5th SASTech
Iran |Mashhad
May 12 - 17 | 2011
5th Symposium on Advances in Science & Technology

Offering an approach on the basis of Bayesian network to evaluate the reliability of architecture in a family of software products

Reza Vaziri, Islamic Azad university-Khoozestan Science and Technology Branch, Iran; eng.rvaziri@gmail.com
Reza Javidan, Islamic Azad University – Beyza Branch, Iran; reza.javidan@gmail.com
Mohammad Hosein Yektaei,Islamic Azad university-Khoozestan Science and Technology Branch, Iran; mh_yektaie@yahoo.com
Paper Reference Number:
Name of the Presenter: Reza Vaziri

Abstract

Software assembly-line is an important issue which puts a lot of emphasis on reusability. In the process of software assembly-line, as in the software production, architectural design begins after needs assessment and the possible alternatives to the architecture of the software are also considered.
Nowadays, there are some methods such as feature RSEB, CONIPE and etc to assess the software architecture. Mostly these methods are not compatible with the principles of assembly-line architecture which is based on reusability. But the change set model is a model which is the most important one because it takes the different arrangements of items into account to achieve reliability. Each change set consists of some items which are next to each other in different products and each product consists of a combination of these change sets. State Machine can be used to display the general behavior of the system and Bayesian network can also be used to assess the reliability of each architectural.
This paper presents a model which applies the Bayesian network to assess the reliability of assembly-line architecture. To achieve this purpose, the architecture is simulated first with an appropriate method, and then the assessment is done. The simulation results show significant improvement in comparison of the previous methods.

Key words: software architect, architectural evaluation, software assembly line, reliability, modification set.

1. *Introduction*

Product line is an important concept which has long been used in the industrial production and manufacturing, and now it has received lots of attention in the software industry. A software product line in an industry like cell phone product line contains softwares which are commonly used in cell phones. Music and message sending and receiving softwares are of these types.
The product line of software engineering is valuable for the developing companies. Long time reusability, quality, the product supply in short time are attributes which make their production economical. In the process of software production be it one product or a family of software products, architectural design starts after needs assessment (figure 1) and the possible options and different architectural decisions are presented for the architecture of

software systems. It is quite clear that the role of architecture in software product line is very important and crucial, since a good architecture can attract lots of customers and sell the family of products in large volumes for any companies. On the other hand, choosing and designing an inappropriate architecture result in low-quality products and jeopardize the investment and the market of not only one product but also a family of products.
To save time and reduce cost, proposing a formal description from the architecture product line and its assessment before the start stages of system installment is very useful and helpful. In recent years, different models and methods have been proposed to assess the reliability of architecture which are mostly based on one product.
These methods hardly assess the family of software products.
In addition to high price and time, they are also in conflict with the design principles and product line evaluation which are based on reusability.
In this paper, Bayesian model is used to evaluate the reliability of each architectural item at the time of the architectural design. Moreover, the architectural evaluation model is expanded and a new model for the architectural evaluation is offered. The proposed model has considered not only the items but also some relationships in the evaluation of reliability and this in turn, turns out quite useful in dynamic environments and working systems in which relationships play an important part in the total reliability of systems.

