Integrated modeling method of Cyber physical system based on extended AADL

He Renya, Wu Lijin, Tang Longli, Han Xinyu, Xu Zhaowei and Gu Guoqiang
China Institute of Marine Technology$Economy,Beijing,100081，China

Abstract: The concept and connotation of CPS are analyzed firstly in the paper, then the challenges of system modeling brought by heterogeneous feature of the CPS system are pointed out. How to describe the CPS model are studied. We propose a description method of CPS integrated modeling framework based on extended AADL (architecture analysis and design language); and realize the unified description of the computing entity, the physical entity and interactive entity through extending the standard and appendix of AADL.

Keywords. Cyber-physical system (CPS), Modeling&Simulation, AADL,Modelica

1. Introduction
With the rapid progress and development of computer technology, communication technology and automatic control technology, the interaction between computer system and physical world is more and more in-depth, and the systematic integration between them is more and more complex. The emergence of the Internet of things, big data, cloud computing and other concepts further exacerbated this trend. Modern engineering systems are very complex systems that combine knowledge from many fields such as computing, information technology and physical environment. Traditional embedded technology adopts single-point solution to solve the problems related to physical systems. This mode no longer ADAPTS to the requirements of new physical equipment being informationize and cyberization, because the existing network transmission and communication technology cannot fully meet the new requirements of real-time, controllable and extensible network of new physical equipment. In this context, a system that integrates the physical world and the information world is produced, namely, the information physical fusion system, or CPS. The high complexity of CPS is the result of the development and integration of different technologies in many fields. In order to improve CPS system model building capability, finite state machine, hybrid automata and data flow models were extended on the basis of scene structure and behavior collaborative modeling. A transformation rule between Modelica model and AADL model is proposed to transform the Modelica model of physical system into AADL model.

CPS focuses on the dynamic integration of discrete computing process and continuous physical
process, and pursues the organic combination of computing, communication and control. It is a new form of complex embedded system. Cps-like integrated electronic control systems are emerging in some areas, such as aviation, automotive, chemical process, infrastructure, energy, health, manufacturing, traffic control, entertainment, and consumer electronics. While embedded systems place more emphasis on the computing power of machines, CPS places more emphasis on the connections between physical devices and computer computing networks. The essence of CPS system is the fusion of human, machine and matter. CPS include in computing science, network communication, control theory, sensor and other disciplines, which are closely related to human life and social development, covering small to nanoscale biological robots, large to global complex systems. A large number of dynamic, uncertain factors exist in the CPS, contains a large amount of heterogeneous network nodes, using a variety of types of network technology, information system and physical system produces a large amount of data in the process of interaction. When we design and analyze CPS, how to model the dynamic, uncertain factors, in order to tolerate the the effects on the real-time performance, safety and reliability, is the enormous challenge which we should be faced in.

2. CPS implications and modeling challenges
   2.1. CPS connotation
   CPS is an engineering system that integrates computing, communication and control system depth into the physical system, perceives and controls the physical process based on the computing process, and realizes the seamless integration of information space and physical world. CPS is a new technology that integrates computing, network and physical environment to form a complex system of three-dimensional integration of human, machine and object, as shown in figure 2-1. Computation, Communication and Control technologies are used to realize real-time perception, dynamic Control and information service of large-scale engineering systems. It is considered to be the core technology to drive the next industrial revolution.

![Figure 2-1 3C collaboration concept diagram of CPS](image)

Different from the traditional embedded system, the organic integration between computing components and physical environment is emphatically considered. Independent existing equipments are intelligently connected, to realize the adaptive networking and interaction, so as to realize mutual awareness, effective coordination among the system, automatically adjust the calculation logic and configuration according to the task. Computing equipment can accurately obtain external information
and make targeted and intelligent responses in real time, improve computing performance and quality, provide timely, accurate, safe and reliable service and control, then realize the integration and unification of the physical world and the information world.

The connotation of the physical system is to build a set of information space and physical space based on the data flow between the state of the perception, real-time analysis, scientific decision-making, accurate implementation of closed loop can assign system, solve the complexity and uncertainty problems of production manufacture and application service, improve the efficiency of resource allocation, resource optimization, as shown in figure 2-2.

