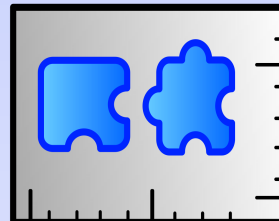




TOPCASED

Toolkit in Open-Source for Critical Application & Systems Development

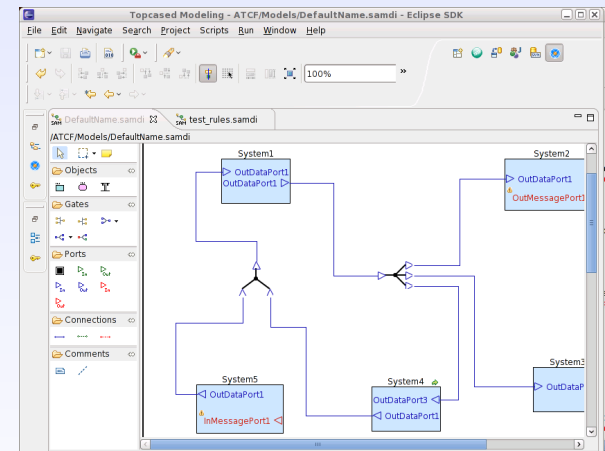


Spices

AADL Experimentations at AIRBUS

by Pierre GAUFILLET <pierre.gaufillet@airbus.com>

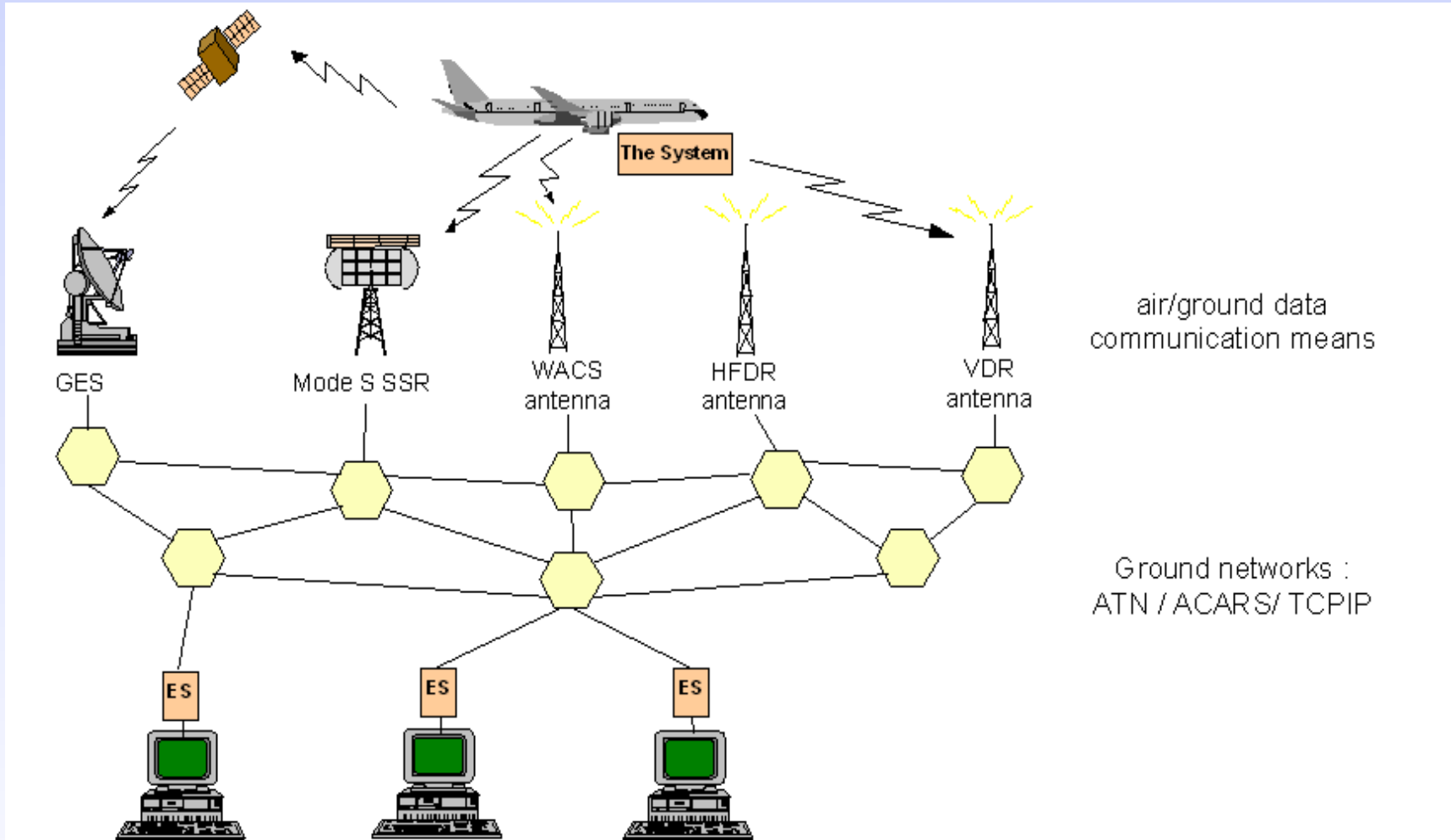
- Toolkit in **OP**en source for **C**ritical **A**pplication and **S**ystem **D**evelopment
- Started in 2004
- Covers Specification, Design & Coding stages, including common tools like configuration & changes management, interoperability.
- MDE oriented, but language agnostic : Ecore, UML, AADL, and several DSL.
- More than 35 partners (big companies, SMEs, universities, laboratories).
- Around 60 committers and contributors
- Based on Eclipse
- Around 80.000 downloads in 2008
- More information at www.topcased.org



