Formal performance analysis - from networked systems to MpSoC

R. Ernst

TU Braunschweig

Overview

- introduction – performance models and applications in embedded systems
- specific MpSoC challenges
- extending formal analysis beyond traditional process-to-process communication
- example
- conclusion

Introduction

- formal embedded system platform analysis and optimization
  - based on abstract formal system models that describe platform load and activities
    (task activation and communication frequency, execution times,...)
  - originally developed in the context of schedulability analysis
  - alternative or complement to simulation
- different mathematical formalisms
  - FSM networks (timed automata)
  - systems of equations for task response times

Performance models and applications

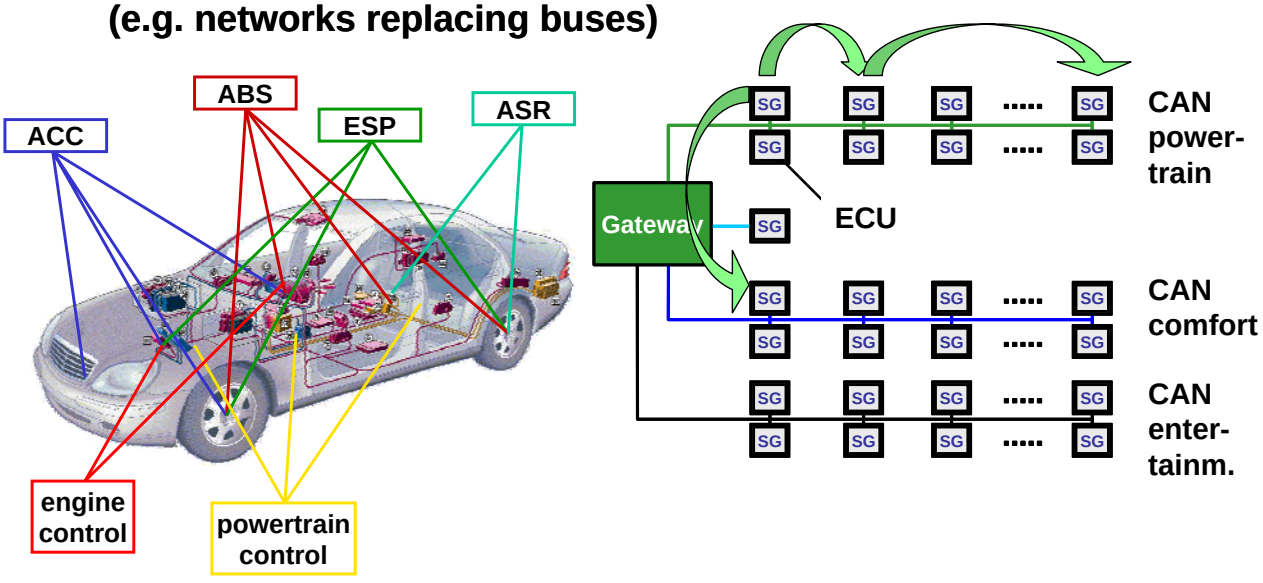
- formal performance model application
  - can be used in early design stages (no executables)
  - can determine design robustness and sensitivity to changes or inaccurate estimations
  - fast – applicable to optimization
  - supports IP component integration from different sources
- different formal models
  - simple load models (average case)
  - worst case models
  - statistical models

Worst case vs. statistical models

- worst case models
  - well established formal semantics (schedulability analysis)
  - composition of worst case component models well understood - *new generation of tools for compositional analysis*
  - can be improved considering execution scenarios (system modes - AADL)
  - currently used for predictable system composition (automotive)
- statistical models
  - potentially higher system utilization
  - derivation of reliable statistical execution models and their composition very difficult (data dependencies) and not fully understood
  - more research needed

