Reliability Evaluation based on the AADL Architecture Model

Dongyi Ling, Shihai Wang*, Bin Liu and Xiaoqi Xing
Beihang University School of Reliability and Systems Engineering, Beijing, China
*Corresponding author, Email: lingdongyi@foxmail.com, {wangshihai, liubin}@buaa.edu.cn, xingxiaoqi@live.cn

Abstract—The structure of the embedded system gets much more complicated. Current basic architecture analysis and design language (AADL) reliability model cannot meet the requirements of software reliability being evaluated while being designed. For the present, reliability evaluation needs abundant fault analysis which can not be realized in the early of software development. The article has come up with a methodology based on system architecture using AADL to perform reliability evaluation at early development with good understanding of the reliability rules and transformation from AADL based system architecture to Petri Net, a one-to-one mapping was achieved between AADL elements and Petri Net elements. Using current mathematic model of Petri Net to evaluate the reliability of software architecture. At last, a flight control system of AADL model was given as an example to validate the availability of the given method.

Index Terms—AADL; Reliability Evaluation; Petri Net

I. INTRODUCTION

As the development of embedded system, its structure gets much more complicated, also requires heavier investments and longer period to complete the development as well as higher expectation to the non-functional requirement (scheduling, reliability and security). The earlier method for developing embedded system can no longer meet the current requirement. Therefore, the industry has come up with a methodology based on Model Driven Architecture (MDA) [1], system development has been brought to a higher level – the level of model. Coding can be done autonomously according to different computation platforms. Modeling is now the core of development. We can perform analysis of non-function requirements on the model directly, and greatly reduce the development time, thereby lower the cost. At present, system reliability evaluation method is mainly divided into two types: one method is system reliability evaluation based on code, the other is system reliability evaluation based on component. The method of system reliability evaluation based on code can be performed easily, but the assess accuracy is not high for the method influence of human factors. To the embedded system with high reliability requirement and more complex, this method can only evaluate reliability on the software code, but not on the period of system design. Compare these two method, system reliability evaluation based on component is more suitable for the system development based on the model.

To incorporate this trend, Society of Automotive Engineers (SAE, United States) has published new standard AS5506 – Architecture Analysis and Design Language (AADL) [2] [3]. AADL is used to describe the architecture of complex real time embedded system. AADL doesn’t concern about how the system will be realized specifically, rather, it describes the system by defining parts, the interaction between them and the binding between software and hardware. References [4] [5] [6] have proposed the theories of transforming the reliability model of AADL based error to GSPN reliability computing model. With these theoretical analysis, one can achieve the transformation from GSPN reliability computing model from library of basic error appendices, basic dependency relations (Out-in, Propagation), and advanced dependency relations (Guard In, Guard Out, Guard Event, Guard Transition). The result of this transformation can be evaluated by using basic reliability test based on AADL reliability model. In real application, a software reliability model based on AADL error model requires software to perform well-round analysis under failure mode. This is hard to be done during the design period of software. For the evaluation requirement during the design period of software, which needs carried out on the basis of software architecture reliability evaluate during the design period. Current basic AADL reliability model cannot meet the requirement of software reliability being evaluated while being designed. The article has come up with a methodology based on system structure using AADL to perform reliability test. We study the distinguishing feature between AADL architecture model and Petri Net model, analysis the feasibility of the AADL model transfer to Petri Net. With good understanding of the rules of transformation from AADL based system model to Petri Net, a one-to-one mapping was achieved between AADL elements and Petri Net elements, the component of AADL is transfer to the Place and the Transition in the Petri Net, and the connection is transfer to the arc in the Petri Net. Definitions for transitional relations were also made by formalizing the problem to simplify the transformation and the evaluation of system reliability based on the current Petri Net related mathematical model. On this basis, transfer from the Petri Net model to the system architecture of Petri Net model in formal transformation, and done the mathematical proof of the work, makes the transformation of
architecture Petri Net model can make use of the mathematical properties of Petri Nets for reliability calculation.

Which designers can be carried out on the architecture of the system reliability and power consumption, weight and cost trade-off, validate and optimize the system structure design of the model. Prior to that based on the reliability of the structure model of the AADL evaluation can only be working with a abundant of error analysis by system designers, which need plenty of manual labour and working experience on the system. In this paper, the reliability of the evaluation method can ignore the effect of error data in the system, by using the basic reliability of components for architecture to evaluate the system and reliability, the reliability of the system is analyzed with the weak links, and system structure for an early assessment of the work. Therefore, the reliability analysis for system model based on AADL architecture model provide theoretical basis for the design and validation is given.

