http://ansinet.com/itj

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (22): 6627-6630, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6627.6630


# Study of Ima Software Dynamic Reconfiguration Based on Aadl

Yumei Wu, Weipeng Wang, Zhengwei Yu and Bin Liu
School of Reliability and Systems Engineering, Beihang University, Beijing, 100191, China

**Abstract:** A method to model IMA software dynamic reconfiguration with AADL and verify the model's deadlock-free with Petri Net are proposed. Compared with current research, the model can describe the reconfiguration caused by both fault tolerance and requirement change and the verification has good performance on checking deadlock-free of AADL model. This paper gives some basic concepts of IMA dynamic reconfiguration and AADL behavior annex, presents the method of modeling IMA reconfiguration with AADL, the model transformation and the verification of its deadlock-free using Petri Net.

**Key words:** IMA software, dynamic reconfiguration, AADL, verification

Integrated Modular Avionics (IMA) as the new generation of avionics architecture is becoming trends of avionics systems. Possessing the characteristics of highly resource sharing, rapid transmission of information, integration of hardware and software and etc., the IMA software system attracts numerous researchers' attention. Dynamic reconfiguration as one of the important characteristics of IMA software system is mentioned in 'ARNIC 653' (AEEC, 2010). However, there are few related studies of IMA software dynamic reconfiguration at present. The dynamic reconfiguration of an IMA could be initiated under two situations: 'mode change request by the operator' and 'the fault occurred in the software system. (Jibb and Walker, 2000) Some studies focus on the reconfiguration of mode change requested by the operator, use AADL to model IMA software system and its reconfiguration and translate AADL model to Time Petri Net (TPN) model to verify the real-time constraints of the system during dynamic reconfiguration (Suo *et al*., 2011a). In addition, some researchers focus their views on the safety of IMA software system. They contribute to proposing a framework which could be used to perform safety analysis of reconfiguration avionics and help engineers to make safety-related decisions in the whole project management process (Suo *et al*., 2011b). In this paper we propose an approach on modeling and verification of dynamic reconfiguration of IMA for both mode change and fault tolerance.

The article is organized as follows. In first section some basic concepts about IMA software dynamic reconfiguration and AADL behavior annex are given. The following section describes how to use AADL to mode IMA dynamic reconfiguration. And then, the next section is devoted to model transformation and verification of AADL model. At last a conclusion is presented.

## BASIC CONCEPT

**IMA software system dynamic reconfiguration:** The ability to reconfigure software at runtime is one critical aspect of achieving continuously availability. The complexity of the software execution environment is increasing tremendously. Software should be able to execute even with more complexity particularly in ubiquitous computing environments. It is a challenge for the software to detect and resolve the problem or meet changing requirement and reconfiguration is such an approach to meet this challenge (Deshan *et al*., 2013).

Compared with traditional avionics, the IMA refers to the architecture that separates hardware and software. Besides, dynamic allocation of software applications to hardware becomes possible. The IMA software could use different configuration according to required functionalities and external environment. According to the cause of reconfiguration, the reconfiguration of IMA could be classified into two categories. IMA dynamic reconfiguration can be triggered by both requirement change and fault occurrence (Karimpour *et al*., 2013). And reconfiguration of IMA software system could be divided into three types: minor reconfiguration, major reconfiguration and degradation (Seeling, 1996). All of them involve the reloading or restarting of application on processor. What's more, there is another definition of IMA software reconfiguration it is 'the transient activity between two ultimate states of the software system.'

**AADL behavior annex:** AADL (Architecture Analysis and Design Language) is a standard developed by SAE based on the MetaH language. Its main purpose is to describe the dynamic architecture of an embedded system its real-time constraints and the mapping of software to

**Corresponding Author:** Yumei Wu, School of Reliability and Systems Engineering, Beihang University, Beijing, 100191, China

```
Annex bahavior _specification
{**
    <state variables.?
    < initualization>?
    <state?
    <transitions>?
    <connections>?
```

Fig. 1: Behavior annex specification

hardware components. An AADL specification defines various kinds of software and hardware component types (systems, processes, threads, subprograms, data, processors, buses, memory), their real time properties (period, WCET) and how they interact.

The AADL behavior annex is an extension of AADL to offer a way to specify the local functional behavior of the components. It supports precisely describing the behaviors, such as port communication, computation, timing and etc. In particular, behavior annex can accurately and clearly describe mode transition. Behavior annex can be attached to a thread or a subprogram: Threads or subprograms start from an initial state with a transition to a complete/return state. Transitions may be guarded by conditions and actions may be attached (SAE, 2011).

The AADL behavior annex mainly includes five parts, namely state variables, initialization, states, transitions and connections. The structure of behavior annex is described in Fig. 1.

As the core of behavior annex, transitions mean change from initial state to target state with specific condition and execution of a series of operations in the target state. A transition contains initial state, transition guard, target state and action. Initial state presents the state before transition, target state is the state after transition, transition guard is the sufficient condition of transition and action defines the achieving function after completing transition.

