AADL standards meeting Jan 29-Feb1, 2018

* Location Toulouse, France.
  + Meeting information
    - At the IRIT, Institut de Recherche en Informatique de Toulouse – CNRS
    - Rue Charles Camichel, 31000, Toulouse
  + <http://maps.google.com/maps?q=Rue+Charles+Camichel,+31000+Toulouse,+France&z=16>
  + Please send Marc Pantel, your name, nationality, current residence and email so he will have you on the list to enter the campus. Marc Pantel <Marc.Pantel@enseeiht.fr>
  + Time differences Eastern Time to Toulouse Time 6 hours, Central – 7 hours.

Bruce Lewis, Eric Jenn, Jerome Hugues, Yvon Kermarrec, Denis Buzdalov, Alexey Khoroshilov, Pierre Dissaux, Thierry LeSergent, Peter Feiler, Brian Larson (Remote), Frank Singhoff, Denis Buzdalov, Joe Seibel (Remote), Philip Alldredge (Remote), Rob Edman (Remote), David Alexander, Tyler Smith (Remote), Alex Boydston (Remote), Charlie Payne (Remote), Dave Gluch (Remote), Steve Vestal (Remote), Cary Pool, Eugene Vasserman (remote), Lutz Wrage (remote), Mike Whalen (remote), Raphael Faudol