II. RELIABILITY ANALYSIS BASED ON AADL MODEL

A. Introduction to AADL

AADL is a graphical modeling language used in the embedded system. The core language elements include components, component type, component implementation, property sets, packages, and annex libraries.

Components are the cores of AADL. System software and hardware are treated as combinations of components, and thus a whole system is described as a set of interactive components. Three types of components in AADL are software components (data, subprogram, process, thread, and thread group), hardware components (RAM, bus, devices, and processing unit), and composite components (system). Software components are connected to hardware through its individual properties. Components are defined according to its type and realization, and can be inherited through the use of keyword ‘Extend’. A type of components can have one more multiple realizations. Components type and realizations can also use property sets and package [7].

Mode refers to the running state of system, mode and its conversions describe the dynamics behavior when system is running. AADL expands through attachment [8] [9].

The interface specification of a component is called its type. It provides features (e.g. communication ports). Components communicate one with another by connecting their features (the connections section). Each component describes their internals: subcomponents, connections between these sub-components [10], etc.

An implementation of a thread or a subprogram can specify call sequences to other subprograms, thus describing the execution flows in the architecture. Since there can be different implementations of a given component type, it is possible to select the actual components to be put into the architecture, without having to change the other components, thus providing a convenient approach to application configuration.

AADL allows properties to be associated with AADL model elements. Properties are typed and represent name/value pairs that represent characteristics and constraints. Examples are the period and execution time of threads, the implementation language of a subprograms, etc. The standard includes a predefined set of properties and users can introduce additional properties through property definition declarations. For interested readers, an introduction to the AADL can be found in.

Other languages can be integrated in AADL models by means of annex libraries. These languages can be added on each component to describe other aspects. Some annex languages have been designed, such as the behavior annex or the error model annex. The error model annex [11] defines states of a component, its potential faults and errors and their propagation in the system.

AADL provides two major benefits for building safety-critical systems. First, compared to other modeling languages, AADL defines low-level abstractions including hardware descriptions. Second, the hybrid system components help refine the architecture as they can be detailed later on during the design process.

B. Reliability Analyze Method Based on AADL

During the process of AADL reliability analysis, first step is to establish the AADL reliability model. The model used in this article is generated based on AADL structure model.

As a high level modeling language, AADL cannot be used directly in the reliability computation and analysis. Therefore, it is required to transfer AADL based model to a lower level formalized model, and computation and analysis can be conducted [12]. Foreign scholars combine EMA with different kinds of analytic methods such as dependency diagrams (DD), Fault Tree Analysis (FTA), and Generalized Stochastic Petri Nets (GSPN) to perform qualitative analysis. However, these models and techniques can only describe system static structure and behaviors. Basic formats of DD are series, parallel, redundancy, but it ignores system time variant dynamic characteristics; FTA does not consider the order of events when error happens, even later on with the concept of dynamic fault tree, it still awaits further exploration and verification. Petri Net is a model without any global control, it is a graphical and mathematical tool to describe and simulate a system. Petri Net is intuitive and visualized. Being capable of describing a complex system and its dynamic behavior in such a way as well as possessing many other mathematical properties makes it convenient to describe and analyze asynchronous simultaneous system.

The reliability analysis process used in article can be expressed in terms of the flow chart (Figure 1).

1) Evaluate the needs during reliability analysis
2) Understand each component
3) Map each functional components, data, and even stream onto AADL components and connections
4) Categorize the function of each components into structural parts and connection parts
5) Convert AADL structure model according to Petri Net mapping relation
By using Petri Net based reliability model and the Petri Net model, perform reliability computation. Collect figure from reliability analysis, review with the requirement and obtain final result for system reliability.

III. AADL MODEL TRANSFORMATION ORIENTED PETRI NET

A. Introduction to Petri Net

Definition 1: Petri Net consist of triple, which is $N = (S, T, F)$, and to meet:
1. $S \cup T$;
2. $F \subseteq (S \times T) \cup (T \times S)$;
3. $\text{dom}(F) \cup \text{cod}(F) = S \cup T$.

$S = \{S_1, \ldots, S_n\}$ is assemblage of token, $I$ is number of token, the token represent the state or condition of the system; is assemblage of transition, $m$ is number of transition, the transition represent the event which change the system state, such as component fault and fixed active; F is key element to connect system state and event, which with two-way [i], the visualization is expressed as the direction arc which connect with token and transition; $\text{dom}(\cdot)$ and $\text{cod}(\cdot)$ are assemblage of define and domain.

Petri Net can describe dependencies between various events, analysis the state of the system dynamic evolution, conflicts and deadlocks [13].

