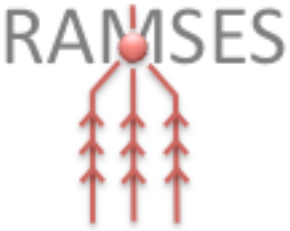
What is RAMSES ?

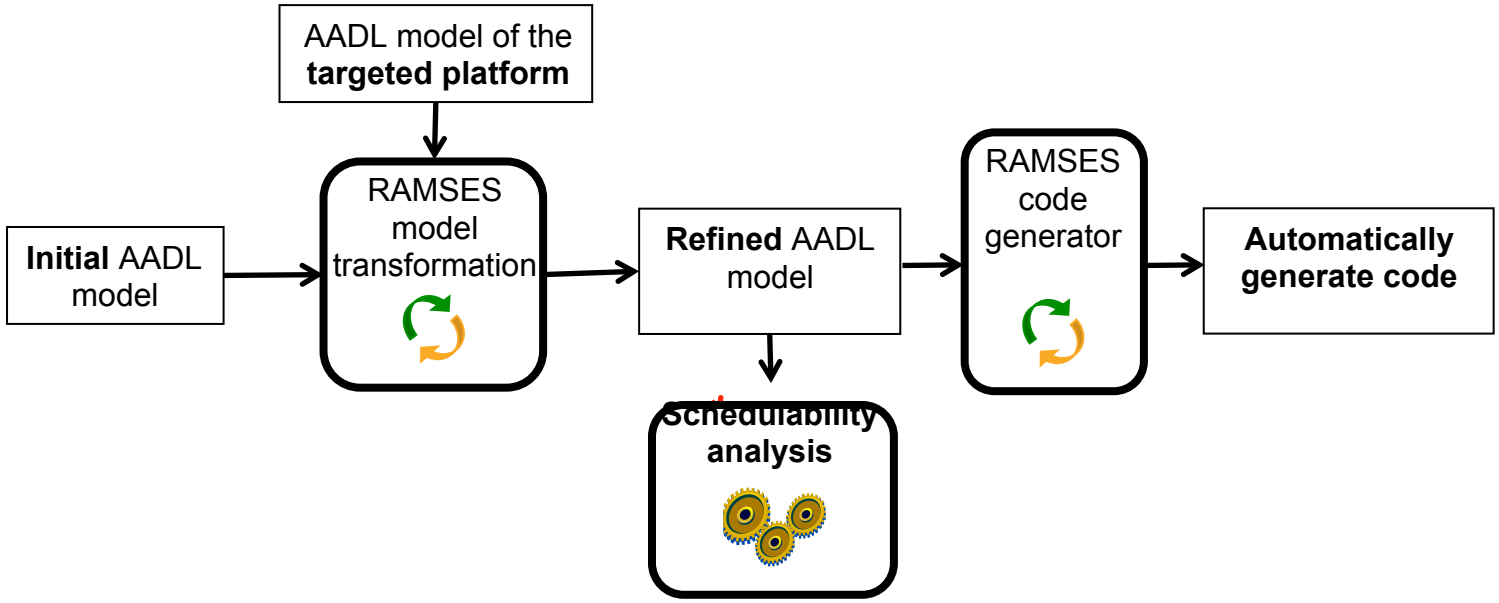
Refinement of AADL Models for Synthesis of Embedded Systems



- **An AADL model transformation framework**
- **A code generation plugin (C, Ada)**
  - ARINC653 (avionics)
  - OSEK (automotive)
- **A set of analysis plugins**
  - Response time analysis (using AADL Inspector)
  - Memory consumption

What is the effect of code generation?

