

The ARTEMIS Platform

H. Kopetz
April 2006

The Message

Computer technology is in the middle of a major paradigm shift, similar to the transition from the *Mainframe* to the *Personal Computer* twenty-five years ago.

- ◆ Introduction
- ◆ What is ARTEMIS?
- ◆ The Technology Landscape
- ◆ Ambient Cosmic Radiation
- ◆ The Seven Key Challenges of ARTEMIS
- ◆ Conclusion

ARTEMIS

In preparation for FP 7, the EU has initiated the ARTEMIS (*Advanced Research & Technology for EMbedded Intelligence and Systems*) Platform, a public-private partnership led by European industry with the goal *to establish and implement a coherent and integrated European research and development strategy for Embedded Systems.*

On June 30, 2005, ARTEMIS has published a *Strategic Research Agenda (SRA)* that outlines the objectives and the research topics that need to be investigated in the field of embedded systems.

One important research domain of the SRA of ARTEMIS relates to the development of generic *reference designs and architectures* for embedded systems. The ARTEMIS subgroup on *Reference Design and Architectures* has identified the following seven key challenges in this area.

The Three ARTEMIS Expert Groups

- ◆ Reference Design and Architecture
- ◆ Seamless Connectivity and Middleware
- ◆ Design Methods and Tools

SRA-- Reference Design and Architectures

According to the SRA of ARTEMIS (p.16) the objective of the WG on *reference designs and architectures* is

The creation of a generic platform and a suite of abstracts components with which new developments in different application domains can be engineered with minimal efforts.

Methodology to Arrive at the Research Priorities

The working group has adopted the following procedure to define these research priorities:

1. Collection of concrete requirements and constraints that were considered relevant from the point of view of different application domains
2. Classification of the collected requirements and constraints in order to find communalities and differences among the different application domains and to establish a limited number of categories of appropriate abstract requirements and constraints
3. Evaluating the state of the art in meeting these requirements and constraints
4. Identification of research priorities and their importance in time and relevance

Members of the Expert Group

Sergio **BANDINELLI**, European Software Institute

Andrei **Bartic**, IMEC

Christian **Bettstetter**, Alpen-Adria-Universität
Klagenfurt

Michael **Borth**, DaimlerChrysler AG

Ed **BRINKSMA**, Embedded Systems Institute

Tom **CLAUSEN**, CEC

Jean-Luc **Dormoy**, Commissariat à l'Energie
Atomique

Gilbert **Edelin**, THALES

Christian **EL Salloum**, TU Vienna(Editor)

Alun **Foster**, STMicroelectronics

Laila **Gide**, Thales

Magnus **Granström**, Volvo

Riccardo **Groppo**, CRF

Peter **Heidl**, Robert Bosch GmbH

Knut **Hufeld**, Infineon

Hermann **Kopetz**, TU Vienna (Editor)

Kimmo **Kuusilinna**, Nokia

Vera **Lauer**, DaimlerChrysler AG

Per **Lindgren**, Lulea University of Technology

Roman **Obermaisser**, TU Vienna

Ton **PEERDEMAN**, Thales

Ian **PHILLIPS**, ARM

Peter **Puschner**, TU Vienna

Christophe **RENAUD-BEZOT**, Thales

Herbert **Rödig**, Infineon

Fulvio **RUSINA**, COMAU

Hans **Schurer**, Thales

András **Tóth**, Ericsson AB

Theo **Ungerer**, University of Augsburg

Mateo **VALERO**, Technical University of
Catalonia

Sjir **van Loo**, Philips Research

Johan **VOUNCHX**, IMEC

The Technology Landscape

- ◆ Hardware
- ◆ Communication
- ◆ Embedded Software
- ◆ User Expectations

The Technology Landscape--Hardware

- ◆ The limits of *Moore's* law are becoming visible: power dissipation, physical feature size, reliability.
- ◆ The *memory wall* gets higher and higher (about 80 % of the transistors in the *Itanium* chip are for caches).
- ◆ The performance increase of a single processor from one generation to the next is proportional only to the square root of the increase in silicon area (*Pollack's* rule).
- ◆ The transient failure rate of sub-micron devices is increasing [both single-event upset (SEU) and single-event transient (SET)].
- ◆ Multi-computer chips (SoC) area appearing. The development cost of such a chip can pass the 100 Mio \$ wall--mass markets are needed to justify this level of investment.