## MODELING IMA DYNAMIC RECONFIGURATION

Modeling of IMA dynamic reconfiguration mainly uses structure model, behavior annex and ARINC 653 annex of AADL. The structure model can describe software and hardware composition, hierarchy and connection of the system. But it cannot model partition of IMA system, so the partition, process and thread of IMA system are modeled by ARINC 653 annex. Both structure model and ARINC653 annex can only model IMA system statically. In order to represent dynamic behavior of the system, behavior annex is indispensable. The mapping relationship between IMA dynamic reconfiguration and AADL is shown in Fig. 2.

As the core part of AADL modeling of IMA dynamic reconfiguration, the formal representation of dynamic behavior is described in details as follows.

As stated previously, the dynamic reconfiguration of IMA is in fact the transition from an initial configuration to a target configuration. And the definition of mode in AADL perfectly describe configuration of IMA system. Mode of system is a state of IMA software system. At one moment, IMA software system is composed by a series of partitions, processes and threads with specific connections, the mode of system of this moment means a set of state of all the partitions, process, threads and their connections. In addition, mode of process or thread means a state of process or thread. For the IMA software system, mode transition changes the structure of IMA software system by transforming model composition, connection and properties. In addition, the transition between two modes is based on some rules.

In the process of mode transition, the threads in the system and connections between them will be rearranged. Some threads will change from activation to deactivation and some threads will change from the opposite. The thread and thread connection in mode transition is shown in Fig. 3.

In this figure, control_algorithms thread includes two modes which are monitoring and battle. The transition between them is triggered by event port of controller. In mode battle, battle_algorithms thread and its connections are active, monitoring_algorithms thread and its connection are not active. However, in mode monitoring, the situation is totally opposite. Combined with AADL behavior annex, mode is equal to state in behavior annex and mode transition is equal to transition in it.

In addition to indicating thread and its connection, mode can also assign properties of thread. Some detail information is shown in Fig. 4.

The activation of mode transition is equal to transition guard in transition of behavior annex. The properties of mode are some of the action in behavior annex.

In the process of mode transition, we also consider thread dispatch, component communication, data exchange, bus access with behavior annex, ARINC 653 annex and structure model. Detailed information is not mentioned due to the limited space.

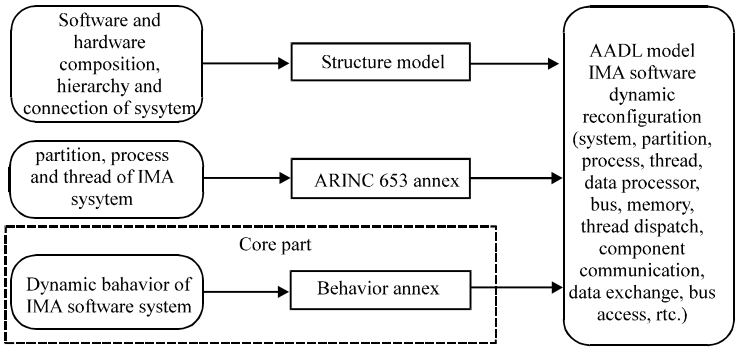


Fig. 2: Mapping relationship between IMA dynamic reconfiguration and AADL

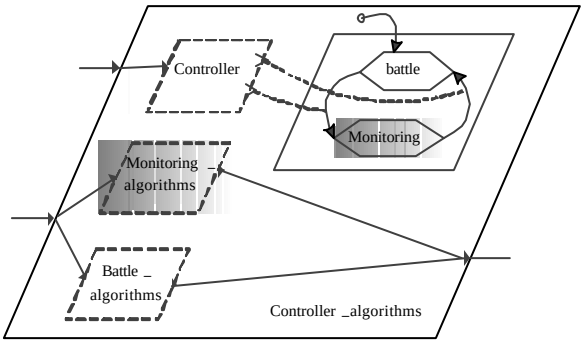


Fig. 3: Thread mode transition of AADL

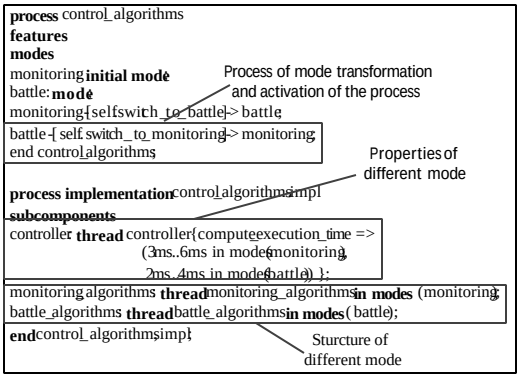


Fig. 4: Code of mode transition

## MODEL TRANSFORMATION

AADL cannot verify its reachability, deadlock-free and real-time by itself. In order to confirm these characteristics of the AADL model for IMA dynamic reconfiguration, we need to transform AADL mode to mathematical model. Verification of deadlock-free is considered in this study with Petri Net.

