

Current JAXA's AADL Activity

2009/4/27@ESTEC

JEDI/JAXA

Masa Katahira

Naoki Ishihama

- I. **Evaluate the Application of AADL technology** (that is considered as one of Model based development method) for onboard component (Avionics) development

- II. Study the **effect and problem of modeling and tool** by adapting to the real space development

AADL training (Jan 2009)

- CMU AADL standard 2day course @Tokyo
- OSATE tool training (1day) @Tokyo
- Mini W/S (2day) @Tokyo
 - Discussion for AADL real project example
 - Try AADL modeling of evaluation system

AADL trial project (Jan – Mar 2009)

- Language evaluation
 - AADL modeling
- AADL model validation
- Tool evaluation

- **Onboard bus network FDIR (Fault Detection, Isolation, Recovery) function**

- **Error detection function**

- Check the BUS port status by software
 - Confirm the command response to the I/F components

- **Error recovery function**

- Root switch function
 - Switch the Bus (disconnection of line, port failure etc)
 - Switch the CPU port
 - Switch the CPU port to Slave (the failure of CPU board and router etc)
 - Degenerate operation
 - Continue the operation at only single component

- **Bus network architecture**

- Computer A

- CPU, memory
 - Network control
 - Redundant CPU board (Cold standby)

- Computer B

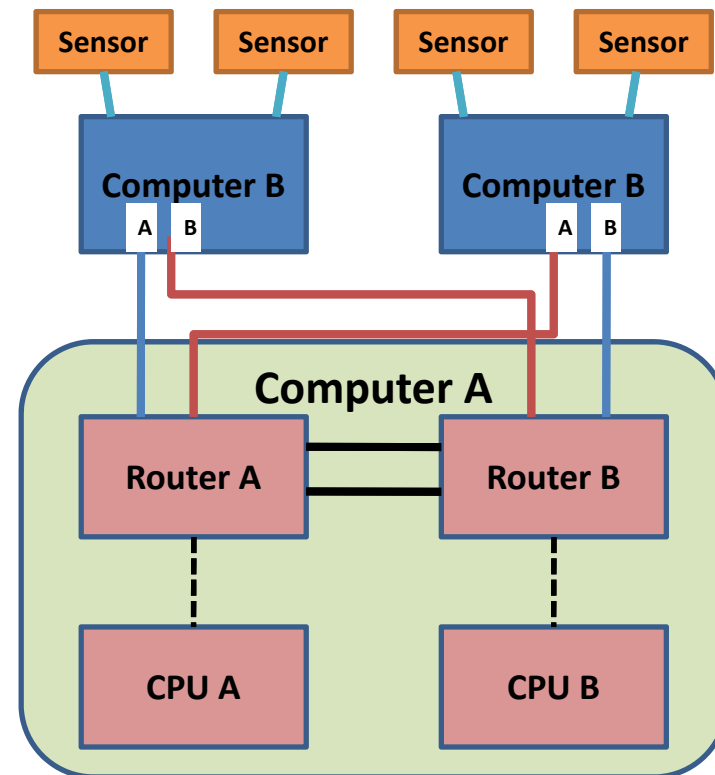
- Sensor handling

- Network

- Serial communication Bus

- Router

- Bus router
 - Redundant configuration



I. AADL Modeling

- ✓ Strictness of system specification description
 - a. Can the system specification be described by using AADL?

