fmi.uni-sofia.bg

[2] We use the terms *module* and *component* as synonyms within this paper.

- Ability to estimate reliability parameters of components into the system – most models currently assume that reliabilities of modules are already known in advance.
- Ability to model uncertainty in system reliability.

The rest of the paper is organized as follows: Section two surveys the current state of the art in white-box reliability modeling with respect to applicability for embedded systems; Section 3 makes an analysis of these models and finally Section 4 concludes the paper and streamlines the directions for further research in the area.

II. How do we take into account uncertainty in reliability estimations?

In this section we are going to survey some the architecture based models that take into an account uncertainty in reliability parameter. According to the broad classification three main groups of white-box models are known: state based models, path-based models and additive models. State-based models use Finite State Machine (usually Markov chain) representation of systems' architecture. They attempt to take into account, all of the possible traces of components execution within the system. Path-based models are similar to state-based models, but consider only finite number of component executions traces. The latter usually correspond to system test cases. Additive models do not concern actual architectural configuration of the software system. Instead, they assume a specific distribution process for components failure behaviour and on that basis infer formulae for calculation of reliability.

Basically, there exist two main sources of uncertainty in reliability estimates calculated using architecture-based reliability models:

- Uncertainty in operational profile – operational profile is a frequency distribution that gives the relative probability that a specific function of the program will be executed. In other words, uncertainty in the operational profile represents uncertainty in the environment, in which a component is integrated and further executed.
- Uncertainty in reliability values – what is the level of confidence we have in the correctness of particular values of reliability

The most traditional approach with respect to the second step in white-box reliability modeling (i.e. creation of architectural model) is to present the architecture with the system as a Markov chain, where states of the chain represent system components, and the edges – transitions between the components. The first model to employ this approach is described in [17] and many of subsequent models are based on it.

Next subsections of the paper make a brief review of different models taking into account uncertainty. We also analyze ability of the models to deal with the other two issues for reliability estimation of embedded software systems as pointed out in the introduction. For ease of reference, we have used the name of the first author as the name of the model, even though these models are collective works of several researchers as indicated by the references. Models are presented in ascending alphabetical order, according to the name of the first author.

A. Adams model

Tom Adams actually presented an approach towards calculation of confidence intervals of reliability with respect to uncertainty caused by unknown operational profile [1]. This model does not take into account uncertainty in reliability parameters of modules within the system. Adams model does not take into account estimation of component reliability parameters nor dependency between component failures.

B. Gokhale model

This model presents reliabilities of components and operational profile not as point estimates but as stochastic values themselves. The key of the model is generation of analytical functions that give the mean and variance of component reliabilities and operational profile and consequent combination of these functions into calculation of the mean and variance of reliability of the whole system. This model also does not take into account estimation of component reliability parameters and dependency between component failures.

C. Goseva-Popstoyanova models [6], [7], [9]

Very similar to Gokhale's is a group of models resulting of research, carried out in West Virginia University, USA. Models presented in [7], [8] calculate system reliability as a random variable is proposed. Authors propose to estimate its value given that the parameters of the model (i.e. component reliabilities and operational profile transition probabilities) are also random variables. First of the two models is based on moments of random variables and the second – on Monte-Carlo simulation. The first one is based on presentation of the reliability in a Taylor series and approximating it by taking into account the first one or two terms in it. However, this approach does not allow taking into account uncertainty in the operational profile of components. Both these facts make it an approximate method for assessing system uncertainty. The method based on Monte-Carlo Simulation allows taking into account both uncertainty in component reliabilities and operational profile. Nevertheless, it assumes that the reliability and transition probabilities follow a given a priori (for instance Beta) distribution. Assumption about such distributions is also a possible source of uncertainty.

In *Kamavaram* model [9] entropy is used to quantify the uncertainty of system reliability, based on uncertainty in operational profile and uncertainty in component reliabilities. However, this method estimates only uncertainty and does not allow for estimation of a value for the system reliability itself. This is the reason that we do not distinguish it in our analysis.

Similarly to previous cases this group of model also takes component reliability parameters as given and does not take into account dependency between component failures.

D. Popic model

Reference [11] presents a model that extends Singh model (cf. subsection F.) and makes it capable to take into account error propagation from one component to another. This way it is able to model dependency between components. For this purpose this model takes into account the error propagation probability in system architectures. It represents the probability that an erroneous state, generated at one component will not be detected but will propagate to other components during system execution. As this is an extension of Singh model it also assumes existence of information about failure rates of components in the architecture.

