AADL Standards Meeting Feb 11-13, 2019 & AADL Tools Demonstration Day Feb 14, 2019

- Location Huntsville AL, USA
    - Hosted by University of Alabama in Huntsville, Rotorcraft System Engineering and Simulation Center.
    - Print a copy of the Campus map and Parking Map that are provided with this announcement, you will need them to navigate and park.
    - Addresses you can plug into you GPS:
      Feb 11-13: Von Braun Research Hall, John Wright Dr NW, Huntsville, AL 35805
      Feb 14 Morning: Charger Union, 4705 Holmes Ave NW, Huntsville, AL 35899
      Feb 14 Afternoon: Training Center, 1410 Ben Graves Drive NW, Huntsville, AL 35816
    - Parking in the right places is critical to avoid tickets (~$50 each).
    - See building numbers referred to in the agenda, easiest to read on the Campus map: https://www.uah.edu/map . Attached are maps highlighting buildings and parking. Building numbers and Parking numbers will be referenced hereafter in this announcement as Bldg ## and Prkg ##, respectively, as per the map. Note that these numbers are not identified at the actual physical address on the buildings or parking lots on campus and are only a map reference.
    - AADL Meeting (Wernher Von Braun Research Hall, Bldg 34, Room M50)
        - Meeting parking in lot south of Bldg 34 which is Prkg w30.
    - AADL Demonstration Day (Charger Union Theater Bldg 19 then lunch, then Conference Training Center (8), internal Exhibit Hall, Bldg 8a)
        - Demo Day parking behind the Executive Plaza, see parking maps for Demo Day buildings and all day parking.
    - All times noted are U.S. Central Time Zone

      Remote – Alexey Khoroshilov, Dave Glutch, Denis Buzdalov, Phil Suematsu, Prachee Sharma, Sam Proctor, Dhruv Monya, Mark Brown, Brendan Hall,

  Present – Kevin Herring, Aron Buehrer, Jeff Obermark, Philip Alldredge, John Hudak, Kyle Litwin, Christopher Cargal, Dominique Blouin, John Hatcliff, Jerome Hugues, Brian Larson, Bruce Lewis, Dorothy Lloyd, Joseph Seib el, Peter Feiler, Mark Brown, Pierre Dissaux, Shawn Kline, O'Neill,

Monday, Feb 11 – Wernher von Braun Research Hall (Bldg 34), 2nd Floor, M50. Parking only in the lot on south side (Prkg W30).

- 0900-1000: AADL standardization committee news + action items (Bruce Lewis)
    - Check for sure about ESAC side meeting, otherwise 24-27 Paris at Paristech

- Reaffirmed – for status of Data Annex
- V3 – change to Spring 2020, informal for Oct 2019 for one part,
- Kyle, Kevin, Arron, John H send copies of the FACE, Chris C, John H
- STPA and AADL – guidance document, Kyle, John H, Kevin,
- FMI – an integrated testbench with Modelica, workflow, Testing, What kind of credit can we get from that. Involved to decide what to do to support general use.
- Jerome will cover Space Robotics 30 minutes , issues encountered, at the next meeting.
- SEI is going through what is on AADL Wiki, Bringing it up to date. Will take it off line when the new AADL webpage is ready.

- 1000-1030: AADL v3- Roadmap (Peter Feiler)
  - Long term support (LTS) for OSATE2.x
  - https://github.com/osate for issue reports
  - New AADL v2.2 errata github.com/saeaadl/…
  - Migration path – Instance model representation with minimal changes – most analyses operate on instance model
  - Declarative model – Translation from V2.2 to V3, definitely, Peter will look at going both ways as well. What are the issues.
  - API for instance model – Jerome – would make it easier to plug-in new tools. Peter –we will have it documented as part of OSATE, whether it's needed in the standard we can discuss.
  - Key V3 Changes
    - Packages and general syntax
      - Import of namespaces, property definitions in packages, private classifiers and property definition, simplier syntax, no section keywords, no matching end identifier, case sensitive
    - Composition of Component Interfaces aka. Component type
      - Extends of multiple interfaces
      - Interface without category
      - Eliminates need for feature group type
    - Configuration Specification
      - Finalize design
      - Configuration assignment of subcomponents with implementation, features with classifier/type
      - Assign final property values to any model element
      - Annotate with bindings, annexes, flows
      - Configurations are composable
      - Parameterized configuration limits choice points
    - Jerome – Folding is very useful for graphics, reducing complexity of the specification.
    - Approved case-sensitive, all keywords are

