SysML to AADL Bridge

Automating the Translation of SysML into AADL for Analysis and Refinement

Hazel Shackleton
hazel.shackleton@adventiumlabs.com

Steve Vestal
steve.vestal@adventiumlabs.com

DISTRIBUTION A. Approved for public release: distribution unlimited.

This material is based upon work supported by the U.S. Army Development Command (DevCOM), U.S. Army Combat Capabilities Development Command Aviation & Missile Center (DEVCOM AvMC), ATTN: Security Office, Bldg 401, Lee Blvd, Fort Eustis, VA 23604-5577 under contract no. W911W6-17-D-0003. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Army.
Agenda

• Design goals & supported workflows
• Demo
  – Using the profile
  – Code generation
  – Analysis and round-trip engineering
  – Model refinement & integration
• Ongoing work (behavior & property mapping)
• Questions
Targeted Portions of SysML Only

- Allow users to annotate portions of SysML models using an AADL Profile and automatically translate those portions of the SysML model into AADL.

Most SysML element and diagram types will not be translated.

- Package
- Block Def. Diagram
- Internal Block Diag.
- Sequence

Much AADL modeling will be done by extending and refining the generated AADL.
Clear Semantics

• Transition SysML specification from system engineering phase of development into standardized AADL semantics for ACVIP/AADL virtual integration analysis.

• Provide a common (across multiple programs, contractors) AADL Profile based on the AADL standard that is amenable to extension for additional AADL properties and annexes (for integration of additional analysis tools).

Provided SysML AADL Profiles:
• Core AADL stereotypes and properties
• Extension profiles for ARINC 653, MILS, SESSAF (STPA), FASTAR, MADS

*MagicDraw/Cameo Enterprise Architecture and Sparx Enterprise Architect*
### Mapping of Core AADL Stereotypes

(Excerpt from Modeling Guidelines)

<table>
<thead>
<tr>
<th>AADL stereotype</th>
<th>SysML element</th>
<th>Generated AADL declaration(s)</th>
<th>AS5506C</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADL_Package</td>
<td>package</td>
<td>package in a file</td>
<td>4.2</td>
</tr>
<tr>
<td>AADL_Abstract</td>
<td>block</td>
<td>abstract type + implementation</td>
<td>4.6</td>
</tr>
<tr>
<td>AADL_Data</td>
<td>block</td>
<td>data type + implementation</td>
<td>5.1</td>
</tr>
<tr>
<td>AADL_Subprogram</td>
<td>block</td>
<td>subprogram type + implementation</td>
<td>5.2</td>
</tr>
<tr>
<td>AADL_Thread</td>
<td>block</td>
<td>thread type + implementation</td>
<td>5.3</td>
</tr>
<tr>
<td>AADL_Thread_Group</td>
<td>block</td>
<td>thread group type + implementation</td>
<td>5.4</td>
</tr>
<tr>
<td>AADL_Process</td>
<td>block</td>
<td>process type + implementation</td>
<td>5.5</td>
</tr>
<tr>
<td>AADL_Processor</td>
<td>block</td>
<td>processor type + implementation</td>
<td>6.1</td>
</tr>
<tr>
<td>AADL_Virtual_Processor</td>
<td>block</td>
<td>virtual processor type + implementation</td>
<td>6.2</td>
</tr>
<tr>
<td>AADL_Memory</td>
<td>block</td>
<td>memory type + implementation</td>
<td>6.3</td>
</tr>
<tr>
<td>AADL_Bus</td>
<td>block</td>
<td>bus type + implementation</td>
<td>6.4</td>
</tr>
<tr>
<td>AADL_Virtual_Bus</td>
<td>block</td>
<td>virtual bus type + implementation</td>
<td>6.5</td>
</tr>
<tr>
<td>AADL_Device</td>
<td>block</td>
<td>device type + implementation</td>
<td>6.6</td>
</tr>
<tr>
<td>AADL_System</td>
<td>block</td>
<td>system type + implementation</td>
<td>7.1</td>
</tr>
<tr>
<td>AADL_Feature</td>
<td>port</td>
<td>feature (in a type declaration)</td>
<td>8.1</td>
</tr>
<tr>
<td>AADL_Feature_Group</td>
<td>port</td>
<td>feature group (in a type declaration)</td>
<td>8.2</td>
</tr>
<tr>
<td>AADL_Feature_Group</td>
<td>interfaceBlock</td>
<td>feature group type + implementation</td>
<td>8.2</td>
</tr>
<tr>
<td>AADL_Access</td>
<td>port</td>
<td>provides/requires access (in a type declaration)</td>
<td>8.6-8.8</td>
</tr>
<tr>
<td>AADL_Flow</td>
<td>interaction</td>
<td>flow source, path, sink, end-to-end</td>
<td>10</td>
</tr>
</tbody>
</table>
Workflow Support

• Virtual integration of computer system architecture model occurs in an AADL modeling environment.
• Modularized, textual, BNF-based exchange formats have been proven very useful for model exchange and configuration management.
Modeling Style Support

• SysML standards and conventions, natural for SysML aficionados.
• AADL standards and conventions, natural for AADL aficionados.
• Clear mapping from stereotyped SysML elements with tagged value properties to the AADL generated from those elements and properties.

