

# Integrated Electronic System Configuration Analysis

Lei Liu<sup>1</sup>, Weipeng Wang<sup>1</sup> and Guangding Feng<sup>2, 3\*</sup>

<sup>1</sup>Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China

<sup>2</sup>School of Reliability and Systems Engineering, Beihang University, Beijing, China

<sup>3</sup>Science and Technology on Reliability and Environmental Engineering Laboratory, Beijing, China

\*Corresponding author

**Abstract**—In this paper, we want to make a clear and concise description of the configuration of an increasingly integrated and complex avionics system. We proposed the configuration analysis method on AADL (Architecture Analysis and Design Language) to modelling the highly integrated electronic system. It can give the corresponding textual and graphical descriptions of that according to the different modules and levels. Also, we sorted out the corresponding conversion rules between AADL and real IMA (Integrated Modular Avionics). Then we can make the analysis of IMA clearly. Finally, we gave the AADL modelling example analysis. It was used to assist the analysis of the specific configuration, thus improving the reliability of the whole system later. This whole process can greatly help us to quickly master the configuration of the IMA, to making an analysis of the key system characteristics.

**Keywords**—AADL; configuration; IMA; conversion rules; modelling analysis

## I. INTRODUCTION

The avionics system involves functions such as communication, navigation, identification, detection, flight management, display control, etc. It provides pilots with comprehensive and accurate battlefield situation information to enhance pilots' potential threats in the battlefield environment. The ability to identify, monitor, analyze and react is the central neural system of the entire fighter. The development of avionics has undergone several periods: discrete, joint, integrated, and advanced integration. It aims to enhance the functions and performance of avionics systems, reduce the life cycle cost of aircrafts, and solve the problems of upgrading hardware and software of systems. The software module and the hardware module are not tightly coupled, and the storage of the software set is borne by a large-capacity storage device. According to the system configuration definition, it is downloaded to each corresponding system initialization in conjunction with the storage management task, thus, it can deploy and achieve task management. The avionics system is to realize the communication navigation and display management of the aircraft. Integration is the development direction of avionics systems. [1] It includes system integration, hierarchical structure, functional software and network standardization.

And the hidden dangers of safety brought about by such highly integrated or complex features have gradually attracted people's attention. The trend of the IMA system is to adopt more and more comprehensive system architecture. The comprehensive range is becoming wider and wider, the integrated functions are more and more, and the modules are

highly coupled. As a result, the complexity of the system increases dramatically, and the range that system faults spread because of resource defects are increased. The integrated electronic system consists of core systems and non-core equipment. The core system is usually a series of avionics racks. It is a set of standardized interconnected Common Function Modules (CFM). [2] Therefore, the configuration of the integrated electronic system is actually the specific link between the modules of that. The analysis of the configuration status of a specific integrated electronic system should use the AADL modeling analysis method. The model is very important and indispensable throughout the life of the product, covering development, deployment, work, and maintenance. As a result, when it comes to how to describe the configuration of integrated electronic system which is critical and real-time control system, we can use the MBE to make an analysis. [3] AADL is an architecture modeling language that conforms to MDA thinking. It was jointly proposed by the SAE Architecture Description Language Subsidiary Committee, the Embedded Computing Systems Committee, and Avionics Systems Corporation to design and analyze the software and hardware architecture of security-critical embedded real-time systems. The AADL is an important way for us to analyze the IMA configuration process.

## II. IMA CONFIGURATION MODELING

AADL regards the component as the core, regards system software and hardware as the collection of components, and regards the entire system as a set of interactive components. [4] The center of the AADL model is the component type description and component implementation description and they make up the software package. The components are divided into hardware and software components and composite components. In the description of the component type, the category and interface (feature) of the component should be defined. This corresponds to a single specification of the component. In the implementation description of this type of component, the internal structure (composition and interaction) of the component should be defined. This is equivalent to a blueprint based on the construction of each component. The AADL consists of software packages and attribute sets. All AADL descriptions must be included in a package or attribute set description. A batch of AADL packages and optional attributes represent the system specification in the AADL, called the AADL declarative model, and a system instance model is an example of a system model that uses the system component implementation with an illustrative system model as its source. [5] The component categories in AADL are

