THALES

Embedded Systems at Thales: the Artemis challenges from an industrial perspective

« Artist Summer Scool 09/09/2009"

Gilbert Edelin
Research Group Director

Thales profile

Three core businesses

Defence

50 %

- Air
- Land
- Naval
- Joint

Aerospace & Space

25 %

Security

25 %

A Worldwide Group

- 68,000 employees worldwide
- Presence in 50 countries

World leader for mission-critical information systems

THALES

Innovation and technological excellence

- R&D at Thales totals €2.4bn (19% of revenues)
- 25,000 researchers on cutting-edge technologies
- 300 inventions per year
- Over 15,000 patents
- Over 30 cooperation agreements with universities and public research laboratories in Europe, the United States and Asia

THALES

Thales and Embedded Systems

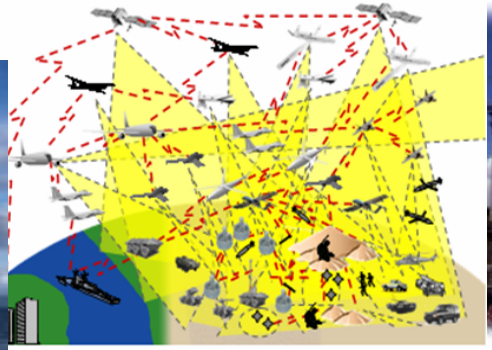
Thales leader in Mission Critical Systems. Most of them are Embedded, trend toward more autonomy

Continuing need of High Performance Embedded Systems: innovative sensor algorithms, parallel architectures, engineering from algorithm to parallel implementation,

Common issues: life cycles, ITAR compliance





THALES

Embedded Systems life expectancy far exceeds that of their components

Processor generation

2 to 4 years

Car

10 years

Aircraft

25 years

Life expectancy

The reconciliation between those industrial discrepancies must stem from elsewhere: architecture, methods

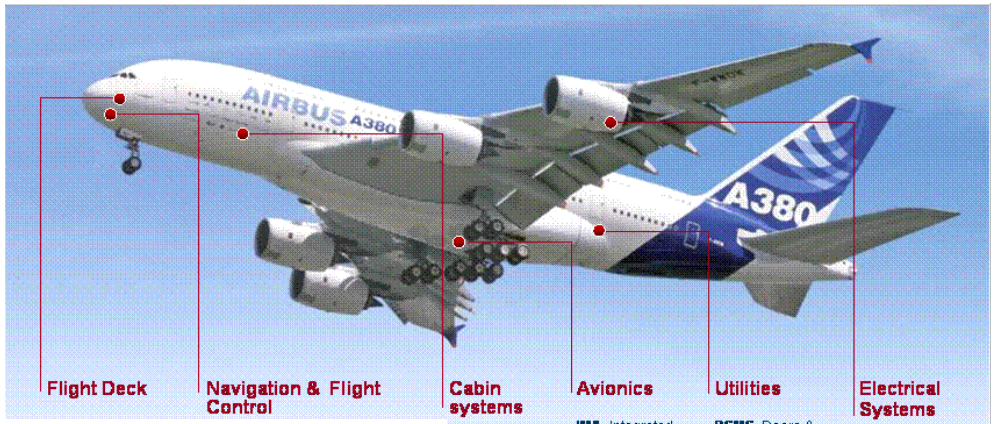
Introduction: Thales and Artemis

Artemis SRA defines 3 research areas with cross-cutting objectives between application domains:

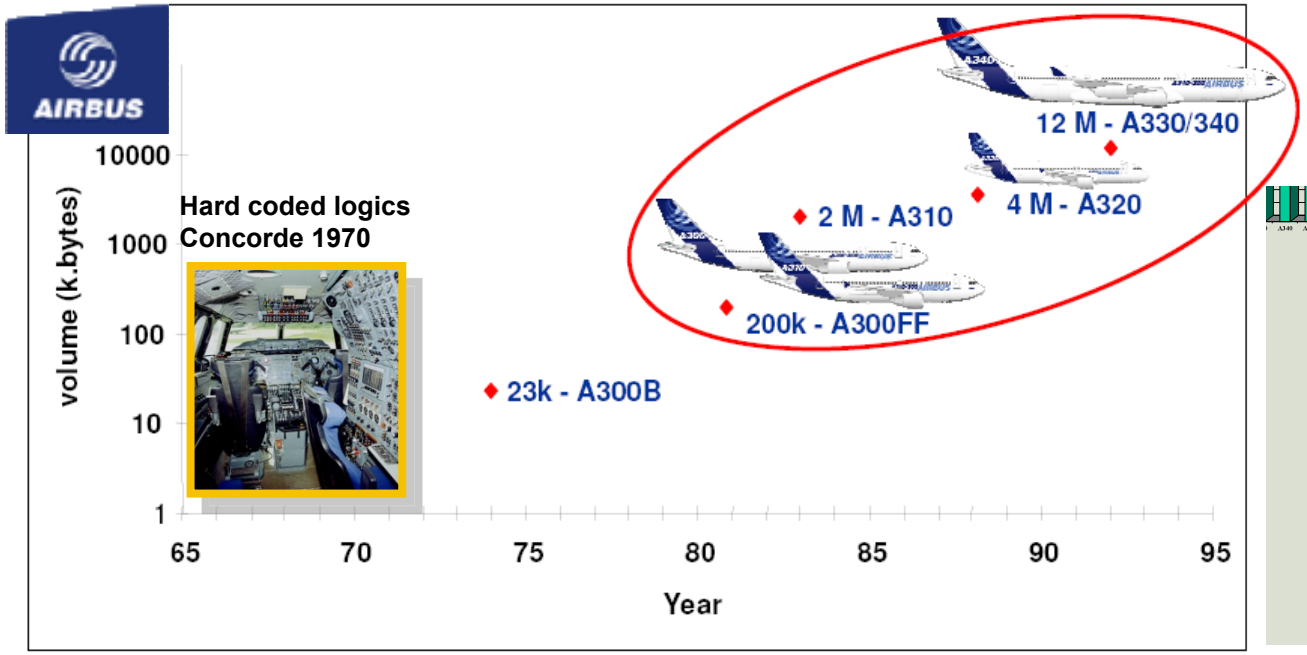
- Reference Designs and Architectures: high performance and dependability in the multi-manycore era
- Seamless connectivity, Middleware: focus on heterogeneous solutions to deal with RT
- Design methods & tools: beyond MDA, Domain specific solutions