Example: Cell Processor

Joint development of IBM, Sony and Toshiba, 90 nm process 250 Mio Transistors, 221mm², 4 Ghz, 250 GFLOPS, Development Cost > 400 Mio \$

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

The Cell contains eight SPE Computers

Specialized Processor for SIMD-type data streams

256 Kbytes of private memory (LS0-LS3), access latency is 6 cycles

32 bit instructions, 128 registers

Ca 15 mm²

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

BIU less than 1 mm²



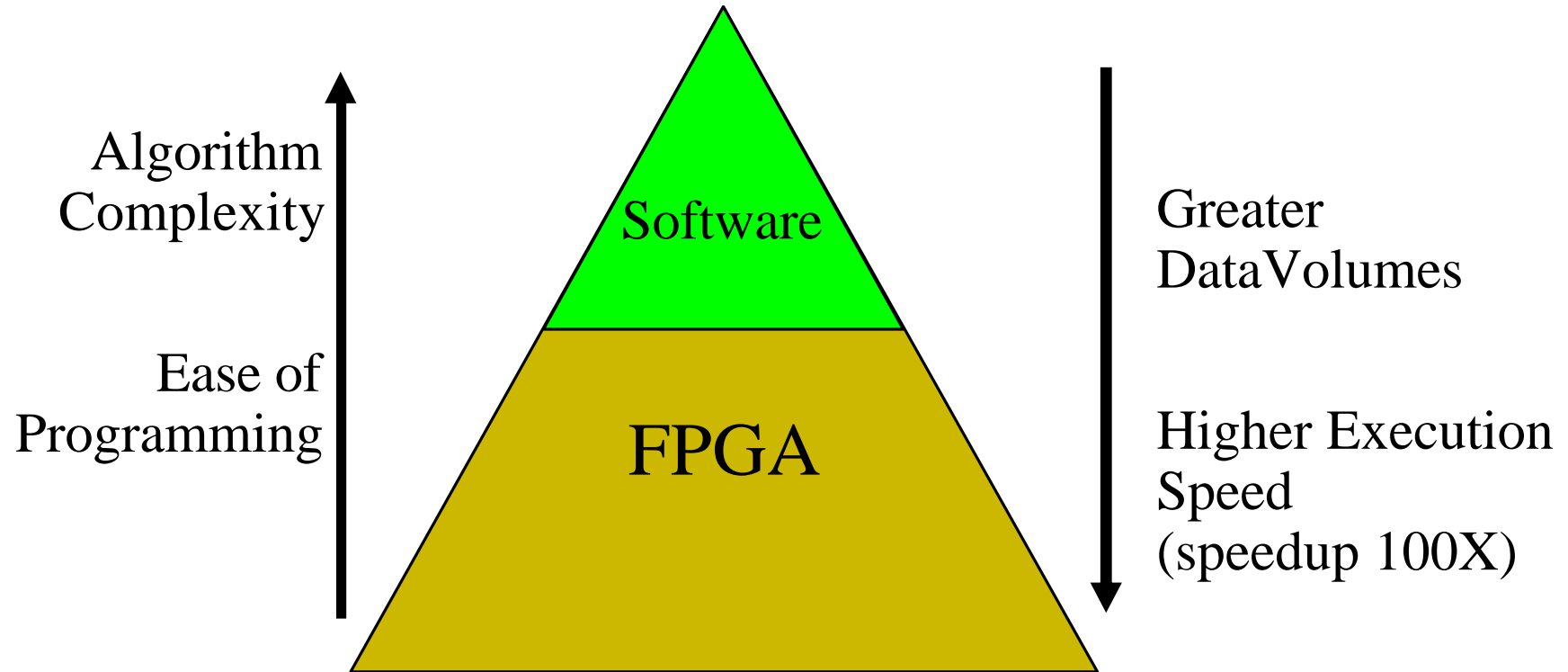
The Technology Landscape--Communication

- ◆ The widespread availability of a wireless communication infrastructure enables the ad-hoc detection and integration of services without any physical action--the coming of *situation aware* systems.
- ◆ Seamless integration of *Radio Frequency Identification* (RFID) technology with embedded devices (e.g., cell phones).
- ◆ Flexible transmission technologies (e.g., *spread spectrum, ultra wide band, frequency hopping*) that allow multiple users to share a given frequency band with minimal interference and reduce the power required per bit transmitted.
- ◆ Waveform-agile transmission methodologies (*cognitive radio*) that enable a wireless device to create ad hoc different types of communication links under software control.