Petri Net is a mathematical object which can be indicated by figures. The related analysis method/technology of Petri Net can be used for dynamic behavior analysis. A normal Petri Net model is defined as a set of six elements.

Where, S is the set of place; T is the set of transition; F maps the flow relation conducting by S and T; K, W and are capacity function, weight function and initial identification, respectively.

For the transformation from AADL to Petri Net model, mode is equal to element S with the symbol of O and mode transition is equal to element T with the symbol of □. The value of K and W are determined by specific mode transition.

A mapping relationship between AADL and Petri Net is shown in Fig. 5.

The deadlock of Petri Net is if and only if the Petri Net achieves an identification M, no transition can happen in the condition of M.

Based on the definition, we can conclude that if a Petri Net is bounded, the correlation matrix of the Petri Net is C, there is a vector *y* which makes yC>0 or yC<0, in that way the Petri Net will has deadlock at some moment.

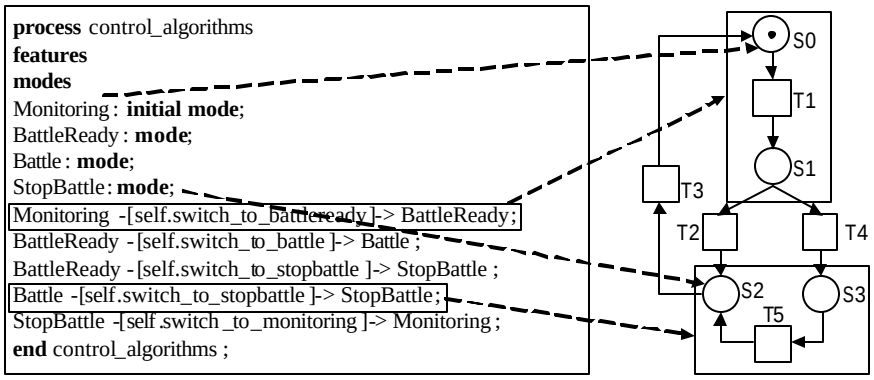


Fig. 5: Mapping relationship between AADL and Petri Net

As previously described, the steps to verify deadlock-free of AADL model are as follows:

**Step 1:** Solving the correlation matrix of the Petri Net
**Step 2:** Judging boundedness of Petri Net
**Step 3:** Judging whether there is a vector y which makes yC>0 or yC<0, where C is the correlation matrix of the Petri Net
**Step 4:** If there exists a vector y, the AADL reconfiguration model has deadlock in it

## CONCLUSION

It can conclude from the above statement that the method can construct AADL model on IMA software dynamic reconfiguration in both two situations of fault tolerance and requirement change. To verify deadlock-free of the model, we translate the AADL model to Petri Net model. The result of verification can be used in the early stage of IMA software system development, help to reduce time and cost of development and improve reliability of IMA software.

## ACKNOWLEDGMENTS

The authors would like to thank for the support by Ministry Advanced Research Foundation under the Grant 9140A19021813HKXXXX.

## REFERENCES

AEEC, 2010. Avionics application software standard interface, part 1, required services. Applied Environmental Education and Communicatio.

Deshan, C., K. Ehsan, M. Sam and R. Roshanak, 2013. Proactive self-adaptation for improving the reliability of mission-critical, embedded and mobile software. IEEE Trans. Software Eng., 39: 1714-1735.

Jibb, D.J. and J.B. Walker, 2000. Avionics architecture standards as an approach to obsolescence management. Proceedings of the Symposium on Strategies to Mitigate Obsolescence in Defense Systems Using Commercial Components, October 23-25, 2000, Budapest, Hungary, pp: 16-1-16-12.

Karimpour, J., R. Alyari and A.A. Noroozi, 2013. Formal framework for specifying dynamic reconfiguration of adaptive systems. Inst. Eng. Technol. Software, 7: 258-270.

SAE, 2011. Architecture Analysis and Design Language (AADL) annex volume 2. SAE AS 5506/2, Supervised Agricultural Experience. http://standards. sae.org/ as5506/2/

Seeling, K., 1996. Reconfiguration in an integrated avionics design. Proceedings of the 15th AIAA/IEEE Digital Avionics Systems Conference, October 27-31, 1996, Atlanta, GA., pp: 471-478.

Suo, D.J., J.X. An and J.H. Zhu, 2011. A new approach to improve safety of reconfiguration in integrated modular avionics. Proceedings of the 30th IEEE/AIAA Digital Avionics Systems Conference, October 16-20, 2011, Venue Seattle, WA., pp: 1C4-1-1C4-12.

Suo, D.J., J.X. An and J.H. Zhu, 2011. AADL-based modeling and TPN-based verification of reconfiguration in integrated modular avionics. Proceedings of the 18th Asia Pacific Software Engineering Conference, December 05-08, 2011, Ho Chi Minh, Vietnam, pp: 266-273.