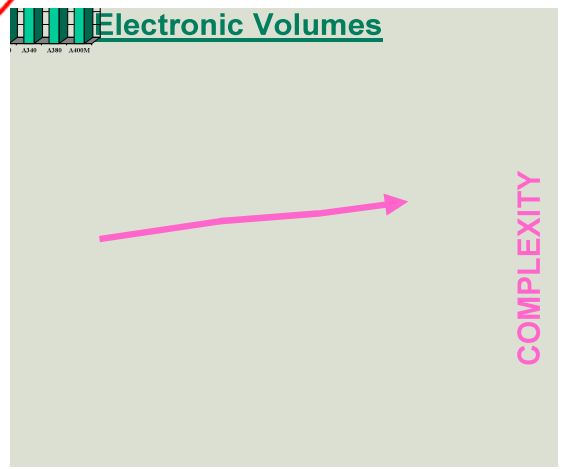
The diversity of the Thales embedded applications urges to increase in-house cross-cutting solutions between domains

Avionics Systems have gone to “generic”processors



A 380 : 80 Mbytes

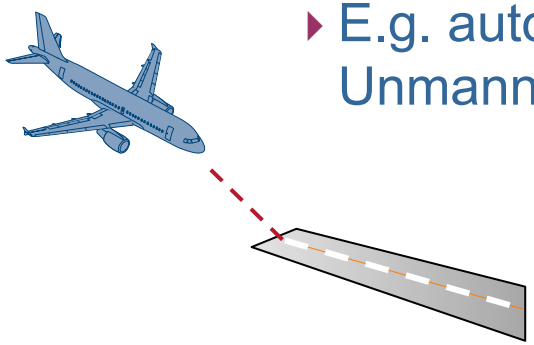


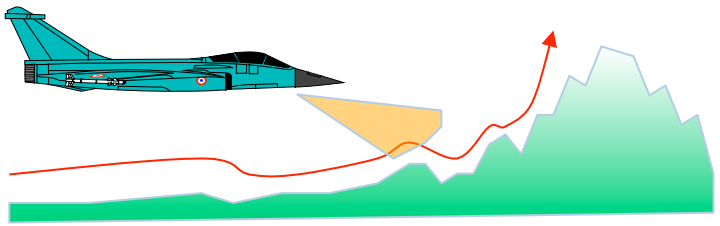


Software has become “intensive” to bring more functional value

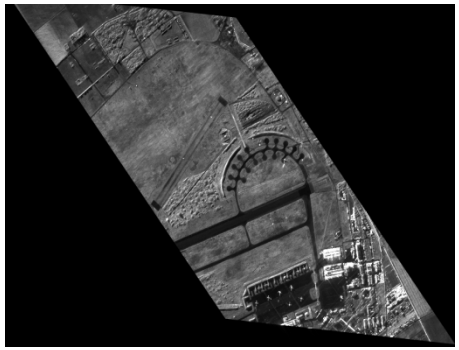
New hard real-time needs keep on appearing

- growing automation (transportation, industrial processes, ...)
  - E.g. auto-Pilots & Flight Control, Fighters Terrain Following, Unmanned Air Vehicles …





- new real-time media/interaction demanding QoS[1] (video, Positioning..)
  - E.g. Cameras & SAR radar video for observation, In-Flight Entertainment Video on Demand, anti-collision Systems (TCAS)…



[1] QoS: Quality of Service

Source : Jean-Luc Voirin
Aerospace Division

THALES

Some (new) challenges for future aerospace systems

- Extensive **Safety** constraints (raising due to Automation)
- Increasing **Vulnerability** of systems, due to complexity and openness
  - One system depends on **external Service Providers** (e.g. anti-collision TCAS, collaborative self-Protection, crisis management…)
  - Many **heterogeneous components** interact in a complex manner (see your own PC !)
  - **Different 'worlds'** begin to merge & interoperate (e.g. in-flight entertainment Vs avionics, Homeland Security…)

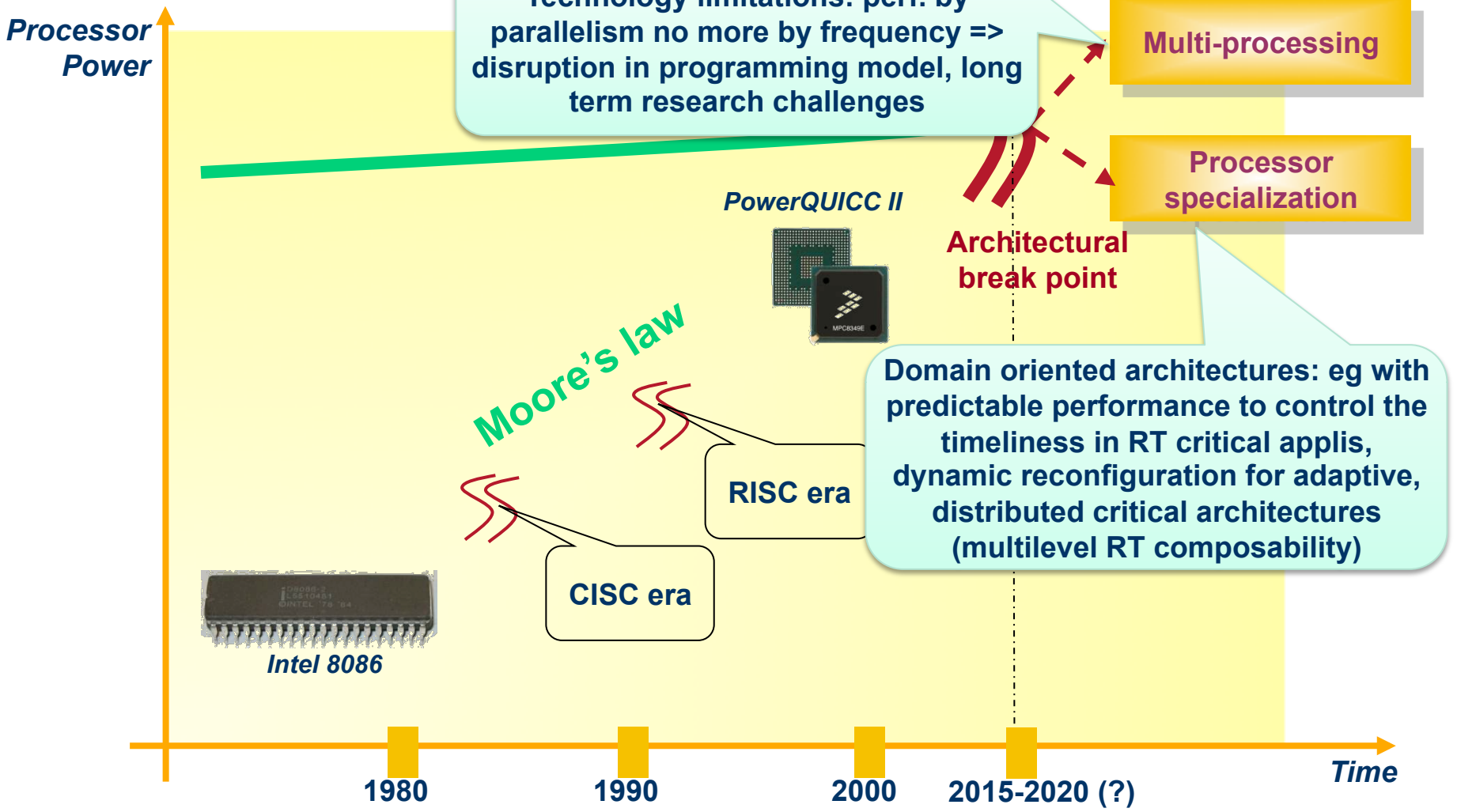
- along with growing **necessity of Security** due to sensitive information, automation or actions to be managed
  - E.g. Transportation Safety Vs Terrorism or unintentional jamming