Definition 2: In $SA = (C, L)$, $C$ is a set of place, $L$ is a set of transition, construct a net which is $N = (S, T, F)$, $N = (P, T, F)$ is a SA net, $F$ is a collection of edge which exist relationship between P and T, $M_0$ represent initial mark of SA, then the net $N$ which is just construct is a Petri Net.

Proof: (1) $\because P$ is a set of Software Architecture(SA), it can learn that is limited from P is limited, so P and T is limited set.

(2) It can be known from the place and transition definition clearly: $P \cap T = \Phi, P \cup T = \Phi$.

(3) $\because F$ is a set of edge which have relationship between P and T, while $(P \times T) \cup (T \times P)$ represent a set contain all the edge between P and T.

(4) $\forall x \in \text{dom}(F) \cup \text{cod}(F)$, it can be learn form the definition of $\text{dom}(F)$ and $\text{cod}(F)$ in Petri Net that $x \in P \cup T$; to $\forall x \in P \cup T$, x in the net N, we are sure to find a node of associated with it clearly: $\exists y, (x, y) \in F$ or $(y, x) \in F$.

(5) We can learn from performability of software system that the Net $\sum_i M_i$ is performace from initial mark $M_0$. That is the Net $\sum_i M_i$ is active.

Definition 3: A Petri Net System $SA\text{PN} = (N, W, M_0)$, $N = (P, T, F)$, $W = \{P_i\}$, $P_i$ is transition probability, $M_0$ is initial mark of N, is considered to be a Software Architecture Petri Net, If and only if:

1. N have two special place s and e, $s = \Phi, e = \Phi$, N have two special transition $t_s$ and $t_e$, $s = \{s\} \land s = \{t_s\} \land \{t_e\} = \{e\} \land e = \{t_e\}$.

2. If we add a transition connect s and e, $s = \{e\}, t_e = \{s\}$, said it as $N'$, which called extend net of N, according $N'$ as strongly connected.

The SA reliability model: to analyze the reliability of the system is composed of several parts. To establish the reliability model through structural decomposition is a very important step. According to the hardware system reliability model can derive the two typical SA reliability model in series and parallel.

(1) Series SA model

In order to facilitate the calculation, consider each element as a sub system $S$. The reliability model of the series SA can be regarded as a series system composed of N subsystems. Assume the $i$ subsystem for the lifetime of $X_i$, the reliability of it is $R_i(t)$ at the time of $t$, and $X_1, X_2, \cdots, X_n$ mutual independence, and the lifetime is $X_s$, the reliability of it is $R_s(t)$ at time $t$, then:

$$R_i(t) = P(X_i > t) = P(\min(X_1, X_2, \cdots, X_n) > t) = \prod_{i=1}^{n} P(X_i > t) = \prod_{i=1}^{n} R_i(t)$$

(1) parallel SA model

Assume $SA$ is consist of $n$ elements, a function as a treatment to a certain element of complete, That is to say, If an element of $E$ in normal working, SA is called parallel SA which consists of $n$ elements. Here, we put each element as a sub system, Assume the $i$ subsystem for the lifetime of $X_i$, the reliability of it is $R_i(t)$ at the time of $t$, and $X_1, X_2, \cdots, X_n$ mutual independence, and the lifetime is $X_s$, the reliability of it is $R_s(t)$ at time $t$, then:

$$R_i(t) = P(X_i > t) = P(\max(X_1, X_2, \cdots, X_n) > t) = 1 - P(\max(X_1, X_2, \cdots, X_n) \leq t) = 1 - P(X_1 \leq t, X_2 \leq t, \cdots, X_n \leq t) = 1 - \prod_{i=1}^{n} P(X_i \leq t)$$

(2) parallel SA model

$$1 - \prod_{i=1}^{n} (1 - P(X_i > t)) = 1 - \prod_{i=1}^{n} (1 - R_i(t))$$
B. AADL Architecture Model to Petri Net Transformation

The basic transformation rules is map the most basic components of AADL into the place of Petri Net model, putting connections of software architecture components are mapped into transition of the Petri Net, and putting components with only out-data/event feature mapped into place with token of Petri Net, then put the connections in AADL mapped into connection in Petri Net. As shown in figure 2.

![Figure 2](image.png)

Figure 2. Transfer rule from AADL model element to Petri Net

Different component transformation can be used for different levels of modeling in a system. In the system level, it can transfer the subsystem component as part of the model transformation; and the hardware component such as bus, device, memory and processor can be transferred as model transformation as well. In the subsystem level, it can transfer components process, thread, data and so on as part of the model transformation.