Application example: Automotive

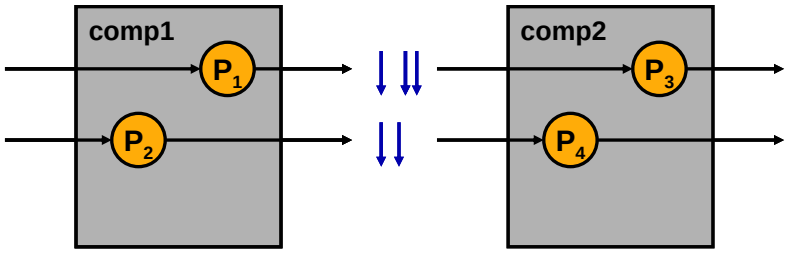
- non-functional dependencies of different subsystems – problem grows with system size (e.g. networks replacing buses)



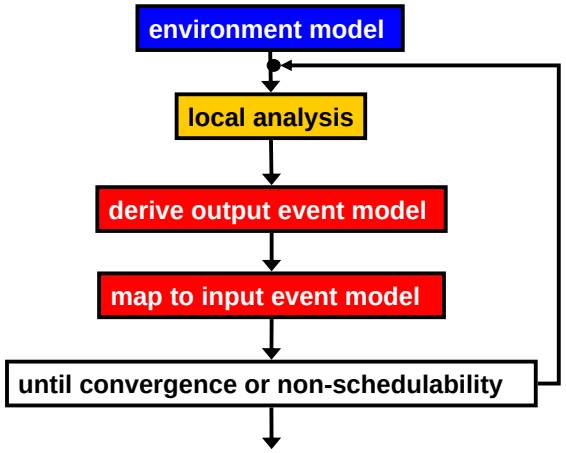
Worst case design for automotive systems

- established worst case design for systems integration
  - development of predictable architectures + software + analysis e.g. by Volcano (Volvo) or LiveDevices (ETAS)
  - communication parameters (e.g. priorities) part of supplier – OEM agreement
  - Time Triggered Architecture
- trend towards heterogeneous networked architectures with different scheduling (FlexRay, MOST) and flexible mapping requires new approaches
  - software standards
  - new analysis and optimization

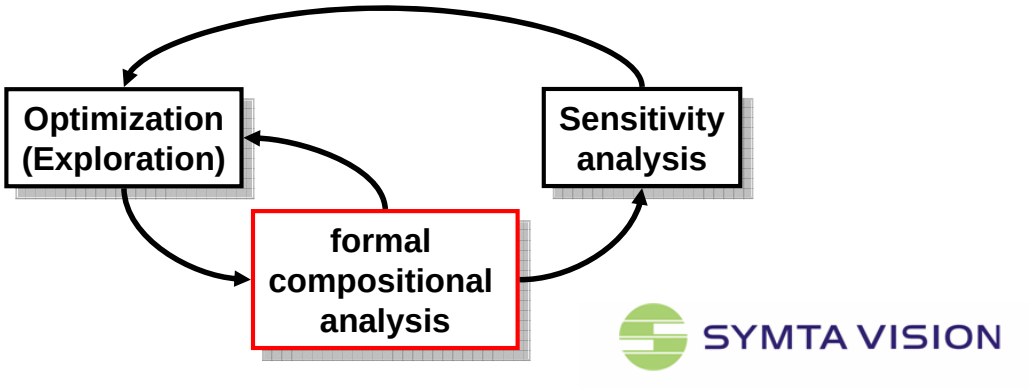
Compositional analysis



- subsystems coupled by streams
- coupling corresponds to event propagation
- Tools, e.g., from ETH Zürich and TU Braunschweig (SymTA/S)



