Pierre Dissaux

AADL Demo Day

UAH, Huntsville, 14 Feb 2019

20 years tool support for major industrial projects:

- HOOD Software design tools for Ada and C
- Eurofighter Typhoon
- Airbus A340, A380, A400M, A350
- Tiger Helicopter (mission calculator)
- Rafale (engine control)

15 years investement in new tool technologies:

- SAE AS-5506: Architecture Analysis & Design Language
- AADL graphical modeling tools: Stood for AADL
- AADL analysis framework: AADL Inspector
- European Space Agency: TASTE Editors (Space SW development tools)
- European Commission: ERGO and MOSAR Projects (Space Robotics)
- Generic model processing technologies: GMP, LMP: LMP Dev Kit

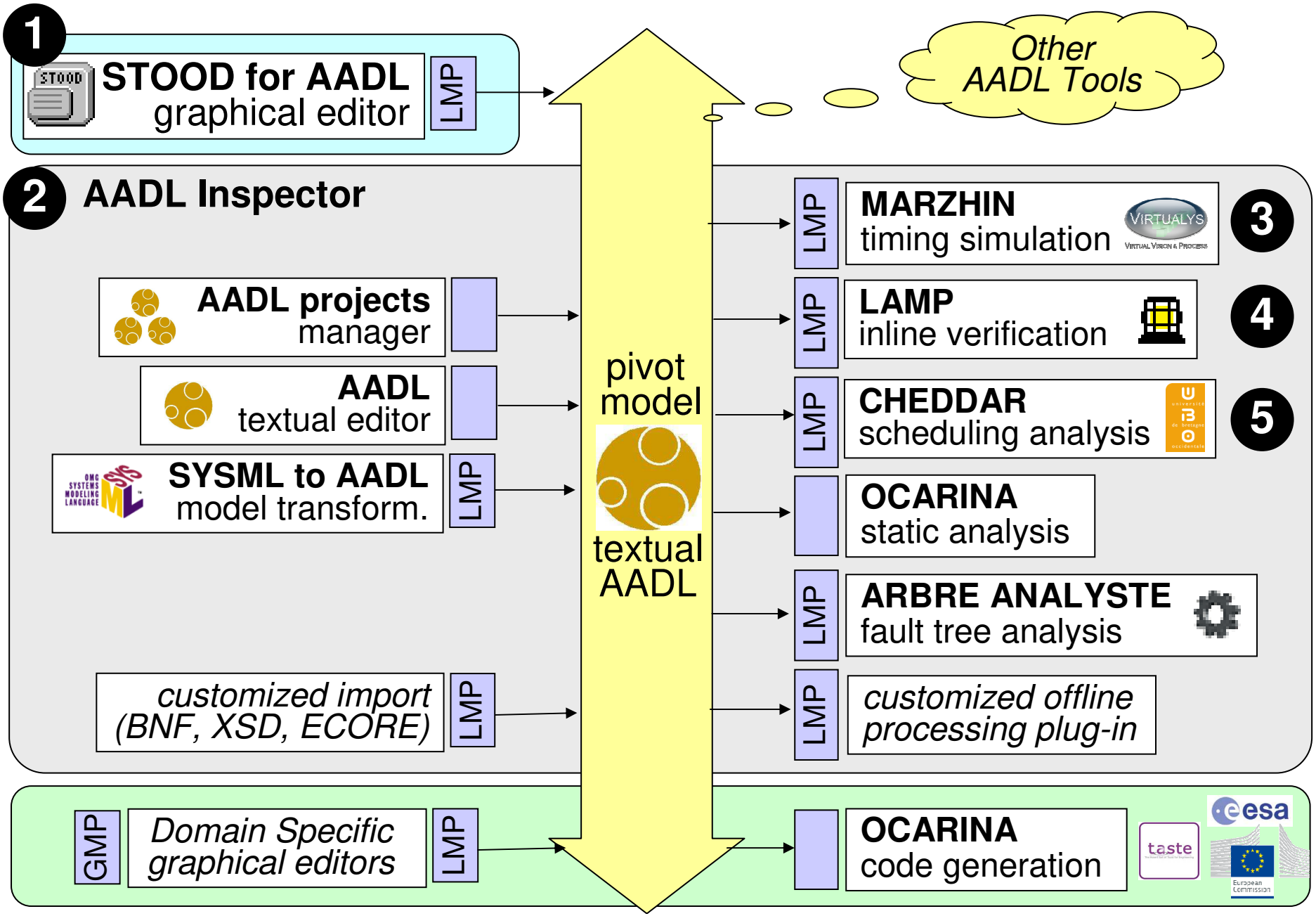
HOOD

AADL

LMP

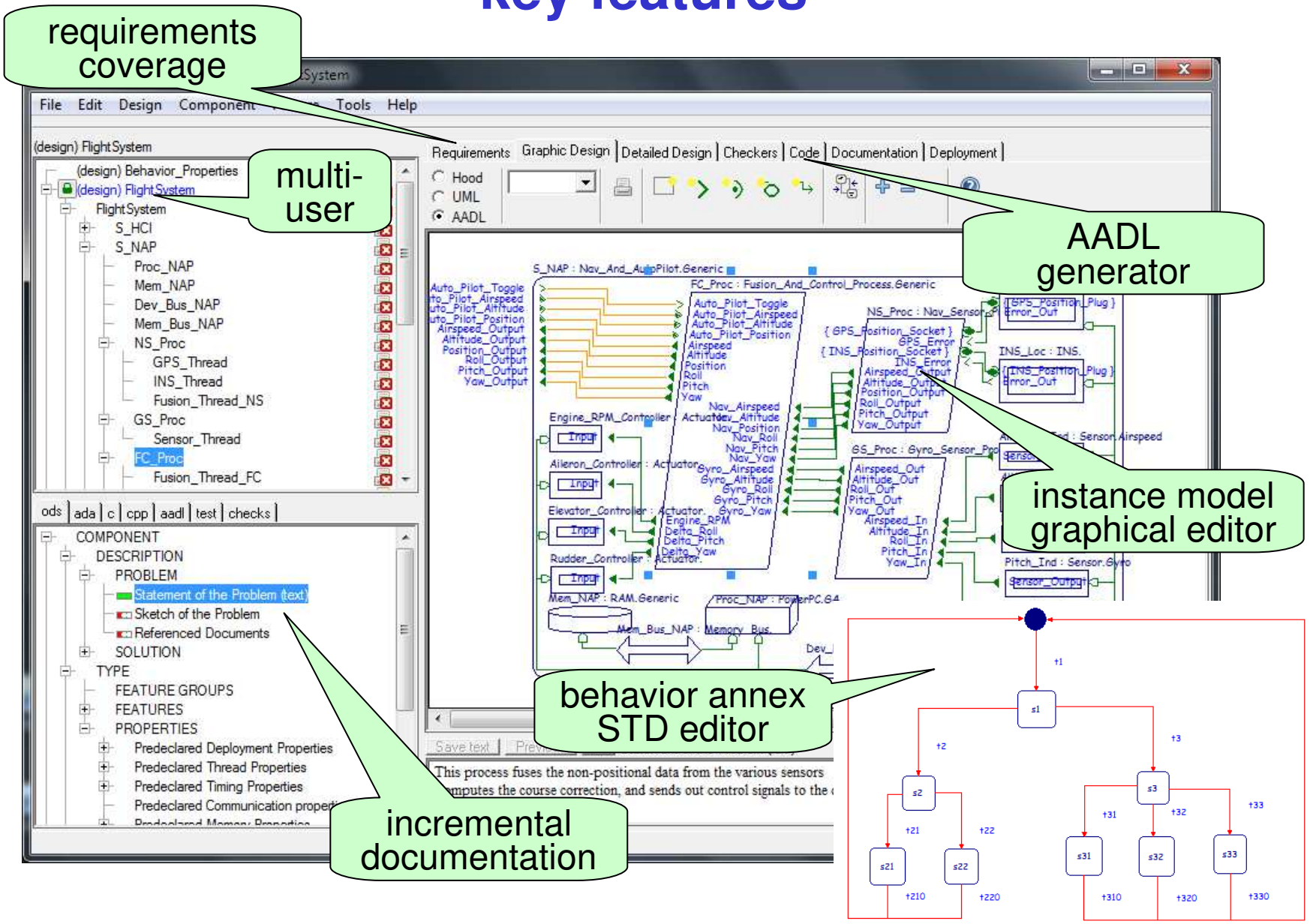
Ellidiss Technologies
www.ellidiss.fr

AADL centric tool-chains



Ellidiss Technologies
www.ellidiss.fr

Stood for AADL key features



Ellidiss Technologies
www.ellidiss.fr

Top-Down modeling process for AADL

Hierarchical Object Oriented Design (HOOD)