Source : Jean-Luc Voirin
Aerospace Division

THALES

THALES

All ES are impacted by architectural breakthroughs

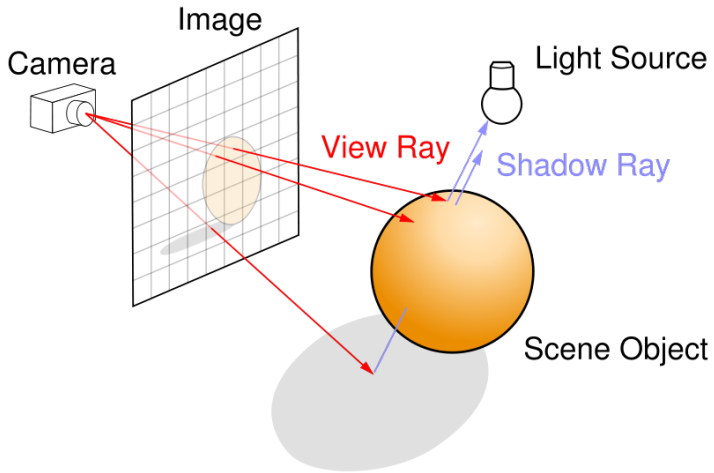


A major architecture disruption: multiprocessing and specialization will have a strong impact on software

THALES

Needs in High performance Embedded Computing

- New generation of complex algorithms: many future applications **will not yield to conventional stream processing** approaches
- Model-based algos will require physics-based computation and inferencing ideally suited for dedicated co-processors (e.g., GPUs and FPGAs)
- Not clearly super-computing. Rather computation intensive with short interaction with environment (typ. milliseconds)

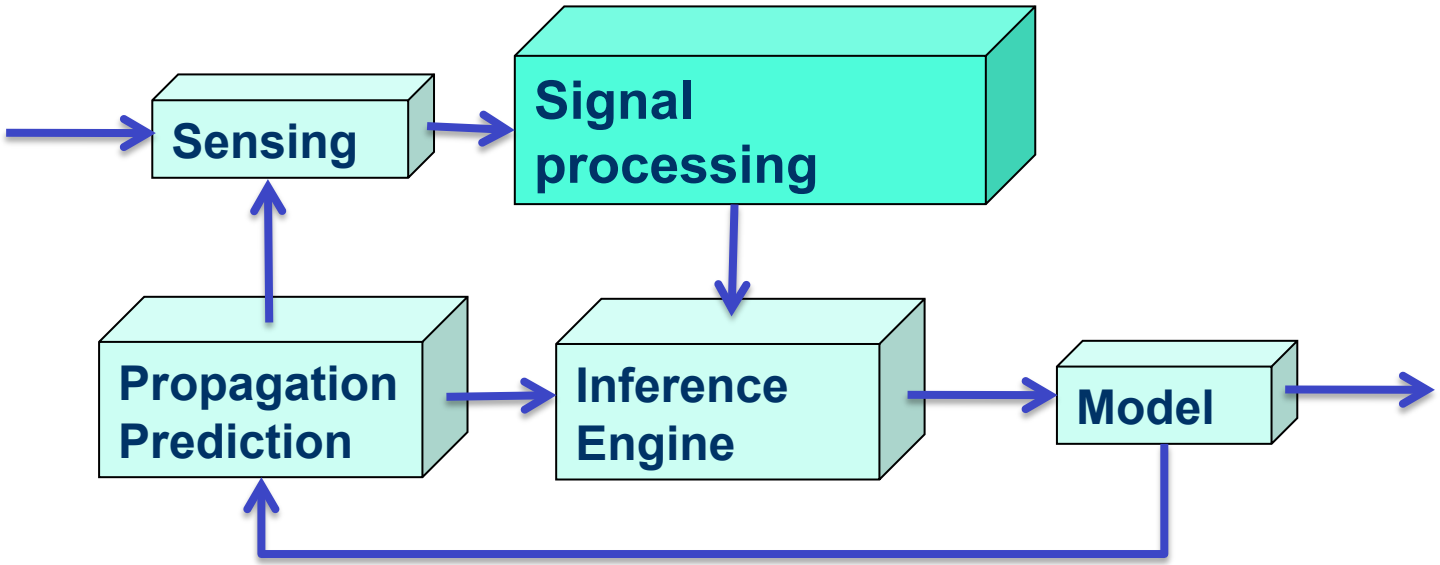


http://en.wikipedia.org/wiki/Ray_tracing_(graphics)

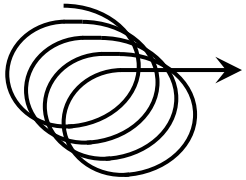
THALES

Next generation Signal Processing: new capabilities

from Baranovski (HPEC 2008)



Model-based reasoning



Model-based approaches might require many iterations on both the data stream and model-hypothesis generation