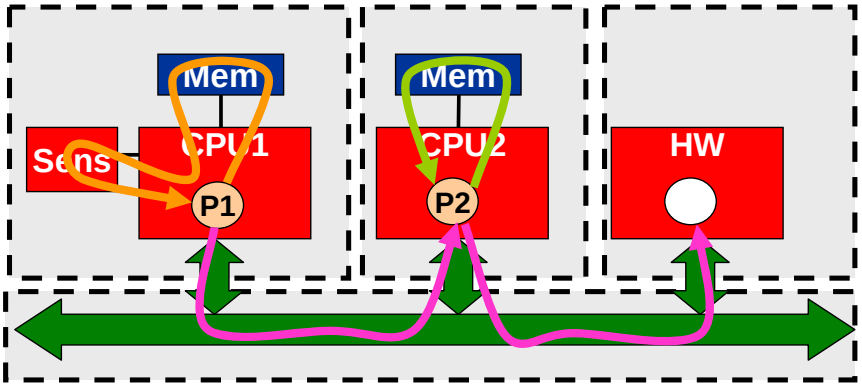
Tool SymTA/S



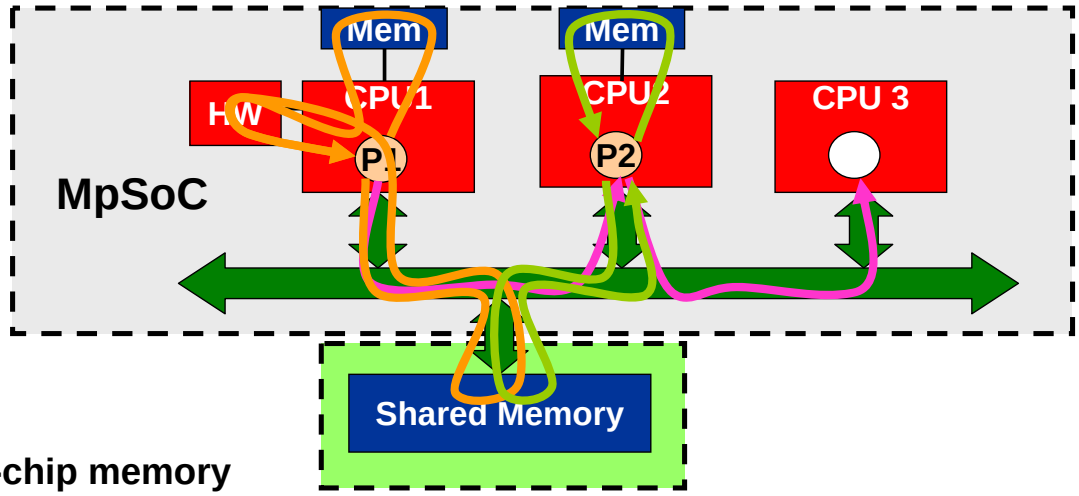
- commericalized by start-up *www.symtavision.com*
- used e.g. by several customers for automotive systems optimization (incl. robustness optimization, planning of upgrades, ...)

From distributed systems to MpSoCs 1/2

- distributed embedded systems
  - local computation and memory resources
  - network mainly used for process communication
  - simple communication model (read at process start, write in the end)
  - simple activation models (time or event triggered)