        - Kevin - Can you support asymmetrical feature group (interface) connections, Peter – we have thought about it, could have it.
        -
        - Unified type system
            - Single type system for properties and data types
            - Records, lists, sets, maps, unions
            - International system of units
        - Properties
            -
        - Explicit deployment binding concept
        - Virtual platform support
            -
        - Flows
            - At the platform as well
            - Flow graphs – merge points
        - Nested component declarations
            - Define nested components without explicit classifier
        - Array support revision
            - Exposure if index dimensions/sizes in interface
        - Connections
            - Distinguish feature mappings
            - Reach down of connection declarations
                - Into named interfaces (aka feature groups)
                - Into subcomponent hiearchy
        - No more category refinement
            - Abstract component to other component
            - Abstract feature to other features
    - Abstract component, abstract feature – how will functional binding work. Kevin
    - Physical feature – how do we handle that. John Hattcliff
    - Github – a place for version 3 exists.
    - Modes – robotic project how to handle fault modes, reconfiguration, need more definition – Jerome.
-
- 1030-1100: break
-
- 1100-1230: AADL v3 Packages, imports, general syntax (Peter Feiler)
    - Allow property definitions and type definitions in packages
    - Allows local properties, also sharing
    - Use <dot> as separator instead of ::
        - Decision: Deferred

        - Peter, now prefer to keep :: because it has a different meaning after the package name, <dot> introduces ambiguity
            - Yes
        - Kevin - ::* could also be ::alias (for a package name)
            - Yes
        - No ordering would allow it to be different points but would apply to the whole package
        - Chris, Brian - Alias can resolve multiple imprted name conflicts
        - Qualify if local definition with same name (indicator to user, warning)
        - Decision: Yes including alias support
        - Qualified name references are not required to be in listed in import declaration
            - Decision yes (Jerome?, Alexey?)
        - Public and Private Sections in Packages
            - Proposal – eliminate public and private sctions in packages
            - Proposal – allow classifier definitions to be marked as **private**
            - Decision: Yes, it's a scoping decision, not whether it can not be seen by a human.
        - One package per file, name of file is the same as the name of the package. Nested packages are in one file. Decision: Yes
        - Document on model interchange – Pierre Dissaux, should consider.
        - Making AADL Case Sensitive – Yes, Keywords are all lower case. Allows for identifiers with mixed case (Yes)
        - Subcomponent Refers to Interface
- 1230-1400: Lunch
- 1400-1530: AADL v3- Interface Composition (Peter Feiler)
    - Interface Extension
        - Interface must have a component category or no category
        - Combining interfaces – could be logical and physical
        - Allows extension, can extend multiple interfaces with the same elements
    - Composition of Directional Interfaces
        - Can include a reverse interface
    - Composition of Named Interfaces
        - All the rules are the same as feature groups
    - Nested Interfaces
        - Same as nested feature groups
    - Subcomponent Refers to Interface
    - Composition of Interface Property values
    - Composition of Flows
    - Composition of Modes
        - Can only have in one interface.
        - Could have dual modes but we need someone to work on it.

- Annex Composition – can have only one- simplest rule best.
  - Then extends rules we have now.
- Feature Name Mapping for Connections
  - Inline mapping - WE CAN REACH DOWN MULTIPLE LEVELS WE WE DO NOW.
- Use as Aggregate Port
  - Interface elements interpreted as elements of aggregate data
  - L1: out aggregate Logical;
  - Multiple options, like a protocol, handshake
  - Could be an aggregate in a flow through hardware.
  - Should this be a port property or a protocol issue.
  - Decouple output rate of enclosing component
  - Later in the design you decide when it starts
  - For implementation architecture use virtual bus as a aggregator. Its binding indicates over what part of the HW flow it stays aggregated.
  - This is an undecided issue
-