The connotative features of CPS mainly include:

1. Discrete information processing process and continuous physical control process deep integration;
2. CPS embodies multi-scale, high credibility;
3. CPS is a new idea to solve various problems in the application system by analyzing the internal operating mechanism of the system;
4. CPS is a typical spatio-temporal event-driven distributed system;
5. CPS has dynamic characteristics such as dynamic reorganization, reconfiguration and autonomy;
6. CPS is a high security and low delay system;
7. CPS can unify heterogeneous elements in the standardization system downward and provide consistent interactive information and coordination services upward.

In CPS, the time and space range of system operation is distributed and asynchronous. It is a physical system composed of structure, function of different subsystems dynamic hybrid system, including various types resources and programmable component which of perceptual decision-making and control, through wired or wireless communication technology, based on mutual coordination of network infrastructure, It achieves real-time perception of physics and engineering systems, remote coordination, accuracy and dynamic control and information services.

2.2 Challenges faced by CPS integrated modeling
In order to adapt to the interaction and integration of the computational process and physical process of the CPS system, the system model needs including multiple physical models and computational
models, and the appropriate model integration framework must be determined to realize the organic integration and integration modeling of heterogeneous models. In the process of CPS behavior modeling, it involves the physicalization of computing process and the information of physical process, and the semantic and physical process of serial execution of computing process should also be considered. The communication network model needs to be integrated into the CPS system model to analyze the influence of communication delay on the precise control of computation process and physical entity network. The temporal and spatial properties of CPS system need to be extended in the system model, and its influence on system function and performance should be analyzed.

Currently, a number of modeling languages have appeared in academia and industry aiming at the characteristics of CPS, such as PTIDES[1] language designed by Edward Lee and others, Simulink and Stateflow developed by MathWorks, and OpenModelica, an open source tool developed by Modelica association, have been successfully applied in mechanical, thermal, control and electric power. In addition, according to the hybrid duml and SysML, a hybrid of the original modeling language, such as UML, MARTE also supports the modeling of CPS characteristics [2]. But there are few researches on the modeling of AADL on information physical fusion system.

At present, the evaluation, modeling and simulation of computer hardware and software, computing system and physical system are completed in a separate way, lacking of necessary collaborative design and collaborative analysis. The separation of hardware and software leads to excessive or insufficient constraints; it is difficult to evaluate the impact of design decisions. The separation of computational and physical systems makes it impossible to fully capture the interdependencies between them, but understanding these dependencies is a key prerequisite for accurate representation and modeling of CPS. In addition, there are many models, description languages and tools in each subsystem of CPS to support its design, but when the subsystem is integrated; it faces great challenges to support the whole CPS system-level analysis and validation.

Firstly, CPS has such characteristics as joint dynamics of computation-communication-control, multi-scale integration of computing and physics, uncertainty of system behavior, spatio-temporal interaction of system environment and state and diversity of non-functional requirements, which is one of the problems faced by fusion modeling of CPS.

(1) Joint dynamics. CPS is usually composed of multiple physical processes, perceptron and controllers, concurrent computing entities and real-time communication networks. The changes of computing, communication and control entities will affect the dynamic behavior of the whole system.

(2) Multi-scale integration. The hierarchy of CPS structure and the heterogeneity of its composition make the integration of discrete computing process and continuous physical evolution process show multi-scale characteristics in system scale, logical level, constituent elements and non-functional attributes.

(3) The uncertainty of system behavior. The uncertainty of state and behavior of CPS is mainly caused by the factors of concurrent computing process, random physical process, disturbance of physical environment and network communication delay.

(4) The spatio-temporal characteristics of environment and process. The response of CPS behavior is not only related to the time experienced, but also to the space it is in. It shows that a computational process requires a strict time constraint, while a physical process is spatially dependent.
Diversity of non-functional requirements. Most information physics departments belong to mission-critical or security-critical applications, which not only have distinct time constraints, but also require high state security and operational reliability, and the system presents multidimensional service quality requirements. The control process involves physical process, computational control process, sensor transmission and other time delays, which poses great challenges to the accuracy of CPS modeling. The analysis of reliability and security of CPS system is the key problem in CPS design.