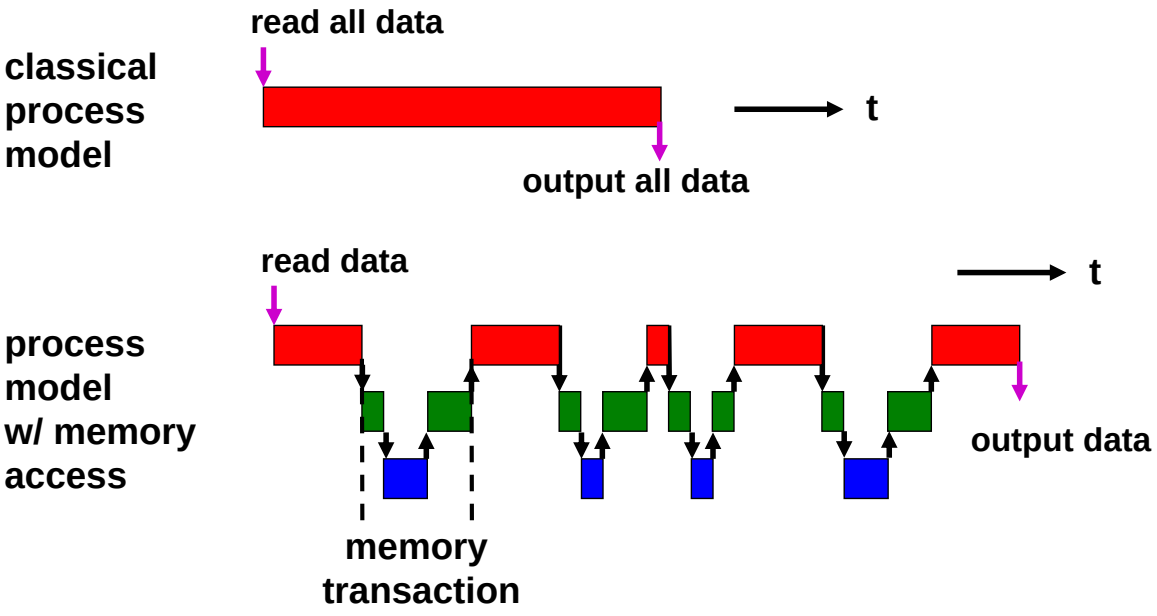


From distributed systems to MpSoCs 2/2



- on-chip memory
  important cost factor
- larger memories off-chip
- data and program memory accesses on same network as task communication - more complex traffic

MpSoC process execution



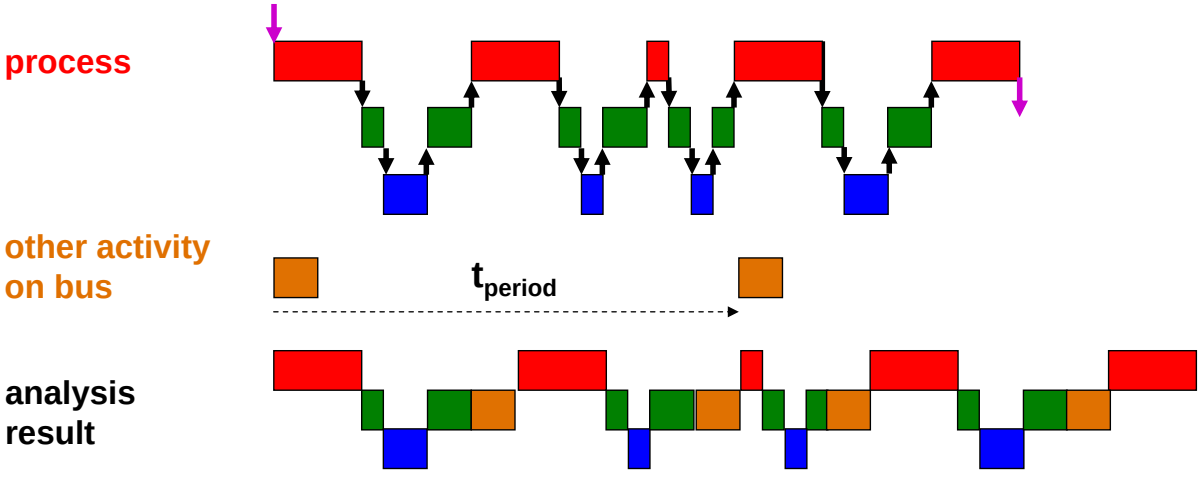
Memory transactions

+ individual transactions can already be modeled in SymTA/S (like in other tools, e.g. MAST)
+ no. of transaction can usually be bounded
– transaction distances are processor and path (i.e. data) dependent – difficult to identify

+ cache accesses can usually be bounded
– caches further “distort” access patterns

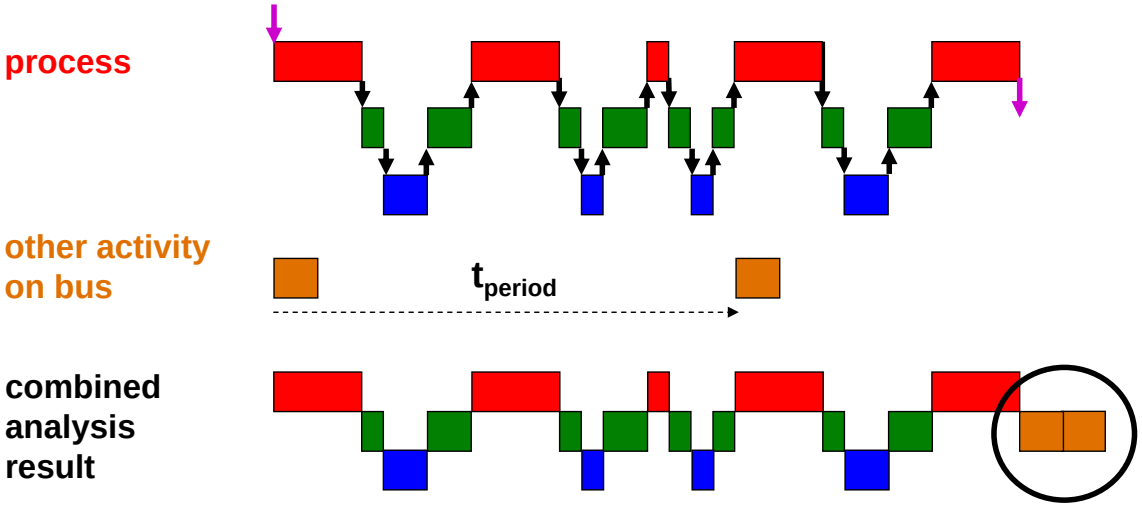
Memory transaction modeling alternatives 1/2