The Technology Landscape--Embedded Software

- ◆ *Uncontrolled system complexity*: The costs of design, verification, integration and maintenance are getting prohibitive.
- ◆ *Component-based design* elevates the design process to a higher level of abstraction--but many key issues are still open, e.g., the precise specification of component interfaces, the identification of *fault-containment units*.
- ◆ *The investment in software* is more long-lasting than the investment in hardware.
- ◆ *Security becomes a key issue*-- as embedded devices are integrated into the Internet, particular in wireless systems.
- ◆ The clear distinction between software and hardware is disappearing, e.g., *power-dissipation* is becoming also a software issue, not only in battery-operated devices.

New Implementation Choices



Example: Look for keywords in a set of documents:

(A and B) or (C and D)

Search for the occurrence of A,B,C,D in FPGA, connect the results in software

From: R. Chamberlain, Embedding Applications within a Storage Appliance Proc. HPEC 2005, p. 2

User Expectations

- ◆ In a large embedded system that consists of a vast assembly of networked components that must operate 24 hours per day for 365 days per year the *occurrence of transient and permanent failures* of components and interconnects must be *considered the norm*, not the exception. Future system must thus include strategies and mechanisms that assure that the reliability of the *user-perceived system services* remains at an acceptable level despite the occurrence of these failures.
- ◆ In an ambient intelligence scenario, where a multitude of diverse embedded devices is fielded in a home, it cannot be expected that the end-user is willing to spend her/his time and effort to troubleshoot a misbehaving distributed embedded system. A system must thus be capable to diagnose its own faults and guide an untrained user to repair the system with minimal effort.

The Effects of Ambient Cosmic Radiation

The neutrons of the ambient cosmic radiation interact with the atoms of the semiconductor devices, giving rise to *soft errors*:

- ◆ Create *electron-hole pairs* that interfere with the electric charge that denotes the information contents of cell (*bit-flip*). Since this electric charge decreases with smaller feature size, disturbances become more probable when the feature size shrinks.
- ◆ Affected area in the order of a few μm^2 . Duration in the nano-second range (i.e., the duration of cycle time of a modern CPU).
- ◆ Single bit-flip most probable, double-bit failures possible, both storage (SEU) and logic (SET). No permanent failure of the device.
- ◆ At present, the sea-level failure rate is about 1000 FITs/megabit (1 FIT = 1 Failure in 10^9 hours, about 100 000 years)
- ◆ SoC failure rate of 1 000 000 FIT: one event per month, but not every event causes a severe failure (e.g., pixel on a screen).
- ◆ Increase: 3-5x at 1500m, 10x at 3000m, 100x at 10 000m

Mitigation Strategies w.r.t. Soft Errors

It is hardly possible to shield a device from the ambient cosmic radiation, however the effects of this radiation can be mitigated on different levels:

- ◆ Material selection to reduce the neutron interaction coefficient (Example: *SoI Silicon in Insulator*)
- ◆ Layout of electronic devices: larger devices
- ◆ Radiation-hardened circuit design
- ◆ Error detection and correction for SEU (e.g., 64 bit words requires 8 additional bits). Mitigation of SETs more difficult.
- ◆ High-level architectural means: triple modular redundancy (TMR) in space and/or time.

Integrity-Level of Application Domains

Application	System MTTF w.r.t. permanent failures (in years)	System MTTF w.r.t. transient failures (in years)	Data-integrity requirement	Market volume	Examples
Low-Integrity	> 10	> 1	low	huge	Consumer Electronics
Moderate-Integrity	> 100	> 10	moderate	large	Present-day automotive
High-Integrity	> 1000	> 100	very high	moderate	Enterprise server
Safety-Critical	> 100 000	> 100 000	very high	small	Flight control

The Dilemma

- ◆ The consumer electronics (CE) domain has the size to support the large development costs needed to build powerful SoCs.
- ◆ Since in the near future there is no need to harden CE chips to mitigate the consequences of ambient cosmic radiation, the CE industry will not pay extra for hardening their chips.
- ◆ Architectural mitigation strategies have to be developed such that replicated mass-market chips can be used to build high-integrity embedded systems.