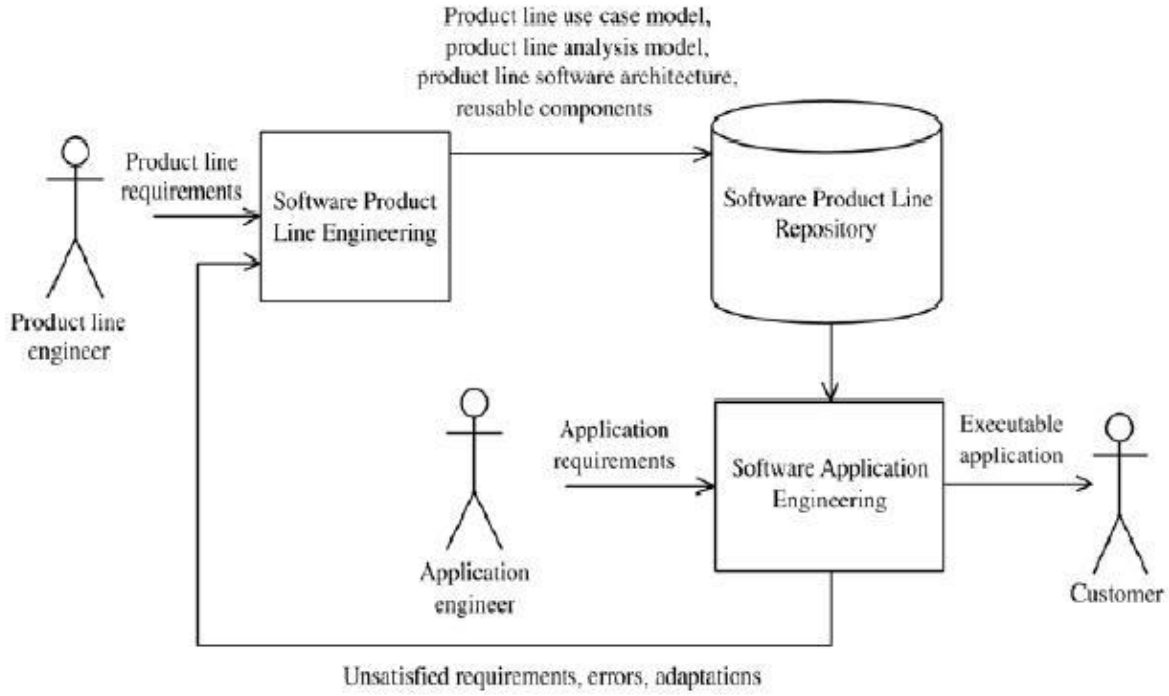

Figure 1

2. . The statement of the problem
The early predictability of the soft ware reliability in the early stages of its development for example architectural design , has helped a lot in the promotion of quality level in an effective and low-cost manner.
In the software product line, this requirement is strongly felt. Lots of works have been done on the predictability of the architectural reliability but all have been at system level and have accepted as default that the reliability of items is determined in advance and also more importantly, none of these works have concentrated on the architecture of the software product line.

In the product line of the software and the family of products, diversity and adaptation, task setting and non-task setting cell phone softwares are of this type.
The diversity in the requirements of the task setting is in fact the development and the adaptation of the product which also compasses the diversity in the requirements of the non-task setting [7]. One of the issues facing the researchers today is the development and adaptation effect of the software family on the quality attributes such as efficiency and reliability. The question here is, with each change in the product line of one product to another, how the quality attributes change?
Many of the modern software families are either installed as software or are distributed in the network. In such an environment, the intended quality attributes for the interested people are reliability and accessibility [8], [9].
The reliability is the amount of the probability that the system can perform its tasks under some specific architectural limitations.
To reach a product line with an appropriate reliability, the quality attributes should be considered from the early stages of product development. Software architecture is the first stage which can make decisions about and can analyze the quality requirements. To do this, in the performance of architectural designs we need to do them with architecture analysis method which evaluates the software reliability. In recent years, lots of methods have been proposed to evaluate the reliability of software architecture. From a very advanced point of view, they are divided into quality and quantity methods. Quality methods [3] have been used since 1970. From another perspective, quality methods are also divided into two categories:
Black box method which does not take into account the internal structure of the software during evaluation [1] and white box method which involves the internal composing items in evaluation.
A common part in all methods of reliability evaluation is the presence of methods which are useable in a single architecture and none of these methods support the diversity and adaptation which are discussed in case of product line architecture [10].
Applying the available methods to evaluate the product line architecture has been very costly and also they do not provide a complete set of information about the reliability of the different configurations of the product line architecture.
In this paper, Bayesian model is used to evaluate the reliability of each of the architecture items at the time of the architectural design and also the researchers are to propose a method to evaluate different configurations of architecture and to combine it with change set-based modeling to evaluate.

Product line architecture
There are some points of importance which should be considered in this regard: since there are different methods for the product line architecture, change set method has been used and due to the attributes of this modeling method, a method is proposed for the evaluation of product line architecture. Each change set consists of some items which are next to each other in different products and each family of software consists of a combination of these change sets. To evaluate the product line architecture, first each change set is evaluated and then the combination of these change sets is evaluated with each other.3. Algorithm and its methodology

3.The proposed method for the evaluation of product line architecture
In the first stage, we start with an architectural modeling method which is a basic part for architecture evaluation and a high reusability. This is done to evaluate the reliability of the product line architecture.