# Monday, Jan 29

* 0900-1000: AADL standardization committee news + action items (Bruce Lewis)
  + 14-17 May at the SEI in Pittsburgh. Week planned for Baltimore not good for Jerome, Peter, or Alexey.
  + SAE Nov 6-8 London, may be interest in papers but we need to focus on standard.
  + Feb 2019, Southern US would be nice, Jerome does not like Florida but Key West OK.
  + V3 Jun 2019 target date is good for Peter as a working estimate.
  + Upgrade to 2A would be 2B. Others would be 3A, 4A, etc. So Network Annex 3A, Security 4A.
  + David Alexander – On core and annexes, can be reasonable for the set and be on a subscription basis so updates automatically provided.
  + Requirements Annex should be an Assurance Annex. Will likely need to changed, not just added to for Assurance.
  + Pierre D – we need a workflow for the AADL from requirements with assurance. This should be provided as a guide for users. AIE?
  + Denis – a volume 0, would be the workflow document.
  + Bruce – I like this.
  + Jerome – take a look at the original requirements document. Extend it if necessary. Not to be system engineering.
  + Brian – we need requirements for the requirements annex to bound it to what we want it to do. The system specification.
  + Thierry – You can combine Hybrid Annex and Synchronous, could also do it just through FMI since it simulates the input of continuous physical systems.
* 1000-1045: AADL v3 roadmap review (Peter Feiler)
  + Compositional Interfaces –
  + Configuration and Choice points –
  + General binding concept – Open issues: binding & Arrays
  + Nested processors – settled
  + Unification of type systems and expression languages
  + Property sublanguages
  + Flow trees and graphs – not this time but will work out a description to move forward
  + More Candidates – interrupt handler (Jerome)
  + Data aggregation via protocol
  + Data mapping via new binding/mapping concept
  + How much of a clean-up, wordiness? Should we refresh the syntax?
  + Pierre – must be consistent and stable, it’s much more important.
  + Denis – much more liberal on level of change, say less and mean more
  + Thierry - Textual is important for the exchange across the tools. But we use the graphical.
  + Bruce – we have many people learning the AADL now, should not change it more than we need to. Complexity in concepts is also an issue.
  + Some comments - Type inference - Relax requirement for component type separate from the implementation.
  + Eric Jenn – separate teams looking at code do not want type inference. The code needs to be understood. Concerned about correctness more than wordiness.
  + Bruce – we want to use this as a communication mechanism across teams.
  + Pierre –where are the rules for type inference? They would have to be very clear, we don’t have them.
  + Jerome – we need to understand as we read that a subcomponent is a thread.
  + Peter – architecture description languages have to have a strong concept of containment but we want to be able to move things with ease.
  + Eric – workflow is typically from the top down, you work through these levels naturally.
  + Thread groups is a top down approach. It’s a process orient decision. Common rates and safety properties. Containment is for interface enforcement.
  + Containment means interface enforcement.
  + Pierre – don’t confuse language with methodology. You can choose how you group.
  + Eric - Type inference must be very limited, it should not do the engineering aspect of the job.
  + Readability and avoiding mistakes in expression are reasons to not to do type inference.
  + Renames, use clauses. We separate the concepts. Typically we use it for renaming,
  + Eric - Contracts are very important. We should enforce the contractual, not ease of use. Dependency is important, because we must review it when we make changes.
  + Denis – I agree with the need for contracts. But packages do not express dependency clearly. Often we use more than we need.
  + Peter – Perhaps we need a grouping, a list of packages that we are dependent on.
  + Pierre – our current state is not bad at all, let’s not break things, we must be able to address the user who is starting a new project. Modularity is quite good. We do not need to add methodology to the language but to the tools. To save time.
  + Bruce – with our real time and runtime orientation we have a good approach for virtual integration analysis. We should strengthen our use for analysis and formal methods. But there is still much to build on for analysis purposes. We should be careful not to break the value of the language for these purposes.
* 1045-1100: break
* 1100-1200: AADL v3 - Categories of features (port, physical features) (Peter Feiler)
  + Features in AADL – Some clean-up
  + Data, event data, event port, - but no way to specify just a port.
  + Generic feature – can be refined to any other feature categories (was it useful, usually know but with ports useful), usable as generic interaction point with type indicating need for matching.
  + In EMV2 – propagation point (observable feature)
  + Coming in V3 binding points (interaction across layers)
  + Feature – generic – directional, non-directional (bi-directional?)
  + Need for ability to refine feature category?
  + Jerome – a generic feature until I define the component I am talking to. May be different type of port.
  + Deferred decision as to whether port based or by shared data
