

Discussion

Frank Singhoff & all workshop
participants

26th Ada-Europe International Conference on Reliable
Software Technologies (AEiC 2022), Ghent, Belgium

ADEPT 2022 ... (mostly) closed

- Thanks 😊
- 8 speakers, about 20 participants
- Post workshop : papers, slides, return of experience, ADEPT 2023?
- Returns of experience of each speaker
- Summary on R.E.
- Discussion proposal?

Return of experience about AADL:

Gianluca Bardaro, Matteo Matteucci

1. About the AADL language

- Why did you choose AADL? What were the motivations to use AADL in your projects?

We started using model-based development to solve some of the problems of robotic software. AADL was the best fit because of its support for both hardware and software components, textual and graphical syntax, inheritance and modularity, and the available tools.

- What are the language features that are missing today in AADL and that you expect to see in the next version of the standard?

At the moment, there are no particular features we would add to the language.

Return of experience about AADL:

Gianluca Bardaro, Matteo Matteucci

2. About the AADL tool

- What are the tools you are using?

We are using OSATE as an editor and Ocarina for code generation.

- What are the main features you need? Do the tools provide them?

Our main interest is in code generation, which is supported by Ocarina.

- What were the main issues you faced?

Some of the features of the language we heavily use (e.g., prototypes) are not supported by the tools. Moreover, it is quite difficult to extend the functionalities of the tools to suit our needs.

- What are you expecting in the next tool versions?

Easy to use APIs and more customization. Full support of the language features.

Return of experience about AADL:

Gianluca Bardaro, Matteo Matteucci

3. About the AADL community

- What do you expect from it? How do we organize it?

Reaching out to other people working with AADL is difficult, there is no place for the community to share problems and solutions. An Internet forum (e.g., Discourse) or a Q&A page (e.g., Stack Exchange) would help significantly the community.

Return of experience about AADL: *Alberto Bombardelli, Marco Bozzano, Roberto Cavada, Alessandro Cimatti, Alberto Griggio, Massimo Nazaria, Edoardo Nicolodi, Stefano Tonetta*

1. About the AADL language

- Why did you choose AADL? What were the motivations to use AADL in your projects?
 - COMPASS tool: HW/SW co-design, textual specification language
 - TASTE tool: chosen by ESA
- What are the language features that are missing today in AADL and that you expect to see in the next version of the standard?
 - Specification of timed/hybrid behavior in state machines
 - Bridge between behavioral and safety annexes: fault injection for state machines

Return of experience about AADL: *Alberto Bombardelli, Marco Bozzano, Roberto Cavada, Alessandro Cimatti, Alberto Griggio, Massimo Nazaria, Edoardo Nicolodi, Stefano Tonetta*

2. About the AADL tool

- What are the tools you are using?
 - COMPASS, TASTE (AADL as an input language)
- What are the main features you need? Do the tools provide them?
- What were the main issues you faced?
- What are you expecting in the next tool versions?
 - N.A. (We mostly develop tools based on AADL, rather than using the existing ecosystem)

3. About the AADL community

- What do you expect from it? How do we organize it?
 - Advertise the existence of COMPASS and TASTE. COMPASS usage is free for ESA member states, TASTE usage is free (open source)

Return of experience about AADL:

Ellidiss

1. About the AADL language

- AADL appeared to better cover the need to design and analyze real-time software architectures than existing HRT-HOOD or COTRE, 20 years ago.
- There are probably too many features already. The most important need is to provide a better formal specification of the language and to support well-defined subsets.

2. About the AADL tool

- One of the successes of AADL is the ability to build heterogenous tool-chains (e.g., TASTE)

3. About the AADL community

- Changing the lead from a Technical Group to a User Group ?

Return of experience about AADL:

Patrick Denzler, Daniel Scheuchenstuhl, Daniel Ramsauer, Wolfgang Kastner

1. About the AADL language

- We explored the capabilities of AADL in the context of a fog computing platform.
- We needed hardware and software in one model.
- We missed a proper documentation, some constructs/options we had to figure out by try and error.

2. About the AADL tool

- We used as a modelling environment: the Open Source AADL Tool (OSATE), and for code generation, the Ocarina tool suite.
- Modelling, code generation and model checking, analysis tools (e.g. RT, Security etc.)
- Code generation did not work for and other tools as well.
- Remove tools, not maintained anymore. Better documentation.
- Alternative to Eclipse? Plugin to Visual Studio Code?

3. About the AADL community

- Needs to be open source, workshops, teaching material or working examples.

Return of experience about AADL:

Xiong Xu, Shuling Wang, Bohua Zhan, Xiangyu Jin,
Jean-Pierre Talpin, Naijun Zhan

1. About the AADL language

- Its capability to capture various aspects of embedded systems
- Validation of system architectures and implementations
- Prediction of runtime characteristics
- Missing feature: No component is oriented to physical environment

2. About the AADL tool

- OSATE
- Graphical interface (✓)
- Transformation graphical models to other forms (?)
- How to co-model and analyze heterogeneous CPSs using the tool?

3. About the AADL community

- More materials that can help transition AADL to newcomers
- More case studies that can exhibit the desired features of AADL
- A formal semantics for the AADL language

Return of experience about AADL: *Zhibin Yang, Zhikai Qiu, Yong Zhou, Zhiqiu Huang, Jean-Paul Bodeveix, Mamoun Filali*

1. About the AADL language

- Why did you choose AADL (powerful architecture modeling of embedded systems, especially the description of the dynamic architecture and timing properties)? What were the motivations to use AADL in your projects? (architecture modeling and analysis)
- What are the language features that are missing today in AADL and that you expect to see in the next version of the standard? (property sets for the modeling of multi-core and machine learning components)

2. About the AADL tool

- What are the tools you are using? (OSATE, OCARINA, Cheddar, AGREE MACAerospace(NUAA))
- What are the main features you need?(Modeling, safety analysis, schedulability analysis, compositional verification, code generation) Do the tools provide them? (Partially)
- What were the main issues you faced? (compositional verification of asynchronous execution model)
- What are you expecting in the next tool versions? (modeling and analysis of multi-core, etc.)