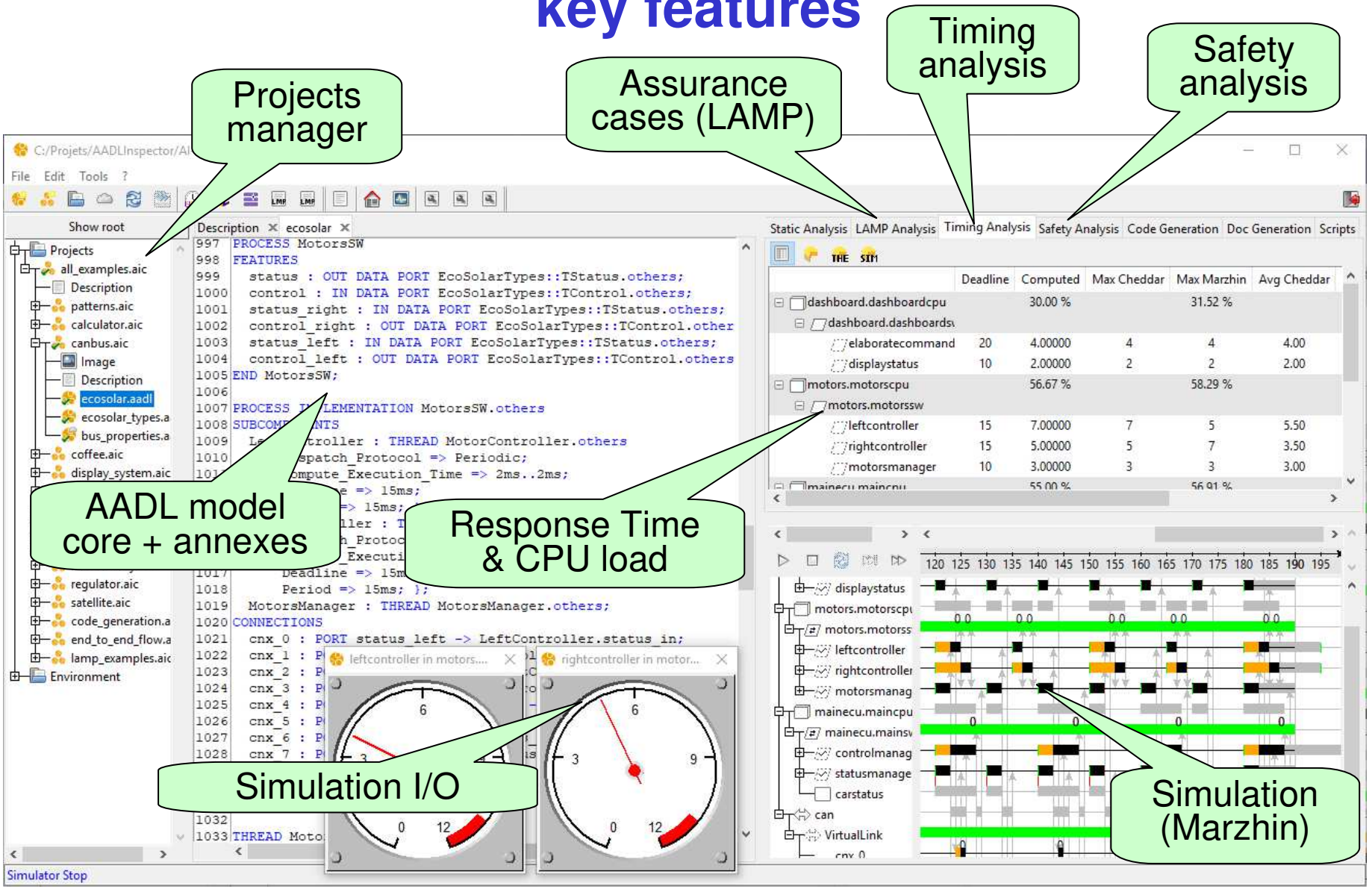
- Used by the biggest European avionics projects (Airbus, Eurofighter)
- Architectural Design (diagrams):
  - hierarchy of components with rigorous visibility rules
  - enable safe subcontracting (sub-trees)
  - ease testing, integration and maintenance
  - prevent from producing "spaghettiware"
- Detailed Design (structured text):
  - keep track of design decisions
  - requirements coverage
  - supporting framework for design documentation, coding and testing