The figure below is a basic AADL model, the model contain a system component called complete. Five process component in the complete which are init, data_check, bit, cds and start cds. The system implement a function that is active bit or cds. When the event is active, the system will perform from init till bit return a result or cds start normal.

The AADL model transform follow the rules as below:

![Figure 3](image.png)

Figure 3. A basic AADL model

C. Reliability Calculation Based on the Transformation of AADL System Model To The Petri Nets

Software architecture Petri (SAPN) net is a reliability assessment model for modeling based on the software architecture. The model uses software components and connections [14] and the correlation of them to structure a Petri Net model [15], abstract the Petri Net which transfer from the AADL model, express the component of SA in place of Petri Net, express the connection of SA in transition of Petri Net, hence the interactive of the component and the connection in SA can be abstracted as process of one place to another place transition [16]. Then weighted in the SAPN model for each component elements do weighted processing, its model abstraction are defined as follows:

Definition 3 (weighted SAPN): Weighted SAPN is eight dimension ordered couple \((P, T, H, S, EN, Pr, R, R)\), \(P\) assemblage of place, \(T\) is assemblage of transition, total function \(H: T \times P \to [0,1]\) is component and connection reliability metric domain, \(R\) is transition process reliability metric domain, \(R, R \subseteq [0,1]\) in general.

This paper treats components and connections as black box, so the reliability of components and connections are given by reliability test and the history data.

Assuming the start component is additional component, which doesn’t have any actions, the reliability of it is 100%. Assessing the reliability of the SAPN steps are as follows:

Step1: Set up the weighted SAPN model according to the SA model [17];

Step2: Set up the test path (PW) according to the transition propriety of weighted SAPN model, if cycle test path consists in the weighted SPAN, thus the path do not be calculated repeatedly. Then work out the transition propriety of PW. In the weighted SAPN, the test path is executed, the breadth-first search (BFS) algorithm can be used to work out test paths from the start to the final,
transition properties of the PW can be calculated as follow:

\[ P_{PW} = \prod_{i=1}^{n} P_i \]  

Step3: calculate the reliability of the test path P.  
Assuming the PW is \( p_1, p_2, p_3, \ldots, p_i, \ldots, p_n \), thus the reliability of this test path can be calculated as follow:

\[ R_{PW} = \prod_{i=1}^{n} R_{pi} \]

\[ R_{pi} = \prod_{i=1}^{n} (1 - \alpha_i \ln C_i) \prod_{i=1}^{n} (1 - \alpha_i \ln L_i) \]

where, \( R_{pi} \) is reliability of \( C_i \), which is component of SA. \( R_{li} \) is reliability of \( L_i \), which is component of SA. \( R_{pi} \) is reliability of transition process.

Step4: The reliability of SA can be calculated as follow:

\[ R_{SA} = \sum_{i=1}^{n} R_{pi} \cdot P_{pi} \]  

\[ P_{pi} \] is transition property which along the path \( P_i \).

IV. EXAMPLE VERIFICATION

This paper takes a flight control system as an example. The State selector and the Operating are interface of flight control system, pilots can set status of the system function and indicate status of the system function. The system consists of the stability augmentation/stabilization (PCS, LCS), automatic flight (ap_manage) and automatic balancing (pap, lap), stability augmentation/stabilization system configuration for more than three degrees (rm). In addition the system owns line control subsystem. The subsystem consists of flight state selector (CDS), build in OS, bit management, and several other main modules.

First AADL model set up on the flight control system is shown in figure 3:

Then evaluate the reliability of the Petri Net which transform from AADL architecture model.

According to structure of AADL model transformation rules, the AADL structure can be mapped into GSPN graphics as shown in figure 4:

Generate test path according to the transition probability weighted SAPN, use BFS algorithm calculate paths from S to EN:

- P1: Operating, key_value, State_selector, POWERON-IOP-os_startup
- P2: Operating, key_value, State_selector, scheduleBit-BIT-pNVM-RM-air_startup-CAS-show_value-Operating
- P3: Operating, key_value, State_selector-NVM-RM-air_startup-CAS-show_value-Operating
- P4: Operating, key_value, State_selector-air_startup-CAS-show_value-Operating
- P5: Operating, key_value, State_selector-scheduleBit-BIT-pNVM-RM-air_startup-lap-pro_l_nav_sub-apmanage-ap_man_lap-show_value-Operating
- P6: Operating, key_value, State_selector-scheduleBit-BIT-show_value-Operating
- P7: Operating, key_value, State_selector-pNVM-RM-show_value-Operating
- P8: Operating, key_value, State_selector-air_startup-CAS-show_value-Operating
- P9: Operating, key_value, State_selector-air_startup-lap-pro_l_nav_sub-apmanage-ap_man-lap-show_value-Operating
- P10: Operating, key_value, State_selector-POWERON-IOP-os_startup-OS-StartLogic-startlog-scheduleBit-BIT-show_value-Operating