- single transaction at-a-time
  - straightforward approach
  - worst case timing per access
  - overestimation of real bus and memory timing



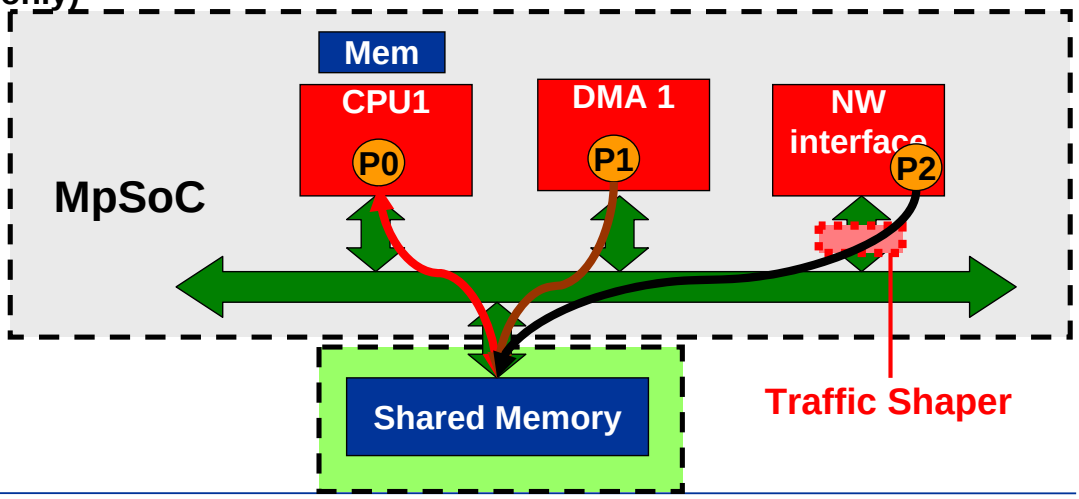
Memory transaction modeling alternatives 2/2

- combined analysis of all process memory transactions
  - add all delays that can occur during all transactions of a process in the worst case
  - more realistic bus and memory timing