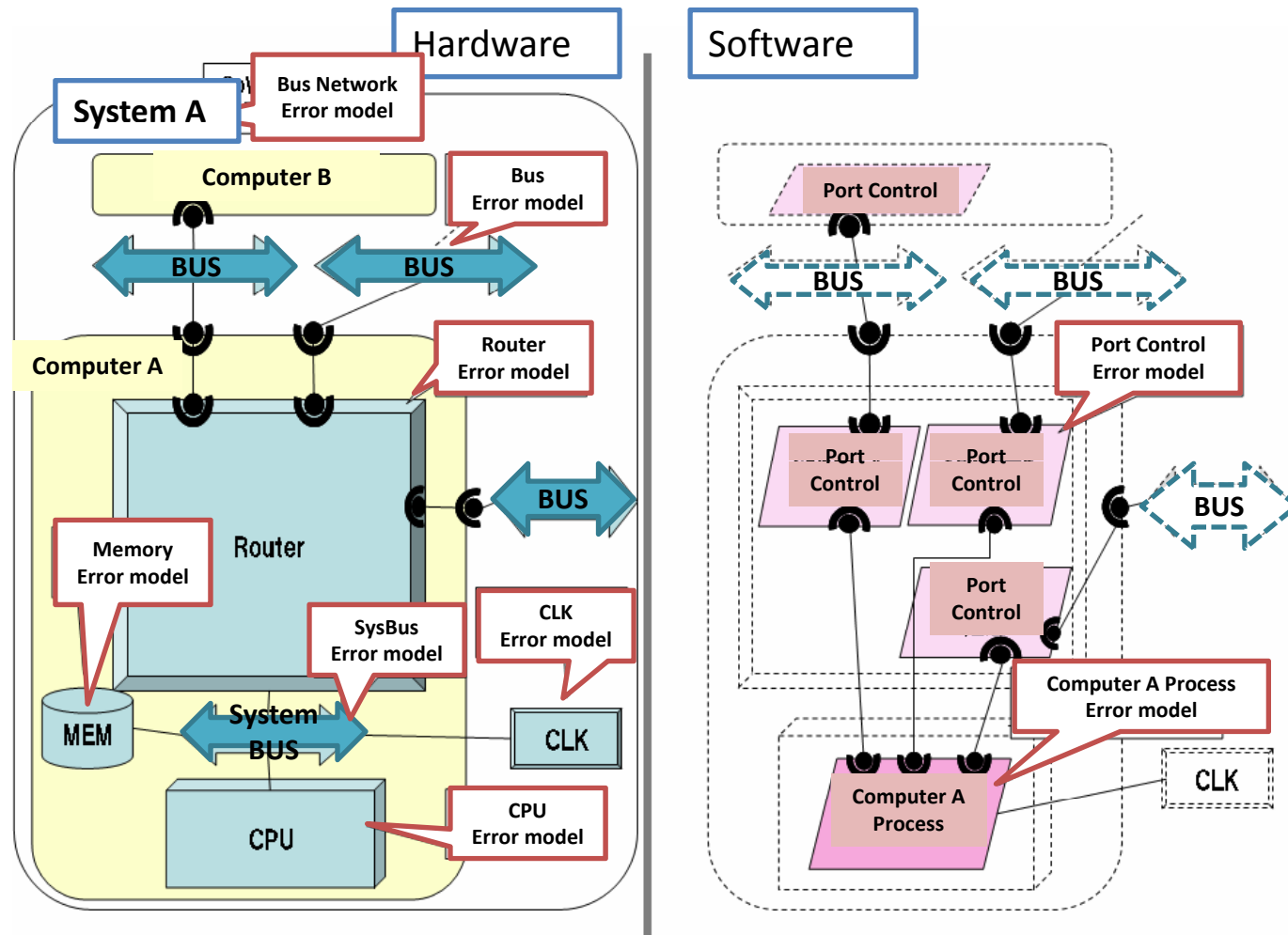
II. AADL model validation

- ✓ FDIR function evaluation
 - a. Is the AADL model correct compared with scenario?
 - b. Can the model be easily evaluated?

III. Tool (OSATE) evaluation

- a. Is it easy-to-use?
- b. Is the OSATE model analysis function effective?
- c. What is the necessary point ? (when we use OSATE at real project)

- Hardware model: component architecture
- Software model: CPU and Router control



- **State transition description**

- **[Advantage]**

- Can do exhaustive verification of the state transition as system

- **[Faced Problem]**

- Not make the total state transition as figure from individual state transition automatically

- **I/F specification description**

- **[Advantage]**

- Can make the model of Input-output relation and state propagation (extent of the impact)

- **[Faced Problem]**

- Not add the some semantic information to one I/F
 - Not describe the value data like register value (substitute the event port)
 - Need the each event port at state transition
 - » The modeling impact is high

- **Dynamic specification description**

- **[Faced Problem]**

- difficult to make the sequence model
(Can describe only at “state transition”)

- **FDIR function description**

- [Advantage]**

- Make the FDIR function description AADL model

- [Point to be improved]**

- Necessary to be easy to understand how the inherent state transition of component influence the state transition as the system
 - (Because) Difficult to pick up the total system state transition from error generation

- **Error state definition**

- [Advantage]**

- Can define the explicit error state by using the error model
 - Can do the model verification automatically

- **Determine of the state propagation area**

- [Point to be improved]**

- Need the function that can pick up the propagation result automatically to ensure the work efficiency
 - (Because) we can track the state propagation only by myself (hand power)