- **far more computations, and parallelism**
- **less static**
- **a chance of more data locality (Computations/IO BW)?**

THALES

Potential Impact of Multi-core processors in avionics

- In safety-critical applications
  - E.g. IMA for civil aircrafts
  - Drivers for safety-critical applications:
    - Determinism
    - Partitioning
    - Certification
    - And performances too, to address systems evolution

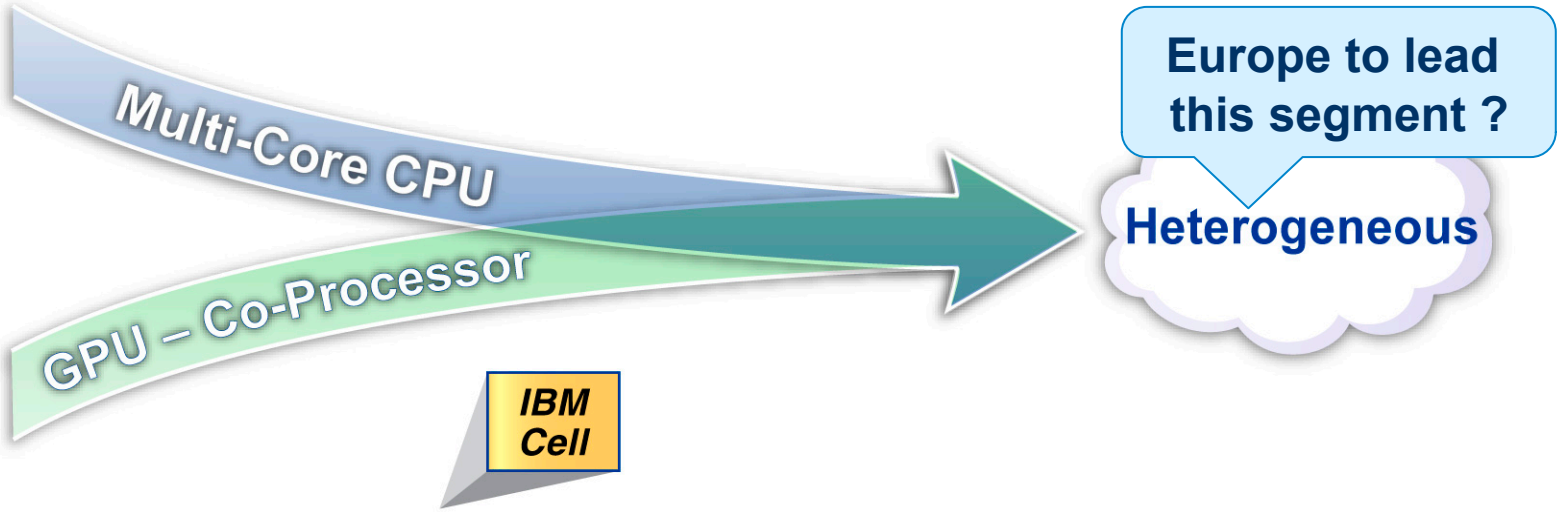
- In very high-performance systems
  - E.g. radar or electronic warfare for fighters
  - Drivers for these non safety-critical applications:
    - Very high performance
    - COTS compatibility
    - Standard software interfaces

Two main targets for multi-core use in Aerospace

THALES

An Industrial perspective in Embedded Computing

- **High performance COTS computing is moving to multi-core and heterogeneous silicon**



MULTI-CORE CPU	CURRENT	FUTURE
Multi-core CPU with smaller individual cores GPU co-processor	Multi-core CPU with 1-3 GPU co-processors Heterogeneous multi-core (IBM Cell)	Smaller, heterogeneous cores on same silicon

Technology access cost and versatility have pushed the use of FPGA accelerators

Source : General Dynamics HPEC07

The turning point : manycore and specialisation

- **Turning point for Turing machines** *(from MDR Report May 09)*
  - No general-purpose architectures anymore: "general-purpose“ = architecture not optimized for a class of workloads or embedded application
  - Moore's Law, the need to match speed with the real world, and the knowledge and tools available to create new architectures have turned the handful of engines of many years ago into thousands of specialized architectures and implementations
  - Today's architecture are composed of processors and accelerators. Mobile electronics and the need to conserve power will make next generations even closer to workload execution by dedicated hardware.
  - Moreover , economical conditions could drastically change the landscape of silicon solutions after 2015, beyond stand alone global foundries! *(iSuppli)*

Stake for Europe: lead embedded computing tomorrow: new directions are possible and necessary: simpler, with reliable performance

…and for Thales: participate to this renovation, conciliate this change in scientific and business conditions with operational requirements like life cycle time and costs, certification, technology access, etc.

THALES

It is thus up to Embedded SW to take up (at least) these challenges

Allow lifecycles "re-synchronization" by hiding (abstracting) electronic components on which they are built, while preserving the non functional properties

Convey the product added value by allowing continuous functionality addition (and taking advantage of the platform in an optimized way)

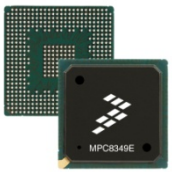
Leverage the "capability" provided by new (disruptive) computing architectures

Parallel programming of multi-cores « is the biggest challenge in computer science today » *(D. Patterson UC Berkeley)*

Leverage the capabilities provided by new architectures

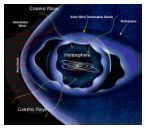
Challenging Embedded Computing

How to take advantage of strong parallelism of processing architecture to increase drastically the computation power density?



How to benefit from parallelism while maintaining "acceptable" programming principles?

How to ensure predictable performance and dependability on these complex architectures ?



