A Modeling Method for Integrated Modular Avionics Dynamic Reconfiguration Process Based on AADL

Lei Liu¹, Wei Zhao¹, Zeyong Jiang² and Shihai Wang²,³

¹Beijing Electro Mechanical Engineering Institute Beijing, 100074, P. R. China
liuleilarry@126.com; qiuhuabb@163.com
²School of Reliability and Systems Engineering, Beihang University, Beijing 100191, P. R. China
jiangzeyong0106@163.com; wangshihai@buaa.edu.cn
³Science & Technology on Reliability & Environment Engineering Laboratory, Beihang University, Beijing 100191, P. R. China

Abstract. With the great development of integrated modular avionics (IMA), the dynamic reconfiguration of IMA not only provides great advantages in resource utilization and aircraft configuration but also acts as a valid means for resource failure management. It is very vital to ensure the correction of the IMA dynamic reconfiguration process. The analysis of the dynamic reconfiguration process is a significant task. The Architecture Analysis & Design Language (AADL) is widely used in complicated real-time embedded systems. The language can describe the system configuration and the execution behaviors, such as configuration changes. Petri net is a widely used tool to conduct simulation analysis in many aspects. Therefore, we proposed a modeling method of IMA dynamic reconfiguration based on AADL mode, behavioral annex and error model annex in the system level. The model is our first step to analyze and solve the safety problem of IMA system. It can benefit deep analyzing about dynamic reconfiguration of IMA system as the foundation.

1. Introduction
Integrated modular avionics (IMA) is a new architecture for avionics. The system has more open and complex architectures compared to other federal avionics systems. The IMA system executes functions based on common functional modules (CFMs). The CFMs help to reduce the weight and size of a plane. In an IMA system, various software functions run on CFMs. The software system is highly integrated because of its complex structure.

For safety purposes, the planes usually restart the applications by redundant backups once some failure occurs. In this paper, dynamic reconfiguration refers to configuration changes conducted when failure occurs during flying. Dynamic reconfiguration can help in creating new backup areas to restart an application [1], which makes the plane more flexible and utilizes the hardware resources more effectively.

There are many studies on dynamic reconfiguration from a static perspective [2]. However, there are few studies on the dynamic reconfiguration process [3]. The correctness of the dynamic reconfiguration process should be considered. Understanding the constraints pertaining to dynamic reconfiguration can help in completing the process correctly and smoothly. However, how to model for analyzing dynamic reconfiguration is a challenge. To analyze the dynamic reconfiguration process, we proposed a model-based method in this study. Modeling of IMA dynamic reconfiguration can benefit the analysis. The IMA system is a real-time embedded system. The Architecture Analysis &
Design Language (AADL) is quite effective in modeling embedded systems and is widely applied in the aerospace field [4]. Zhang F [5] applied AADL to model F-16 ‘Auto Pilot Controller’ and analyzed the behavior properties of liveness and trace refinement with various fairness assumptions, considering time capacities and deadlines. Zhao Z [6] built an AADL model for the complex hardware structure and robust software of avionic display system.

AADL not only describes the system components but also describes the system behaviors and other elements, such as mode and all types of annex. Modes can represent different configuration states of a system or component when an event triggers a mode change. All of these features make AADL a good method to describe the transition process of systems [7], especially, the dynamic reconfiguration of IMA.

Above all, during the state transition of IMA, safety problem of dynamic reconfiguration is very important and the research has done is so much. But the process of the dynamic reconfiguration is quite complex and hard to analyze. We consider that AADL modeling solves a lot of problems in various real-time embedded system. So we proposed a modeling method of IMA dynamic reconfiguration based on AADL mode, behavioral annex and error model annex in order to provide a basis for deep analyze.

The rest of this paper is organized as follows. Section 2 provides a brief introduction about IMA dynamic reconfiguration and details about model based on AADL. The case analysis is provided in section 3. The section 4 is the conclusions.