- 1530-1600: Break
-
- 1600-1730: AADL v3 - Configurations (Feiler)
  - Architecture Design & Configuration
    - When you want to focus on a product, an instance of, completing as you go down, or if I want to give someone a subsystem I can give them some ability to tweek it. Parameterized configuration, I let you configure.
  - Evolution of System Design –
    - Addition of features, flows, etc.
    - Assignment of types/classifiers to existing features
    - Override with type extension or any type – change the type that flows through a system with checks along the way for type matching.
      - Decision
    - Assignment of property values
  - Component Implementation Extension – if it is only refinement it can work.
    - Addition of subcomponents, connections, etc.
    - Revision of existing subcomponents
      - Assign implementation for specified interface
      - Override existing implementation with extension
      - Override existing implementation with alternative
      - Assign interface extensions and their implementations
      - Enables us to get rid of matching rules
      - Supports partial specifications and checking

- Configuration of a System Design
  - Configuration Specification elaborates and annotates component hierarchy
  - Associated with an implementation/interface via extends
  - Configuration assignment assigns
    - K
    - K
  - Assign “final” property values with within existing component hierarchy
  - Specify bindings
  - Add flow specifications
- Configuration Assignment
  - Elaborate and annotate subcomponent substructure
    - Annotate substructure with “final” property values, bindings, annex subclauses
- Nested Configuration Assignment
- Assignment of Configuration Specifications
- Configuration of Property Values
- Composition of Configurations
  - Rules for configurations – interface must be the same, single extension linage for implementation, must be at the leaf
- Organizing by modes? Brian, Peter considering. Would work only with properties. What things are mode specific, annexes. Food for thought.
- Parameterized Configuration –
  - You as the configuration designer can set how deep
  - User can only use what you have provided
  - Configuration of subcomponents via configuration parameters only
    - Assignment of formal parameter to one or more subcomponents
  - No direct configuration assignment to subcomponents by user
  - Usage – supply parameter values
- Explicit Specification of Candidates
- Property Values as Parameters
  - Can do specific properties or can do just the values
- Complete Configuration
  - Everything has been configured, can I add anything
  - Users are able to add “missing annotations”
    - Additional flows, error model specification, property values
    - User can declare extensions
- Configuration Assignment Patterns
  - Replace vs replace all type of pattern
  - GPS *=> GPS.secure;
  - Can we do properties
    - #Period * => 20 ms;

            - Period for all elements within scope of associated implementation that require a period. (what about changing both thread periods, processor periods) Only those that have assigned properties. Does not need inherit.
        - Configuration Assignment Patterns
        - Generic Configuration Patterns
        - Is implementation specific pattern needed if we can do it with the generic
        - **Action**: Brian will provide product line capabilities of a custom version of MetaH. (PLE terms).
        - Peter – three approaches, attributes is one.

Tuesday, Feb 12 - Wernher von Braun Research Hall (Bldg 34), 2nd Floor, M50. Parking only in the lot on south side (Prkg W30).

- 0900-1030: AADL v3 Property definition & assignment (Wrage/Feiler)
    - Brian taking notes – insert here
    - Property Associations
        - # starts all property references
        - Can qualify with a package name pl#Size=>3;
    - Property Association in Annexes
        - ^Process[1].thread2@Failstop#Occurrence=>2.3e-5;
        - ^ escape to core model as context
        - @ enter same annex type as original
        - @(BA) enter specified annex: if we have annex specific properties in the annex rather than core (we may not need this)
        - [x] array index
        - We can have mode specific values 2.3e-5 in modes (m1), 2.4e-4 in modes (m2);
        - => {m1=>2.3, m2=> 2.4};
        - One value per mode for multiple modes
    - Property Values
        - Property values can be overridden many times in V2
        - Property definitions can be public or private
        - Override keyword, of a default value?
        - Scoped value assignment – if it is not assigned, this will be the value, scope of configuration, implementation, or interface
            - This gets rid of inherit
        - If no default values, then the model is intended to not have a value
        - You can configure default values in and out.
        - Forced override of default then allows a final change