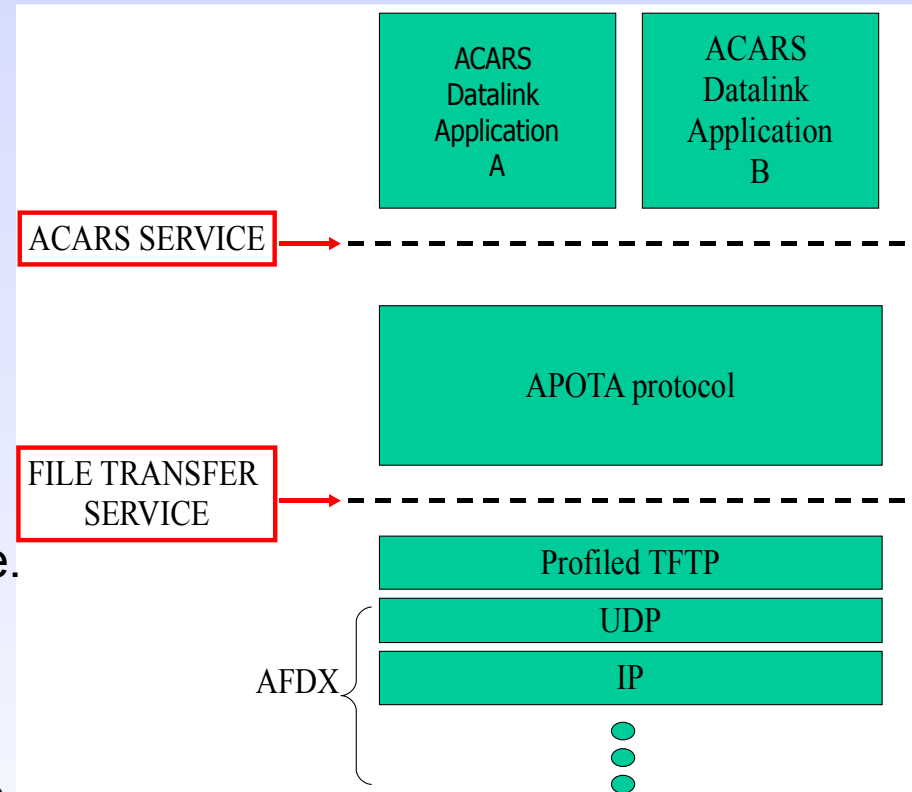
- **Support for Predictable Integration of mission Critical Embedded Systems**
- Started in 2006
- Cover AADL improvements (hardware support, behavioral modeling, etc.) & experimentations (avionics, telecom), models verification (behavior) and analysis (schedulability, power consumption).
- Involves several tools (ADELE, Tina, BIP, AADS, CAT, etc.) around the TOPCASED platform.
- 13 partners from Belgium, France and Spain.
- More information at <http://www.spices-itea.org>



