AADL standards meeting Jan 25-27 2016

* Location Toulouse France. IRIT University.
  + Google map link - <http://maps.google.com/maps?q=Rue+Charles+Camichel,+31000+Toulouse,+France&z=16>
  + Address: IRIT/ENSEEIHT, 2 rue Charles CAMICHEL, 31071 TOULOUSE, FRANCE
  + Room location: Building E+F, room F501 (fifth layer, 2nd lift right in the hall of the building).
* Not collocated with ERTS2016 - http://www.erts2016.org/
* Information on hotels <https://wiki.sei.cmu.edu/aadl/index.php/AADL_meetings>

Attending in person: Marc Pantel, Bruce Lewis, Brian Larson, Alexey Khoroshilov, Denis Buzdalov, Etienne Borde, Jerome Hugues, Julien Delange, Jean Pierre Talpin, Serban Gheorghe , Pierre Dissaux, Pierre Gaufillet, Thomas Noll,

Remote: Shaber Highdex, Dominique, Thomas Driessen, Brendan Hall, Steve Vestal, Roger Champagne, Tiyam Robati, Todd Carpenter

# Monday, Jan 25

* 0900-0945: AADL standardization committee news + action items (Bruce Lewis)
  + Germany in Summer, Embedded Systems, same week at SEI,
  + Check on whether we can upgrade the document rather than ballot to get clean copy
  + Peter has updated action items.
* 0945-1030: AADL v3 roadmap review (Peter Feiler)
  + V2.1 errata to go into current OSATE release stream
  + Will discuss what will go in 2.1 and 3
  + Incremental prototyping of V3 implementation
  + Issue of compatibility – what if we remove, but support
  + Separate release of the experimental, different file extension
  + End User – what he needs, virtual memory, etc.
  + Meta Model did not change much, especially in the instance model
  + Scalability of generated system operation modes
  + Serban – recursive instantiation tree, need to agree on textual instance model representation, Pierre agrees on the need, supports composition, Could be supported in OSATE, no official document on meta-model,
  + Better if syntax is the same as the declarative model, Peter will show.
  + Error reporting update on OSATE
  + Plugin support: run & run as job, read-only/read-write
  + **Action:** Peter – an errata - The ability to instantiate types (well as any classifier now supported) to implementations, is the textual implementation for any classifier?
  + Categories on connections on connection specification– make them optional? Since we know we are connecting ports.
  + Virtual processor -> processor by binding only
    - Currently we allow them as subcomponents as well
    - Or define the difference in the semantics
  + **Decision** - ports on buses and memory – to trigger modes – decision to accept event port to trigger modes and potentially dispatch, there is a property to say it’s only for a mode.
    - If it has an enclosing entity, it can inherit the mode.
    - If it has a failure, it senses and changes with an internal mode.
* 1030-1100: break
* 1100-1230: AADL v3 discussions (Nested processors, virtual memory & memory configurations) (Peter Feiler)
* 1230-1400: Lunch
* 1400-1530: AADL v3 discussions (Configuration & Binding) (Peter Feiler) Alexey presented
  + Multi-level Architecture
  + Logical flows, (software threads, processes, connections), (Logical execution with virtual processors), (execution platform processor, memory, bus, device)
  + Levels and layers (time of development) – (layers of abstraction including refinement of binding, extension + refinement, implemented as) goes from abstract to detailed.
  + Bindings looks like levels – (consumer – uses resource – provider), (Requires – quantity and characteristics – can provide), (Uses – uses != provides- provides)
  + Processor resources, communication …
  + Semantics of multiplicity (how many providers)
  + Analysis Targets – Resource consumption (AADL Core)
  + Memory binding – 10MB Ram, 5 MB Rom, TLB entries, ECC, duplication (?loss detection, lost probability?)
  + Semantics of multiplicity: one of, partly one, partly another, duplication-triplication