- **Code generation interprets an abstract model to produce its implementation:**
  - Connections are mapped into queues, potentially using protected shared data, additional threads,
  - Operational modes may require additional threads to manage mode transitions,
  - Health monitoring requires faults detection and recovery mechanisms,
  - etc, etc.
- **Proposed approach in RAMSES:**



Input and platform AADL models

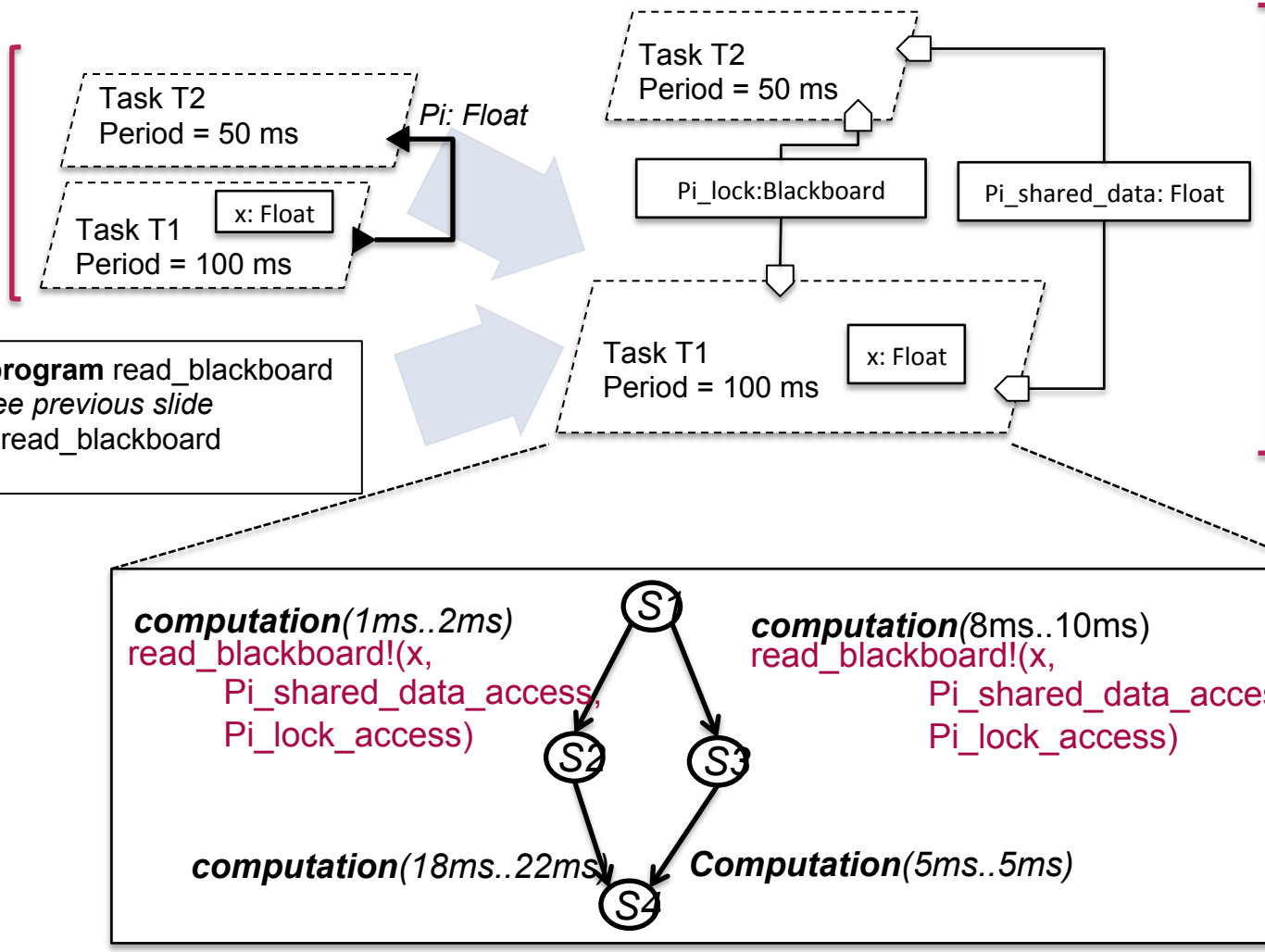
- **Input model**
  - Interconnected periodic or sporadic threads with timing properties
  - State machines for threads behavior (with timing properties)
- **Platform model**
  - Data types and Subprograms of the underlying middleware or OS, with protection protocol for data accesses
  - State machines for subprograms behavior (with timing properties)

```
subprogram Read_Blackboard
features
  BLACKBOARD_ID: requires data access BLACKBOARD_ID_TYPE
                                {Concurrency_Control_Protocol => Priority_Ceiling; };
  …
annex behavior_specification {**
  states
    s: initial final state;
  transitions
    t: s-[]->s{computation(4 us .. 5 us) in bindings (RAMSES_Platform::POK_X86);
                BLACKBOARD_ID!<;
                computation(1ms..2ms) in bindings (RAMSES_Platform::POK_X86);
                BLACKBOARD_ID!>};
**};
end Read_Blackboard;
```