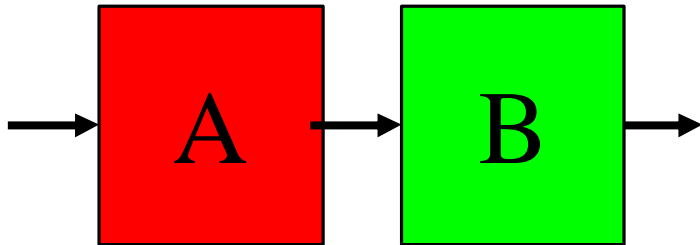
Mitigation of Soft Errors by Architectural Means

Architectural means to mitigate the consequences of component failures might become a necessity when using the upcoming submicron devices, as stipulated in the latest 2005 *International Roadmap of Semiconductors* p.6:

Relaxing the requirement of 100% correctness for devices and interconnects may dramatically reduce the costs of manufacturing, verification and test. Such a paradigm shift is likely forced in any case by technology scaling, which leads to more transient and permanent failures of signals, logic values, devices and interconnects.

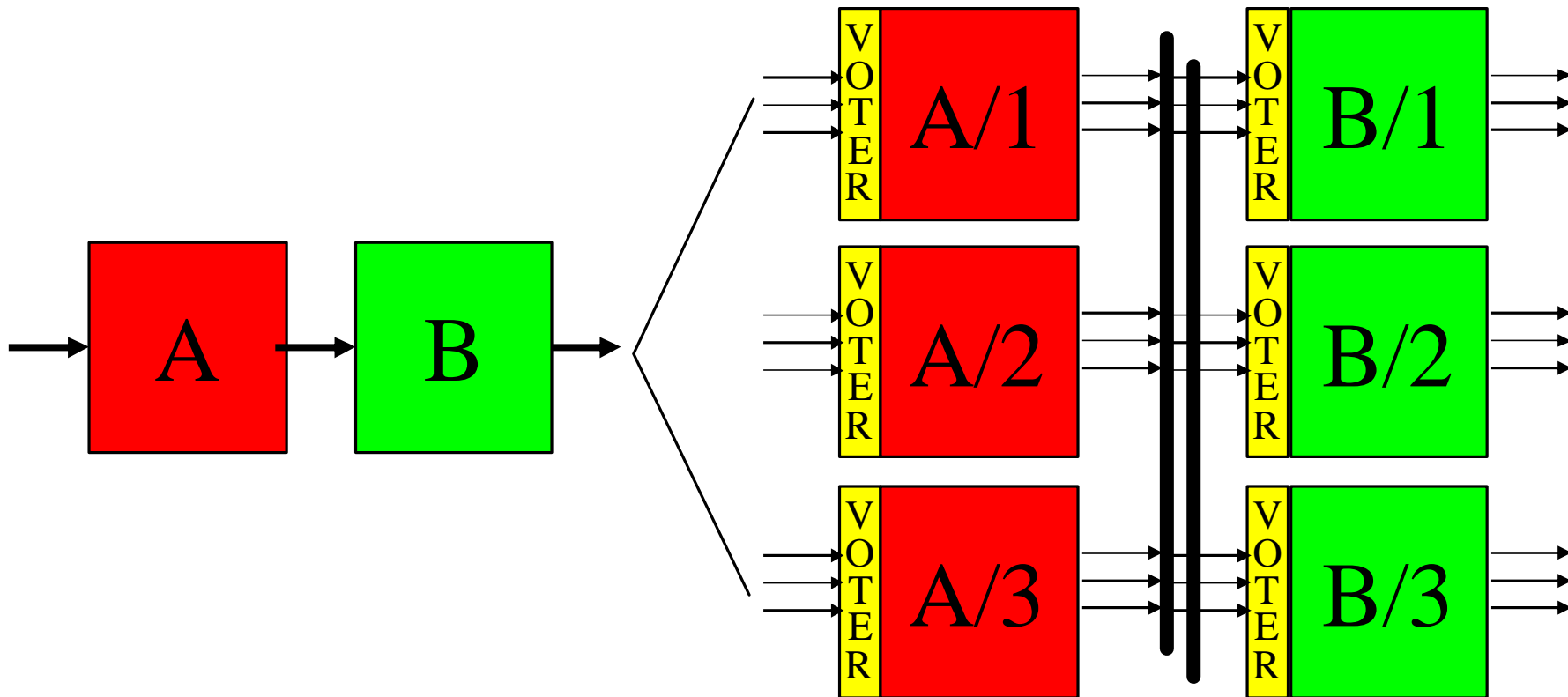
Mitigation at the Architecture Level: TMR

Triple Modular Redundancy (TMR) is the generally accepted technique for the mitigation of component failures at the system level:



Mitigation at the Architecture Level: TMR

Triple Modular Redundancy (TMR) is the generally accepted technique for the mitigation of component failures at the system level:



What is Needed to Implement TMR?

