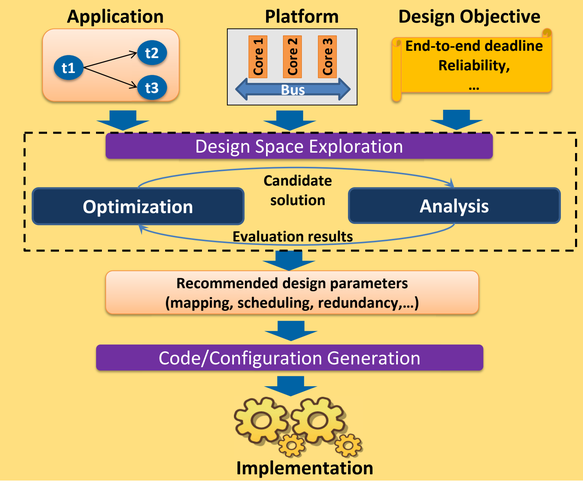
A Framework for Embedded System Design for MPSoCs

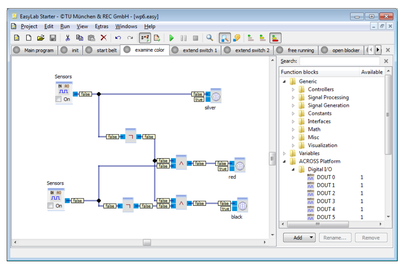
Simon Barner, Jia Huang, Andreas Raabe and Alois Knoll

Overview

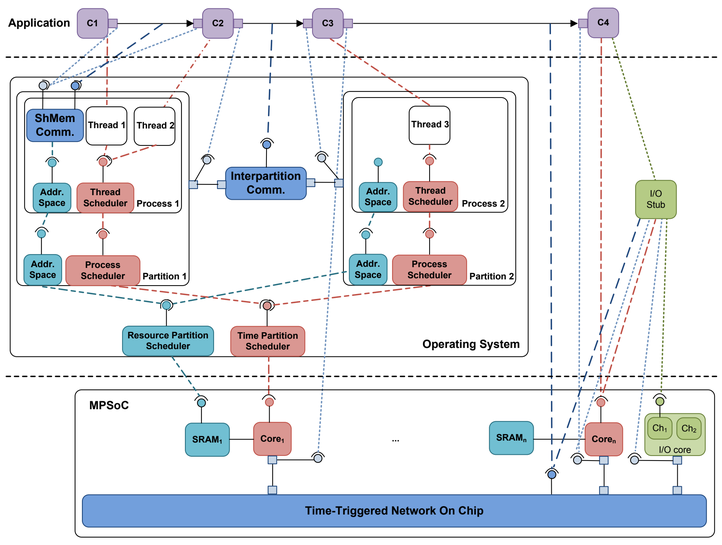
- **Framework** for implementing model-driven development tools for complex embedded platforms based on Eclipse EMF.
- Automatic Analysis & Optimization, Code Generation & Platform Configuration.
- Evaluation of approach:
  - Application of framework in **tool-chain for Multiprocessor System on a Chip (MPSoC) platform provided by ACROSS project**.
  - Validation using **industrial control demonstrator**.



Modeling Approach



[IEC 61131-3 FBD & system model of case study (ACROSS MPSoC, OS)]



- **Platform components** represent system **resources** (further classified by resource type & arbitration strategy). E.g., time-triggered network-on-chip, general purpose and dedicated system cores (I/O, …) , OS (partitions, processes, threads, inter-partition communication).
- **Application models** are annotated with **requests** for platform resources. The behavior of domain-independent **Kahn Process Networks** (with timing annotations) can be specified with **domain-specific models** (IEC 61131).
- **Allocation models** describe mapping & scheduling.

Reliability Analysis & Design Space Exploration

- **Automated design optimization** that evaluates trade-off between user-specified **design objective models** (end-to-end latency, reliability, resource consumption).