THALES

Challenging requirements in Defence Systems

- **High-performance combined with environment constraints**
  - Massive parallelism emerging (Tilera, Monarch, Intel), making a breakthrough wrt current multi-core approach
  - Reconfigurable computing as candidate solution with challenges of insertion in multi-processing systems and dynamic reconfiguration: accelerators

- **Timing predictability (real-time & safety critical applications)**
  - Current approaches of more and more complex processor architectures and OS/middlewares may lead to uncontrollable designs if applied to multi-cores.
  - Major concern for many domains (commercial avionics, automotive, ..)
  - Possible solutions at architecture level (eg PRET machines, Merasa, Predator projects)

- **Dependability**
  - Build Reliable systems from unreliable components

Challenging requirements in safety critical systems

- **Avionics**: Additional functionality on board, higher comfort by better control of sensors and actuators, absolute guarantees on the timing (Do178B)
  - Solution: mixed application (hard and soft real-time constraints). Challenge: partitioning on multi/ many-core processors
  - Scalability and adaptability to a wide range of aircraft types while providing 100% availability through fault tolerance and reconfiguration capabilities. Reduced ownership, weight and size costs
- **Space**: Built-in self-test and self-repair, adaptive and reconfigurable hardware, related evolutionary algorithms, etc.
- **Rail Signalling**: one common modular platform for multiple applications, ensuring fault tolerance, safety and high availability, scalability of performance and costs
  - mechanisms for fault tolerance / redundancy handling in SW and HW

Increasing Challenge: repeatable timeliness

Temporal behavior is as important as logical function in ES

- Using standard IT Computing Platforms for high performance in Embedded is risky: optimisation is less cost effective than anticipated
- Performance of Computing Architectures are less and less predictable
- SW Interacting processes : too low level abstractions, complex, error prone, not compositional.
  - "The standard concurrency model, based on threads, semaphores, and mutual exclusion locks results in programs that are incomprehensible to humans". (Ed LEE)
  - SW abstractions (layers) discard the temporal properties provided by the HW

Future directions for improvement

- Precision networks (TTP , time synchronisation)
- Languages with timing (eg. Timing Definition Language, SNET)
- Predictable concurrency (synchronous languages?)

Combined with a new generation of predictable platforms

Software in Thales: key features

- Large range of applications
- SW critical information systems
- 12000 SW engineers
- Often (very) complex (millions lines of code)
- Cost and duration of SW development is crucial
- Large effort on structuring system and software engineering flow (MDA/MDE..)
- High cost and risk when changing languages or methods&tools
- Long life cycle of products: several releases, different programmers, different targets ..
  - strong pressure from Customers on life cycle costs (IMA, SW portability, technology transparency, ..)
- Only a part of applications goes to parallel, but constraints still apply...

Is parallelism SW or HW?

- Mapping/scheduling on parallel machines was originally an Hardware task:
  - hand-crafted code
  - needing specialists of the machine
- Large gains in productivity, portability, flexibility .. when moving towards architecture-naive, unifying Software approach
  - a strong request from Customer
  - enabled by middlewares and architectural tricks (HW caches, SMPs..)
  - collateral damages: processing efficiency, real-time guarantees
  - not applicable as such to multicores
- Sparse hardware-minded niches with FPGA (VHDL) evolving towards « System Level Design » from enriched C or Matlab

Embedded systems now need a new **mixed SW-HW education curriculums under the system banner**

Programming multicores

- « The biggest challenge … »*(David Patterson UCB)*
- multicores re-introduce SW-controlled scratchpads, distributed memories, memory hierarchy, ...Parallel complexity comes back to the surface.
- no longer 2 or 3 but many cores

- requirements from embedded systems industry (at least Thales):
  - programming should not be reserved to top gun specialists of every technology/machine: **the more tricks-free and straightforward the programs are, the better**
  - but reasonable performance and **timing guarantees** are needed

- how must we live with multicores?
  - seems that no magic tool exists so far
  - what must be the involvement of the programmer in the design flow?

The bad new: compilers can't do all of it for us

- Complete and standard tool chain providing both predictability and acceptable performance on multi/manycores for current general-purpose C++ or Java applications would be the ideal solution
- ... but we need to be patient and/or believe in miracles as seen from today's state-of-the-art: solutions must be reliable, no doubt is allowed
- Compilers don't (and probably won't) accept programs in current imperative languages (C/C++, Java, ..)
  - but probably not a good idea to propose a brand new language for universal adoption