Moreover, CPS is a system that exists in both discrete subsystem and continuous subsystem. How to unify discrete subsystem and continuous subsystem is another problem that CPS system design faces. The research and design of CPS must be carried out under the unified theoretical framework. However, there is no unified theoretical framework that can deal with computer system, communication network system and physical dynamic system, so it is necessary to develop the unified modeling theory of information system and physical system.

AADL can be used to describe the software components which are mapped onto the execution platform system structure, components, functional interface components (such as data input and output) and the performance of the key attributes (such as real time) and the interaction between components, add in view design factors of the implementation platform, solve the problem of the separation of software and hardware, model driven design method based on AADL can play performance test of integration system in modeling phase, solve system level analysis and validation issues in after subsystem integration.

3. Integrated modeling technology based on extended AADL
In order to depict the CPS system functional component and the interaction relationship between components, it needs to model the CPS system architecture in the early design stage. Existing method is using components and the connection to model the communication of calculating component and others. Although some architecture have the ability to describe the physical artifacts, but physical artifacts and computing components cannot be on the same level. It is difficult to analysis the interaction and influence between them. This section USES the AADL modeling language to further characterize the information physics system fusion modeling framework, providing a common modeling language and related description methods to express the interactions and dependencies between computing and physics component.

3.1 AADL modeling elements
AADL is an embedded system design and analysis language which is centered on system architecture and based on model drive. Thanks to its powerful modeling capabilities such as abstraction, reusability, compensability, and practical analytical capabilities, AADL has become increasingly popular in the process of modeling in the security critical areas. AADL USES components to describe the static properties of the system, and the interactions and data flows of components to describe the dynamic properties of the system. AADL can be used to describe both the hardware of the physical environment and the software execution platform. Moreover, it supports the early prediction and analysis of the key attributes of the system, such as performance, schedulability and reliability, and provides a unified framework for the modeling, analysis and verification of the system.
The AADL modeling component describes the components, interfaces, and connections of a system. AADL uses component types and component implementations to describe the components. Component types describe the interface composition of system components, while component implementations describe the internal structure of system components. There are ten component types in AADL, which are divided into three categories:

- Software components: data, subroutines, threads, thread groups, and processes;
- Hardware components: memory, processors, buses, and devices;
- Composite components: systems.

As the name suggests, AADL's system components combine software and hardware components.

As an architecture description language, AADL can formalize the modeling of traditional embedded systems from hardware and software, and provide runtime semantics for the control flow and data flow mechanism. But AADL core language parts cannot support the detailed modeling of physical systems, the information related to time and space behavior model, the model cannot be directly used in reliability analysis and simulation. Based the advantages and shortcomings of CPS on the above AADL, according to the information mentioned in section 2.2, the requirements of CPS, this paper use AADL on modeling CPS software part, software implementation which is closely related to the and execution platform part of the input/output devices, use Modelica which is good at modeling physical system, through transformation, makes the model eventually convert AADL model, in the aspect of system behavior, with the behavior attachment of AADL [3] describe and model the behavior of the components, use the temporal automaton to extend the behavior attachment on order to the spatial and temporal behavior of the system.

3.2 Transformation between Modelica model and AADL model

Modelica is an object-oriented modeling language, its basic elements are composed of classes and objects, and support inheritance, and AADL component type and component implementation one-to-one correspondence to achieving inheritance through extends keyword these commonalities provides the basis for their mutual conversion. Classes and models in Modelica can be converted into the system type declaration; input and output device and computing platform (memory and processor) in AADL according to the specific situation of the modeling system, and the class instantiation object can be converted into the type implementation of. The restricted class connector in Modelica is a connector with three types of ports: real type, Boolean type and integer type, and the turn is bidirectional. The component port and port group described by feature in AADL are used for the interaction of data and events between components, and there are input port/output port and input output port, which are interchangeable. Parameters in AADL are defined in the form of variable declarations that can be used to represent variables in Modelica. Flinction function in Modelica conforms to the requirements of calling AADL subprogram, and the formal parameters in the function can be represented by parameters in AADL. The data type in the AADL can define various data types related to ports and parameters, including composite data structures that correspond to the various data types in Modelica, and modeling elements such as equations, constants, etc. that are commonly used in Modelica in system modeling but not in AADL are extended through the property set of the AADL so that it can be described by the AADL. The transformation rules of Modelica model and AADL model are shown in table 3-1:
### Table 3-1 transformation rules of Modelica model and AADL model