E. Roshandel models

One of the earliest works taking into account uncertainty in operational profile of software components that build the software system is presented in [12].

Its approach towards the estimation of individual components reliability when both the implementation and the data for the operational profile are unknown. Component architecture is modeled with an extension of classical Markov chain, called Hidden Markov Model (HMM) [16]. This is the main point of ability of this to model uncertainty. In fact it models presents uncertainty of a transition between states of the chain. Transition probabilities of HMM are estimated by an iterative algorithm, which upon convergence provides optimal component reliabilities.

This model is further elaborated in [18], [19], where different views of software architecture, presenting systems structure and behavior are employed. Further, a so-called *Global Behavioral Model* is constructed. It represents the behavior of the software system as a function of the collective behavior of its constituent components. Behaviour of components is modeled using a set of concurrent state machines, which means that the current state of the system is presented with a set of component states. In the next step, a *Dynamic Bayesian Network* is used to estimate reliability of the system, combining the behavioural model with individual components' reliabilities. The approach is able to model a rich variety of uncertainties: uncertainty of individual component reliability values; uncertainty of system startup process, and uncertainties of human-system interactions. However, the model does not give any guidelines to quantitatively estimate uncertainty. Authors only make a sensitivity analysis of the model towards variations in reliability parameters of components and operational profile.

F. Singh Model

Singh model [13] is not specifically aimed at uncertainty, however similarly to Roshandel model it uses a Bayesian approach towards reliability calculation. Goal of this model is to be applicable early in the development process where reliability values of system modules and operational profile are not known. Nevertheless, the model assumes existence of information about failure rates of components in the architecture. This way, it estimates the mean and variance of software failure probability. However for this purpose authors also assume some given distribution (Beta distribution) of reliability parameters of individual components within the system, which as said above may also be a source of uncertainty. Singh model does not explicitly consider component failure dependence.

G. Zhang model [9]

This model upgrades the one presented in [20], which is essentially a path-based model, with ability to model component reliability as a function of transition probability to and from other components within the system. This way it becomes possible to take into account uncertainty due the environment in which a component is integrated and the systems operational profile. In fact, the empirical validation of the model shows that even when it is known in advance, operational profile of the system itself may introduce some uncertainty in the reliability of the components.

III. Discussion and analysis

In this section we analyze architecture-based software reliability models with respect to their suitability for embedded software systems. Table 1 shows a comparison of the models with respect to the three issues that should be addressed when estimating reliability of embedded software systems.

TABLE I

COMPARISON OF ARCHITECTURE-BASED RELIABILITY MODELS

	Modeling of uncertainty		**Dependency of component failures**	**Estimate reliability parameters of components**
	Op. profile	**Comp. reliability**		
Adams model	Yes	No	No	No
Gokhale model	Yes	Yes	No	No
Goseva moments model	No	Yes	No	No
Goseva-Monte-Carlo model	Yes	Yes	No	No
Popic model	Yes	Yes	Yes	No
Roshandel model	Yes	Yes	No	Yes
Singh Model	Yes	Yes	No	No
Zhang model	No	Yes	No	No

As seen from table one, none of the known models take into account all the identified issues for embedded systems. Dependency between component failures and estimation of individual component reliabilities are addressed by only one model each. Although uncertainty is a matter of broad research only one models (Adams) actually assume it is a variance of the reliability estimation within some confidence interval. We believe that reliability should not be presented as a point value, but together with the variance in this value, i.e. with its confidence interval.

Additional issue is that empirical study of some of the presented models have shown that uncertainty in reliability estimation of the overall system is mostly influenced by the reliability of components that build the system reliability [21]. Usage profile, i.e. probability of transition of execution from one component to another has significantly less impact on system reliability. This way it is more important to be able to take into account uncertainty due to inaccurate component reliabilities than inaccurate usage profile.

IV. Conclusion

Reliability is a major concern to be taken into account when designing and implementing embedded software systems. Currently, there exist a lot of unresolved issues with architecture-based software reliability models and uncertainty in estimation is one of them. This paper presents a review of current state of the art in solving the issue of uncertainty.

Analysis of the models has shown that all of the important issues when modeling reliability of embedded systems need further research. In this respect our plans for further research include development of modeling framework that takes into account most of the issues listed in this paper.

Acknowledgement

The work presented in this paper was partially supported by grants from the National Science Fund, part of the Ministry of Education and Science in Bulgaria under the PD01-0106 (ARECS) project.

References

[1] Adams, T., Total Variance Approach to Software Reliability Estimation, IEEE Transactions on Software Engineering, vol. 22, no. 9, Sept. 1996, pp. 687-688.