Benefits for the AADL user (Stood for AADL)

- Graphical editor of the AADL Instance Model (what you design is what you get)
- Data Hiding enforcement (visibility rules, no provides data access)
- AADL Declarative Model generator (textual AADL) for tools interchange
- Complement AADL design activities with detailed design (documentation and coding)

Ellidiss Technologies
www.ellidiss.fr

AADL Inspector key features



Ellidiss Technologies
www.ellidiss.fr

AADL Inspector 1.7

AADL projects manager

- core 2.2 + annex sub-languages EMV1, EMV2, BA 2.0
- hierarchical AADL project structure:
  - AADL environment (libraries, property sets)
  - sharable sub-projects
  - simulation scenarios
  - documentation sections (text, pictures)

Imports XML/XMI models

- generic transformation process for ECore based models using LMP
- existing prototypes for UML/MARTE, SysML, Capella, …
- require precise mapping rules to be formalized (project dependent)

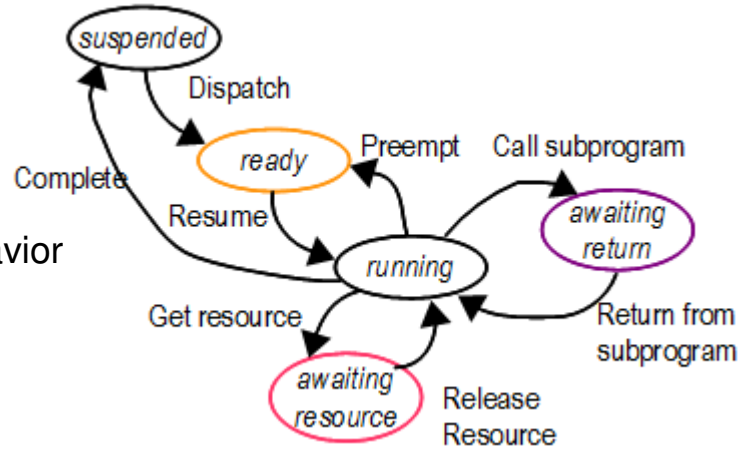
AADL model processing

- turnkey embedded tools:
  - Cheddar (scheduling analysis)
  - Marzhin (event based simulation)
  - Ocarina (AADL compliancy analysis, code generation)
- user defined on-line assurance case checkers with the LAMP annex
- customizable off-line plug-ins using the LMP toolbox

Marzhin
Executable AADL

Multi-agent real-time simulator:

- Based on a pre-existing multi-agent kernel
- Specialized agents to represent real-time software constructs:
  - Processor and scheduler
  - Process and partition
  - Thread and shared data
  - Ports and connections
  - Bus and bus messages
- The agents interact together and exhibit a global behavior



Implementation of the AADL run-time

- Standard AADL run-time semantic
- Behavior Annex interpreter
- Supports multi-processor, multi-partitions and multi-core architectures
- Generates system state changes events

Accepts user interaction

- Can be controlled by scenarios or dialogs
- Used to display simulation traces
- Used to animate 2D/3D graphics

LAMP

Logical AADL Model Processing:

- Constraint language
- Assurance cases
- Inline verification rules
- Fully integrated with AADL models (LAMP Annex)
- Can replace REAL, LUTE, AGREE, RESOLUTE, …by a single one
- Based on the LMP technology (

LAMP features:

- Standard prolog language:
    - No new language to define & learn &maintain
    - Declarative syntax and formal semantics (ISO/IEC 13211)
    - Byte code available for IP libraries
- Exhaustive AADL model accessors:
    - Core language
    - Behavior Annex
    - Error Annex
- LAMPLib: predefined rules library
- Can process other input data sets (requirements, analysis results, …)
- Available in AADL Inspector 1.7 (LAMP Checker)

