Ten Years of Performance Evaluation for Concurrent Systems using CADP

Nicolas Coste[1] Hubert Garavel[2] Holger Hermanns[2,3]
Frédéric Lang[2] Radu Mateescu[2] Wendelin Serwe[2]

[1] STMicroelectronics, Grenoble, France
[2] VASY (INRIA and LIG, Grenoble, France)
[3] Saarland University, Sarrebrück, Germany

What is CADP?

A toolbox for analyzing asynchronous systems

- At the crossroads between four computer-science branches:
  - Formal methods
  - Concurrency theory
  - Computer-aided verification
  - Performance evaluation
- Development started in 1986:
  - Initially, there were only 2 tools (Caesar and Aldebaran)
- Today (CADP 2010):
  - 45 tools
  - 22 code libraries

CADP 2010

- A large functionality spectrum:
  - Several specification languages
  - Code generation, rapid prototyping
  - Explicit-state verification
  - Step-by-step, random, … simulation
  - Test generation
  - Performance evaluation (since 2000)

- Architectural principles:
  - Generic software components
  - Modular, extensible architecture

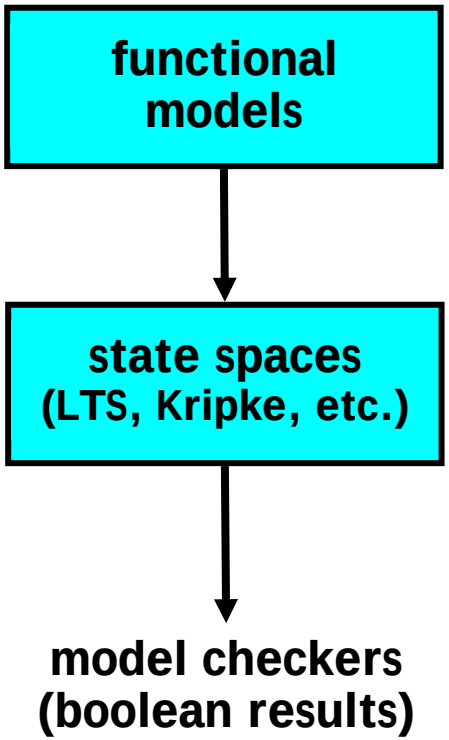
Performance evaluation in CADP 2010

- Answer to *quantitative questions* such as:
  - Is the system efficient? *(performance estimation)*
  - Which probability for a failure? *(dependability)*

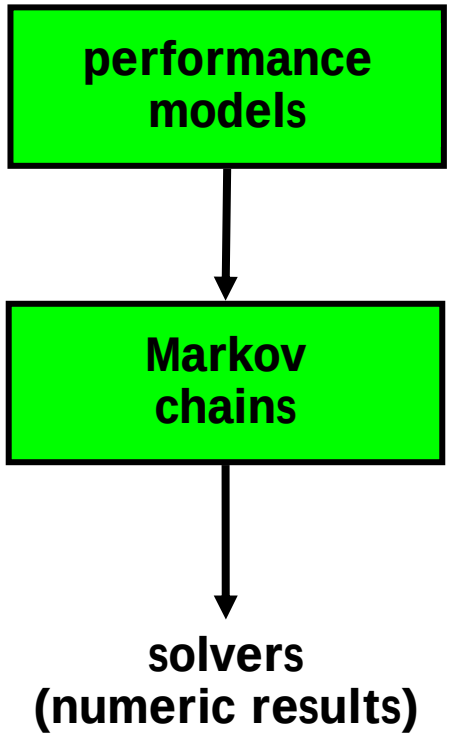
- Use *extended Markovian models* combining
  - **Functional models** specified in high-level languages (e.g. the LOTOS [ISO-88] and LOTOS NT languages for describing protocols and distributed systems)
  - **Performance data** based on (discrete and continuous time) Markov chains