  + Process Binding for execution power – no abstract quantity, only in terms of concrete executor functionality
  + Connection Binding – bandwidth, latency, protocol, Quality of Service
  + Semantics of multiplicity: includes above plus parallelization, sequence, several sequences, diamond (network may create a diamond with two channels between two points)
  + Functional Binding
    - API, protocol, QoS
    - Function to the component that provides the functional capability in the system.
    - Do we need a separate resource concept? Concept of binding point, one side says I need a resource, other side provides. I only support x entities bound to me. Resource management.
    - Needs to address Virtual memory, multi-processor
    - Binding vs subcomponent – nested is it is built in, binding can change. Do you model a multi core – we do it as 4 subcomponent processors, do you allow them to bind to each processor or only to the top processor. Virtual processor to model the operating system, processor is the processor. OS are bound to each core.
* 1530-1600: Break
* 1600-1730: AADL v3 discussions (Compositional Interfaces) (Peter Feiler)
  + Ability to make selection multiple levels down in architecture design.
    - Configuration replaces of an open architecture via selections
    - Can go down configuring to lower components
    - Configurations may be partial, i.e., require additional selections
    - Response to idea positive
    - May still have Prototypes but removes the problem of having to move them to the top.
    - Closing a system for configuration, phasing configuration.
    - Public and private at the classifier level? Denis, do we get rid of it at the package level. If it has some information private, is it reconfigurable? Private implementations? Public contains the configuration types,
    - Control if reconfiguration, we don’t want some things reconfigured. Hiding of the blueprint is another problem. Should they be solved separately? Denis
    - Configuration of a Partially Closed Architecture.
    - Controlling configuration from the AADL model, integrating info in from other models Brendan Hall. Used by the integrator.
    - Issue of public and private, could it map as we would into a repository. So how do I link the spec in the public repository and implementation in the private repository? Need to look at how that works. Yet need to integrate for analysis.
    - Nested Closed Subsystems
      * Exposing the variability points. Partially closed.
      * Everything I need to know as configuration options must be in the open part.
      * Configuration space vs design space. What is remaining on the configuration? Example – car has two options, engine and speakers. They can be done incrementally in two configurations to define fully configured, one after the other. Selection section to state selection.
      * Configuring implementations. Identify the implementations that are to be configured. If you are in the design space, could you do partial?
      * We need the ability to define what is configurable, the second question is can you do it incrementally. In the design space. Need use scenarios.
      * Brendan – will send you some East AADL since they are in the automotive industry.
      * You need to know all that have to be configured, can configure some. The user who configures does not need access to the implementation.
      * Inter-dependencies, how to control, Brendan
      * Should we allow Reconfiguration? Explicit indicator if replacement is allowed (default) or not allowed (final)
      * Configuration of feature Classifiers – similar to refines. Extending to the more complete type for the port.
        + Subcomponent configuration
        + Reach down configuration selection for subcomponent features/prototypes
        + Replacement of existing classifier, scoped multi-layer
      * Array Sizes