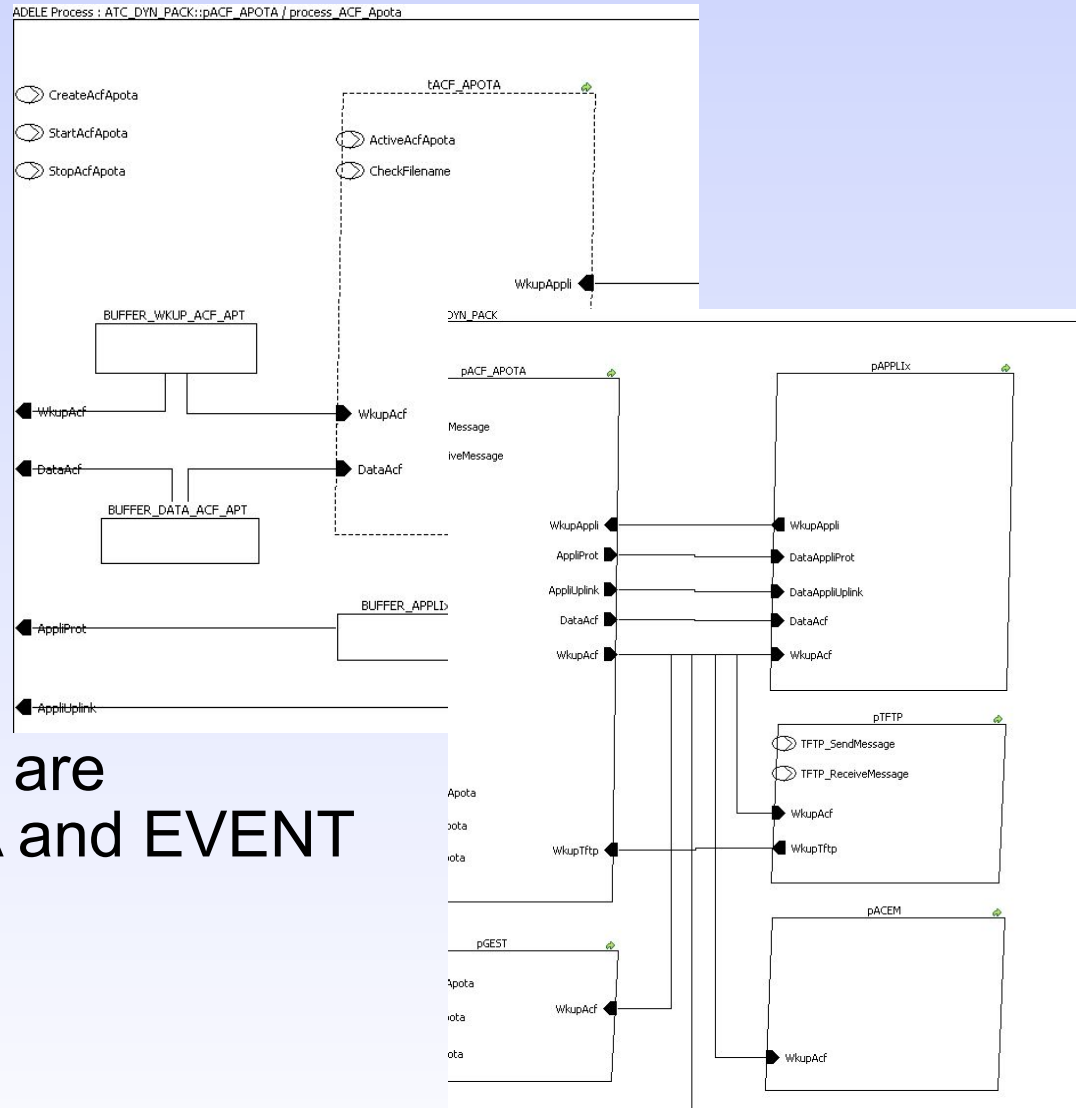
Air Traffic Control (1)