    - Configuration Use
        -

- 1030-1100: break
- 1100-1230: AADL v3- Parametrized Configurations & patterns (Feiler)
- 1230-1400: Lunch
- 1400-1530: Security Annex (Dave Gluch)
    - Dave – will use resolute and ALISA, will use as an example, or appendix.
    - Threads would have access, information security level would be for data, process, processor, and device.
    - Will get out a draft in a couple of weeks.
    - **Action**- Dave will get the MILS tool (SEI) and take a look
    - Device – can have multiple outputs at different levels, how does this work with giving a device a security level. Resolve
    - Data is the key for understanding classification impact in the architecture
    - Time based classification – does not work with solid state memory (persistent)
    - Encrypted data can be carried by a person that does not have security clearance
    - TS data is passing through a public network because it is encrypted.
    - Encryption: a binary (boolean)
    - encryptionScheme
    - test if it is encrypted, then what is the value. The default should be non-encrypted.
    - There should be a key length property. Dave – it's included in Key related properties, pg
    - X509 properties – the source, where it came from
    - Key can have a life.
    - There will be new algorithms – 3 o4 every year.
    - Key Management – consider when it will expire, (generation, storage, Distribution control, destruction, Replacement). No battlefield management capability for keys. Trust boundaries. Cary – I don't know how you would verify it. Would not touch it.
    - X509 in Wikipedia
- 1530-1600: Break
- 1600-1730: Features, connections, bindings (Peter Feiler)
    - Abstract Features
        - No refinement into one of the other categories
        - No specific communication semantics
        - Can be directional
        - Action: Chris will send an STPA with AADL model, issues related to functions. Brendan would like to talk to Chris.
        - Brendan, Chris, Kevin, Kyle – please provide any input you might have on functions and Abstract features.
    - Question : Cyber Physical ports
        - Device is used for the physical

- Ports are to pick up the readings
- Abstract port – you need to be able to describe, or a physical port.
- Bring the architectural that will compose with the physical modeling?
- Peter – I put in an abstract port on a device, error model annex supports a propagation point, can we have that in the core language (John H)?
- Methodology issue, how will we do this (John).
- Do we have any guidance. Not that is in one place. Some work by Julien. Dominique also used abstract ports on devices.
- Understanding how these relate, not redo. How that information flows. How these link, interlink.
- Jerome – what I miss is the to tag a specific port to propagate.
- If we have a failure mode we need to
- Error model Interactions are the things that import this observation.
- **Action:** send Peter ideas related to functions, physical feature – (john) an abstract port may do as long as the property mechanism can support what we need. Need for guidance on use of abstract.

- Ports
  - Predictable received value – IPO semantics (received value not affected by new arrivals) – this is the best way to describe. BA will allow (Brian –this should be fixed)
  - Default send/receive timing
    - Completion/dispatch
    - Explicit service calls
      - Timing spec via property
      - Received value at time of call
  - Queuing – on the receiving port, this is sufficient, bus delays should be on the bus
  - Shared queue
    - Queue serviced by multiple receivers
- Event, Data, Event Data Ports
  - Event: no type, has receive queue
  - Event data: message type, has receive queue
  - Data: data type, Receive queue size of 1
  - Peter suggesting simplification – only port then with a data type if it's a data port. Also now a data port can dispatch.
  - Peter – we still have the connection types that indicate the proper real time semantics.
  - Jerome – case of the sporadic thread, Brian – case of data port to read A/D
  - **Action:** Peter - Verify real-time timing semantics preserved? (could discuss with Steve Vestal, Bruce concerned about preserving real time messaging semantics).

        - Provides (read|write|readwrite) access <data type>
            - Jerome – I like access since it indicates a copy, readwrite is not used on buses
            - Peter – not sure we need it given readwrite, will consider
        - Data Access and Data Components
            - Access connection to component itself. Decision: Yes
        - Other Access Features
            - Bus, Virtual Bus, subprogram, subprogram group
            - Bus, virtual bus: read, write (readwrite implicit of no read_write)
            - Do we need to indicate "bus access"
            - We need the feedback for the reader,allow in the case of data component since we have readwrite
        - Peter will consider direct connect to memory within processors, but there is always a protocol/internal bus.
    - Data access
        - Provides (read
    - Bus/Virtual Bus Access
    - Subprogram (group) access
    - Subprogram parameter
    - Physical?