        + Refined to of subcomponent/feature
        + Use of property constants
        + Need different array sizes in different modes – Denis
        + Scoped “constants” aka. Prototypes for array size
      * Configuration Specification Proposal
        + Extends allows in modes or changes to in modes
        + Configuration should not touch modes
        + **Action**: Brendan will send Peter example if single channel two dual channel brakes.
      * Binding Proposal
        + Typically resources, deployment bindings, seeking to do it not with properties.
        + Define a binding type, User defined name, sets of source and target categories
        + I to N or 1 to 1
        + Hardware flows can be sequences in binding
        + Hardware flows can also be look down. Problem of when we specify the bindings.
        + Channels you bind into a path, protocols you do not. Virtual bus used for both. Do we need a protocol separately? See CR on Safe Bus as example. Brendan
        + Partial binding configurations, configures as well?
        + Will talk about hardware paths separately
        + Binding points, then we can associate properties to them
        + Binding types and points make sense
        + Binding type could represent a sequence, like a flow
        + Binding point would use binding type.
        + Discussion of binding semantics, binding point, binding type by Alexey, Denis, See Peter’s notes.
      * Interface Composition
        + Covered 10 slides, do we still need feature group types? Feature group connections. Will pick it up later.
* 1730-1815: Update on the IRT Rover Project (Patrick Farail, Pierre Gaufillet, Jerome Hugues)
  + Pierre has been working for IRT Saint Exupery for one year.
  + Targeting TRL3-TRL6 bridging the gap of research to use.
  + Ingequip and MOISE projections
  + MOISE is just starting now. Targeting System Engineering
  + Functional breakdown …
  + IP-XACT
  + Capula is the tool, MOISE is the process.
  + Generating AADL from OCARINA
  + Software architecture captured in AADL
  + Vague concept of hardware, most are CPU’s, blue boxes are a software application on something that can execute software
  + Physical architecture breakdown stops when a leaf is allocated to hardware – device or software – application.
  + No concept of bus, interconnected ports implicitly define a bus to generate to AADL
  + Software is allocated to CPUs
  + Physical component =>. AADL processor
  + Physical Component => Device
  + We can do 90 per cent generation to AADL
  + We connect direct and then construct the intermediate ports.
  + Example was fully generated from Capella to AADL, Read-only artifact, modify Capella model
  + With AADL we refine manually by adding internal application objects, …
  + Can use prototype levels and it will change the data type all the way down.
  + Validation with AADL inspector, structural properties, scheduling with Mazarin and Cheddar
  + OCARINA generates code skeletons for each application including task management
  + We can realize the hardware/software design from the system physical architecture
  + AADL analysis and meta-modeling, AADL prototyping, experimentation on TwIRTee
  + Open the door to better exchanges of information between system and software engineers. Can discuss the models.
  + For now it’s not something shared. Will propose.
  + Formal specification of behavior – we will use the behavior annex and rely on FIACRA
  + Benefit – targeting system engineers who want to describe things that are very abstract. Really for a high level of abstraction. Not enough to give the software engineer so this way we bridge.
  + Functional source code – Simulink and hand generated.
  + Its rapid feedback to the system engineers to capture hardware/software impacts. We do not need the source code, this is an architectural prototype.
  + Length of time 4yrs.