LAMP Example


```
THREAD t
FEATURES
  i : IN DATA PORT d;
  o : OUT DATA PORT d;
PROPERTIES
  DISPATCH_PROTOCOL => Periodic;
  PERIOD => 15ms;
  DEADLINE => 8ms;
  COMPUTE_EXECUTION_TIME => 2ms..2ms;
  MY_PROPERTIES::MAX_VALUE => 80 APPLIES TO o;
ANNEX Behavior_Specification {**
  VARIABLES
    v : d;
  STATES
    s : INITIAL COMPLETE FINAL STATE;
  TRANSITIONS
    t : s -[ON DISPATCH]-> s { rand!(v); computation(10ms); o := v };
```

access to AADL property values

```
ANNEX LAMP {**
  checkOverflow(Id,Class) :-
    concat(Id,'.o',F),
    getProperties(F,Class,'MY_PROPERTIES::MAX_VALUE',M),
    getPortValue(F,T,V),
    strToNum(V,W), strToNum(M,N), W > N,
    write('overflow of out data port '),
    write(F), sp, write(' at tick '), write(T), nl,
    write('( '), write(W), write(' > '), write(80), write(' )'), nl,
    fail.
  checkOverflow(Id,Class).
**}
```

access to AADL simulation output

overflow detection

Cheddar: Experimental Multiprocessor & Multicore Scheduling Analysis

F. Singhoff
S. Rubini
L. Lemarchand
H. Nam Tran

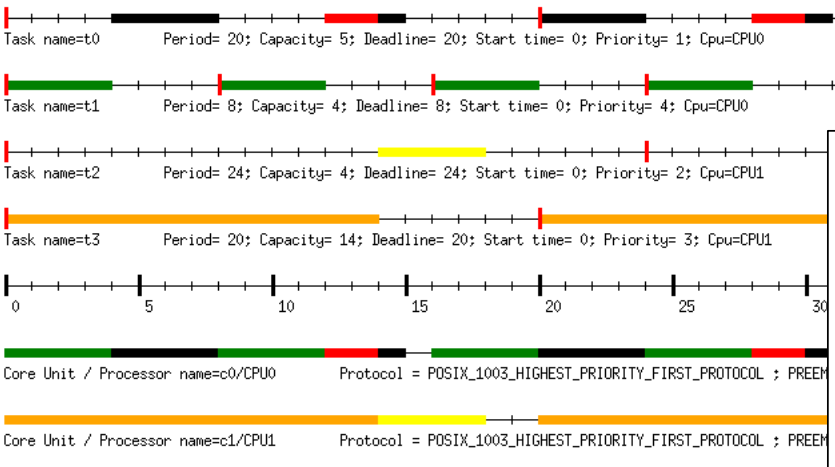
Started during the SMART project (completed in 2014):

- How to model multicore/multiprocessor architectures with AADL
- Choose or design new scheduling analysis methods for such architectures

Main multicore/multiprocessors features of Cheddar 3.x:

- **Partitioned and global scheduling policies:** extension of classical uniprocessor policies such as fixed priority or EDF + specific multicore policies such as Proportional Fair, EDZL, LLREF, …
- **Design of PAES partitioning algorithms:** trade-off between preemption, latency, communication, …
- **Support of shared resources between cores:** cache, network of chips

Example: Cache-Aware Scheduling Analysis



Scheduling analysis for systems with cache

Cache Configuration	Control Flow Graph	Worst-Case Execution Time
Memory Layout	Cache Access Profile	Scheduling Policy

Scheduling simulation with cache:

- L1 uniprocessor instruction caches
- Sustainable CPRD model (Cache Preemption Related Delay)
- And known feasibility interval (prooved): [0, LCM(Pi)]

Cache-Aware Priority Assignment Algorithm:

- Audsley oriented algorithm

Ellidiss Technologies
www.ellidiss.fr

Conclusion

AADL commercial tools

- Stood for AADL: instance model graphical editor for AADL
- AADL Inspector: analysis and simulation

Technology

- LMP: model processing toolbox (prolog)
- GMP: DSL graphical editor framework
- Research collaboration with University of Brest/Lab-STICC

Services

- Tools support
- AADL consulting
- Graphical front ends development
- Model processing tools (rules checkers, generators)
- Model transformations
- Heterogeneous tools integration
- R&D partnerships