Lip6 meeting
Sharing perspectives

20th February 2019
Avionics Products & Simulation - Missions

Airbus **Avionics Equipment** supplier
Develop excellence on the full scope of **hardware** and **embedded software activities**
Deliver Airbus core equipment to all aircraft programs

**Simulations Models & Platforms** provider
Develop innovative solutions to optimize the efficiency of aircraft design, testing and training
Deliver mature simulation products for all aircraft types, from research to commercial operations.

**Chamber of Reference**
Build a reference & set the standards for equipment suppliers.
Leverage high level skills in embedded software, on-board electronics and real-time simulation
Technical support and knowledge sharing with design teams, procurement, customer services

**Business centre**
Sell and support avionics and simulation products to customers.
Avionics Products & Simulation - Dimensions

- 8000 equipments per year
- 12000 electronic boards per year
- 120 software standards per year
- 75% engineers
- 590 employees (530 France – 60 India)
- 380 airlines & training centres
- 250 Full Flight Simulators
- 170 Flight Training Devices
- 4500 repairs and up to 4000 retrofits per year
- 4800 electronic boards per year
- 1000 software standards per year
Product Line approach

Software Product Line Engineering
- Component Based development
- Modular architecture / Re-usable Building Blocks
- Virtual Integration Platform

“Generic Safety Critical Platforms” Product Line
- Multicore architecture
- Versatility/Configurability vs hw context
- In-house kernel

“Applicative“ Product Line
- Design Patterns
- System/Software Architecture
Formal methods applied to critical software design (DAL A) to reduce verification effort.
Formal methods - some examples

Binary static analyzer for Stack use & WCET computing
- Abstract Interpretation based static analysis of the Executable Object Code
- Static analyzer: A3 (AbsInt GmbH);

Static analyses for Unit Verification of components services Unit Proof
- Weakest Precondition (WP) based program proof at C function level
- Proof tool: PHENIX_P (Frama-C/WP based from CEA)

“Local” static analyses (i.e. on subsets of the call graph)
- Data & Control flow analyses
  - Abstract Interpretation based static analysis of C code
  - Static analyzer: Fan-C (Airbus)
- Numerical accuracy assessment of floating-point computation
  - Abstract Interpretation based static analysis of C source code
  - Static analyzer: FLUCTUAT (CEA)

Run-Time Error analysis of C programs
- The ASTRÉE static analyzer
  - Developed by CNRS/ENS and AbsInt GmbH
  - Commercialized by AbsInt

Proved compilation of C source code
- CompCert (INRIA + AbsInt GmbH)
  - Formally verified source / object code semantic equivalence
Automatic Code Generation (relying on Formal Modelling techniques)

**Reactive Systems**

Real-time Control/Command systems (e.g.: Flight Controls)

⇒ Synchronous Language: Subset of Scade (Lustre) – Mainly pure data flow

*Desired 'non functional' properties:*
  - Determinism / Predictibility
  - Direct traceability Scade ‘⇒ Binary file’
  - Fast / Safe / Automated generation process

*Suited for:*
  - Formal verification (e.g.: fully automated computation of safe upper bound of WCET)
  - Parallelisation of treatments

⇒ 80%-90% of LoC are automatically generated

**Communication systems (e.g. ATC)**

⇒ Asynchronous language: LDS for communication protocols
Some technological/engineering trends

**Short term**
- Product Line engineering
- Reduce System/Software gap
- Pursue Process improvement (certification and engineering activities)
- Be competitive, reduce cost and lead time
- Data Security
- Multi-Core for Applications
- Formal methods: Pursue investment

**Medium/Long term**
- Artificial intelligence
- Parallel software engineering
- Data management
- Distributed avionics
- Many-cores architectures & New processing cores