# Tuesday, Jan 26

* 0900-1030: AADL v3 discussions (array connections, unified type systems)
  + Processors
    - Processor as a subcomponent: no. put processors in system (current)
    - Virtual processor inside processor: currently yes.
      * Also relates to binding point
      * Recommendation to only use binding
      * Virtual processor inside processor provided for binding vs. processor is target of virtual processor binding
      * Is there a need to indicate that a virtual processor subcomponent is available as binding target>
    - Virtual processor inside virtual processor
    - Can processor represent OS in addition to HW: Yes,
    - Purpose for the subcomponent virtual processor – to limit external change, its incorporated with the processor rather than bound.
    - Binding only approach how to represent pre-bound partitions? How to distinguish internal and external binding points. Enclosing system with both VP and P system binding point mapped to VP.
    - Is there a need to indicate that a virtual processor subcomponent is available as binding target?
    - White box binding. Offer way to restrict visibility down the hierarchy. But does not block ability to analyze the layered processor.
    - **Action:** Peter will do nested and pre-configured examples of processors for next meeting?
    - No recommendation yet as to whether we need virtual processors as subcomponents
  + Bus and Virtual Bus
    - Virtual bus acts as channel and as protocol
    - Virtual bus can be connected.
    - Bus as a subcomponent of a bus: no. Connected to bus.
    - Virtual bus inside bus/processor: Currently yes.
      * Provided\_Virtual \_Bus\_Class property: supperceeded by binding point
      * Provides side specifying the classifier of the resource that is available for binding. If no classifier then it is the component itself that is available as resource.
      * Virtual bus as protocol: how do we specify that we only support one end of the directional protocol? Do we need to indicate whether the protocol is directional or bi-directional?
      * Use extends to create a family of protocols, not hierarchy of binding.
      * Virtual bus in Virtual bus – used to represent channels. Division of resource. Supported now.
      * **Action**: Peter will do white paper on buses, whether we need subcomponents.
    - Memory & Virtual Memory
      * Enclosing system/memory may have binding point for whole memory
      * Need for representing different section of memory addressed, base address and range. Memory binding point on devices can model device registers without requiring memory subcomponents.
      * Nested memory (subcomponent) we may be able to eliminate. Use system.
      * Virtual memory in inside memory:
      * Virtual memory inside virtual memory:
      * Virtual memory roles
        + Enforcement: property of requirement and specification of whether it is supported. Who is supporting it (processor, memory, VP)
* 1030-1100: break
* 1100-1230: AADL Core Errata (Peter Feiler)
  + End to End Flow Specification – flow internal to a component with no connection so show connection. Keep in V3.
  + Mode Transition Triggers. Memory and buses did not have ports. Is in V2.1. But we need to understand it more. Binding points would support, seems better. Keep in V3
  + Priority on Data Access (new). We can do this on 2.1. Your concurrency algorithm would use the information. To resolve contention. Should we use priority or urgency? Only on requires access. It’s the person seeking the resource that asks for the priority.
  + **Action**: Denis and Alexey – propose a way to understand the semantics of the properties involved in scheduling. White paper for version 3.
  + Virtual bus connections
    - Virtual bus to Virtual processor
    - Virtual bus to virtual bus
    - Allow virtual bus access on virtual processor, virtual bus, allows access connections.
    - Requires consistency rules to insure that the connection exists
    - It is not required. It also can be derived.
    - Component itself or only via provides clause
    - **Decided:** We will put in the errata with the one change for virtual bus connections. It will be balloted with the BA. This will be version 2.2
* 1230-1400: Lunch
* 1400-1530: AADL BA Errata (Etienne Borde)
  + Behavior Annex Errata
    - 53 errata, 42 closed, 11 yet to be decided.
    - Delay modification, accept modification,
    - Three progressing drafts, BA with Errata, BA with Errata plus JP, BA with Errata plus JP plus BL
    - D.7-09 Data used in the RHS of … Accepted
    - D.7-10 behavior variable as a “final” … Accepted, but relationship with data annex needs to be adjusted. Is it an errata for the Data Annex? Accepted.
      * **Action:** Brian to notify Jerome to verify he has the errata. He needs to represent Final in the Data annex.
    - D.7-11 Goes with D-10
    - D.7-12 Assignment of behavior variable with D.710
      * D.7-02 Syntax definition issue – initialize with and property reference) , approved see Etienne’s notes
    - D.7-13 , derived from D.7-11, initialization Second option D.7(14) Approved
    - Require more discussion
    - D.6-12 exceptional behaviors, exception management – Should notify
      * Exactly one, keep the hash tag everywhere (for both property constant
      * Data modeling Annex –Brian. What would this look like in Ada, JAVA?
      * Need examples to discuss more.
      * Holds for function calls but also …
      * Delaying solution? Recommend accepting with small changes that Denis will provide.
      * **Action**- Denis will make small modifications to D.6-12 and Brian, Etienne see if they can resolve.
      * **Decision** – we will go forward with ballot based on what we have resolved. Last two or three have been difficult to find a solution. Tomorrow – 20 minutes extra for BA Annex.