classified as software components, execution platform components, composite components, or general (non-runtime specific) components. Application software components include data, threads, thread groups, processes, subroutines, and subprogram groups. Execution platform components include processors, virtual processors, devices, buses, virtual buses, and memory. The relationship between the input of the system architecture design model and its corresponding description: when inputting the components of the system, it includes the subsystems, modules, the processes and threads of each function; the connection relationships between components include data, events, access and subroutine call relations; system hardware components are mainly the resources for calculation and connection; the binding relationship is the mapping of internal software components to hardware components in the system.[6] In AADL, the functional area in the modeling is represented subsystem, the partition is represented by process, and the application is represented by thread. The operating system can be seen as special software expressed as thread, switches and peripherals can be represented by device. [7] The bus within the partition is represented by bus, and the large capacity memory is represented by memory. The data interaction, or event trigger between applications, is represented by data port, event port, or Data Event port. The data storage communication request is represented by Data Access. The system's dynamic call process can be described in the AADL mode transformation. Model analysis is one of the most important technologies of AADL. In modeling the configuration of the avionics system, we initially use the AADL system, process, and equipment components to build a high-level system. [8] When we build this, we will describe in detail the runtime components of all the elements. The software is allocated to hardware resources; and the attributes of each element are assigned a value that corresponds to the application level of a specific instance, that is, to meet the requirements specification.

When modeling a device, we need to first define a simplified model. In this model, we want to include the unit for input, output, transmission and response. For each of these identified components, we will develop type definitions, specifically defining the component's name, runtime class and interface. Because we originally developed an advanced (conceptual) model, we limited the component categories to systems, processes, and devices.

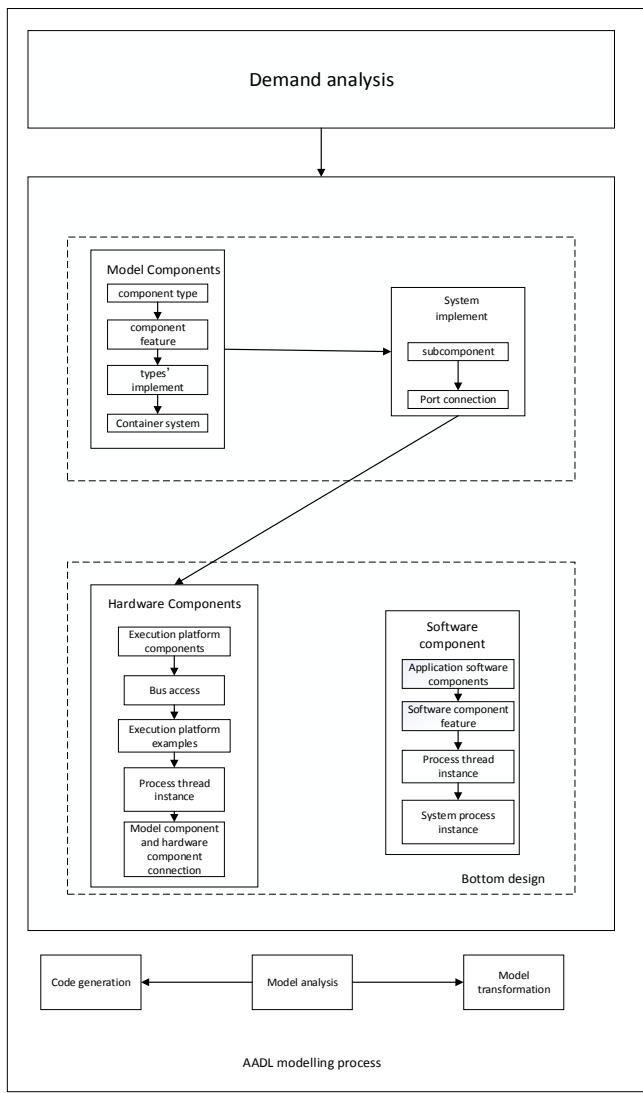
When we model a device, we first break it down into several modules that contain many different types of components. Then we need to describe the types of components, that is, a series of alternative word descriptions. The component type specification defines the interface elements of the component and its external observable attributes, where the interface elements represent the features (interaction points with other components), logic flow specifications, and work patterns. Component properties represent observable properties. In each AADL component category, for example, a thread is a schedulable unit that executes in parallel, and a thread group is used to logically organize the abstraction of a process-wide thread, thread group, and data component. A process is a protected address space that is enforced at runtime. The processor is a scheduling and execution thread and a virtual

processor. The memory is used to store code and data, and the device is a sensor, actuator or other component. The external interface of the declarative component is the explicit entry and exit point for information and component access. This describes them within the feature segment of the component type description, including ports, access (access to data, buses, and subroutines) and parameter. Features can be composited into feature groups and then can include multiple feature groups below it. [9]

