For a reliable and scientific approach in system and software engineering.

**ASSERT: a step towards reliable and scientific system and software engineering.**

**ERTS2008 - 30/01/08**
assert-project / ESA - Dr Eric Conquet – ASSERT Project Manager D-TEC/SWE

**ASSERT: an European and Integrated Project.**

**Industrial partners:**
- THALES ALENIA SPACE (F+I)
- ASTRİUM Satellites (F)
- ASTRİUM Space Transportation (D+F)
- CS – Systèmes d’Information (F)
- DASSAULT Aviation (F)
- EADS Corporate Research Center (D)
- MBDA France (F)

**Academic partners:**
- CNRS-VERIMAG (F)
- UPM Technical University of Madrid (E)
- ENST (F)
- ETH - Swiss Federal Institute of Technology (CH)
- ONERA (F)
- University of Padua (I)

**SMEs:**
- BSSE (D)
- Dutch Space BV (NL)
- ESTEREL Technologies (F)
- INTECS (I)
- PROVER (S)
- SciScys (UK)
- SEMANTIX (GR)
- SoftwCare (E)
- Terma A/S (DK)
- TNI-EUROPE (UK)
- TNI-Software (F)
What is wrong with Software Engineering?

- Any SW system is a black box whose properties cannot be fully evaluated.
- Some samples can be extracted from this black box through testing.
- Consequences:
  - Nobody can guarantee the SW properties.
  - Confidence in SW can only be increased by testing (coverage level increases confidence level).
  - A SW development can be started even when no solution exist to a given problem (Who knows when a problem is undecidable?).
  - Trying to build software for either complex or undecidable problems leads to failure (many examples of SW projects stopped before end exist).

There are known problems...

...and known solutions
Learning from others: an example with the bridge construction.

- A bridge must be within budget, aesthetic, and safe.
- The basic steps of bridge building:
  - Selection of reference architecture: what kind of bridge do we need?
  - Tailoring of the reference architecture: what kind of adjustments do we need to make our bridge.
  - Modelling using mathematics to check the bridge properties:
    - Maximum load admitted.
    - Resistance to external forces (wind, water, ...)
    - Influence of material selection: identification of design margins.
  - Definition of the construction roadmap: instructions to be strictly followed by workers,
  - Controls on the building site.
- Can we learn something from this in the SW industry?

A major objective: build safe systems on safe foundations.

- Key areas for safe systems:
  - Data description: consistency at system level (wrong assumption = data model not needed for RT systems).
  - Performance prediction: not a nice to have feature for RT systems (wrong assumption = optimization will come later on)
  - Guarantee of safe behaviour: proof of key properties (wrong assumption = simulation is enough).
- Key solutions are:
  - Data modelling at system level: guarantees consistency and ease data exchange (ASN.1)
  - Performance modelling and proof at system level: performance guaranteed by the use of proven computational models.
  - Behavioural modelling: done at system and component model, proven at specification/design level, can produce code directly from proven models.
The ASSERT technical achievements.

- **The ASSERT process:**
  - Result of a consensus within the project
  - Integrates the different technologies used in ASSERT
  - Has shown its scalability and independence wrt the technologies (fully implemented on the ESA demonstrator)

- **The tools:**
  - Some tools have been specifically developed for ASSERT (Tools to support data modelling)
  - Other tools have been developed as extensions or new versions of existing tools.

- **The demonstrators and case studies:**
  - Satellites on-board SW with THALES ALENIA SPACE,
  - Multi-platform and highly available systems with ASTRIUM.
  - ESA demonstrator where the ASSERT full process has been implemented in 3 months by a non experienced team.

---

**Overview of the process**

1. **Capture of the system properties**
2. **Capture of the hardware architecture**
3. **Feasibility analysis**
4. **System real-time architecture**
5. **Run-time environment**
6. **Complete system**

---

**Modeling phase**

**Model transformation**

**Automatic code generation**
Are you ready for the new system and software engineering?

- The ultimate goal: to have a system and SW engineering approach enabling a full measure and guarantee of system properties.
- A proposed approach that ASSERT has begun to implement:
  - A given problem must be first proven as decidable,
  - A system solution must be proven as having the expected properties (functional and non-functional).
  - A system solution can also be reused (system families).
  - The system solution must define the acceptable ranges of properties for each component.
  - Each component must prove at any time that it respects the system constraints.
  - The transition to implementation must be fully automatic and guarantee the properties.

Means at system specification and design level.

- Proving that the system is feasible (proving that the problem is decidable).
- Proving that a given solution will exhibit the system properties.
  - Ex from ASSERT: validation of system Dependability with ALTARICA (non-functional properties).
  - Ex from ASSERT: validation of system functional properties with OMEGA.
- Defining the conditions under which the properties will be fulfilled (ex: system performance will be OK if WCET are within acceptable ranges).
- At system level: there is a consistency in terms of data/object manipulated or exchanged.
  - Need of a complete data model, defined at system level and used to specify all interfaces (internal and external).
  - Ex from ASSERT: use of the ASN.1 language.
Means at component level.

- The specification of the component can be proven
  - For components implementing mathematical functions (control laws for example):
    - the automation expert proves the law (or use an already proven one),
    - the computer expert shall demonstrate the correct implementation (taking into account the performance factors and the numerical representations).
  - For others, the used algorithms have to be proven at functional level.
    - Solutions invented on the fly by programmers have to be avoided.
- The properties of the component can be checked.
  - System properties are guaranteed if the system’s components match specific criteria.
  - It is thus important to permanently check the component’s status during development.

From model to implementation.

- Properties of the components shall be preserved at binary code level: use of proven code generators will help.
- The use of code generators will ensure that only acceptable code structures are used.
- Use of code analysis techniques (Abstract Interpretation) will ensure that unwanted behaviour are not present in the code.
- Middleware/OS shall have proven (or demonstrated) properties (ex: the ASSERT Virtual Machine).
A process change goes with cultural change.

- SW people are too much influenced by the “garage culture“:
  - Because programming is easy, anybody can write code!

- SW engineers must switch to a real industrial approach like in other domains:
  - This will increase confidence from system engineers.

- Evolution of tools must be accompanied with training and education programs:
  - First to prepare people for change,
  - Then to train people to the new approach.