Assuming that the reliability of each components C1 - C10 {Operating, State_selector, IOP, OS, startlog, BIT, RM, CAS, lap, apmanage} are {0.99, 0.98, 1, 0.95, 0.98, 1, 1, 0.98, 0.98, 0.97}. The reliability of each connections {key_value, POWERON, os_startup, StartLogic, scheduleBit, PNV, air_start, pro_l_nav_sub, ap_man, show_value} are {0.98, 1, 0.95, 0.98, 1, 1, 1, 0.98, 0.97}. 

Figure 4. AADL model of flight control system

Figure 5. AADL model of flight control system

Figure 6. SAPN of flight control system
1) According to the principle of transfer path weight average path transition probability is obtained as follow:

\[ P1: 0.03125 \]
\[ P2: 0.078125 \]
\[ P3: 0.0625 \]
\[ P4: 0.125 \]
\[ P5: 0.078125 \]
\[ P6: 0.125 \]
\[ P7: 0.125 \]
\[ P8: 0.125 \]
\[ P9: 0.03125 \]
\[ P10: 0.0625 \]

2) According to reliabilities of each component in each path, calculate the reliability of each path.

\[ P1: 0.79 \]
\[ P2: 0.90 \]
\[ P3: 0.922 \]
\[ P4: 0.922 \]
\[ P5: 0.89 \]
\[ P6: 0.90 \]
\[ P7: 0.922 \]
\[ P8: 0.922 \]
\[ P9: 0.86 \]
\[ P10: 0.768 \]

3) Using calculation results above into architecture computing formula, calculate the reliability of software architecture as follow:

\[ R=0.7074085/0.78125=0.9055 \]

V. SUMMARY AND OUTLOOK

Reliability evaluation based on the architecture mainly has two aspects of meaning. On one hand, some software can't implement black box testing at the system level. Then the system structure of the reliability evaluation model is established to calculate the reliability of the whole system. On the other hand, it can establish software architecture model in the early period of software development. A large number of engineering practice shows that many defects of the software appears in the design period. It is meaningful to set up an architecture reliability model in design period to guide the software development. The key component which has greater influence on the reliability, can be found in individual component sensitivity analysis to avoid unnecessary mistakes. The work can improvesystem reliability, thus save a large number of cost in the last stage of verification.

As aviation standard analysis and design language, AADL can model and analyze system function and non-function properties. This paper gives a reliability assessment method based on AADL architecture model. As a high level modeling language, AADL cannot be used directly in the reliability computation and analysis. Therefore, it is required to transfer AADL based model to a lower level formalized model. Petri Net have many excellent properties, it can find a corresponding relationship between AADL, hence transfer the AADL to Petri Net then to analysis the system. This paper put stress on the AADL architecture model transferring to Petri Net rule, made a synthesis of reliability mathematical calculation method on Petri Net. At last take a flight control system as an example to verify the feasibility of this method. The next research direction is:

1) Integrate the rules of automatic transfer function, Petri Net analysis and Evaluation function into the OSATE modeling tool;

2) A new generation of joint type on the new features of the aviation electron system is "high coupling and low cohesion". This is totally different from the traditional distributed software architecture, should do corresponding calibration and how to improve to make the software reliability model is more accurate fit to the reliability evaluation work on new generation of aviation electron system.

3) The scale of the Petri Net model will be grown exponentially by the growth of token and transition, which will lead to the state space explosion problem. How to ease the state space problem, to simplify the model while not reducing its description ability, is the next research direction.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (61300069).

REFERENCES


© 2014 ACADEMY PUBLISHER


Dongyi Ling was born in Changsha, Hunan, China. She is a Ph.D. student in BeiHang University School of Reliability and Systems Engineering, Beijing. Her research interests include software architecture analysis, component-based software modeling, testing, and software reliability evaluation.

Shihai Wang received his Ph.D. in Computer science from the University of Manchester UK at 2010. He joined the school of Reliability and System Engineering, Science and Technology on Reliability and Environmental Engineering Laboratory BeiHang University, as a lecturer from 2011. Currently his research interests include software testing, software fault diagnosis.

Bin Liu is a Ph.D. and a professor. His research interests include system engineering, software testing and software reliability.

Xiaoqi Xing is a Ph.D. student in BeiHang University School of Reliability and Systems Engineering, Beijing. His research interests include software reliability evaluation, software testability and software fault inject.