Among the different methods which are used to model the family of software products, the change set-based modeling method was chosen since the internal behavior of items and relationships and their adaptations are involved in reliability of the software products, in our suggested model, we deal with this issue.
Since we use change set modeling method among other available methods, the architectural evaluation of the product line software consists of the basic change set evaluation and the combination of change sets with them and up-dating the evaluation.

3.1 The principles of the suggested methods
In this paper, the architecture evaluation method is expanded and a modeling based architecture evaluation is offered which is based on change set of the product line architecture .the method presented involves not only the items but also the relationships in the evaluation of the reliability. This makes the method useful in the dynamic environments and moving systems in which relationships play major roles. Figure 2 shows an overview of the method.

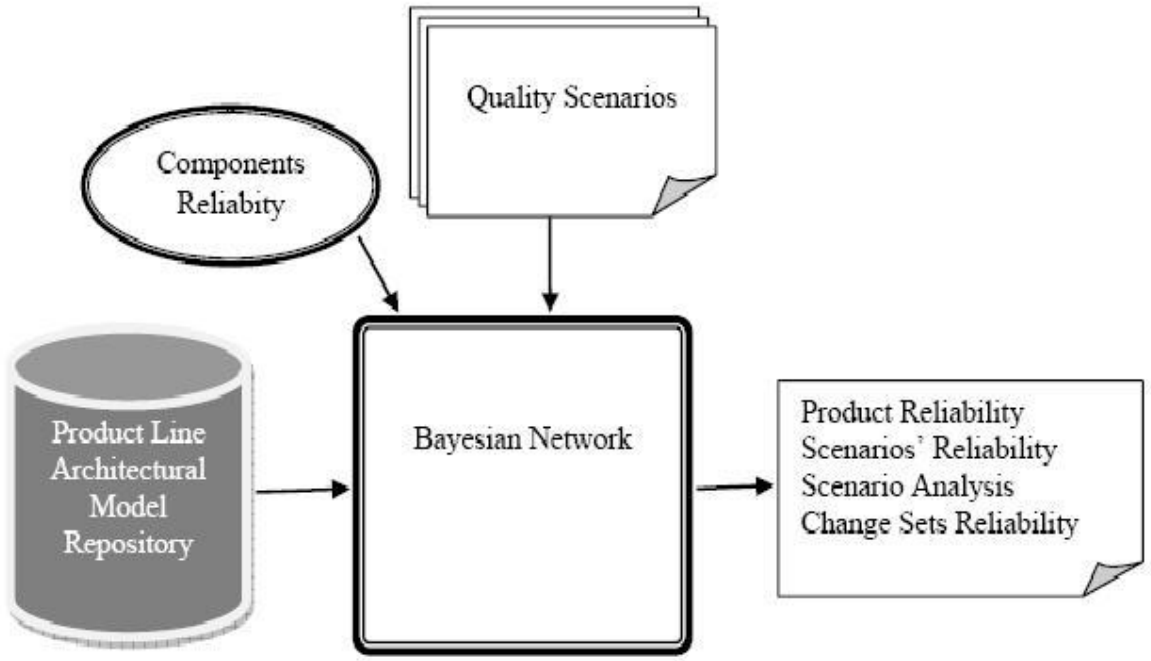

Figure 2

As it is clear from figure 2, the inputs of this method (architectural models), include all the change set adaptations of the designing, the state machine graph of the items and relationships and also the primary reliability of the items and relationships.
The amount of reliability might be available before hand like the reliability of the premade items or it might have been obtained through the available methods which deal with the evaluation of the reliability of the items.
The outputs of the method are as follow: the reliability of each change set, the reliability of the products, the reliability of the extracted quality scenarios and the possibility of the analysis of the extracted scenarios. Analyzing the scenario involves considering cases such as the primary reliability effect of an item on the scenario or the effect of architectural decisions on the reliability in that scenario and so on. Since quality attributes are accounted for in different scenarios and since the general quality of the systems is of