Wednesday, Feb 13 - Wernher von Braun Research Hall (Bldg 34), 2nd Floor, M50. Parking only in the lot on south side (Prkg W30).

- 0900-1030: AADL v3- Type system (Wrage)
- 1030-1100: break
- 1100-1230: AADL v3 – Flows (Feiler)
- 1230-1400: Lunch - start at 1300 at UAH. Bruce pick up Jerome
    - Connections - continued
        - Connections between subcomponents
            - Information flow (out->in, provides read->requires read, requires write-> provides write)
            - Access (provides->requires)
            - Decision Ports in, out, inout, access the same, Subprogram access control flow (requires->provides)
            - Decision – use in out and inout, do not make inout a default.
            - Names interaces – can have both in and out, which would <->
                - -> implies all interface elements same direction
                - Reverse used to create the mating
            - Feature Delegation/mappings

        - Delegate with the same symbol as connection with direction
    - Reach down of Connections
    - Reach down into Component Hierarchy – impacts substitution, considering what to do.
    - Feature, Connections and Modes
        - V2.2 Issue #24 is satisfied by the standard already
            - Connection is only active if both endpoints are active; no need to explicitly specify in modes for connection (already in V2.2) Decision keep
            - Connection not active even though endpoints are active: need in modes on connection (already in v2.2). Needed? Yes
            - Mode specific visibility of feature – need to play with, not sure how to address.
    - Flow Specifications and Sequences
        - No change to flow specification and implementation.
        - Additional flexibility Component.flowspec sequence only
        - Keep in for now, Peter has implemented. Simplifies composition.
    - Flow Graphs
        - We could do multiple inputs to a flow and then get
- 1400-1530: AADL V2.2 errata
    - Allow renames of property sets#32
        - Packages can be renamed. Properties should be too.
        - Property sets were oriented to forms of analysis and had short names.
        - It would require changes to tools, should wait for version 3 where it will be handled.
        - close
    - Flow specification des only allows one level of reference into nested feature groups #29
        - Connections can reach into nested feature groups multiple levels.
        - Is something we could not do before.
        - Accepted
        - Reballot
    - Mode transitions cannot reference a feature in a feature array #30
        - It also cannot reference an array element in a subcomponent array.
        - Accepted, it appears to be an oversight.
    - Add ability to deprecate #31
        - Useful when we migrate properties from e.g. “SEI” to pre-declared properties
        - Considered for version #3.
    -
- 1530-1600: Break

- 1600-1730: FACE Annex – Preliminary Ballot Review (Tyler Smith)
  - Describes translating from FACE to AADL and AADL to FACE.
  - ANSYS plans to implement the annex.
  - UoP and UoC are the same, UoP emphasizes the portability, C the conformance.
  - **Action:** Bruce send FACE annex to Kevin GE, Dominique, Jeff. Kyle, John Hatcliff, Mark . Brown, Chris Carhill.
  - Take out grey boxes and shorten variable names.
  -
- 1730-1800: Model Based Testing with AADL (Shawn Kline)
  - Automated Test and RE-Test (ATRT)
  - IDT – using a model for automated testing
  - Working with the SEI
  - Started with SysML testing and extending into AADL
  - AADL to ATRT interface
  - Do work with the Navy, List of platforms that we are working on
  - TRL 8/9
  - Significantly reduce the time and manpower required for testing and in particular regression testing
  - Near real time turn around on testing results to support fielded system test.
  - Expansive permutation testing
  - Has to be non-intrusive
  - Supports fielded systems and systems under development
  - Technical Approach – blueprints in AADL, drives the tests
  - Automate message chains to drive the system, raw events,
  - You would modify the model to provide information that relates to the recorded data.
  - End to End flow (of data, events, or both)
  - Latency (between/throught logical components, execution of threads)
  - Modes attached to threads
  - Communication bus bandwidth
  - Power bus capacity
  - Resource utilization of bound loads (memory, CPU)
  - Error flow (ensure error types are handled/mitigated)
  - Functional hazard analysis
  - Fault tree analsysis
  - These are what we add through AADL, SysML is used for the more functional aspect.
  - PEO Missile & Space sponsored.
  -