* 1530-1600: Break
* 1600-1800: AADL BA Update (Etienne Borde, Jean-Pierre Talpin, Brian Larson)
  + Jean Pierre
    - Formalization of features of the BA
    - Simplification of the model: no pause statement
    - Incorporation of the formal semantics to the BA
    - Based on a model of automata with related asynchronous, reactive and timed trace semantics
    - Capture of asynchronous, aperiodic, sporadic, multi-rate,
    - A bit harsh to start on this for the reader.
    - Automata with ports and variables
    - Definition of behaviors, result is the post action
    - Force synchronization of two events.
    - Small asynchronous steps
    - Big-step, symbolically time, synchronous semantics, to complete or pause state is reached.
    - Small/big-step, real timed semantics –until complete or pause state is reached.
    - Asynchronous traces – partially ordered
    - Synchronous traces – totally ordered and partially ordered as well.
    - You have sync points, but between you have partial ordering. But it’s not fully partial ordering, in the worst case you have no ordering.
    - Some trace can have some order, asych
    - Some is control of release, some is making sure that the third execution of x and the second read of y occur at the right time and order.
    - We have not written a paper on the semantics, there is a paper on real time semantics in AADL, we will write a paper, to tell the story of how to standardize the semantics.
    - There is no impact on the BA, will not be easy to understand, but could be useful for Polychrony and for model checking. The changes we make would impact the formalism.
    - Brian – still working on time-outs, values and timing behaviors are more specific in the BA. Seems to be pretty close. BA-V2 allows complete code generation. Formalization would be an errata of the BA. It seems that it would be more than an errata. We would have to work through the semantics. We would need to have some guidelines of how to use, such as a book or extended article.
    - Etienne – calling the new parts BA.1 is misleading, we have to call it something else. What would we call it? BA+BLESS. BA+Formalization. The third could be something like BA+Formalization with BLESS.
* 1800-1820: AADL2Fiacre Re-implemented (Marc Pantel, Faiez Zalila)
  + Presented by Faiez
  + Fiacre language as target formal language.
  + Glue process, it manages buffering and …
  + Issue-the gap between AADL and Fiacre
  + Fiacre added port arrays, indexed constructs, genericity, library of semantic elements, like JP is trying to express by defining each capability. Quartex project.
  + Fiacre keeps the librares, then generates with TOM which builds a single unit.
  + Periodic and sporadic threads, ports, port-based communications, Shared data, Behavior Annex V2
  + RT-Fiacre – Connextion controllers, thread controllers, Schedulers added.
  + Next work
    - Cover the whole behavior annex V2 capabilities
    - Extend the considered AADL v2 subset, ie more threads types, …
    - Execution model checking – user driven simulation
    - Will be used by IRT on the Rover project.
    - Faiez and JP working together.
    - Tools can be made available if you have a specification that you want to evaluate and want us to evaluate.
    - You will do execution to show it will work, then later we move verification.
    - Can be supported of the building, its lazy state space exploration.
    - Can you support pre-emptive scheduling? Don’t know.
    - Currently we express constraints when you do communication and shared variables, analyze. It’s an abstraction of the scheduler.
    - We could benefit from the AGREE composition capability to make the verifications small.
    - Checks properties like scheduability, then properties on the model. How to do model checking in any language,
    - Observers one the state space.
    - We have an error, it looks like this component, we cross it in multiple execution paths. For each property we generate counter examples. Useful for safety.
    - Fiacre is very good with asynchronous. Now need to work the Synchronous.
    - JP is working with them, they would like to exercise the formalism.