  + Ports as features
    - Currently is it an event port, data port, etc.
    - Should we have a more abstract port that is data or event that is refined?
    - Aggregate port and protocol for buffering on the output (to be looked at later)
    - **Action:** Peter will look at use cases on default semantics on ports provided by Denis related to 653 ports, impacting also runtime services.
    - Eric: what about continuous? Peter: Physical Features
    - Physical Features – Devices represent the interface to the physical world. Digital sampling of physical world measures is represented by ports or events triggering a device. The continuous time physical behavior may be modeled in Modelica, thus, a reference to this model may be sufficient. Alternatively, a property can specify appropriate equations.
    - Physical features – continuous time physical characteristic of a component – may be directional (flow) or non-directional (temperature)
    - Type associated with feature represents physical characteristic of interest
    - Proposal use generic feature construct (in/out)? Feature <physical type>?;
    - Type definition includes measure type with unit and optionally dynamic characterization of that measure (diff equation)
    - Jerome – FMI will have a description language for continuous and discrete, we should look at it. Peter – agreed. To be released in first half of 2018. **Action:** Jerome to send FMI description language to Peter for review when available.
* 1200-1230: ANSYS SCADE AADL (Thierry LeSergent)
  + SCADE Architect – SysML Engineering tool, extensible to support Domain Specific Languages via “configurator”
  + SCADE Suite – generator for safety critical embedded software for control systems.
  + SCADE AADL – released just last week.
  + Will manipulate with graphics as you do for SysML
  + AADL is quite clear about what it is. This is helpful.
  + Define in the Domain specific modeler
  + Configuration tool will not allow creating thread inside the processor, so follows AADL semantics
  + SCADE Architect Configurations – ICDs for ARINC 653, 429, 664, CAN, and now model editing for FACE, and now AADL. Static analysis is best in AADL. FACE compliance. Independent Meta models. These independent models need to be used together. That is what we are working on now.
  + Exchange models with textual AADL.
  + We have simplified AADL to make it easier to express by directly modeling the AADL component, type and implementation.
  + From SCADE Architect, Configuration of AADL, AADL property sets we exchange with Analysis tools for Safety, Security, Timing …. We do not have AADL analysis tools. We only provide an editor.
  + Most of the time you have one type and one implementation per component in any case.
  + Replica is the instantiation of the component into its instance. Properties can be different. Same for bindings. If you over-ride it, that is the option at its instance use.
  + Thread or process.
  + Graphical information is not exchanged.
  + Denis – what if you have modes, impact on properties and connections. Thierry – will have to check.
  + For primitive types we define a library and make the conversion to the AADL types.
  + Do you support the data model annex. Base types and more complex types? Thierry – will check.
  + Does SCADE suite support time? No. You only define a function.
  + SCADE suite reads the input, executes, output
  + SCADE does not generate the code to launch the execution. We only generate for FACE.
  + SCADE suite event and event data ports? Yes.
  + SCADE suite function – could sync with subprogram, thread or device. (Fix slide for presentation tomorrow.)
  + In AADL you can read with an event data port multiple data items. How would this work with SCADE suite? Thierry there is a way, both inside and outside SCADE Suite.
  + If we make an AADL simulator, we would need to manage this.
  + Annex subclauses – how would I use EMV2, except the data annex. When it is requested, we would do it. Brian, it’s an important part of analysis. Error annex is very important, 653 annex also important.
  + In the same project you can have the SysML, the FACE, and the AADL. But you don’t put the SysML in AADL.
  + You can import and export from Excel.
  + If the constraint is not part of the metamodel then SCADE would not be able to check its correctness. But you could check with an external checker. (IDE)
  + It took several hours to draw the Julien model nicely. Drawing is not easy. Exported to OSATE and did latency analysis.
  + Next feature under study, Multi-view AADL-FACE. To synchronize the two. Combination of transformation and verification between semantics.
  + SCADE solution for AADL -
* 1230-1400: Lunch
* 1400-1530: AADLv3- Configurations (minimize language changes) (Peter Feiler)
* 1530-1600: FACE to AADL - Alternative TSS Models (Robert Edman, Joe Seibel) (9:30AM Eastern)
  + Adventium – Both AADL and FACE are compatible, with strong benefits to using them together. Need the analysis of AADL with the standard data and API’s of FACE.
  + BALSA is
  + FACE verification is focused on data and API and interoperability through that.
  + Adding AADL properties and AADL specification provides the analysis basis
  + Reusable components are called Units of Portability
  + Applications from different vendors running on the same system.
  + Integration model in FACE 3.0 allows for code generation