Thursday, Feb 14 Tool Demonstration Day – See parking map for all day parking and location of Charger Union Theater (Bldg 19) and Exhibit Hall (Bldg 8a). Morning presentations at Theater, Afternoon demonstrations at Exhibit Hall

- 0730 Presenter arrival for set up for presentations at theater (Bldg 19)
- 0755-0800 – Welcome and Announcements
- 0800-0820 – Keynote Address
- 0820-0830 - SBIR Contributions - Dawn Gratz
- 0830-0845 – Overview
- 0845-0900 - Vendor 1 - Software Engineering Institute (SEI)
- 0900-0915 - Vendor 2 - Ellidiss
- 0915-0930 - Vendor 3 – ANSYS
- 0930-0945 - Vendor 4 - Adventium Labs
- 0945-1000 - Vendor 5 - Kansas State University
- 1000-1030 - Break
- 1030-1045 - Vendor 6 – BLESS
- 1045-1100 - Vendor 7 - University of Alabama in Huntsville
- 1100-1115 - Vendor 8 – Georgia Tech
- 1115-1130 - Vendor 9 – WW Technology Group (WWTG)
- 1130-1145 - Vendor 10 – Integrated Defense Technologies (IDT)
- 1145-1200 - Vendor 11 – Physical Optics Corporation (POC)
- 1200-1215 – Vendor 12 - Telecom Paristech
- 1215-1230 – Vender 13 - ISAE-SUPARO

- 1230-1400: Lunch and Tool Demo Table Setup
    - Proceed to Conference Training Center (Bldg 8), with internal Exhibit Hall (Bldg 8a) and Cafe, Parking behind the Executive Plaza. See Parking map.
- 1400-1800: Visit Tool Demonstration Tables - Trade Show Style, Exhibit Hall (Bldg 8a), Parking behind the Executive Plaza. See Parking map.

Friday, Feb 15 – Parking at Executive Plaza behind buildings, as on the 14th.

- 0800-1200: Exhibit Hall open for additional arranged demos/meetings. No formal schedule. This day is optional for your participation depending on your needs for further meetings.

Webex Meeting information:

Monday and Tuesday

AS-2C AADL Committee Meeting
Every day, from Monday, February 11, 2019, to Tuesday, February 12, 2019
8:30 am | Eastern Standard Time (New York, GMT-05:00) | 10 hrs

JOIN WEBEX MEETING
https://sae.webex.com/sae/j.php?MTID=m54a6cdc232972228c7ecb72e501dfb08
Meeting number: 629 227 012

Wednesday

AS-2C AADL Committee Meeting
Wednesday, February 13, 2019
8:30 am | Eastern Standard Time (New York, GMT-05:00) | 10 hrs

JOIN WEBEX MEETING
https://sae.webex.com/sae/j.php?MTID=m135340bf7ceb144af46a1eb2b8da0ffc
Meeting number: 620 970 642

JOIN BY PHONE
1-866-469-3239 Call-in toll-free number (US/Canada)
Tap here to call (mobile phones only, hosts not supported): tel:1-866-469-3239,,*01*620970642%23%23*01*

1-650-429-3300 Call-in toll number (US/Canada)
Tap here to call (mobile phones only, hosts not supported): tel:%2B1-650-429-3300,,*01*620970642%23%23*01*
Access code: 620 970 642

Global call-in numbers:
https://sae.webex.com/sae/globalcallin.php?serviceType=MC&ED=757428717&tollFree=1

Toll-free dialing restrictions:
https://www.webex.com/pdf/tollfree_restrictions.pdf

Add this meeting to your calendar (Cannot add from mobile devices):
https://sae.webex.com/sae/j.php?MTID=m2a3992c197ae855fbbf3057147e268b8

Unable to join the meeting? Contact support here:

https://sae.webex.com/sae/mc

IMPORTANT NOTICE: Please note that this Webex service allows audio and other information sent during the session to be recorded, which may be discoverable in a legal matter. By joining this session, you automatically consent to such recordings. If you do not consent to being recorded, discuss your concerns with the host or do not join the session.