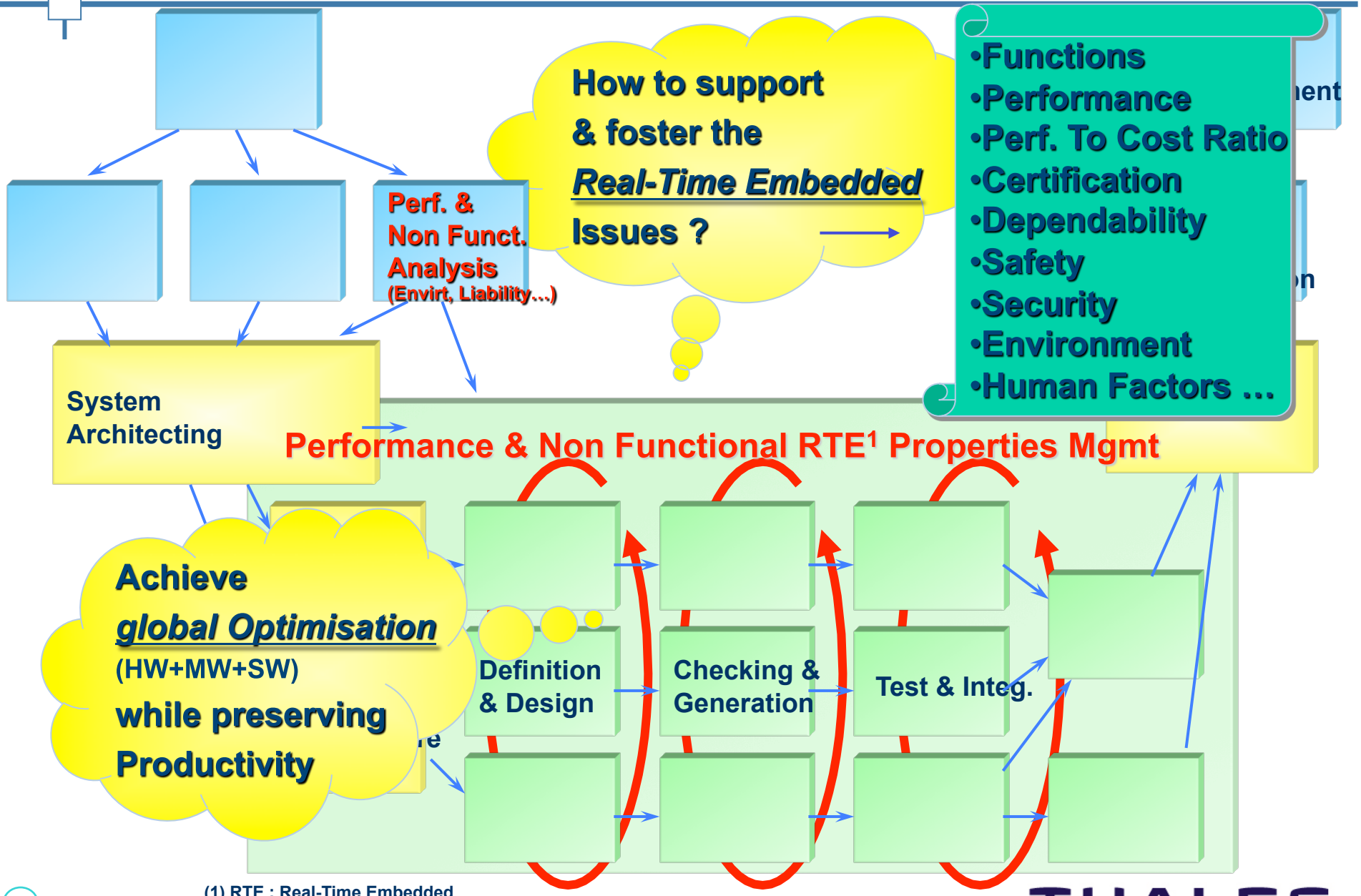
Tools that start to work

- **restrict the scope of applications**, e.g. graphic, data-flow, i.e. where parallelism is relatively present and visible
- **impose strict coding rules on C, plus specific extensions**
  - Tool performance depends on the coding method and the know-how of the tool user wrt the tool

MoC : a pivotal technology in know-how driven engineering

- **Principle:**
  - Engineering for Embedded Systems: mostly isolated islands of engineering activities (multiview)
  - Central concepts and high level frameworks necessary to share across activities and enable support for these shared concepts across tools
  - Abstract modeling approach uses meta-models to define the shared concepts of a domain, and to establish tool chains
  - These domains bring their own (large) set of concepts: MoC
    - provide formal definitions of how system elements (hw or sw or purely behavioral) interact and communicate.
    - MoC allows mathematical reasoning about the properties of the assembled system components, e.g. parallelizing an application
  - The missing link between MDE and ESL
  - A first step toward an extended “Mead & Conway” approach for Embedded Systems engineering?

Real Time Embedded Systems Design : the Challenge

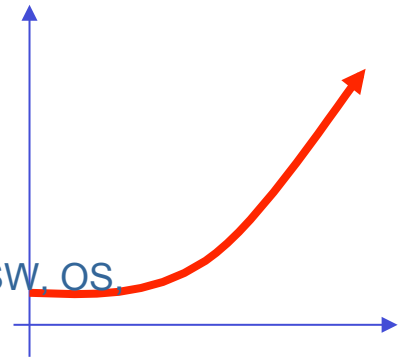


(1) RTE : Real-Time Embedded

THALES

Industrial sources of complexity

- **Internal complexity** gap and **Industry segmentation** will result in huge increase of:
  - Engineering data (Requirements, functions, interfaces, norms, …)
  - Team sizing
  - Massive reuse, of owned or foreign components (HW, SW, OS, MW, tools)
  - Product policy constraints



- **External constraints & Human Factors** will add to the former:
  - Reduced time to market
  - Sub-contracting, Partnership
  - Integration of not mastered foreign components
  - Solutions & tools complexity

This will also lead to Increasing **Engineering Complexity**

Suggestions of transverse Research Themes

To be achieved	*Research Themes*
Formalise, Model, Architect	MetaModels to support RTE, non functional Issues Mgmt Architecting Methods & Patterns to reconcile them
Make Solution Emerge	Methods, Techniques & Tools to aid Solution Definition : Domain Rules, Model Transformation, algorithms, multi-criteria optimisation, auto-organisation, …
Check internally	Meta-Knowledge on Models Conformance and Correctness, Associated Aids & Tools, Model checking; Executable Models; Simulator Generation
Assess, Evaluate **wrt Need/Definition**	Model Transformation / Mapping, high-level Impact Analysis Methods & Tools
Prove, Demonstrate	Formal Languages, Proof Techniques applied to RTE non functional Issues
Exploit, Produce	Model Transformation, Generation, multi-target Generation; Test Model Generation

From Model-driven to Know How-driven Engineering

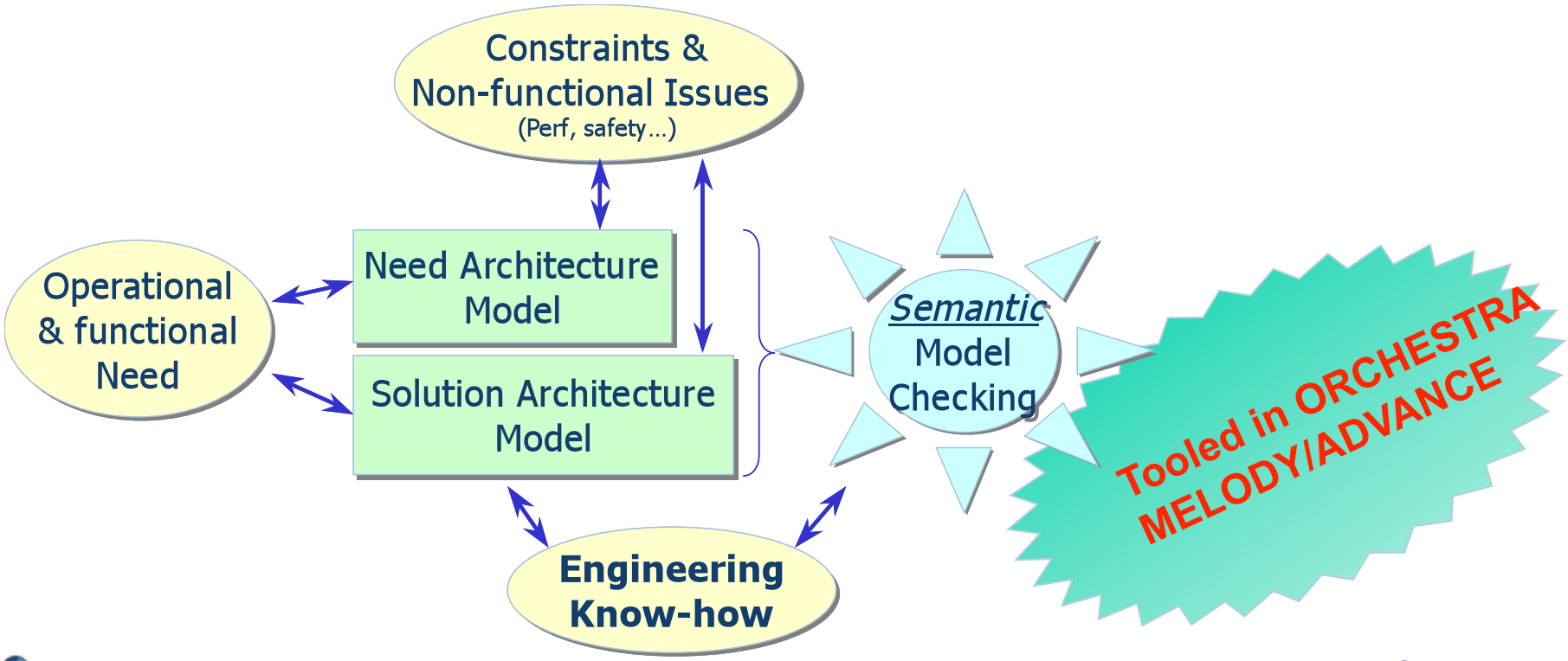
- **Key is capturing the way of reasoning to solve the problem:**