  + ARINC 653 and ways for the applications to communicate.
  + Two independent paths of development make them compatible.
  + TSS is the Transport Services, all the UoPs talk to the TSS to communicate.
  + FACE Portable Components
  + FACE: Data model of a system without physical properties of a system
    - API, Data types, direct connections
  + AADL: Formal methods and model based analysis of Cyber-Physical Systems.
  + Notional translation
    - TSS routing to port connections
    - UoPs to thread group
    - Message ports to port
    - Views (Message types to data types
    - … to …
    - AADL Behavior Annex
    - AADL …
    - Stock AADL with data and mapping
    - UUID’s are aadlstring applies to (all); But we could use an AADL type for this better).
    - BALSA is a desktop computer application. It reads GPS coordinates and packages them for output.
    - A FACE 2.1 does not provide the TSS routing, see example. The TSS is the challenging magic part of FACE. TSS is the hard part.
    - It could be modeled as an abstract, a virtual bus, but works as API calls, so subroutine calls. If you are a UoP creator then an abstract is ok. If you are a system integrator, then a process or a virtual bus.
    - So the UoP is a thread group, the TSS could be an abstract, or more deeply a process or a virtual bus.
    - Binding of the connections to the virtual bus provides the routing table. Then you bind the virtual bus to the hardware. In OCARINA we use the binding to generate the table. Think of it as a virtual channel. Peter - A single connection can generate multiple connections, we need to look at how we handle that.
    - We would like to provide a draft by April, if you would like an earlier draft we can provide it by NDA.
* 1600-1630: Break
* 1630-1745: Network Annex Draft Review and Update (Alexey Khoroshilov, Brendan Hall)
  + Don’t have a draft ready for ballot yet, but we have it on the network
  + Have mark down and syntactic highlighting now
  + We have the current draft on the network, see slides
  + Make file allows updating and re-generating
  + Content Open questions:
    - **Action:** Peter -Data volume used but should we use data rate. Peter will check. Data Rate unit type.
    - Port\_Speed to define all hardware ports in one specification. Could be a minimal required or max or a declarative value. How it relates to actual speed is not quite clear. Enumeration can be used since it’s not the actual speed but the speed that the port is rated for. A range is just one of several approaches. Should we handle for each separately or as a more generic approach? Network could support all three, depending on mode of operation. Or a property constant. Under version 3.0 a stronger solution can be provided but this is for 2.2.
    - Could use property constant or property value. Max\_supported\_VLS is now defined as a property (value).
    - How to answer to Steve? What do you mean by this. **Action:** Alexey will email Steve to see what he intends for Max\_supported\_VLS. Is it budget or design exploration, or… You don’t want to overload a property. Each meaning should have a property. Or in context x it means …
    - All elements have this property. Or a more flexible way to model. If possible use the library element. Is it reasonable? I will add a property for each element in the library, hardware …
    - Draft Ballot I hope to support for May.
    - I have an example of TTE with a property set but we need to build a small example. Let’s see if we can get feedback from Eric Jenn. He has a project that could use. TTE supports time triggered, non-deterministic and supports AFDX like but slightly different. From a model perspective it’s the same.
    - **Action:** Alexey - Contact Eric Jenn and ask him to review what we have, see if it is useful for him, let him know it would take some time to build an example of TTE but we have properties defined. See if they can help with the example but do not fully rely on them.
    - Dinner at 7:30. Frank will send address so we can meet.
* 1745-1800: Proposed SEI/lab-STICC project (Frank Singhoff)
  + Head of lab willing to seek funding for.
  + Numerous AADL tools
  + Need to build tool chains
  + Need to show tool interoperability shone on several projects.
  + Need a first experiment in which they can succeed.
  + Multiple tools but
  + No shared library of reusable AADL components prevents from creating a synergy between various teams, tools, and projects.
  + Output
    - AADL cookbook
    - Materials available in public repository, could be AADL.info, or?
    - Guidelines to help users to start multiple analysis of AADL models (safety, security, scheduling, …)
    - Scope of use.
  + AADL committee would be the steering committee
  + Frank would start this year, funding to come the next year.
  + Scope of interaction, model, analyses, execution platform, documentation.
  + Anyone from AS2C is welcome
  + Project proposal available on request for each phase.
  + Yves – would the junior researcher have the AADL skills to do this.
  + Yves – who would provide the models and keep them up.
  + Yves – How would you order it?
  + Peter – what application domain are you thinking of – do you have someone who is from the Aviation domain.
  + Peter – we have models that are updated, but also those not updated.
  + OSATE, MASIF, ELLIDISS
  + You need someone who understands the analyses and understands AADL