# Wednesday, Jan 27

* 0900-1000: Remaining issues BA, Core, V3
  + Peter – Error annex errata
    - All but one or more
    - All – something
    - Allbut (x,n)
    - See notation from probability
  + Exclusive or – should we use a different operator (xor) since in other context or is not exclusive.
  + Operators need to work together – consider what the full logic required for sets.
  + **Action**: Denis – propose a set of operators for a set of errors
  + Legality rule for a set of initial states, one per mode?
* Alexey –
  + Questions on core
    - Input time and output time for ports – there is a more complex need, another type is dynamic, the component decides if it freezes its ports. - Properties can be extended in AADL Properties. The general principle is you can add to what is described in the Core. The semantics of this should be defined in the core – Denis. Now obligated to freeze. Not every component should freeze every port. Peter – you are mis-interpreting the default, you can identify which. Modes allow changes. Sometimes I have … Denis. Action-Denis will send Peter an example of what he needs. **Action** – Peter, clarify in the description, add dynamic input value, in the core.
    - **Action**: Peter - Add to core legality rule that you can not have cycles in bindings.
    - **Action**: Peter - Usage discussion – on layered architecture, Peter can add in notes.
* Etienne – Support for internal features (across annexes)
  + A guard of a transition may rely on an internal condition in or to abstract the reaction of a component on the occurrence of internal events.
  + Internal\_condition::=On Internal internal\_port\_name
  + D.6-07 ( if you use one you should not use the other support for internal features –internal vs among annex) internal features may be used to represent interactions among annexes. In the scope of a behavior annex subclause, an internal feature may be used to describe under which circumstances in internal event is sent. Recover – Accept
  + D.3-09 Avoid complete states in mode transitions – if anyu complete state indentifier is a mode identifier, if any complete state indentifier is a model identifier, then all complete state identifiers in that annex subclause must also be mode identifiers.

Accept.

D.3-10 Why only one final state for subprograms? Resolution – Remove L2 for now. Accept.

New errata. add each final state of a subprogram represents the end of an nominal behavior: the subprogram managed to compute the expected outputs. For non-nominal behaviors, …

D.6-13 Loop interators: - for future errata.

1000-1030: Expression language and Constraints Annex Draft plus Update (Serban Gheorghe)

Serban

Revised Constraint Annex summary of changes

* + - Constraints declared as MCS Structural theorems in package annexes or external files. New is files.
    - Simplification of previous Constraint Annex spec
    - Declarative model queries removed, replaced with pre-defined model instance element sets
    - Higher level assurance tool interface assumed
  + Constraint Languages Layers
    - Element types hierarchy
    - Basic types
    - Access methods
    - Computational functions and expressions (with scoping rules and context for AADL)
    - Check/prove statements
  + MCS Element Types include the ability to get to annexes – hooks to attach.
  + Adds flows modes, etc not defined in Resolute.
  + Peter - Would want to define a similar table for annexes as defined for the core language, Serban – would be great, did not find a way automate it.
  + Property\_ref – Peter has 3 ways to use this now.
  + Peter – constraints should work on a unified type set. Type set integration – will need to deal with in the future, need a unified system but if you stick with the core then you will be fine. Data modeling annex – record, core has record, underneath both are the same type. Data and property record types.
  + Union type is very needed. List of types that are equivalent. Union construct – can be anyone of these elements.
  + Expressing the language in a simple way first then more detail later – Peter. Use Prototype actuals vs actuals as the name to be clear, elements vs binding when you mean elements. Use AADL terms with AADL meaning so better understood.
  + Mode map, mode\_trigger, mode\_state\_machine
  + Use space instead of dot to make it more readable – Denis
  + Don’t have product – did not see in Resolute, need for probability
  + Should support functional and sets or mathematical or both?
  + Should have the fold operator for functional programming
  + Category is a string using the AADL metamodel, you get all features after extends
  + Flattened set with extends, as does Resolute
  + Issue related to the instance model and classifier name, we use the classifier name but it could be another name, need to use the instance name.
  + Given an instance, give me the instance subcomponents – his subcomponent is a subcomponent instance.
  + Subcomponent should be called … problem distinguishing between the instance and the declarative hierarchy, much debate. Meta-model in the standard (A.3?) for the instance model. Can use I\_name notation to make it clear. That would enable understanding between the declarative and instance model.
  + Theorem should be for specific component, types of components, global,
  + Theorems in classifiers? Peter uses external files but could then put them in line for component. This is important for scalability and readability.
  + Theorem are named constraints, leaf theorems are individual assertions, if I have a group of them, they are named, and can be reused. A theorem body should not be dependent on the architecture, if possible, if designed to be independent.