ambiguous nature, the proposed method (model) not only can evaluate the reliability of a change set or a combination of them, but also can predict the reliability of the scenarios.
In this model, the first step is evaluating the change set to assess the product line architecture.
A change set is nothing but a set of interacting relationships and items which are interacting with each other to access the system properties.
To evaluate the reliability of change sets, the graphic models are used because they possess probable deductibility.
Bayesian network is one of these models. This network is a probable directional graph in which nodes show the variables and edges show the interdependence between these variables. Bayesian network show the probable relationships between nodes through the conditional probability.
By drawing the structure of this network and assigning the probable values of some of the probable distributed nodes the other nodes are deductible.
The state machine can be used to display the general behavior of the system in different configurations. The general behavior of the system is shown in the form of a set of interacting state machines. A set of item state machines and relationships are used to display the general behavior of the system and to construct the Bayesian network to evaluate the architectural reliability. In the constructed Bayesian network, the interdependence between the reliability values of state machines is displayed. Then, this model is combined with the failure states which show the states in which items and relationships have displayed inappropriate behavior. The reliability of the item configurations equals the total probability of entrance to a failure state. The overall steps of this model are as follow:
First, a basic change set (a change set which is common among all the products) is chosen and the following steps are performed:
 Constructing (updating) Bayesian network
The numerical display of Bayesian network: the probable assignment between nodes
 Obtaining the probability of failure nodes
 Using the radar graph to obtain the cumulative effect of available failure nodes in the scenario path.
The next step is the repetition of the mentioned steps for the basic change sets and their combination.
In the combination of the change sets, only the Bayesian network is updated.

References

[1] Gomaa, H., Designing Software Product Lines with UM From Use Cases to Pattern-Based Software Architectures. s.l. : Addison Wesle, 2004.
[2]IEEE Standards Description: 1471, Available:http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html, 2000.
[3] Rasmussen, J.L., and Singh, M., Designing a Security System by Means of Coloured Petri Nets.In Proc. ICATPN, Springer- Verlag, 1996.
[4] Roshandel, R., A Bayesian Model for Predicting Reliability of Software Systems at the Architectural Level. USC, PhD Thesis, 2006.
[5]Eeles,P.,What is a software architecture?, Available at: http://www.ibm.com/developerworks/rational/library/feb06/eeles/index.html, 2006.
[6] Hendrickson, S. A. and van der Hoek, A., Modeling Product Line Architectures through Change Sets and Relationships. In Proceedings of the 29th international Conference on Software Engineering. International Conference on Software Engineering. IEEE Computer Society, ashington, DC, pp. 189-19, 2007.

[7] Clements, P., Kazman, R., Bass, L., Software Architecture in Practice, Second Edition, Addison Wesley, 2003.
[8] Eeles, P., What is a software architecture?, Available at: http://www.ibm.com/developerworks/rational/library/feb06/eeles/index.html, 2006.
[9] Goseva-Popstojanova, K., and Trivedi, K.S., Architecture- based approach to reliability assessment of software systems: Perform. Evaluat, 2001.
[10] Clements, P., Kazman, R., Bass, L., Software Architecture in Practice, Second Edition, Addison Wesley, 2003.
[11] Eden, A. H., Kazman, R., Architecture, Design, Implementation, 25th International Conference on Software Engineering, pp. 149-159, 2003
[12] Hadipour Sanati, H., an Executable model of Softtware Architecture, Thesis, Shahid Beheshti University, 2005.
[13] Jensen, K." Coloured Petri Nets. Basic concepts, analysis methods and practical use. Analysis Methods,". Springer, Berlin, pp. 145-153, 1994
[14] Hotz, L., Krebs, T., Wolter, K., Nijhuis, J., Deelstra, S., Sinnema, M., MacGregor, J., Configuration in Industrial Product Families –The ConIPF Methodology, IOS Press, ISBN 1- 58603-641-6, 2006.
[15] Czarnecki, K., Helsen, S., Eisenecker, U., Formalizing cardinality-based feature models and their specialization, Software Process Improvement and Practice 10 (1), pp. 7–29, 2005.
[16] Wagner, F., Schmuki, R., Wagner T., Wolstenholme, P., Modeling Software with Finite State Machines: A Practical Approach, by Taylor & Francis Group, LLC, 2007.
[17] Cheung, L., Roshandel, R., Medvidovic, N., Golubchik, L., Early prediction of software component reliability, Proceedings of the 30th international conference on Software engineering, Leipzig, Germany, 111-120, 2008.
[18] Birrell, A. D., and Nelson, B. J., Implementing Remote Procedure Calls. ACM Transactions on Computer Systems, pp. 345 – 365, 1984.
[19] Huang, C., Liang, W. Object-oriented development of the embedded system based on Petri-nets, Computer Standards & Interfaces on Elsevier, pp. 187-203, 2003.
[20] Vibhu S., Kishor S., Quantifying software performance, reliability and security: An architecturebased approach, Elsevier, J. Syst. Softw, pp. 493-509, 2006.