➤ *Evaluation method*

- Review of the specification model by using the evaluation model
 - Specification model: component architecture, port control, error model
 - Evaluation model: FDIR scenario
 - Example Scenario (Rooting switch scenario)
 - Ý Fault occurrence (define to the Error model)
 - Þ Fault propagation (track of Event and State transition)
 - Ɔ Fault detect (change of subprogram)
 - à Fault recovery (track of state transition)

- **FDIR function verification**

- Confirm that the detection and recovery function for fault mode is described
- Confirm that can do the model verification for FDIR function

[Advantage]

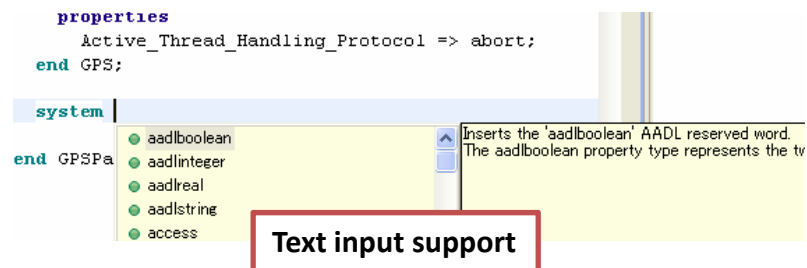
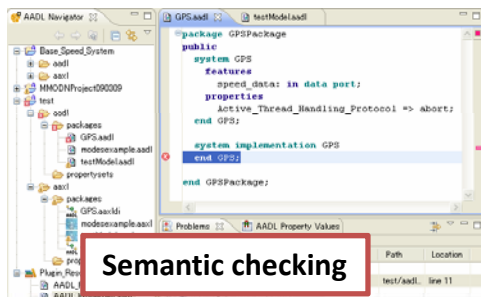
- Can be verified by tracing the state propagation individually defined in hardware and software

[Point to be improved]

- Generate figure of the state transition automatically
 - Be inefficient that the person traces the state propagation

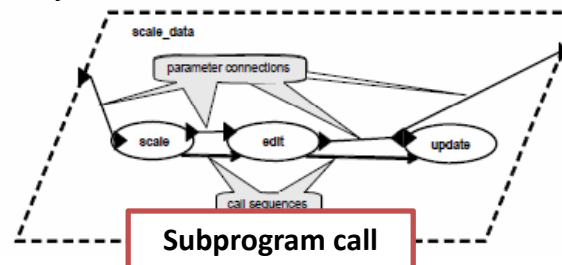
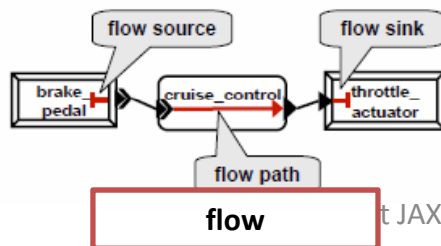
[Advantage]

- Model analysis function
 - Flow analysis, scheduling analysis, architecture analysis etc
 - Semantic checking function
 - Support of text input



[Faced Problem]

- Not support all AADL description at OSATE diagram editor
 - Non support description: “flow” and “subprogram call”
 - Need to make the model by using text editor and object editor



- There is no big lack in the AADL specification description to the model (though there is some restriction)
- Can lost the vague expression
- The error model's application for error propagation is effective (though there is some efficient problem)
 - Ex: Complex FDIR function that hardware and software synchronize
- Possible to do the verification of FDIR function by using AADL model
- Application to development process
 - Possible to adapt to only high risk area
 - Possible to do the early verification (specification analysis, design study, review, model verification) at early phase (requirement and design level)

- It is necessary to maintain or make the tool to improve cost-effectiveness.
 - Ý Support of **Diagram editor** for OSATE tool
 - Ɔ Support of the **Graphical editor and view** as system mode
 - Automatically making the figure or table from text model (be difficult to review the text model)
 - ß Make the **Verification tool**
 - Pickup the information depending on purpose of verification
 - State propagation list as system (or total system)
 - FDIR verification support tool

- Study phase (end of 2009)
 - **Item:**
 - “Where can we introduce to legacy process?”
 - “How can we collaborate between legacy tool (development and IV&V) and AADL?”
 - Ex: SpecTRM, SPIN, UPPALL, Matlab, Rhapsody, ...
 - **Output:**
 - JAXA AADL activity load map