What architectural services are needed to implement Triple Modular Redundancy (TMR) at the architecture level?

- ◆ Provision of an Independent Fault-Containment Region for each one of the replicas
- ◆ Synchronization Infrastructure
- ◆ Multicast communication
- ◆ Replicated Communication Channels
- ◆ Support for Voting
- ◆ Timely and Deterministic Operation

The Seven Key Challenges of *ARTEMIS*

- ◆ *Composability*
- ◆ *Networking and Security*
- ◆ *Robustness*
- ◆ *Diagnosis and Maintenance*
- ◆ *Integrated Resource Management*
- ◆ *Evolvability*
- ◆ *Self Organization*

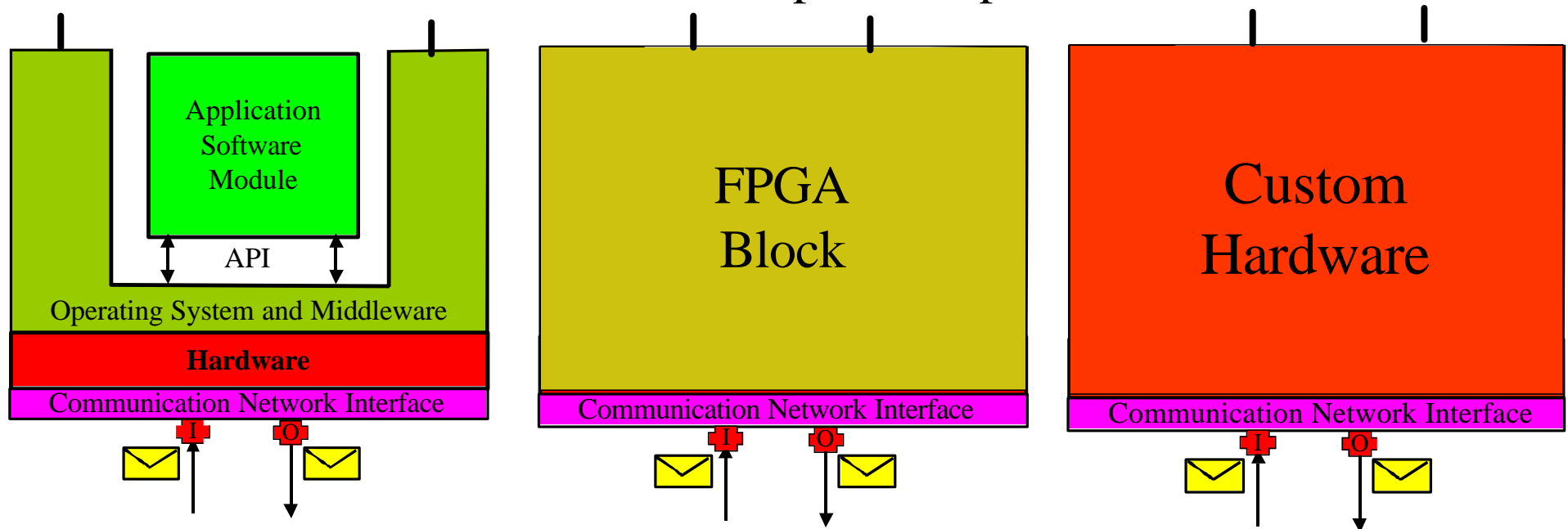
And all of this under the constraint of
Follow established standards

Composability

- ◆ Techniques for the precise specification of the interface behavior of a component (value and temporal) both at the syntactic and semantic level, including formal and semiformal interface models that are expressed in tool-processable metadata with precise semantics.
- ◆ Encapsulation of self-contained subsystems, comprised of assemblies of communicating components, such that any unintended interference (temporal, value) between unrelated subsystems is excluded by architectural means.
- ◆ Conceptualization and characterization of the aggregate properties of communicating component assemblies in order to form a *new self-contained subsystem*, thus elevating the level of abstraction to a higher level.

Different Types of Components

Local Interfaces--Open Components



The Communication Network Interfaces (CNI) of all three different types of system components should have the same *syntax, timing* and *semantics*. For a user, it should not be discernible which type of system component is behind the CNI.