# Tuesday, Jan 30

* 0900-1030: AADL v3 Bindings & Resources (closure) (Peter Feiler)
  + Configuration – out of order now
    - Configuration of previously configured subcomponent
    - Provides extends to reach down
    - Provides for replacing a previously assigned implementation by an extension of the implementation.
    - Configuration of Property Values – finalizing a set of property values via extends
    - Composition of configurations – combine structural configuration with different extensions that represent different annotations/”data sets” , ensure that all model element references being composed remain valid.
    - Parameterized Configuration – Explicit specification of all choice points, only the choice points can be configured by users. No direct external configuration of elements inside. Explicit specification of where choice points are used.
    - Parameter Match and Replace within a scope- specific to the context where a pattern is specified. **Action:** Denis to provide Peter some use cases for Parameter Match and Replace. Can you reuse the pattern? You need to be explicit about the effect of the extension.
    - Complete Configuration – Finalizing an extension. Can you add an error annex to it? Yes, it is external to the component itself. But if there is already an error model you could not. Don’t know the best answer yet. Need reasonable rules. You can add things that don’t exist. No overrides. New properties for new analyses would be ok. New things need to not contradict.
    - Configuration of Annex Subclauses – Adding in annex specifications, added in via with. (Slide 16)
    - Composition of Flow Configurations – Adding in end to end flows. End to end flows may be declared in a separate classifier extension.
    - Alignment with Core V2 Syntax.
      * Configuration vs. implementation
      * Question of granularity of final, down to subcomponents?
      * Configuration parameter vs. prototype – Configuration parameters specifies mapping of parameter into the model across multiple levels. Allows for parameterization of existing model without modification.
      * Can stay with “applies to” or without “applies to”
    - If you parameterize then everything else is final.
    - Could consider two levels of change
      * Not changing what is defined but can add, not changing or adding.
  + Binding and Resources
    - Instead of using properties for binding, we use allocations.
    - Resources are providing by binding
    - Directional features to be used as source or target of binding
    - Predefined binding points
    - There are consistency rules and here are the rules for understanding use of resources.
    - Understanding if the resource is exhausted. Requires 200 MIPs.
    - Verification plan vs. analysis of verification, sometimes you are allocating and sometimes it’s the output of verification. Spec value or actual. Do we have separate properties?
    - Binding of from source to target hierarchies, multiple bindings for same configuration.
    - May or may not go with a pattern language for binding, default is inherit.
* 1030-1100: break
* 1100-1130: AADL v3 -Unified Type System & Units (closure) (Peter Feiler)
* 1130-1230: AADLv3 - Compositional Interfaces (closure) (Peter Feiler)
  + Composition of Directional Interfaces
    - Named interfaces, should we have the keyword interface?
    - Interest in using # to indicate properties and put them anywhere.
  + Annex Composition
    - Configuration of annex specification into an AADL model – see configuration discussion
    - Composition of annexes from different interfaces
  + Type System
    - Types – Data types, property types, constraint language variables types
    - Replaces data types with the new type system
    - Do we need to support specific aggregates, those that we use for architecture, or ability to declare aggregates?
    - View as a protocol or a pragma?
    - Expression of types with data representation, bounds.
* 1230-1400: Lunch
* 1400-1430: Behavior Annex Errata (Pierre Dissaux)
  + **Action**: Peter will open a Git hub repository for the Behavior annex where he also has V3. Done!
* 1430-1500: AADL Core Errata (Peter Feiler)
  + What if we don’t have actual input for prototypes? In some cases it is ok. A user could invoke a check for completeness. For instance, in some software architectures you do not have a processor, at some point, with analysis. Should be in user guidance for use of AADL but not in the standard.
  + Inconsistent relations between devices or processors and subprograms. A device driver may need to call it’s driver. I could be a thread in the application. Agreed. Useful to require or provide program access from/to a device, processor, or virtual processor. Is there a need to offer subprogram access for buses and virtual bus (e.g. represent a API).
  + Reaching into feature groups. For the moment we don’t have deep reference. **Action**: Peter will look into whether we have ability to reach down. We allowed one level. Done!