Provided Documentation:
• SysML AADL Profile and Modeling Guidelines
• SysML example models
 SysML AADL Profile
 and Modeling Guideline Topics

- Names and Name Spaces
- Allocation and Binding Relations
  - Packages, Blocks, Types, and Implementations
  - Tagged Values and Properties
  - Ports, Features, and Connections
  - Interactions and Flows
  - Annex Profiles

- Not covered today
- At least briefly discussed
(Quick Shout Out/Thank You)

• Our work was accomplished with oversight by a SysML AADL Working Group consisting of Army Aviation system integrators and suppliers.

• Other work we looked at:
  – MARTE
  – Open Archive Toulouse Archive Ouverte (OATAO)
  – ExSAM Profile, 2011 (University of Oslo, Norway)
  – SCADE Architect to AADL translator
  – Rockwell Collins SysML to AADL translator v1.1 2016
  – SLICED AADL Profile for UML state machine diagrams
Selecting Elements for Translation

• Using AADL Profile to select Packages and Components for translation.
• Annotating the Block Definition Diagram.
Annotating with Tag Values

- Adding detailed port and connections information using Stereotype Tags.
- Annotating the Internal Block Diagram.
Adding End to End Flows

• Creating a sequence diagram to add a flow through existing components.
Round-trip Engineering

- Generation of AADL project.
- Mapping back from AADL to SysML using GUIDs.
Split Between SysML & AADL

- Detailed AADL Analysis: where should additional property values and model details be added?
  - Option 1: Expand SysML model with refinements, new components, new tags/properties.
  - Option 2: Refine generated AADL model.

---

DISTRIBUTION STATEMENT A: Approved for public release; unlimited distribution.
Option 1: SysML Refinement

Example of use: Want to provide the property at the SysML level.

- Generalization/Inheritance refinement all supported by the translator.
- Create custom profiles for a missing AADL Property Sets.
  - Detailed instructions in provided documentation.
  - Any tag defined in a profile named “AADL_<something>_Profile” will be translated into an AADL property if a SysML model element has that tag.

Provided Example Model:
- Architecture_Refinement_Example

Provided SysML AADL Profiles:
- Core AADL stereotypes and properties
- Extension profiles for ARINC 653, MILS, SESSAF (STPA), FASTAR, MADS

MagicDraw/Cameo Enterprise Architecture and Sparx Enterprise Architect
Options 2: AADL Refinement

*Example of use:* Complex properties defined during the embedded design phase with added analyses as the fidelity of the model increases.

- Refine Mission System & Air Vehicle to include Error Modeling using EMV2 annex.
AADL Integration

• Integrate existing project feature types into generated AADL model.
Highlighted Ongoing Work (1 of 2)

- Behavior & Annexes under GUMBO (Grand Unified Modeling of Behavioral Operators) SIBR (Adventium and Kansas State).

```
State Machines

--- from SysML TempControl_19_R_3_31c48641_1681579128489_221139_43824
thread [implementation TempControl_impl]
from SysML behavior_specification_19_R_3_31c48641_1608321832653_136478_42944)
annex behavior_specification (==
variables
lastCurrentTemp: TempSensor::TemperatureImpl; lastSetPoint: SetPointImpl
states
start: Initial State; run (State_Type = complete)
run_SLM: SLM (State_Type = complete)
run_GTN: GTN (State_Type = complete)
waiting: complete state
transitions
start => run_SLM
run_SLM => run_GTN
run_GTN => waiting
waiting => lastSetPoint
```

behaviorSpecification annex

```
import [State Machine TempControl] TempControl

aAAL_SLICED_State
run
(State_Type = complete)

aAAL_SLICED_State
checkTemp

<from fanA/a / lastCurrentTemp = currentTemp;
lastSetPoint = setPoint

all [currentTemp.degrees < lastSetPoint.low.degrees] /
fanCmd[FarEnd[FarEnd:FanEndImpl]]

all [currentTemp.degrees = lastSetPoint.low.degrees] /
fanCmd[FarEnd[FarEnd:FanEndImpl]]

all [currentTemp.degrees < lastSetPoint.high.degrees] /
fanCmd[FarEnd[FarEnd:FanEndImpl]]

all [currentTemp.degrees = lastSetPoint.high.degrees] /
fanCmd[FarEnd[FarEnd:FanEndImpl]]

fanA/a = fanA/a
```

2/24/2021
Highlighted Ongoing Work (2 of 2)

- Mapping other existing profile tag values or properties to generated AADL properties.
- User must ensure mapping is semantically correct.
Unsupported AADL Features in Bridge

- Connection bindings that are ordered lists (routings)
- System operating modes
- Prototypes
- Subprogram parameters
- Subprogram call sequences
- Annex sub-languages (e.g., EMV2 Annex, Behavior Specification Annex)*
- Multiple implementations for a type must be declared as extensions to both the type and the implementation.

*Ongoing work
Questions?

https://camet.adventium.com

- SysML AADL Profiles
  - Core AADL stereotypes and properties
  - Extension profiles for ARINC 653, MILS, SESSAF (STPA), FASTAR, MADS
  *MagicDraw/Cameo Enterprise Architecture and Sparx Enterprise Architect*
- Documentation
  - SysML AADL Profile and Modeling Guidelines
  - SysML example models
- *SysML to AADL Bridge Tool*
  *MagicDraw/Cameo Enterprise Architecture and Sparx Enterprise Architect*