Models hardly cover this part!

- **Domain-specific languages** (DSL) to express
  - Domain-related concepts (and not generic 'meta-concepts')
  - Their contribution to elaborate or evaluate solution

- Domain-dedicated **Knowledge base** of 'Solution Patterns'
  - Not only reusable model parts but rules & means to use
  - Quantitative metrics, 'merit factors'…
- Domain-dedicated **engineering 'Rules'**, to aid assist or automa
  - Expression of expectations/constraints on solution
  - Means to build solution (incl. Model generation/transformation)
  - Means to evaluate & assess it (incl. Model analysis rules)
- Ability to *customise, capitalise, check & update* this Know-how

The ARCADIA Method: an attempt to go beyond Modelling



- **Ability to capitalize architectural patterns, engineering rules & Know-how**
- **First Decision Aids, by**
  - **Automatic analysis of architecture properties**
  - **Multi-viewpoint reconciliation to find the best optimum**
  - **Semantic impact analysis thanks to dedicated metamodel & Rules**

Embedded Systems Engineering is a new discipline

- Although futuristic in some cases, these issues are **definite enablers** for Model-driven Engineering success in complex, critical Systems
- They show that MDE should not be considered as an end in itself, but one means among others to solve new engineering challenges
- In order to really manage complexity, **accept to give up genericity, and to address complexity on a *semantic*, domain by domain basis**

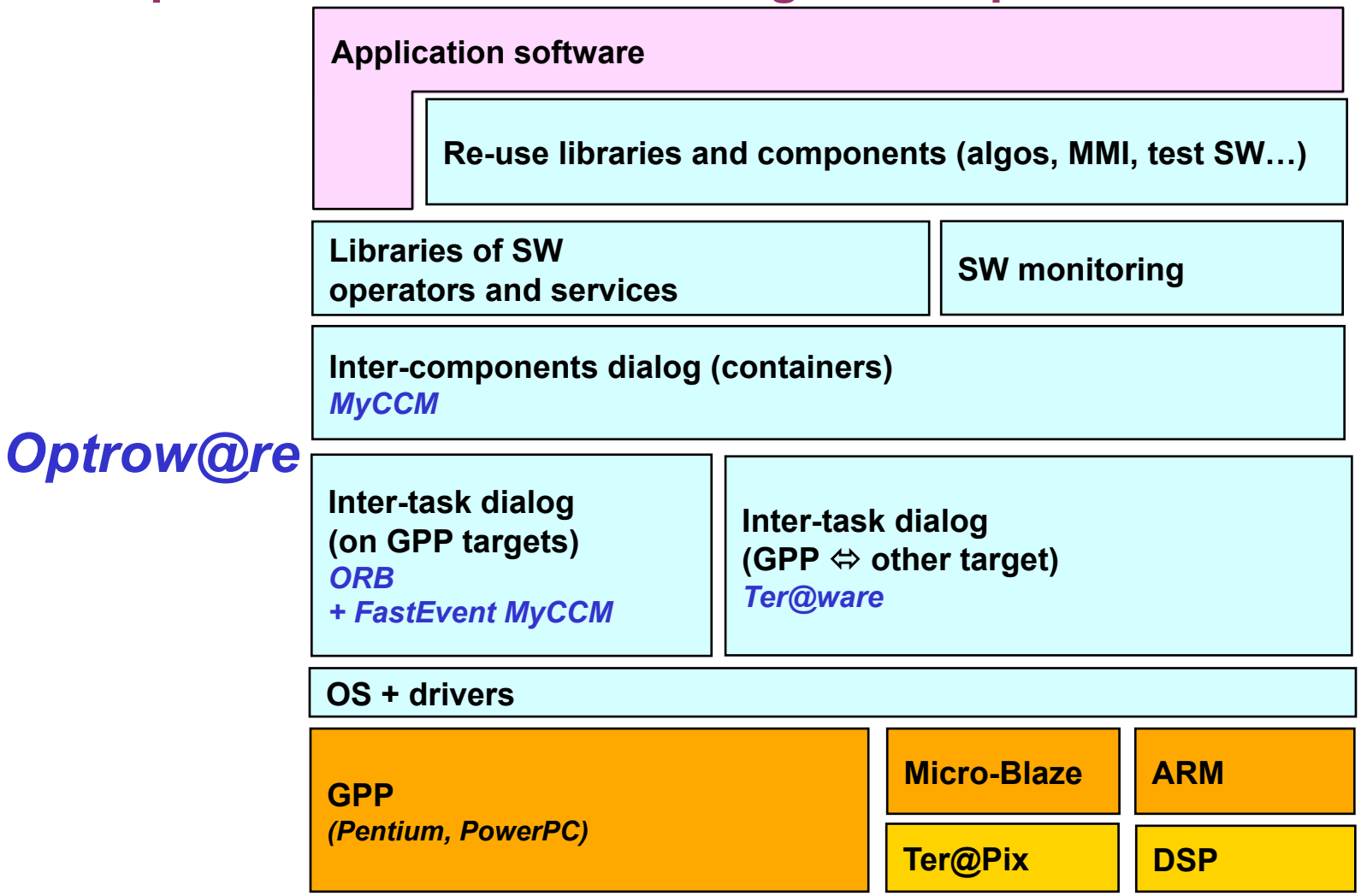
- **Hopefully they might help in**
  - broadening views about engineering, far beyond MDE,
  - and encourage to consider the whole 'big picture' of complex systems &software engineering,
- **to get the best out of MDE Research must go on!**