  + Steve’s question on Priority. **Action:** Peter will contact Steve to discover purpose of priority extension.
* 1500-1600: AADL OSATE Graphics Tutorial and Update (Philip Alldredge) (8:00 AM Central)
  + Major Enhancements
    - Diagram Configuration – subset of the architecture or multiple diagrams
    - Appearance Customization
    - Layout improvements – open source eclipse, user interface
    - Properties view (nightly)
    - Flexible Editing Rules (nightly)
    - Predefined Diagram types (Next release) (what shows up in the diagram, starting point and can be edited)
    - Demonstrated creating an architecture. Recorded the Demo.
* 1600-1730: Security Annex (Dave Gluch) (10:00 AM Eastern)
  + No comparable annex, more of an guidance like the ARINC653 annex
  + Supplemental Elements
    - ALISA, Resolute, EMV2, Security Architecture Modeling Guidelines, Transport Aircraft Example System, others like MILS, attack tree
    - Verification of security policies – like requirements, multiple levels, will demonstrate with ALISA
    - Ensure there are mechanisms that enforce security policies/requirements
    - Security Policies and Requirements
      * High Level Security Policies (organizational)
        + Employ stakeholders and relevant organizations – optional
        + Naming to identify (TransportAircraftSecurityPolicies.goals)
      * Low Level Security Policies
      * Information/Data Protection
        + Cryptography a& Encryption – property sets
      * Security Resolute Library
      * There is also a need to know – do we what to reflect that as well. Authorization is need to know.
      * Systems themselves need to have separation as well, systems have roles, can keep from communicating.
      * Detecting intrusion, we have to keep intrusion
      * We record a public key and validate the private exists?
      * Propagation of effects, distribution pattern of keys
      * Security Architectures (Modeling) MILS, DMILS
      * Vulnerability Modeling and Analysis – Fault tolerance and fault recovery. What is the effect of an error, on the system from a security perspective? Which part of the system can be reconfigured? How do we isolate the system?
      * Eugene - We can model what the adversary can do, in EMV2. We can formally define this. Based on security properties of a change. We can provide information through Adventium to show what to do, when they can release it. Network adversary model on the architecture. Not a lot of modifications required to EMV2. Pierre - Behavior annex can also be used to simulate the effect of an attack.
      * Threat/Attack Modeling -
* 1730-1800: STOOD and AADL Inspector (Pierre Dissaux)
  + Recorded session.
  + Address merging models, SysML, Capella, MARTE – we do it with a template since they are not well defined in terms of usage. Capella is more firm than SysML.
  + HOOD gives us a top down design methodology, we apply it to AADL.
  + LMP is a set of prolog libraries to process the AADL model. We also parse Ada, C, XML, XMI, SDL. So this helps us merge approaches.
  + LMP designer helps us develop the processing rules.
  + Graphical front end GMP, simple graphics to express behavior annex from Hood but it could be annotated.
  + Graphical model for the AADL instance model – we work on the instance like ANSYS. What you see is what you get.
  + We generate the AADL Declarative model (textual AADL) for inter-operability.
  + Requirements traceability through Rectify.
  + Hood works on a subsystem of the system, allowing easy work between teams. One high level component and down the tree. We apply this with the AADL.
  + STOOD 5.5 new features – AADL export and import improved, now trying to give high level representations, do a BA template. IDEF (0) like Diagram with call sequences, graphical properties preserved in the AADL model to preserve layout. Reqtify 2018 import/export. Reqtify was originally developed by Airbus for STOOD.
  + LMP helps us merge new analysis tools.
  + OCARINA compiles AADL precisely, AADL Inspector, less precisely.
  + Simulator supports a subset of the Behavior Annex.
  + Simulator only works with a single time clock, but now can run faster when nothing happens. Skip to the next significant time tick. (next version)
  + Simulator can input a scenario, or a VCD trace, if you have timed input you must keep time constant. We need absolute time in the scenario. We can have random input. Simulator can provide a random number but of course not good for regression testing.
  + Capella is available and TAHLES is really promoting. Training, etc. CAPELLA is easier than generic SysML, which is just a block.
  + Multi-core in Cheddar, next input could be better but not settled. CACHE timing yet to come. Need a user to work with on what is needed. Cheddar has a simple model of hardware but not cycle accurate. For that we do system C. Cache modeling requires scheduling. No one solution. Ellidiss takes BA to create input for Cheddar, but more work needs to be done to fully map to the input of Cheddar. Tuning to get more out of the tool.
  + Github.com/SAEAADL