2. Modeling of IMA Dynamic Reconfiguration Process

The complex process of dynamic reconfiguration is difficult to analyze without modeling. AADL is an effective modeling tool for a real-time embedded system. Dynamic reconfiguration is a process involved in a human operator and automation. The analysis range is quite wide. However, in this study, the events and conditions that change the process are simplified as some triggers. The object is just the process itself. Thus, a detailed decomposition and formalized expression of the simplified process is discussed here.

1) Dynamic reconfiguration process

When one or several failures occur on a module of IMA, a health manager detects the failure and informs a fault manager to handle it. The fault manager can handle a series of failures under all types of mechanism. Then, the fault manager determines the type of failure to take actions to solve it, for example, closing dynamic reconfiguration, or reporting to the upper layer manager. If a failure causes dynamic reconfiguration, then the system stops the failure application and backs up the data. The connections are destroyed. Then, the target module of reconfiguration is selected based on the functional and non-functional requirements such as minimizing the cost of communication. The next step is to create a new partition on another module for the application. Subsequently, application reloading and connection rebuilding are conducted.

A typical process of dynamic reconfiguration is presented in figure 1. The arrows indicate that messages are being sent during the process. Rectangles represent the important actions that occurred. Compared with the reconfiguration process mentioned in another study that always has redundant modules, dynamic reconfiguration discussed in this study refers to a system without spare modules, especially when reconfiguration in the case redundancy is not designed or is used in the system when dynamic reconfiguration begins.

2) Modeling of the dynamic reconfiguration process

A mode of a system can be associated with the logical configurations. Mode transitions imply that the configuration state changes from one to another. A system or a component has different static structures and properties in different modes. A property can describe task scheduling, real-time characteristics, communication, memory, etc. Then, modes at the system level represent the content of a system configuration. A system has its own modules, partitions, processors, and communication bus in each mode. Thus, the static structure of the system in one mode is built by ARINC 653 annex in AADL.

Figure 2 reveals a series of substates between two modes. The substates and transitions between the modes can be described in the behavior annex. The initial state in the annex corresponds with the prior
mode, whereas the last complete state is the latter mode. Other substates can describe a definite state of the system when a transition is finished during the dynamic reconfiguration. The transition and action in the annex can describe the transition of the modes. The behavioral annex can describe the ceasing and restarting of applications, establishing and destroying processes and their threads, creation and deletion of communication interfaces, building and breaking of transfer connections and virtual channels, and sending and receiving messages with other GSM components.

![Figure 1. An IMA dynamic reconfiguration process.](image1)

![Figure 2. IMA Dynamic Reconfiguration Modelling Approach.](image2)

The error model annex represents triggering conditions in the reconfiguration caused by failures. An error model type may declare error states, error events, and error propagations. Error model implementations declare error state transitions. Transitions are declared to present the errors that are propagated out of a component based on the current error state of that component. An error property of
a guard event may specify that certain patterns of error states and propagations are detected and cause an AADL core event, for example, triggering a mode transition.

The modeling method proposed in this study is presented in figure 2. Mode change represents that dynamic reconfiguration occurred. More details and substates between the modes are described using the behavior annex. The trigger condition of the mode transition is declared in the error model annex.

Properties are added to the model, especially to the behavior annex for the following analysis. As a basis of the multiconstraint analysis, elements such as time properties, memory size, and data states are essential elements of the system.

3. Case Study

Here, in the case of the IMA system, a series of functional modules including navigation, display, communication, and integrated radio frequency sensors (IRFS) are integrated. The navigation module provides the place of the plane and guides the plane in a definition router. The module for an aircraft cockpit display provides the man–machine interface for a pilot. The communication module is responsible for the communication between an aircraft and a ground unit. IRFS integrates all the RF sensors in the aircraft for sending and receiving signals at all frequency ranges.