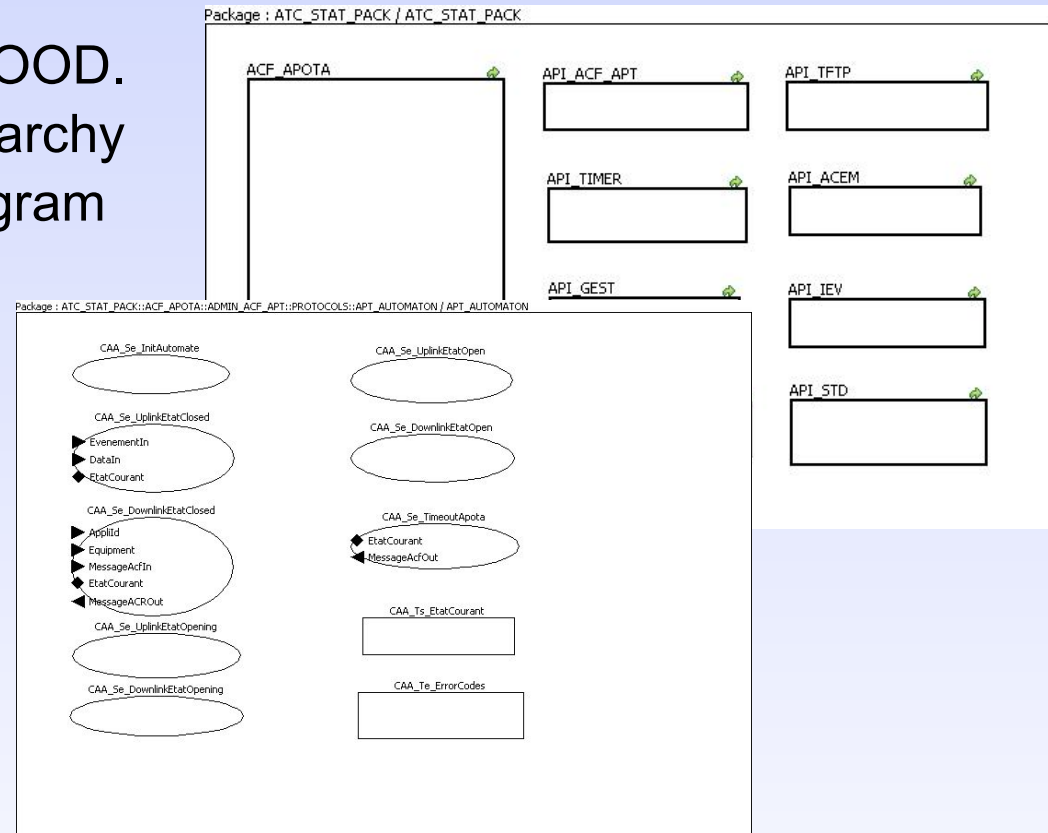
- Target specification, design, coding activities and connection to product verification.
- Validate AADL for static software architecture and dynamic software architecture.
- Test and improve ADELE.
- Define design rules.
- Prototype semi-automated tools between dev. stages – SAM to AADL, AADL to code.
- Static model verification using OCL.
- Dynamic model verification/simulation using SPICES tools – Tina, BIP, MAST, etc.



- Direct mapping for threads and process
- ARINC 653 Partition are represented by virtual processors
- Intra-partition objects are represented by typed DATA
- Inter-partition objects are represented by DATA and EVENT DATA ports



- How to represent Software Components ?
 - ...When you come from HOOD.
 - Using AADL package hierarchy
 - Ok for data types, subprogram
- Caveats
 - no implemented_by relationship between packages interfaces
 - not really adapted to represent constants...
- Others approaches
 - Use System ?
 - Use Abstract ?