# Wednesday, Jan 31

* 0900-1030: AADL v3 - Property Language Revisions (Peter Feiler)
  + Property Language
    - No more aadlinteger, …
    - Lists & sets for properties
    - Types that would be acceptable in the BA
    - Types that are currently in the language
    - Integration of proposed Units system(ISO, SysML)
    - Union of types? – for application types useful, do we want for property language? Compute entry point – Jerome is this or that different, execution time estimated, measured and values. Includes a definition with the value.
    - Record: map can represent record
      * Require all fields?
      * Require naming of fields in assignment or assume ordering?
    - Property set Name path (dot) for property sets or just in a package?
    - Nested property sets needed?
    - Distinguish between types to be used in model specification (BA, constraint, property values ) and types used as application data
    - Properties of flows useful (Denis), do not represent flows with a property.
    - A list is a sequence, which is fine, but also a bag, (need rationale for adding each to properties) map – Denis.
    - Brian – Explicit types: on value assignment same syntax for lists/sets of values. It catches my errors.
    - Question reasonable for the user vs very fancy in what can be expressed.
  + Property Language
  + Property Definition
    - Identify scope of application (applies to) – should it be for categories only?
    - No need to list enclosing categories for inherit for properties?
    - Partition period and thread period.
    - Merging package and property set?
    - Need a better way to limit scope of properties below the category.
    - Scoped defaults
  + Property Association
    - Need to look at different styles of expression. Applies\_to provides lists.
  + Property Values
    - Value in terms of another property: Needed?
  + Property Applicability
    - “Stereo” type or profile to associate a set of property definitions to a component, component can have multiple profiles/stereo types.
    - Stereo type for periodic threads? Which properties must have values. “View specific values.”
* 1030-1100: break
* 1100-1230: AADL EMV2 Errata (Peter Feiler)
  + How do we version management from these.
  + Officially published set of Errata, vs. re-ballot of the standard
  + Name Path Consistency – Useful to change but not critical.
    - Inn EMV2 “escape” to core. [^sys.a.b.c@ep.et](mailto:%5esys.a.b.c@ep.et) escape character and @ without annex name. Should use {} as we do for propagation constraints. Et is a property, so Ep{et}#Occurrence=>2.5
    - Users should be able to use what is in the standard until it’s changed in the standard. Post in errata on the standard. Support both. Do document update on next major release of AADL in a cleanup of syntax.
  + Error Type Library Name – currently error type library inherits the name of the package. Can do backward compatible. Proposal – Provide explicit name for type library. Allows for multiple type libraries in the same error library declaration. Useful enhancement, but not critical. I have features where I have not declared it as a propagation point, but now I’m concerned about it and want to declare a propagation. Yes, either unnamed in package, or named only in package, not both in the same package. Should support both until a cleanup of the syntax.
  + Error Propagation Declaration – Considered.
  + Practical Set of Operators – not yet published.
  + Propagation Paths – need some use cases to consider.
* 1230-1400: Lunch
* 1400-1530: Approaches to FMI with AADL Demo, Final Brief (Jerome Hugues, Raphael Faudou)
  + FMI-AADL Study
    - AADL has multiple capabilities for analysis but not for physical modeling, this way we can connect with the physical models.
    - We use FMI to interface to these capabilities, to model continuous time, physical
    - We have several extensions – now WP3, we presented WP1 and WP2 in the past. Now this is exporting AADL model into a FMU.
    - Very interesting option by providing more precise simulation of AADL.
    - Causality of actions must be preserved.
    - ERTSS’18 paper available. Jerome will send.
    - We are missing access to the FMI plug-ins.
    - Mid –Feb should have.
    - SSP System Structure and Parameterization – allows the interconnect of the FMU’s to help keep them consistent. Struggling with initialization, configuration and keep the output. Unfortunately not available until April or May. XML so need GUI.
    - WP3 – Define the interface of the component to export, a process. Solver is to run the model and provide data. Will run the simulation for a period of time.
    - Oak Ridge – discrete event simulation. Provide a way to advance time. Machine emulator, execute binary. X86, Spark, etc. Can be used to run the O/S including RTEMS. Supports scenario 2.
    - Will work on a purely periodic application. FMI 2.0 is periodic, FMI 2.1 is new. Supports events.
    - Brian – ISOSOLES, with Adventium, we are also working QEMU and AADL to simulate. We should get together.
    - QEMU –
    - We could coordinate with Adventium to support another operating system.
    - QEMU time is controlled by FMU: run/play
    - We can do functional testing and inspect events. Replay is very valuable. You can check how your operating system is behaving so very interesting to test the operating system.
    - ANSYS is updating to 2.0. That will help.
    - SysML side start with, then build FMU for physical part, build AADL then bring back. Can allow aligning the work flow. Could do early prototypes generated from AADL. Earlier you can work with the model the better for finding issues. Simulations bring elements together. However, timing is a big issue to keep co-simulation working.
    - Alexey on POK. In two years we will release the version we have now of POK. Then we will have production quality code for our customers. Supports many processors now. We will use with our students up to then.
    - We use AADL when we do real time simulation and have FMU’s that can keep up.
    - We would use AADL as an FMU when we want to time control it.
    - We use OCARINA as is, a few make file changes, so we get the real simulation of the system running on the O/S emulating the processor.
* 1530-1600: Break
* 1600-1700: AADL Runtime Updates (Jerome Hugues, Etienne Borde, Brian Larson) (9:00 AM Central)
  + Mike Whalen – HACMS, Air Team AADL was the integration technology, both the commercial quad copter and the military Little Bird
  + Great models but had to have trusted build, we had to build to what we designed, so we had code generation from AADL. All the air vehicles and ground vehicles, $100M build with AADL.
  + Ensure fidelity between models and system image – to avoid mismatch, etc. Many, many ways to shoot yourself in the foot as you hand build.
  + eChronos, CAmkES/seL4, VxWorks, Linux all used with AADL generation.
  + Often working with external code, so had to add the ability to call. According to the Code Generation Annex.
  + Single command build to build all the platforms. Run the analysis. Continuous Integration server called TRAVIS. We could rebuild the system with new changes within an hour with confidence.
  + No high-assurance pedigree, except proofs on some connectors, connector components for data/event ports (seL4), code has been inspected by multiple groups including
  + CASE will use connector proofs on seL4
  + Generating code was target dependant.
  + Some of the issues
    - Proper way of modeling interrupts/drivers, interrupts have 1st and 2nd levels, strictly speaking, 1st level handler does not run on associated AADL thread.
  + Everyone wanted to be the master, so did not match AADL dispatch.
    - External types/threads, three different groups wanted to create the types. We let Ivory, VMs/Containers - wrote up ways to map.
  + VMs/Containers
    - Maintaining VM memory/device mappings
    - Schedulability?
  + Peter- map variable to port and use a asyc thread
  + Could have done in a device using it as an interrupt handler.
  + Could use output rate to add more outputs from a single incoming event.
  + We would like to be completely aligned to code generation annex.
  + Running OSATE “headless”
    - For code generation: want single command build
    - For AGREE/Resolute analysis
    - I think this will be a standard mechanism for use.
    - It’s not for efficiency, but rather that it is easy to script.
    - Tyler – we have a scripting approach for headless analysis – described.
    - They took over the original Little Bird in 45 minutes. They did not succeed on HACMS upgraded Little Bird, a big part of it was the trusted build and having a good understanding of the architecture.
    - AADL Code Gen Annex Questions
      * We need to revisit the runtime services to make them type safe.
      * We should report error types.
      * Send output – can we detect it and what is the right info.
      * How are periodic dispatches managed?
    - **Action:** Jerome will coordinate with Etienne, Brian, and Mike for issues to be discussed. **Action:** Mike Whalen will send further detail on runtime needs to Jerome. Done.
* 1700-1730: Planning for next meeting (Bruce Lewis)