| Modelica model | AADL model               |
|----------------|--------------------------|
| package        | package                  |
| class, model   | system, device, memory, processor |
| connector      | port, portgroup          |
| equation       | attribute extension      |
| variables      | data or parameters from a port or port group |
| Connect ()     | connection               |
| Function       | parameter, subprogram    |
| constant       | attribute extension      |
| initial value  | attribute extension      |
| basic datatype, record, array, Modelica.SIunits | data, datatype |

Next, the attributes of AADL are extended so that it can describe modeling elements such as equations and constants in Modelica.

```plaintext
Property Modelica_property is
Equation : aadlstring applies to (device/system/memory/processor); // equation
Const : aadlstring applies to (device/system/memory/processor); // constant name
Const—value : aadlstring applies to (device/system/memory/processor); // constant value
end Modelica_property;
```

#### 3.3 Transformation from behavior model to time automaton model

AADL behavior model is a model that describes the detailed behavior inside AADL standard components based on the behavior attachment of AADL. In this section, we will use the timed automata to construct physical objects, through the physical object of timed automata model, extending the AADL behavior attachments enable AADL behavior model to modeling the CPS behavior of space and time, and gives the extended AADL behavior model and the UPPAAL (UPPAAL timed automata [4] is a kind of modeling and simulation tool) under timed automata model mapping rules.

Timed Automata (Time Automata, TA) by Alur and Dill [5] and others in 1992 put forward an extended finite state machine, compared with the traditional finite state machine, on the basis of the original state machine in the process of state transition, it adds time variables and constraints, so as to make the state machine can describe the state of continuous time changes. Time Automata can be applied to the description and analysis of real-time system behavior.

AADL behavior model is the further essence of AADL standard language model, which is based on AADL behavior attachment and describes the detailed behavior inside AADL standard components. The behavioral attachment of AADL is proposed based on the transition theory of automatic state machine. The grammatical definition of behavioral attachment [6] is as follows:

```plaintext
Annex behavior_specification {**
< state variables >?
The < initialization >?
The < states >?
The < transitions >?
The < connections >?
```
The `<composite_declaration>`?

In the definitions described above, `<state variables>` means to declare the state-type variables involved in the behavior, and `<initialization>` means to initialize the declared variables. `<initialization>` represents the set of states of behavior changes within a component. Each state can have its own state initialization, complete state, return state, urgent and so on. `<states>` In a behavior artifact, transitions between state variables are declared via `<transitions>`, where the transitions description should follow the following format:

```
< state 1 > -[< guard >? ] - > < state2 > {< action > *}
```

Where `<guard>` refers to the condition that causes the state 1 change to transition to state2, and `<action>` represents the specific action behavior during the state change. `<guard>`

The behavior component also includes the `<connections>` section, which describes the actual computation data or the propagation path of the triggered event in the behavior component during operation. `<composite_declaration>` describes a composition of states in detail that describes very complex behavior implemented. `<composite_declaration>`

Modeling of spatial aspect describes the spatial dynamic characteristics of information physical fusion system, and the design of behavioral attachment is also used to solve the dynamic problems in the modeling process. Through the study of the physical object's time automaton model and AADL behavior model, the corresponding relationship as shown in table 3-2 is obtained:

**Table 3-2** comparison of time automaton model and AADL model of physical objects

| Physical object time automaton model | AADL behavior model |
|-------------------------------------|---------------------|
| State set L                         | states              |
| state variables for the properties of the physical object Attr | state variables |
| Initial state l0                   | initialization      |
| clock set X and clock constraint Inv | delay(min., max), computation(min., max) |
| A finite transitions set T          | transitions         |
| The assignment function Dom of physical objects and the action set A | action |
| the energy enable condition g of Transitions and the physical object set Ev | guard |

When analyzing spatio-temporal problems in information fusion systems, we will use time automata to construct correlation models. In order to achieve the purpose of analyzing the model with time automata, the AADL model needs to be extended correspondingly. Through the above comparative analysis of the time automata model and AADL behavior attachment model, the time...
automata model can be transformed into the behavior attachment model in AADL.