* 1030-1100: break
* 1100-1230: Expression Language and Constraint Annex Update (Serban Gheorghe)
* 1230-1400: Lunch
* 1400-1500: Network Annex Update (Alexey Khoroshilov, Tiyam Robati, Brendan Hall)
  + Network Annex – presentation reviews and provides current information.
    - Pre-Synthesis model – AADL bus connects components
    - Basic model – virtual links represented, types of options, like queue links, with virtual bus, binding to HW devices. Describes symentric networks. If we have a virtual bus inside then we have a binding to the outside bus.
    - Virtual bus provides resources to virtual link. According semantics of AADL 2.1 netA and netB are bound to VL. netA of VL is subchannel of VL then net A could be bound to VL.
    - Virtual Bus Implementation – declaring bindings rather than using links, seems to have both according to picture. Virutal Buses in the VL, then virtual buses bound to hardware devices.
    - Vendor specific protocols put on top of the channel in order to detect.
    - Decouple – Brendan to add more protocols to make it interesting.
    - **Action**: Brendan will get with Peter to discuss the network bindings, subcomponents vs bindings.
    - Could also be bindings virtual bus to bus, then binding virtual bus to hardware.
    - Brenden – this could be 5 gigabytes, we need an effient way to specify it. We need configuration management, concise, linked to a data base to get the tables.
    - **Action:** Alexey will send his presentation and Tiyam’s
    - **Action:** Brendan/Peter to experiment with representation and database.
    - Need the correction in the Error Annex
    - Transitions
      * Fail : Operational – [ 2 ormore (connection{NoService}) }-> Failed;
    - **Action**: Alexey will provide AADL model asymmetric to start from. Need more details to show why you need two buses to link to.
    - **Action**: Brian with Brendan to take this model and illustrate end to end error model over lunch. Some time in Feb.
    - May not fail stop, may fail
    - Need to have a message description in the error model.
    - Symmetric model in TwinWire, asymmetric on two wires.
    - Detailed model
      * Represents internal structure of AFDX
    - Wilfred sent the information.
      * This should give us all the paramenters we need for TTE.
      * Synchronization services added but we need to meet with TTech to do the next step. Go through Wilfred’s material. Would stress the error model for TTE. Would build it from the AFDX model.
      * Brian - Saturday lunch meeting until evening.
      * Roger Champagne is the TTech that should be included. Tyiam at the same university. Add to the working group.
* 1500-1530: Plan next meeting
* 1530-1600: Break
* 1600-1800: D-MILS and MILS-AADL with Discussion (Thomas Noll)
  + MILS is now a proper name – A component based approach for the construction, assurance, and certification of dependable systems that encourages a commercial marketplace of off-the-shelf high assurance components.
  + MILS platform – the architecture must strictly be implemented on the target.
  + A single policy architecture may span multiple DMILS nodes
  + TTE time-triggered Ethernet.
  + Semi-automated creation
  + SK – separation kernel
  + Consortium – Aachen, Univ of York (assurnace cases), INRIA, Grenoble (Univ Fourier), FBL (IT), Fortiss, Frequentis, …
  + MILS/AADL
    - Extended subset of Architecture Analysis & Design Language
    - Plan to come back to the standard AADL so the tools can be used more widely
    - Includes Hybrid modeling, using Compass AADL
    - Extended type system, symbolic names for type and value constants
    - Time units
    - Asymmetric and sym encryption and authentaion, Uninterperted functions, event data ports, …
    - Cryptographic Controller to split off secure info
    - Then re-merge
    - Asymmetric key pair, public part, private part. Use public to encrypt.
    - Adding encryption on top of Compass AADL
    - Pure mathematical function encrypted
    - Error Modelling – includes transient and permanent
    - System Analyses
      * Contract-Based Design
      * Means of composing components
    - Compositional Verification Tool Chain
      * nuXmv extended with … model checker
      * OCRA tool used to verify contracts (security properties example)
      * Property Specification Language LTL, First-order, real time properties
    - Implemented in an extended COMPASS toolset, COMPASS Toolset II and III
      * Fault-Tree Analysis
    - System Deployment
    - MILS to Prolog translator takes MILS-AADL specification to generation MILS Platform Configuration Compiler (MPCC)
    - Configuration Correctness, compares original and reconstructed MILS-AADL model
    - Policy Architecture – equivalence as architecture is made secure
    - Fortiss Smart Microgrid – power distribution and power generation to home. Make sure that all private info from home is not compromised
    - Only autheorized prosumers
    - Security and safety properties examples, battery shall not be overloaded. Should not charge when fully charged, …
    - Property Verification using COMPASS –
    - Compositional verification. Safety property – no overload, not easy, you have to have well specified behavior of the component. Contracts for the subcomponents are significantly more complex than the description in LTL
    - Must compare the performance of the two systems. Can be increased by 20Msec but must not violate the worst case requirement.
    - Re-integrating into the AADL mainstream
      * We have quite a number of projects that are following on
      * COMPASS – 2008, HASDEL – 2013, D-MILS -2013, CATSY – 2015, other projects AUTOGEF (2011), …
      * May previous projects, many ideas
      * New project COMPASS3
        + Update SLIM language to AADL standard
        + Update tools
      * Re-integration with AADL
        + Core AADL V2/V3
        + Error model annex
        + Behavioural annex
        + Easier to get industry acceptance
      * SLIM DATA types
        + AADL allows the definition of custom data types
        + SLIM annotations change introduce AADL properties system into SLIM. Allow contracts. {
      * Possible properties are defined in AADL Property Sets
      * COMPASS property patterns only available from a limited predefined set. Patterns used so simplify interface to engineers – Instantiated patterns which are translated to CTL, LTL, MTL, … About 16 patterns.
      * A new set of patterns has been defined with a grammar.
      * COMPASS integrates the behavior annex and error model, these are in the patterns.
      * We would re-define the syntax, perhaps have a cross complier.
      * This is a funded project by ESA. Starting now.
      * May be able to start discussions if we can come to Germany.
      * You are welcome to come.