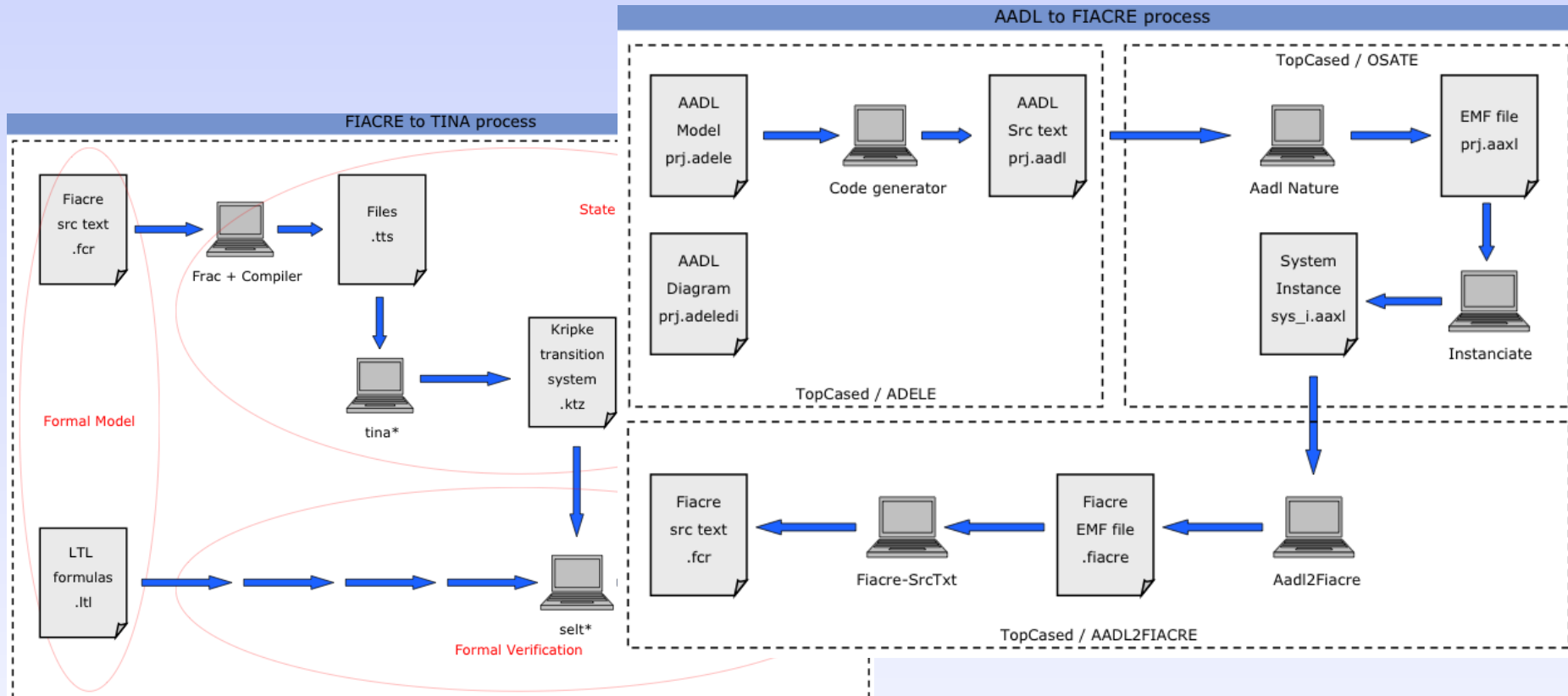


- For now, behaviors partially modeled using textual behavior annexes
- But a graphical tool is required to go further.
- Use also standard AADL properties

```
exemple.aadl
FEATURES
  dr2 : IN EVENT DATA PORT behavior::boolean;
  ds2 : OUT EVENT DATA PORT behavior::boolean;
END thrvc;

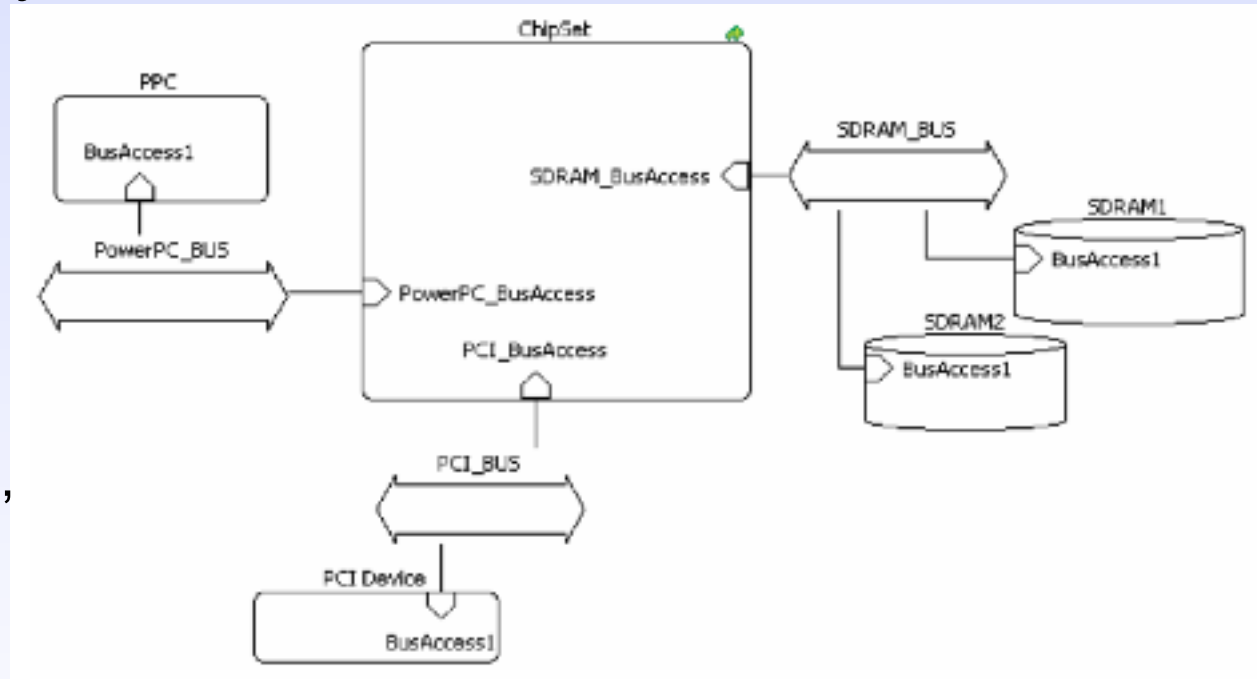
THREAD IMPLEMENTATION thrvc.others
SUBCOMPONENTS
  b2 : DATA behavior::boolean;
PROPERTIES
  Dispatch_Protocol => Sporadic;
  Compute_Execution_Time => 1 ms .. 1 ms;
  Period => 5 ms;
ANNEX Behavior_Specification {**
  states
    st: initial complete state;
    sf: complete state;
  transitions
    st -[dr2?(b2)]-> st;
    st -[dr2?(b2) when b2]-> sf { ds2!(b2); };
    st -[dr2?(b2) when not b2]-> st { ds2!(b2); };