**Figure 3-1** extension of AADL behavior attachment by Time Automaton

The extension process approach as shown in figure 3-1, based on timed automata model file, extract the related elements, and the information stored in a custom class, depending on the rules of the we define the class of information stored in converted to meet the requirements of AADL XML file format, and the function of the interface provided by Dom4j, designing XML function to write the XML file.

4. Application sample

Unmanned Aerial vehicles (UAVs) are Unmanned Aerial vehicles (UAVs) that are capable of autonomous flight control, completing a given flight mission, or handling remotely piloted commands launched by ground control centers and performing operations accordingly. UAVs are mainly composed of the following parts, including aircraft systems (such as high-frequency communications, GPS sensors and others), self-guided control systems (or flight control systems), electrical and power systems, take-off and landing systems and other systems. The flight control system is one of the most important parts. Flight control system consists of hardware and software. Unmanned aerial vehicle (UAV) through collecting the real-time data from the airborne sensors, save the system execution environment and the related data, and constantly monitor airborne radio sensors instructions, for calculating and processing, then feedback to the airborne steering gear control system, so as to realize the three rudder surface, airborne engine control, complete flight control of unmanned aerial vehicle (UAV) [7, 8].

**Figure 4-1** working principle diagram of UAV

As shown in figure 4.1, the sensor system of UAV mainly includes vertical gyro, angular rate sensor, heading, altitude sensor and GPS receiver, while others not shown in the figure include
airspeed sensor and radio receiving sensor. The flight control system mainly includes three modules: sensor input and reading module, controller module and navigation control module. The input of the sensor module comes from the real-time solution of the aircraft model. The state value of UAV model solution is fed back as the sample value of sensor. Controller module is the main object in this paper, mainly including navigation mode and control law of decoding module, decoding module is responsible for handling the navigation module was sent data to control law, and the navigation control module mainly complete unmanned aerial vehicle (UAV) flight control modal and the control law of output control parameters to decoding module. The following uses AADL to model and analyze the flight control system of the airborne control system in the UAV information physical fusion system.

The first step is physical system modeling: the UAV information physical fusion system is regarded as a rigid body with constant mass. Hardware system modeling includes three parts: input equipment modeling, computing platform modeling and actuator modeling. Firstly, establish the Modelica model of the physical system, and then, according to the transformation rules of Modelica model and AADL model described in the previous section, transform the Modelica model of the UAV physical system into the AADL model with extended attributes. The results of the overall AADL model of UAV are as follows:

![Figure 4-2 overall AADL model of UAV](image)

The second step is the modeling of software system: the flight control software of UAV mainly realizes the functions of automatic take-off, three-axis attitude maintenance, three-dimensional track maintenance, automatic approach landing, etc., including the control of attitude (pitch/roll), altitude and track during the flight of UAV. Flight control law is the basis for UAV to achieve the above functions. Different control strategies are adopted for different motion modes to ensure good performance of UAV. In this paper, two subsystems of UAV control system UAV-system modeling, namely flight mode judgment and switching subsystem FMC and flight control solution operator system LCC, are shown in figure 4-3.
The third step is to model the spatial behavior: Unmanned aerial vehicle (UAV) flight process of physical information fusion system is made up of takeoff, autonomous navigation and landing three parts, the take-off and landing take the glide phase, triggered by events to switch between state, unmanned aerial vehicle (UAV) to receive instruction, take-off by sliding state stopped into the state, reached after take-off point, by sliding into the flight state, received after the landing, on arrival at the landing site, unmanned aerial vehicle (UAV) flight from sliding mode, finally to stop state. The continuous climbing and descent phases of a UAV after it reaches its take-off point and after receiving landing instructions are critical and unique phases of the entire range and are modeled as separate states, i.e., climbing and descent. In view of space, the space behavior model of the airborne part of the UAV information physical fusion system is not listed in detail. Convert the model mapping rules according to the last under the UPPAAL timed automata model as shown in figure 4-4, transitions between states in the model has a corresponding time constraints, a state after the conversion, automatic zero time, when the triggering event or receives the instruction to the current state of the state and phase at the same time, will not happen state change, such as in stop state receives the instruction, the unmanned aerial vehicle (UAV) stop.