Seamless Connectivity and Middleware for RT ES

- Fragmented research and domain specific standards in an multiform, non stabilized market: Autosar, Arinc, SCA, etc.
  - Various Qualities of Services: RT constraints, safety, modularity, certificability, etc.
  - Few generic standards: LwCCM, DDS
    - identify cross domain research, and improve interoperability
- Layered approaches usable for distributed systems with poor RT constraints must be carefully evaluated regarding timeliness to take care of not discarding all properties of the underneath layers.
- Challenges include:
  - Composability: long term challenge for non functional properties
  - System integration: Interoperability between different domains
  - Integration of MDE and component frameworks like SCA
  - Hierarchical services: application specific (PF independence), domain specific + domain independent (basic services, API)
  - Solution for Self adaptive systems

An example of use in a real time application

- **A multiform approach to combine standards and performance on an heterogeneous platform**



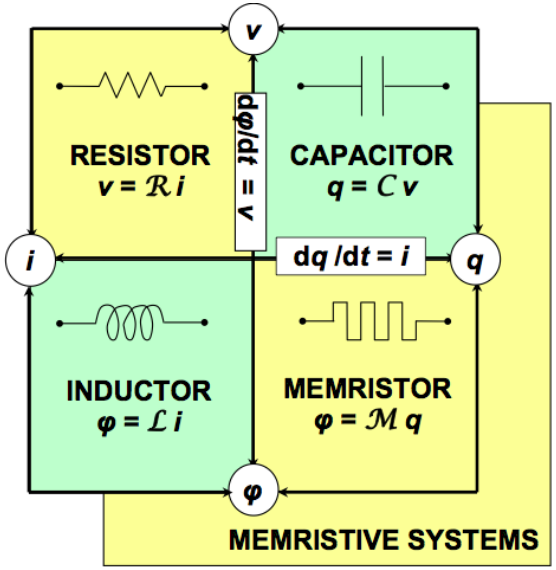
- **The architecture evolution toward parallel and domain specific architectures change the landscape**
  - Stakes: harnessing this new capabilities for high performance and safety critical applications (timeliness, dependability), developing optimised platforms, etc.
  - Challenges: programming, programming model, legacy, etc.
  - Opportunities: new algorithms become affordable, etc.
- **The multidomain context and the trend toward more complexity urges to explore advanced tracks in designing high end mission critical embedded systems:**
  - A multidisciplinary model based engineering process
  - Using the most appropriate and sound semantics to capture the know-how: formalisms, DSML, etc.

We have to learn how to build robust systems from less and less reliable technologies and roadmaps!

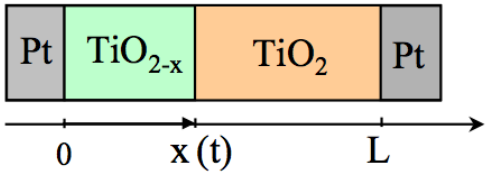
To conclude…

- **Future is not written!**
- **E.g.The architecture evolution toward parallel and domain specific architectures change the landscape**
  - Stakes: harness this new capabilities for high performance and safety critical applications (timeliness, dependability), develop optimised platforms by markets, affordable technology access etc.
  - Challenges: programming, programming model, legacy, etc.
  - Opportunities: new generation of algorithms become affordable, MoC & DSML promise to change the picture in engineering, etc.
  - Highly disruptive promising technologies appear which could change the computing game in 5 to 8 years: **memristors**

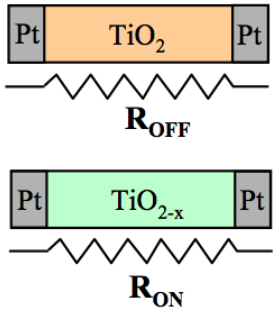
Memristor: the promising missing Element



L. O. Chua, "memristor – the missing circuit element" IEEE trns. Circuit Theory 18, 507-519 (1971)



$$R = R_{ON}\frac{x}{L} + R_{OFF}\left(1 - \frac{x}{L}\right)$$



Nanoscale devices with unique properties:

✓Variable *Non volatile* resistance

➔ Used as memory elements

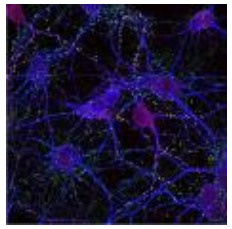
THALES

Disruptive potential capabilities!

- **Cognitive computing! Artificial Neural Network**
  - E.g. New capabilities for autonomous sytems

Synapses

10^{10}/cm^2 (human cortex)

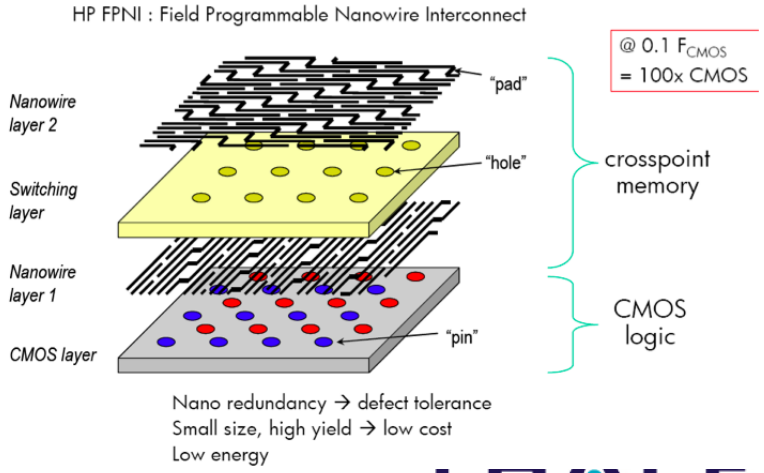


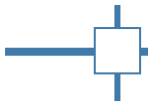
Memristor

10^{11}/cm^2 (30x30nm^2)

- **High density and power efficient reconfigurable circuits (next generation FPGAs?)**

  - **Memristors used as nonvolatile configurable switches**
  - **Density and power >>**
  - **Nonvolatile configurations**





Thank you for your attention!

THALES