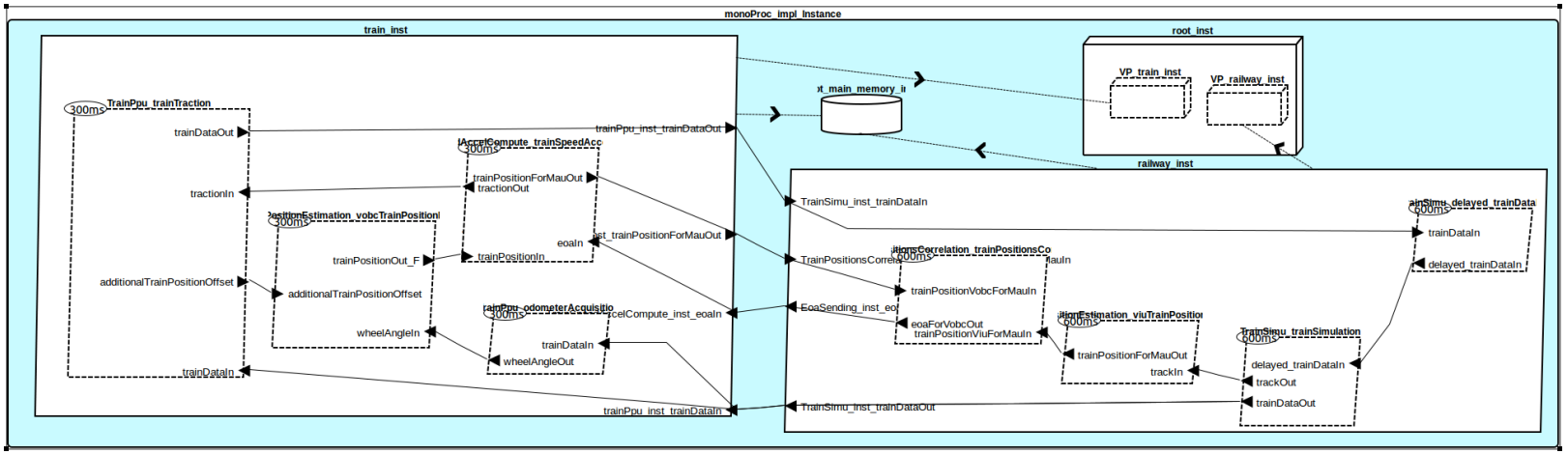
Architecture model refinement

A
Input model

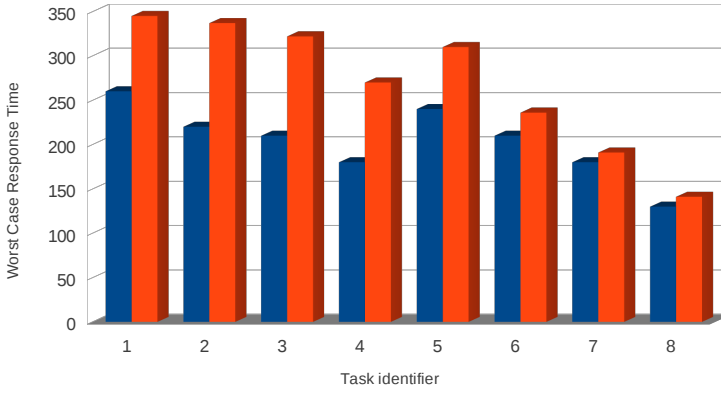
B
Platform model



Case-study



- **This project was evaluated on a case-study of**
  - 2 partitions
  - 8 threads, two branches with locks per thread
- **Leading to 64 scheduling configurations to analyse**
  - Less than 3 minutes computation (parallelized, Intel Core i7-3740QM; 4 cores)



Open questions (1/2)

- **Integration of AADL runtime services**
  - In its current version, RAMSES uses a minimal runtime environment (sometimes directly the OS services).
  - What about using standardized AADL runtime services?
    - Here is an example of AADL runtime service as defined in the standard.

```
subprogram Send_Output
features
   OutputPorts: in parameter <implementation-dependent port list>;
   -- List of ports whose output is transferred
  SendException: out event data; -- exception if send fails to complete
end Send_Output;
```

    - No reference implementation of AADL runtime services

AADL runtime services (draft proposal)

- **Define error codes for AADL runtime services: every runtime service returns a code which identifies an error type. This would replace the « exception » seen in previous slide.**

```
data error_code
Properties
  Source_Text => ("aadl_runtime_services.h");
  Source_Name => "error_code_t";
  Data_Model::Data_Representation => Enum;
end error_code;
```

- **Conventions**
  - 0/OK → no error
  - -1 → wrong param
  - …

AADL runtime services (draft proposal)

- Define the signature of services with AADL subprograms

```
data port_reference
properties
  Source_Text => ("aadl_runtime_services.h");
  Source_Name => "port_reference_t";
  Data_Model::Data_Representation => Struct;
end port_reference;

subprogram Send_Output
Features
  PortVariable: in parameter port_reference;
  ReturnCode: out parameter error_code {Generation_Properties::Return_Parameter => true;};
end Send_Output;

subprogram Put_Value
Features
  PortVariable: in parameter port_reference;
  DataValue: in parameter request  {Generation_Properties::Parameter_Usage => By_Reference;};