In addition, the component implementation specification defines the internal structure of the component. The alternate words used for the various parts of the component implementation description are mainly composed of extend, prototypes, subcomponents, calls, connections, flows, modes, require modes, properties, annex, and end. [10] The actual component implementation description includes at least: a set of attributes that will distinguish this implementation from other implementations of the same component type. If the component implementation represents a composite component (a component is composed by many other components), the component implementation includes descriptions for subcomponents and connections.

In our work to specify system components with AADL, we developed descriptive models that describe the various software, hardware, and system components, as well as their interactions and hierarchical organization. In the illustrative model, a pattern of the types of components and implementations of the software and hardware component instances is constructed.

AADL model analysis includes analysis of AADL model and analysis of AADL instance model. The AADL instance model is an example of a system in the AADL model and has SAE attribute values such as AADL attribute values and bandwidth capabilities, processing capabilities, and service quality agreements such as processor bindings, memory bindings, and thread processing protocols. [11] The contents of some analysis objects not only support the AADL model but also the AADL instance model. The contents of other analysis objects only support the AADL model or the AADL instance model, and are mainly determined by the attribute values of the analysis object content. The figure1 below gives the direct display of modelling process in AADL.



FIRGURE I. AADL MODELLING PROCESS

As we can know from the figure1, when the AADL models an embedded system, it will model from the requirements analysis (functional or non-functional attributes) at different levels and stages as a whole. When doing top-level design, it will study modules, model components, and systems. When it comes to the underlying design, it will study software and hardware, that is, hardware components and software components. The whole process is to map the model of the top-level design to the model of execution platform components and application software components. AADL as a model-driven new standard supports model-driven architecture. AADL uses the method that a single model support multiple analysis, it integrates system design, analysis, verification, automatic code generation and other links under a unified framework, providing sufficient structural information for system analysis and implementation, and greatly deepening our understand to the system configuration. AADL describes the component features, sub-components, attributes and so on through system modeling to complete the configuration analysis.

### III. AADL MODELING EXAMPLE ANALYSIS

According to a modeling process described above, we model the implementation of a data processing module on an IMA.

Figure 2 shows an AADL text representation. Figure 3 shows the AADL graphical representation of this part of the model.

```

system implementation DP_BLOCK.impl
subcomponents
    ARM_1: processor ARM_1.impl;
    ARM_2: processor ARM_2.impl;
    DDR3_1: memory DDR3_1.impl;
    DDR3_2: memory DDR3_2.impl;
    MSB: process MSB.impl;

connections
    c_ddr3_1: port ARM_1.A_DDR3_1 -> DDR3_1.DM_1;
    c_ddr3_2: port ARM_2.A_DDR3_2 -> DDR3_2.DM_2;
    c_arm_1: port ARM_2.A_2 -> ARM_1.A_1;
    c_arm_2: port ARM_1.A_1 -> ARM_2.A_2;
    c_msb_1: port MSB.M_ARM_1 -> ARM_1.A_MS_B_1;
    c_msb_2: port MSB.M_ARM_2 -> ARM_2.A_MS_B_2;
    c_srio4_1: port SRI04_1 -> ARM_1.A_SRI04_1;
    c_sgmii_1: port SGMII_1 -> ARM_1.A_SGMII_1;
    c_srio4_2: port SRI04_2 -> ARM_2.A_SRI04_2;
    c_sgmii_2: port SGMII_2 -> ARM_2.A_SGMII_2;
    c_pmbus: port PMBUS -> MSB.M_PMBUS;
end DP_BLOCK.impl;