**Figure 4-3** UAV flight control system subsystem

Step 4 is to internal system behavior modeling: It is behavior modeling of flight model control of
CPS. The modal calculation model have three flight control, namely the initial state, state, and the idle state, every 50 time unit will trigger a calculation process of the current mode to calculate. Its code is as follows:

```plaintext
System implementation FMC.imp
Annex behavior/**
States
Start: initial state; Compute , Idle: state;
Transitions
Start-[t == 0] -> Compute;
Compute-[t <= 20] -> Idle;
Idle-[t <= 50] -> Idle;
Idle-[t > 50] -> Compute(t = 0);
** j.
End FMC.imp
```

Convert this model into the time automaton model under UPPAAL according to the mapping rules in section 3, as shown in figure 4-5:

![Figure 4-5 model of time automaton for modal computation of UPPAAL](image)

**Figure 4-5** model of time automaton for modal computation of UPPAAL

5. Conclusion

Fusion modeling and simulation of CPS, on the one hand, aims at the integration of physical process and computational process of entity, on the other hand, aims at the integration of architecture model and dynamic behavior model. This paper solves the problem that single modeling language lacks the ability to describe CPS, extend AADL to support the structural modeling of CPS computing entities, interactive entities and physical entities. Describe the interaction relations between discrete information domains and continuous physical processes. Based on the collaborative modeling of scene structure and behavior, extend the computational models such as finite state machine, hybrid automaton and data flow; improve the system model construction ability of CPS. Madelia, a multi-domain modeling language which has advantages in physical system modeling, is used to model the physical system, and the conversion rules between Modelica model and AADL model are proposed to convert the Modelica model of the physical system into AADL model. The temporal automata is used to expand the behavior attachment of AADL, so that the behavior attachment can describe the spatial behavior of the information physical fusion system under the time constraint, and the mapping rules of the behavior attachment of AADL to the temporal automata model of UPPAAL are given, so that the correctness and rationality of the established AADL behavior model can be verified by UPPAAL.
Reference

[1] EDISON J C, LEE E A, MATIC S, SESHIA S A and ZOU 2012 Distributed real-time software for cyber-physical systems Proceedings of the IEEE 100(1) 45-59

[2] LIU Z, LIU J, HE J and Ding Z 2012 Spatio-temporal UML state chart for cyber-physical systems ICECCS 137-46.

[3] Ranca RB, Bodeveix JP, Filali M, Rolf and JF 2007 The AADL behaviour annex-Experiments and roadmap, Proc of the 12th IEEE Int Conf on Engineering Complex Computer Systems. Washington. 377-82

[4] Behrmann Q, David A and Larsen K 2004 GA turorial on UPPAAL Proceedings of the 4th Intl School on Formal Methods for the Design of Computer, Communication, and Software Systems Bertinoro: Springer Verlag 200-36.

[5] Rajeev A, David L D. 1994 A theory of timed automata Theoretical Computer Science 126(2) 183-235.

[6] SAE Aerospace.SAE AS5506 Annex: Behavior Specification V1.6 [EB/OL].
   http://www.aadl.info/aadl/documents/Behavior_Annex1.6.pdf, 2006.

[7] Yuan suozhong 2003 Development of digital UAV flight control system Computer measurement and control 11(1) 64-8

[8] Zhu chenxi 2014 Analysis and design method of information physical fusion system based on AADL Guangdong university of technology 5

[9] Mkaouar H, Zalila B, Hugues, Jérôme and et al. 2019 A formal approach to AADL model-based software engineering International Journal on Software Tools for Technology Transfer.

[10] Cao Y, Dong Y, Wei X and et al. 2019 AADL Vulnerability Modeling and Security Analysis Method 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C).