The ACROSS Project

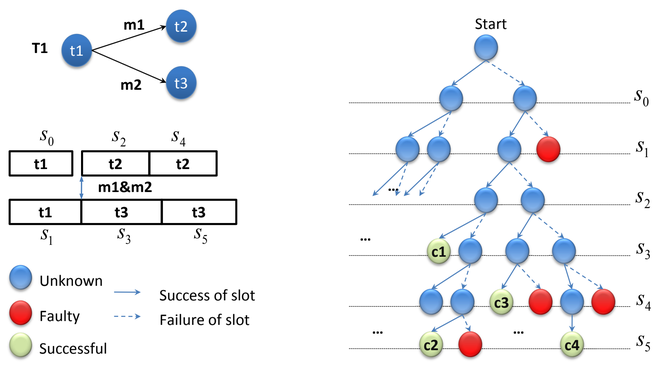
- **Problem Statement:** Exchange and reuse of technology between different application domains of embedded systems is difficult.
- **Approach:** Cross-domain HW/SW platform for the realization of mixed-criticality applications.
- **Results:** Time-triggered MPSoC, software environment (OS, middleware) and tool-supported development methodology. Assessment in automotive, aerospace and industrial control case studies.

- Tree-based **system-level reliability analysis** in the presence of **active redundancy** and **voting**.
- Visualization of tradeoffs & recommended deployments.

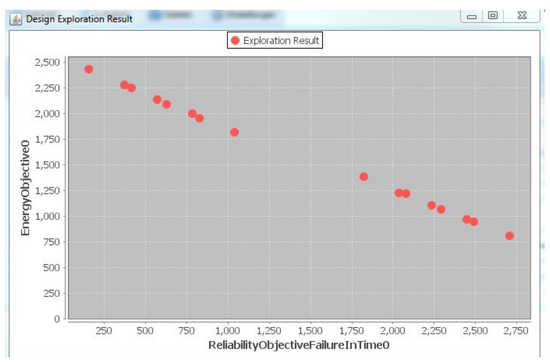
Platform Configuration & Code Generation

Facilitate deployment to target platform by automatic generation of:

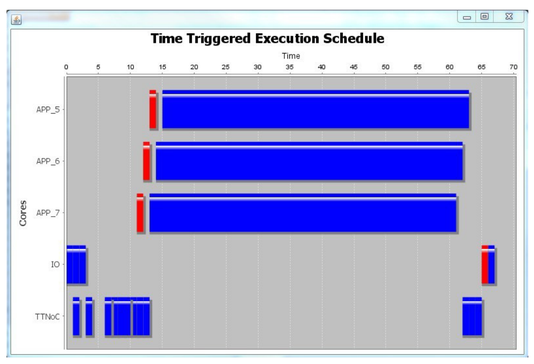
- **Functional C code** for application behavior.
- **Structural C code** for component organization, communication, etc.
- **Platform configuration**: hardware configuration (network & task schedule), OS configuration.



[Reliability Analysis Procedure]



[DSE result: Pareto Optimal Design Alternatives]



[Recommended schedule for case study]

Industrial Control Demonstrator

- **Objective:** Demonstrate tool supported implementation of safety critical automation task on ACROSS platform.
- **Function:** Sort work-pieces by material properties.
- **Reliability Goal:** tolerate single fault of a random component.
- **Application of Development Methodology:**
  - **Input**: IEC 61131-3 **models of sorting task** embedded into Kahn process network model as well as **models of ACROSS MPSoC** and **PikeOS operating system**, **reliability goal**.
  - **Output**: Fault-tolerant deployment (here: Triple Modular Redundancy scheme). Automatic generation of application code as well as configuration of ACROSS MPSoC and PikeOS instances.

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
ARTEMIS

The ACROSS project is funded by the BMBF (German Ministry of Education and Research) and the ARTEMIS Joint Undertaking under grant agreement No ARTEMIS-2009-1-100208.

fortiss GmbH, An-Institut Technische Universität München

Contact: Simon Barner • **Tel.:** + 49 89 360 3522 22 • **Mail:** barner@fortiss.org • **Web**: www.fortiss.de

www.across-project.eu