```

FIGURE II. AADL TEXT REPRESENTATION

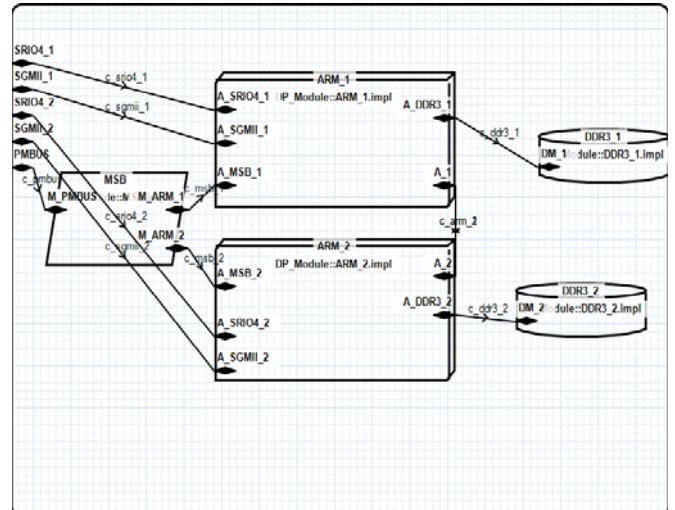


FIGURE III. AADL GRAPHICAL REPRESENTATION

In the above description of the component implementation, it describes the sub-components and connections of the implementation of the data processing module in the IMA. Of course, those ports that do not specify the connection will also be connected to the ports of other modules, but they are not described here. This modeling is a gradual refinement process that will be refined through step-by-step additional text descriptions. In short, AADL is suitable for modeling the IMA configuration state, and AADL is also very suitable for the modeling of dynamic software systems with a large amount of

data interaction. It is very convenient and easy for us to carry out the simulation test and the reliability analysis after the AADL modeling process is completed. [12]

#### IV. CONCLUSION AND FUTURE WORKS

AADL is applicable to high-security real-time embedded systems such as avionics systems, and has the ability to describe these characteristics such as real-time, practicality, security, confidentiality, reliability, and system throughput that are of primary concern when describing such kind of system. AADL uses models to describe the model. Component types include data, subroutines, thread groups, processes, devices, memory, and processors. AADL can describe the functional interface of components (such as data input/output), and can also describe performance aspects (such as time requirements); the relationship between components is determined by the nature of references and subcomponents; component properties describe model attributes; The communication relationship uses port features and connection expressions; the mode and mode conversion functions enable dynamic behavior binding of the runtime system. In the example given above, AADL's textual and graphical representations which are of the two real-time interactive description ways give a good model representation of the part of the system.

But we should notice that IMA modelling on AADL cannot make the reliability analysis, we need to do model conversion, and simplify the process of reliability analysis as possible as we can. In addition, to generate the code of AADL automatically, the code generation tool should be implemented. So these two problems will be studied in our future works.

#### ACKNOWLEDGMENT

This work was supported by a grant from the National Natural Science Foundation of China (No.61300069).

#### REFERENCES

- [1] Kleppe A G, Warmer J, Bast W, et al. The model driven architecture: practice and promise[J]. 2003.
- [2] Romanski G. Management configuration data in an IMA system[C]/Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th. IEEE, 2008: 1. B. 5-1-1. B. 5-10.
- [3] Luxi Chen. Integrating behavior analysis into architectural modelling[J]. Frontiers of Computer Science,2015.
- [4] Moir I, Seabridge A G, Jukes M. Military avionics systems[M]. John Wiley & Sons, 2006.
- [5] Gu Q, Wang G, Xu W. Research on resource fusion for Integrated Modular Avionics system[C]. IEEE, 2012.
- [6] Sei A T. An extensible open source AADL tool environment (OSATE)[J]. Software Engineering Institute. 2006.
- [7] Aerospace S A E, AS5506 S A E. Architecture analysis & design language (AADL)[J]. AS-5506, SAE International, 2004.
- [8] Jolliffe G. Producing a safety case for IMA blueprints[C]/Digital Avionics Systems Conference, 2005. DASC 2005. The 24th. IEEE, 2005, 2: 14 pp. Vol. 2.
- [9] Schmidt D C. Model-driven engineering[J]. COMPUTER-IEEE COMPUTER SOCIETY-, 2006, 39(2): 25.
- [10] Kai Hu, Teng Zhang, Zhibin Yang,Wei-Tek Tsai. Exploring AADL verification tool through model transformation[J]. Journal of Systems Architecture.2015.
- [11] Frank Singhoff, Alain Plantec. AADL modelling and analysis of hierarchical schedulers[J]. ACM SIGAda Ada Letters, 2007.
- [12] Wang G. Integration technology for avionics system[C]. IEEE, 2012.