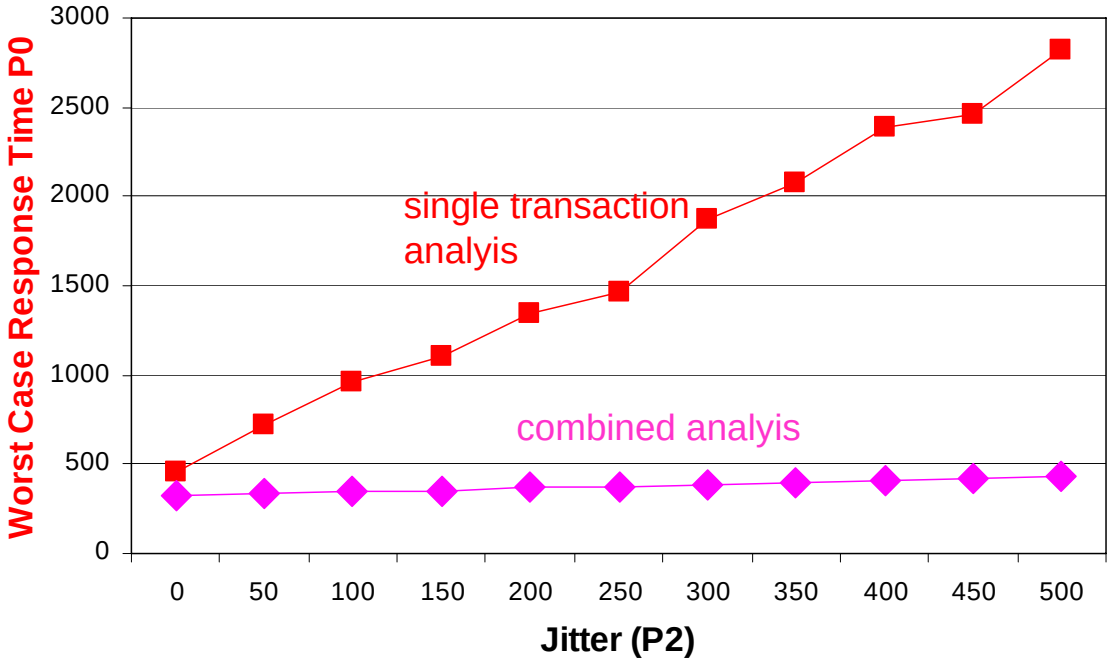


Example: periodic process system

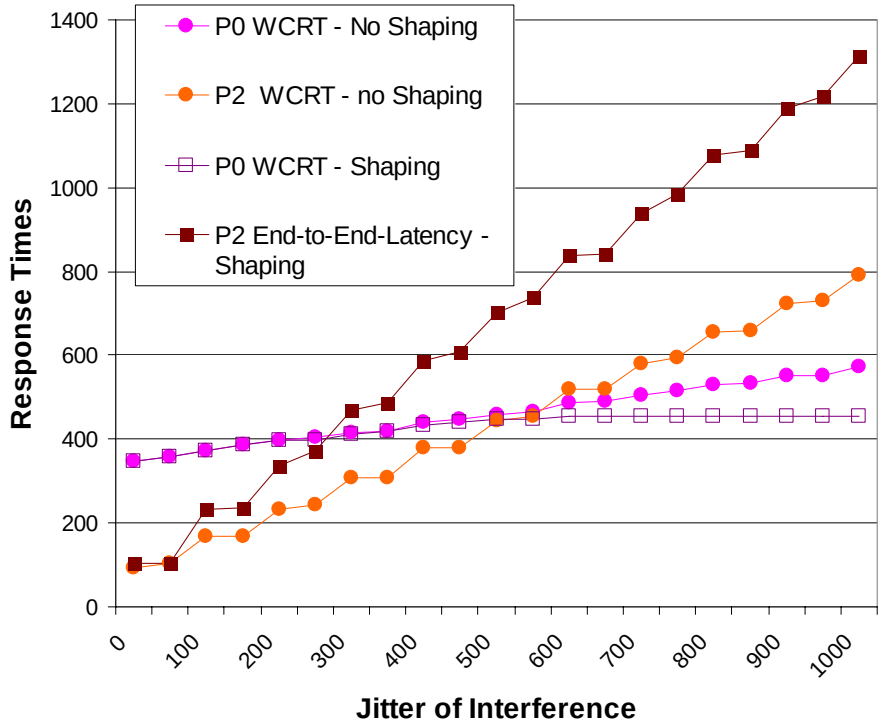
- process P0: t_{period} = 1000, t_{jitter} = 0,
  - 10 memory accesses: t_{bus} = 2, t_{mem}= 1
- process P1: t_{period} = 100, t_{jitter} = 0, t_{bus} = 2, t_{mem}= 1 (write only)
- process P2: t_{period} = 100, t_{jitter} = 0..1000 (Burst), t_{bus} = 5, t_{mem}= 3 (write only)



Single transaction vs. combined analysis



Traffic shaping effect



Conclusion

- formal performance analysis and optimization are gaining momentum in distributed embedded system design
- worst case design successfully used for predictable and robust systems integration – supported by tools
- more complex behavior of MpSoC due to conflicting task communication and memory access
- new technique presented to more efficiently include memory access in formal analysis

Acknowledgement - Literature

- Literature can be found
  - www.spi-project.org
  - www.symtavision.com
- The experiments and some of the slide figures have been provided by Simon Schliecker