3. About the AADL community

- What do you expect from it? How do we organize it? (good)

Return of experience about AADL: *Eric Senn, Lucie Bourdon*

1. About the AADL language

- motivations: robotic applications → need to represent 1st the SW architecture, & 2d the HW
- need to check on deployment options → understand problems / identify bottlenecks / find a solution
- why to choose: I knew it (since SPICES) and used it / from this year teaching it
- missing: *actual_processor_binding* to a set of *processor* components (CPU cores)
- missing: *bwcapacity* applying to *virtual bus*, or better, to a new *software bus* component (cf ROS middleware)

2. About the AADL tool

- tools used: OSATE2, cheddar plugin, cheddar
- features needed: resources budget analysis (cpu load & bus load) / schedulability analysis / timing-latency analysis
- issues faced: many bugs in the diagram view / bug in analysis generation report
- expected: bug correction / automatic completion / easy descent into the hierarchy of sub-components (while binding ...) / generation of more compact diagrams / case sensitivity

3. About the AADL community

- expecting: feedback on our research activity / ease sharing our libraries

Return of experience about AADL: *Dominique Blouin, Paolo Crisafulli, Françoise Caron, Cristian Maxime*

1. About the AADL language

- Why did you choose AADL? What were the motivations to use AADL in your projects?
 - Domain coverage, maturity of language and tools, industrial language to test new MDE approaches.
- What are the language features that are missing today in AADL and that you expect to see in the next version of the standard?
 - Flow specification is cumbersome and could be simplified

2. About the AADL tool

- What are the tools you are using?
 - OSATE, AADLInspector, Cheddar, RAMSES, ALISA
- What are the main features you need? Do the tools provide them?
 - Model editing and analysis, code generation. Yes the tools provide them.
- What were the main issues you faced?
 - For OSATE, some elements from the declarative model are missing in the instance model such as subprograms and annexes. Also a bi-directional transformation between declarative and instance models would be useful so that tools like RAMSES can only update the instance model. Some minor issues such as errors in Latency computations.
- What are you expecting in the next tool versions?
 - A bi-directional transformation between AADL declarative models

3. About the AADL community

- What do you expect from it? How do we organize it?
 - It would be good to continue the AADL users days that were taking place during the committee meetings, especially that those committee meetings no longer take place. Adept is excellent for that.

About the AADL language, *Summary tentative*

- Why did you choose AADL
 - Support for both hardware and software modeling, deployment, inheritance and modularity
 - Its capability to capture various aspects of embedded systems, including the dynamic architecture and timing properties
 - AADL appeared to better cover the need to design and analyze real-time software architectures
- Textual and graphical representation.
- Available tools.
- Maturity of language and tools
- AADL ability to build heterogeneous tool-chains (e.g. TASTE)
- Already knew the language or part of a needed framework (e.g. TASTE).

About the AADL language, *Summary tentative*

- What are the language features that are missing
 - Timed/hybrid behavior in state machines
 - Bridge between behavioral and safety annexes
 - New properties on processor and bus components. Property sets for the modeling of multi-core and machine learning components
 - Flow specification is cumbersome and could be simplified
 - No component is oriented to physical environment
- Provide a better formal specification
- We missed a proper documentation.
- Probably too many features already. Support well-defined subsets.

About the AADL tool, *Summary tentative*

- What are the tools you are using?
 - Tools & related languages : STOOD, OSATE, Ocarina, AADLInspector, , RAMSES, ALISA, COMPASTA, COMPASS, TASTE, AGREE, RESOLUTE, MACAerospace(NUAA), MARS, RAMSES, others ...
- What are the main features you need? Do the tools provide them?
 - Modelling
 - Code generation
 - Analysis: model checking, safety, schedulability/timing analysis, resource budget analysis, security, compositional verification
- What were the main issues you faced?
 - Bugs
 - Tools that are not supporting all/required AADL features.
 - Difficult to extend the functionalities of the tools to suit our needs.

About the AADL tool, *Summary tentative*

- What are you expecting in the next tool versions?
 - Features
 - Full support of the language features.
 - Automatic completion / easy descent into the hierarchy of sub-components (while binding ...) / generation of more compact diagrams / case sensitivity
 - News domains
 - How to co-model and analyze heterogeneous CPSs?
 - Modeling and analysis of multi-core
 - Tool ecosystem
 - Alternative to Eclipse? Plugin to Visual Studio Code?
 - Remove tools, not maintained anymore. Better documentation
 - Interoperability, customization
 - Transformation graphical models to other forms
 - A bi-directional transformation between AADL declarative models
 - Easy to use APIs and more customization.

About the AADL community, *Summary tentative*

- What do you expect from it? How do we organize it?
- *Reaching out to other people working with AADL is difficult, there is no place for the community to share problems and solutions.*
- *Changing the lead from a Technical Group to a User Group ?*
- Having:
 - Feedback from users
 - More materials that can help transition AADL to newcomers
 - Open source material: case studies or working examples to show AADL features, libraries, teaching materials
 - Workshops, continue AADL users days
 - Documentation, formal semantics for the AADL language
 - Advertising the existence of tools (especially when open source)