For simplification, an IMA system with four modules is modeled based on AADL in this section. We denote each module with the first letter of its name—navigation module (N), display module (D), communication (C), and IRFS (I). There are several partitions on each module according to their functions. An application runs on a partition. Process refers to the application here. Moreover, it is assumed that there is one partition in module N and module D. Three partition are set up in the module I. The other two partitions are in module C. The application on each partition communicates with GSM to define the operation of connections and applications.

![Figure 3. Model of system structure in one mode based on AADL.](image)

First, the configuration state of a system can be described by AADL. The logic configuration structure needs the ARINC 653 annex in AADL. The ARINC 653 entities fabricated using the system architecture correspond to the AADL components. The model of the IMA system is presented in a graphical manner based on AADL, as shown in figure 3.

When the module N fails, the GSM detects the failure and starts the failure management. In this case, the failure causes process 1 to break down and the system reconfiguration. Thus, dynamic reconfiguration is triggered. The process is presented in figure 4.

1) After data backup for the process 1, the process 1 is shut down and the connections of process 1 in the module N are destroyed.

2) The system selects a proper module to establish a new partition to run process 1. The strategy for selecting the target module is not introduced here. The target module is module D in this case.

3) A new partition is created in the target module D. Moreover, new channels and connections are set up. Process 1 is reloaded and restarted on the new partition in module D.
Figure 4. Case of the dynamic reconfiguration.

Behavior annex applied between two modes presents the mode transitions with a series of actions, triggers, and conditions, such as the data backup for process 1 and the creation of a new partition on module D. The properties defined are appended to the behavior annex. A modification is made to define the substates between mode 1 and mode 2. The declaration of the substate set is ‘composite state between mode 1 and mode 2 compstate.’ Then each state in the compstate is defined as ‘mode 1: initial state, Backup: complete state, Stop_Process: complete state,’ and so on. A transition between state Backup and Stop_Process is presented in the statement ‘Backup-[data_backup]-> Stop_Process; {RealtimeProperty::ProcessTime => 10.0; memory size ≥ 12MB}. Time and memory properties in this transition are appended to this transition.

The error model annex is used to represent triggering conditions caused by failures. In this case, one failure event occurs to cause the system to dynamic reconfiguration. The annex describes that the state of the system changes from the initial one without error to an error state. The statement is ‘error_free -[error_occurred]->error_state.’ Then the transitions in the error model trigger the system to start reconfiguration. A statement can be ‘Error1_trigger ≥ self[detected_state] applies to mode_transition_event.’

4. Conclusions
IMA dynamic reconfiguration brings great flexibility and reduces redundancies of the system. Meanwhile it increases the complexity for us to analyze. Safety problems deserve studying in all aspects. It is effective to model dynamic reconfiguration process in the embedded system, especially IMA, based on AADL. Therefore, in this paper, we propose a method to model with AADL and its supplement to abstract the complex dynamic reconfiguration process. The AADL model is our first step to analyze and solve the safety problem of IMA system. It can benefit deep analyzing about dynamic reconfiguration of IMA system as the foundation.

5. Acknowledgments
This work was supported by the Defense Industrial Technology Development Program (JCKY2016204A102) and Foundation of No.61400020404.
6. References

[1] Shukla J, Das B and Pant V 2018 J. Int. J. Electr. 99 121-133
[2] Ellis S 1997 J. Microprocess. Microsyst. 21 29-39
[3] Montano G and McDermid J Conf. on 27th Digital Avionics Systems Conference (St. Paul: USA) pp 26-30
[4] Yang Z, Hu K, Ma D, Bodeveix J, Pi L and Talpin J 2013 J. Syst. Softw. 93 42-68
[5] Zhang F, Zhao Y, Ma D and Niu W 2007 J. IEEE Access. 5 27421–27438
[6] Zhao Z, Zhang J, Sun Y and Liu Z 2018 Conf. on 30th Chinese Control and Decision Conference (Shenyang: China) pp 4121–4126
[7] Suo D, An J and Zhu J Conf. on 18th Asia Pacific Software Engineering Conference (Ho Chi Minh city: Vietnam) pp 266-273