Webex Info:

SAE AADL Committee Meeting

Monday, January 25, 2016

8:30 am | Europe Time (Paris, GMT+01:00) | 10.5 hrs

JOIN WEBEX MEETING

https://sae.webex.com/sae/j.php?MTID=m442f9c21e43f6cd5b2a3c59b94523bd7

Meeting number: 625 590 259

Meeting password: Monday1

SAE AADL Committee Meeting

Tuesday, January 26, 2016

8:30 am | Europe Time (Paris, GMT+01:00) | 10.5 hrs

JOIN WEBEX MEETING

https://sae.webex.com/sae/j.php?MTID=md04825dfe50c1f5b533fa77139cc0e87

Meeting number: 625 986 379 Meeting password: Tuesday1

SAE AADL Committee Meeting

Wednesday, January 27, 2016

8:30 am | Europe Time (Paris, GMT+01:00) | 10.5 hrs

JOIN WEBEX MEETING

https://sae.webex.com/sae/j.php?MTID=m13920f81f9b3d60540dc98c6616737ae

Meeting number: 629 135 239

Meeting password: Wednesday1

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: 625 986 379

Global call-in numbers:

https://sae.webex.com/sae/globalcallin.php?serviceType=MC&ED=441241227&tollFree=1

Toll-free dialing restrictions:

http://www.webex.com/pdf/tollfree\_restrictions.pdf

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.

Nothing in this message is intended to constitute an electronic signature unless a specific statement to the contrary is included in this message. Confidentiality Note: This message is intended only for the person or entity to which it is addressed. It may contain confidential and/or proprietary material. Any review, transmission, dissemination or other use, or taking of any action in reliance upon this message by persons or entities other than the intended recipient is prohibited. If you received this message in error, please contact the sender and delete it from your computer.