[2] Avižienis, A., Laprie, J-C., Randell, B., Basic concepts and Taxonomy of dependable and secure computing, IEEE Trans on Dependable and Secure computing, Vol. 1, Issue 1, Jan -March 2004.

[3] Dimov, A. and Punnekkat, S.: “On the Estimation of Software Reliability of Component-Based Dependable Distributed Systems”, In Proc. of the 1st International Conference on the Quality of Software Architectures (QoSA 2005), LNCS 3712, Springer-Verlag, September 2005, Erfurt, Germany, pp. 171-187.

[4] Gokhale, S., Architecture-Based Software Reliability Analysis: Overview and Limitations, In IEEE Transactions on Dependable Security Computing 4(1): 32-40 (2007).

[5] Gokhale, S., Quantifying the Variance in Application Reliability, in Proceedings of Pacific Rim Dependability Conf., Mar. 2004, pp. 113-121.

[6] Goseva-Popstojanova, K. and Kamavaram, S., Software reliability estimation under certainty: generalization of the method of moments, In Proc. of the 8th IEEE International Symposium on High Assurance Systems Engineering, 2004, March 2004, pp. 209-218.

[7] Goseva-Popstojanova, K. and Kamavaram, S., Assesing Uncertainty in Reliability of Component-Base Software Systems, 14th International Symposium on Software Reliability Engineering (ISSRE’03), 2003.

[8] Goseva-Popstojanova, K., Trivedi, K.S.: Architecture Based Approach to Reliability As-sessment of Software Systems, Performance Evaluation, Vol.45/2-3, June 2001

[9] Kamavaram, S., and Gosseva-Popstojanova, K., Entropy as a Measure of Uncertainty in Software Reliability, In Proceedings of the 13th International Symposium on Software Reliability Engineering, 2002 (ISSRE 2002), pp. 209-210.

[10] May, J., Testing the reliability of component-based safety critical software, In Proceedings of 20th International System Safety Conference (S. Thomason, editor), System Safety Society, August 2002, pp. 214–224.

[11] Popic, P., D. Desovski, W. Abdelmoez and B. Cukic, Error Propagation in the Reliability Analysis of Component Based Systems, In Proceedings of the 16th IEEE international Symposium on Software Reliability Engineering (ISSRE), November 2005, Washington, DC, pp. 53-62.

[12] Roshandel, R., Medvidovic, N.: Toward Architecture-based Reliability Estimation, In Proc. of the Workshop on Architecting Dependable Systems, International Conference on Software Engineering (ICSE 26), Edinburgh, UK, May 2004

[13] Singh, H., Cortellessa, V., Cukic, B., Gunel, E., Bharadwaj, V.: A Bayesian approach to reliability prediction and assessment of component based systems. In Proc. of 12th International Symposium on Software Reliability Engineering (ISSRE’01), 2001

[14] Zhang, F., Zhou, X., Chen, J. and Dong, Y., A Novel Model for Component-Based Software Reliability Analysis, In Proceedings of the 11th IEEE High Assurance Systems Engineering Symposium, (HASE) 2008, pp.303-309, 3-5 Dec. 2008.

[15] Zhang, X and Pham, H., An analysis of factors affecting software reliability, In Journal of Systems and Software, 50(1), 2000, pp. 43-56.

[16] Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, In Proceedings of the IEEE, Vol. 77(2), pp 257-286, 1989.

[17] Cheung, R. C.: A user-oriented software reliability model, IEEE Transactions on Software Engineering, 6(2): 118-125, 1980.

[18] Roshandel R. et al. A Bayesian Model for Predicting Reliability of Software Systems at the Architectural Level. In Proceedings of 3rd International Conference of Quality of Software Architecture (QoSA 2007), Boston, MA, July 2007.

[19] Cheung, L., Roshandel, R., Medvidovic, N., and Golubchik, L. 2008. Early prediction of software component reliability. In Proceedings of the 30th international Conference on Software Engineering (Leipzig, Germany, May 10 - 18, 2008). ICSE '08. ACM, New York, NY, 111-120.

[20] S. Yacoub, B. Cukic, H. Ammar, “A scenario-based reliability analysis approach for component-based software”, IEEE Transactions on Reliability, Vol. 53, No. 4, pp. 465-480, December 2004.

[21] Goseva-Popstojanova K. and Hamill, M., Architecture-Based Software Reliability: Why Only a Few Parameters Matter?, In Proceedings of the 31st Annual International Conference on Computer Software and Applications Conference, 2007, vol.1, pp.423-430, 24-27 July 2007.