  ReturnCode: out parameter error_code {Generation_Properties::Return_Parameter => true;};
end Put_Value;
```

- The mapping fo subprograms and types defined in the code generation annex show how to map this to source code
- N.B: there should be variants for different programming languages

Open questions (2/2)

- **Conformance to the code generation annex**
  - RAMSES tends to conform to the naming rules of the code generation annex
  - The code generation annex defines a notion of context for accessing ports within a subprogram
    (57) This opaque type is a record whose members are the ports available in this particular context, but also access to data. The name of these members follows identifiers mapping rules defined in this annex.

  ➔ The naming convention of the data structure is described, but not its content.

  - It is not clear when this contextual information needs to be generated:
    - *A.6.3 The Code_Generation::Convention property controls the generation of specific structures to manage port variables.*
    - *(64) One instance of the context type is passed as parameters to each user's subprograms that need to interact with ports.* What does "subprograms that need to interact with ports" mean?
    - It is not clear what should be the signature of subprograms in source code:
      A.7.2 *(68) The first parameter is the AADL context information, discussed in section A.6.3* … What are the other parameters? Why other parameters?

What is missing (summary)?

- **On the runtime:**
  - agreement on data type definitions of the runtime (AADL definition) + conventions (e.g. error codes)
  - agreement on subprogram definitions of the runtime (AADL definition) + conventions (e.g. which parameters may be null)
- **On the code generation annex**
  - A clearer definition of when context is generated, and what it contains
  - what is the signature of the runtime services usable by programmers when using the AADL convention (may differ a bit from the core runtime services definition)
- **More generally**
  - A reference implementation of the runtime services is needed

Thank you for your attention

etienne.borde@telecom-paristech.fr