    st -[on timeout 5 sec]-> st { ds2!(false); };
    sf -[dr2?(b2) when not b2]-> st { ds2!(false); };
    sf -[dr2?(b2) when b2]-> sf { ds2!(true); };
**};
END thrvc.others;

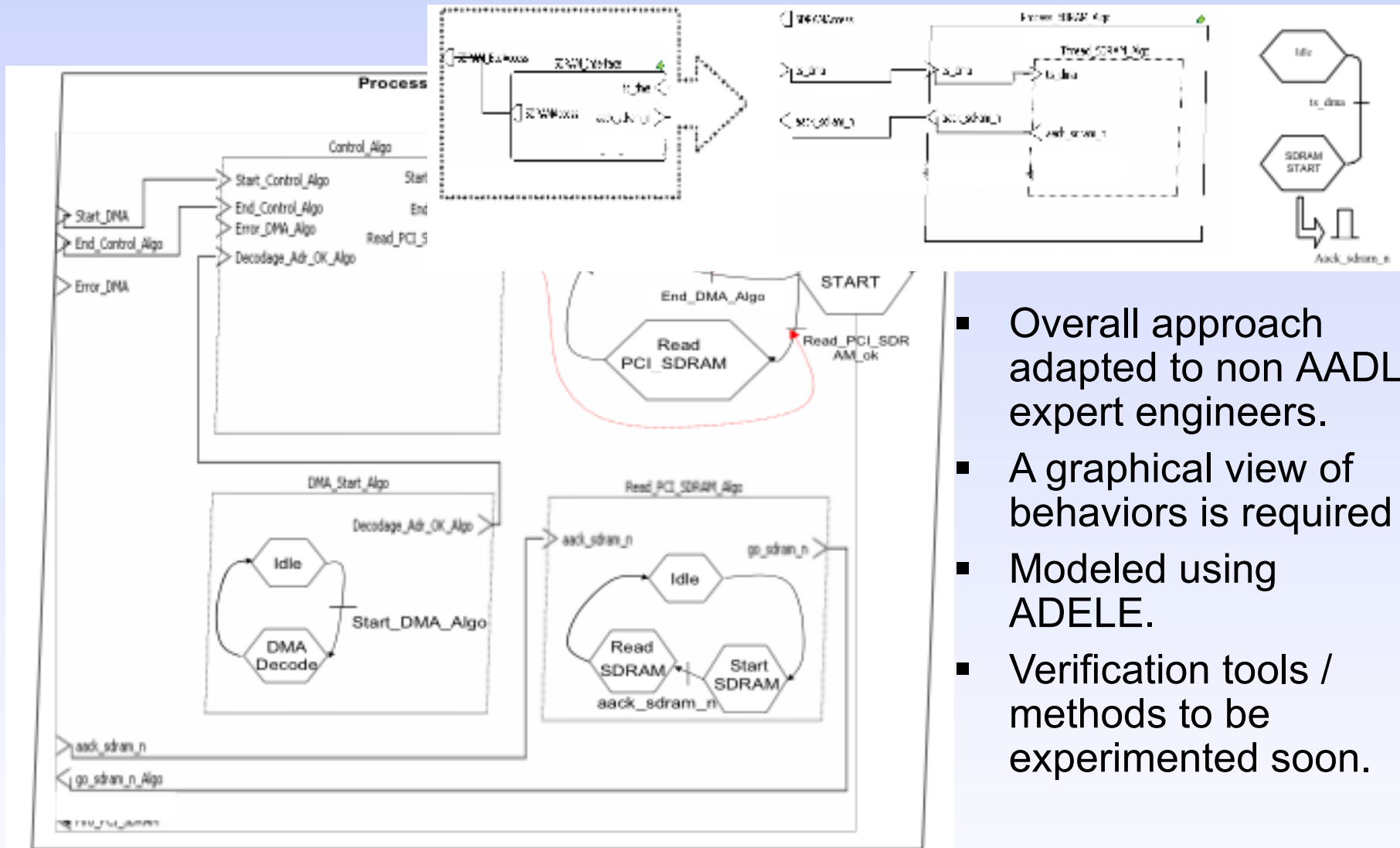
MEMORY mem
PROPERTIES
  Read_Time => 0 ms .. 1 ms;
END mem;
```



- Automated transformations from AADL to FIACRE, and FIACRE to Tina / Selt and verification of contracts.
- Behavioral contracts : looking for something simple.

- A mean to describe and compare architectures
- Adapted to industrial constraints
 - Requirements management
 - Code / Doc generation
- A support for analysis and verification
 - power consumption
 - stacks size
 - absence of common pathologies (deadlocks, buffers overflows, starving, etc.)





- Overall approach adapted to non AADL expert engineers.
- A graphical view of behaviors is required
- Modeled using ADELE.
- Verification tools / methods to be experimented soon.

- **Modeling**
 - AADL is definitely a first class real time architecture modeling language.
 - But combining static and dynamic designs in a unique model still requires some work.
 - Seems also to be adapted to some hardware contexts like globally digital systems.
- **Behavioral Verifications**
 - AADL > FIACRE + FIACRE > Tina is currently being experimented on simple use cases.
 - Interesting perspectives for real time architecture verification
 - But there are open points : modeling to verification tools is beginning to be ok, but what about verification to modeling tools ?
 - Engineers without formal-methods expertize should be able to express contracts.
 - Scalability still to be experimented...