Networking and Security

- ◆ Architectural support for *structured name spaces*. Dynamic and trusted *name address binding* such that moving entities can maintain a disruption-free connection in a mobile environment. Routing algorithms that are based on *names* rather than *addresses*.
- ◆ Communication protocols that give the user the option to make a choice between latency, jitter, throughput, determinism, reliability, security, and power efficiency.
- ◆ Architectures that provide a *global time service* and predictable transport mechanisms for *pulsed real-time data streams* within control loops across different types of networks.

Networking and Security (ii)

- ◆ Ad hoc wireless networks that provide sustained and trusted performance to smart sensors, actuators and other embedded system nodes in harsh environments (e.g., *car-to-car* and *car-to-infrastructure* networks).
- ◆ Automatic security management (authentication, authorization, privacy, confidentiality of information, protection of intellectual property, digital rights management) of trusted embedded environments considering the limited resources (energy, bandwidth, power) of embedded nodes.
- ◆ Secure authentication infrastructure for embedded entities that cannot be forged by malicious intruders and can be used by the authorized applications to control access privileges.

Robustness

- ◆ Development of fault-tolerant architectures that mitigate the effects of *soft errors* (e.g., the transient corruption of a single bit by low-energy cosmic neutrons) at the system level. Architectural support for error masking, such as error-correcting codes or triple-modular redundancy (TMR), for critical system services.
- ◆ Self-configuration, reconfiguration and self-commissioning of systems in a novel environment without explicit user interaction.
- ◆ State-aware design techniques that always keep track of the *critical system state* and provide mechanisms to detect and repair corrupted state within a short error-detection latency.
- ◆ Quantitative fault models, including failure modes and failure rates for sub-micron VLSI devices, components and subsystems.

Diagnosis and Maintenance

- ◆ Provision of a framework that supports the observation of component behavior and the identification and isolation of faulty components. Independent on line checking of *pre-conditions on input data* and *post-conditions of output data* of components.
- ◆ Architectural techniques for in-system test generation and on-line testing without the probe effect. Provision of the capability for built-in test (BIST) at the system level.
- ◆ Architectural means and novel algorithms for the on-line analysis of diagnostic data to reduce the diagnostic data volume and arrive at reliable information about the current *health state* of the components.
- ◆ Support for non-intrusive on-line auditing: error detection, intrusion detection, detection of misconfigurations. Provision of an auditing framework that is customizable by the user.

Integrated Resource Management

- ◆ Development of a framework where multi-voltage zones and multi-clocking zones can coexist on a single system-on-a-chip (SoC) and where the voltages and clock speeds in the individual zones can be controlled by software.
- ◆ Algorithms that accurately estimate system-level power requirements, both the peak power and the integral power needed to complete a task.
- ◆ System-level resource management algorithms (power, execution time, bandwidth, memory) that dynamically allocate resources to tasks such that the deadlines of all time-critical tasks are met and the given budgets for resource–usage are observed.
- ◆ Power-aware algorithm design: Development of algorithms and execution environments that optimize the power usage. Power aware scheduling algorithms.

Evolvability

- ◆ *Generic platform evolvability* versus *artifact evolvability*: Finding a proper level of abstraction for the architectural style such that the architectural style is much more stable than the artifacts that are instantiated within this architectural style.
- ◆ Legacy management: Integration of an existing application into a novel architecture and the related issue of integrating a novel architecture into an existing application.
- ◆ Technology obsolescence management: In a number of scenarios (e.g., aerospace industry, process control) the embedding system (the airplane or plant) has a much longer life-cycle than the embedded system.

Self Organization

- ◆ Emergence of complex behavior out of an assembly of simple components that are context aware and act autonomously within their limited context.
- ◆ Situational awareness, such as location, time, power and discovery of cooperating and adversary entities and the capability to generate autonomously plans for goal attainment, taking the situational awareness as input.
- ◆ Decentralized management: autonomic embedded systems that are capable to install and configure themselves, learn about their context, maintain themselves and reconfigure themselves in case of failing components without any user intervention.

Conclusion

- ◆ We are in a period of dramatic change--such a period offers many opportunities, but many of the present views and approaches have to be reconsidered.
- ◆ There are many common requirements and constraints across different application domains of embedded systems.
- ◆ The economic scenario of the semiconductor industry requires a common approach across application domains.
- ◆ ARTEMIS tries to establish such a common approach by creating a public-private partnership which supports a common strategic research agenda (SRA) in order to maintain the European lead in the embedded system area.