# Webex Info for the meeting:

**MONDAY and TUESDAY:**  
  
AS-2C AADL Meeting   
Every day, from Monday, January 29, 2018, to Tuesday, January 30, 2018   
9:00 am | Europe Time (Paris, GMT+01:00) | 9 hrs   
  
JOIN WEBEX MEETING   
<https://sae.webex.com/sae/j.php?MTID=mb3952846bd38985c6e4cf7082f03817a>   
Meeting number: 624 857 526   
Meeting password: Monday8   
  
JOIN BY PHONE   
1-866-469-3239 Call-in toll-free number (US/Canada)   
1-650-429-3300 Call-in toll number (US/Canada)   
Access code: 624 857 526   
  
Global call-in numbers:   
<https://sae.webex.com/sae/globalcallin.php?serviceType=MC&ED=632500662&tollFree=1>   
  
Toll-free dialing restrictions:   
<https://www.webex.com/pdf/tollfree_restrictions.pdf>   
  
**WEDNESDAY:**   
  
AS-2C AADL Meeting   
Wednesday, January 31, 2018   
9:00 am | Europe Time (Paris, GMT+01:00) | 9 hrs   
  
JOIN WEBEX MEETING   
<https://sae.webex.com/sae/j.php?MTID=m82d97dc8ba366f81c14c79077e504f22>   
Meeting number: 629 009 939   
Meeting password: Wednesday8   
  
JOIN BY PHONE   
1-866-469-3239 Call-in toll-free number (US/Canada)   
1-650-429-3300 Call-in toll number (US/Canada)   
Access code: 629 009 939   
  
Global call-in numbers:   
<https://sae.webex.com/sae/globalcallin.php?serviceType=MC&ED=632501117&tollFree=1>   
  
Toll-free dialing restrictions:   
<https://www.webex.com/pdf/tollfree_restrictions.pdf>