The initial picture

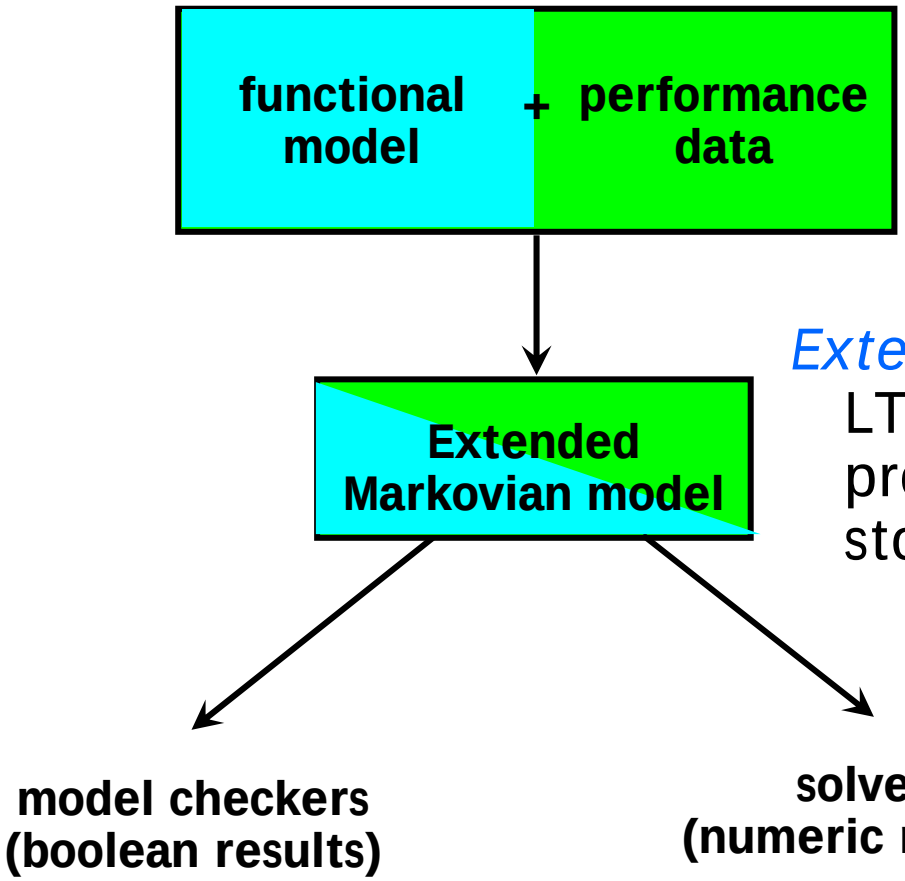
functional verification



performance evaluation



Extended Markovian models (1/2)



Extended Markovian model =
LTS (*Labeled Transition System*) +
probabilistic transitions ("**prob** 0.8") +
stochastic transitions ("**rate** 3.1")

Extended Markovian models (2/2)

Model	LTS transitions	Stochastic transitions	Probabilistic transitions
LTS (*Labeled Transition System*)	✓	✗	✗
CTMC (*Continuous Time Markov Chain*)	✗	✓	✗
DTMC (*Discrete Time Markov Chain*)	✗	✗	✓
IMC (*Interactive Markov Chain*) [Hermanns 99]	✓	✓	✗
IPC (*Interactive Probabilistic Chain*) [Coste 10]	✓	✗	✓
Extended Markovian models [CADP]	✓	✓	✓

Models subsumed by CADP's extended Markovian models (among others)

Two approaches for performance evaluation

- Approach #1:
  - Generation of a Markovian model
  - Analysis using a Markovian solver

State explosion sometimes occurs

- Approach #2:
  - Random simulation and on-the-fly analysis

CADP tools for Markovian model generation

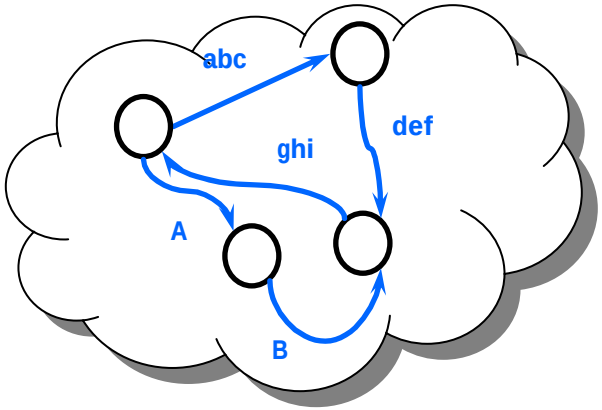
The BCG format and library

- A file format for storing huge state/transition models (up to 2^{44} explicit states)
- A set of tools for handling BCG files: format conversions, LTS info, visualization, hiding, renaming, …
- Five kinds of transitions
  - *ordinary transitions* *a*
  - *stochastic transitions* "**rate** *r*" ($r \in \mathrm{R}^+$)
  - *labeled stochastic* transitions "*a*; **rate** *r*" ($r \in \mathrm{R}^+$)
  - *probabilistic transitions* "**prob** *p*" ($p \in]0, 1]$)
  - *labeled probabilistic transitions* "*a*; **prob** *p*" ($p \in]0, 1]$)

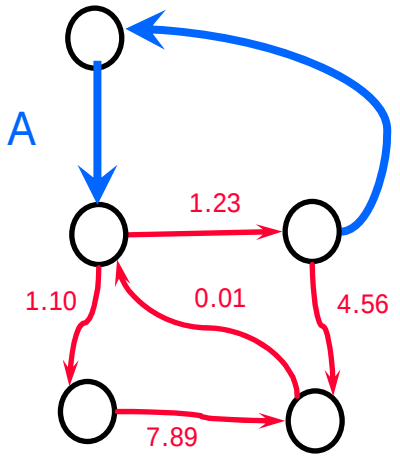
High-level specifications

- Functional model specified in LOTOS or LOTOS NT
- Two ways to model performance aspects
  - Model rates symbolically using ordinary labels, later on instantiated (i.e., renamed) with actual values
  - Or use constraint-oriented specification style, a safe and compositional technique to insert delays in a functional specification

Example: insert between successive actions A and B a delay represented by the red CTMC



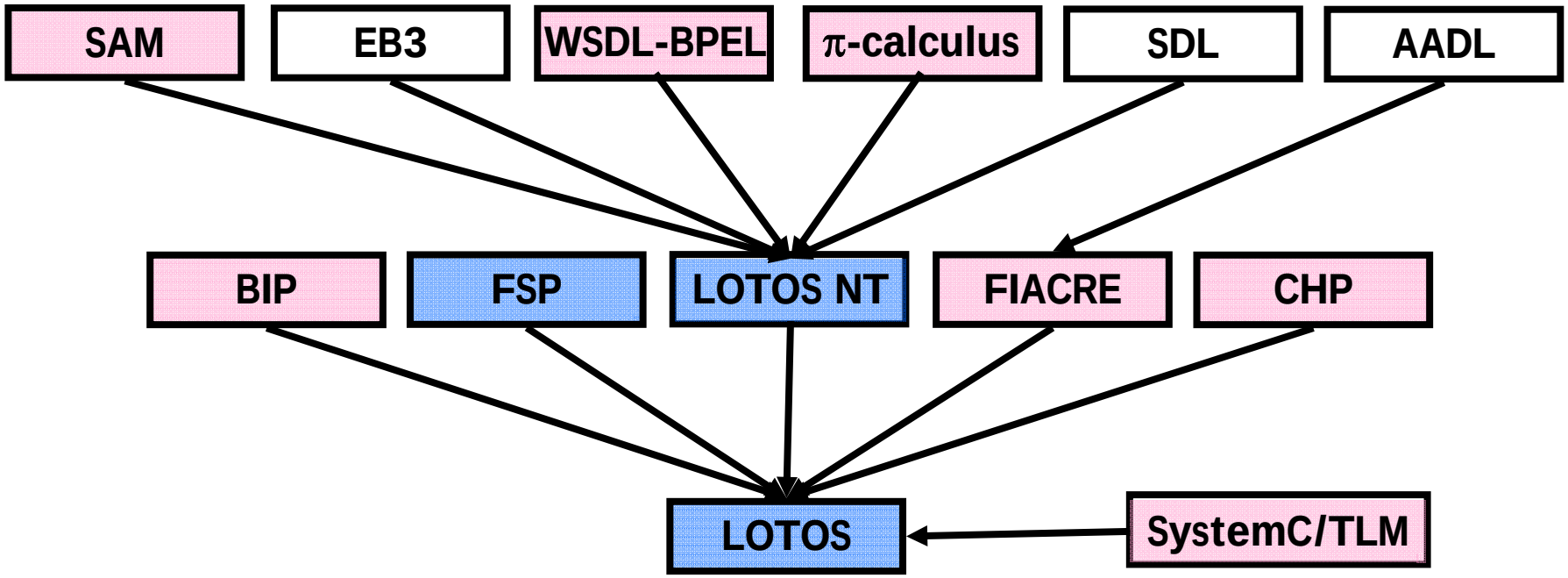
|[A, B]|



The CAESAR.ADT and CAESAR tools

- Compilers for LOTOS
- CAESAR.ADT: a compiler for LOTOS data types
  - generates C code (compiles pattern-matching)
  - optimizes both memory and time
- CAESAR: a compiler for LOTOS processes
  - translates LOTOS into Petri nets, then into LTS
  - generates C code for on-the-fly state exploration

Connections with other languages



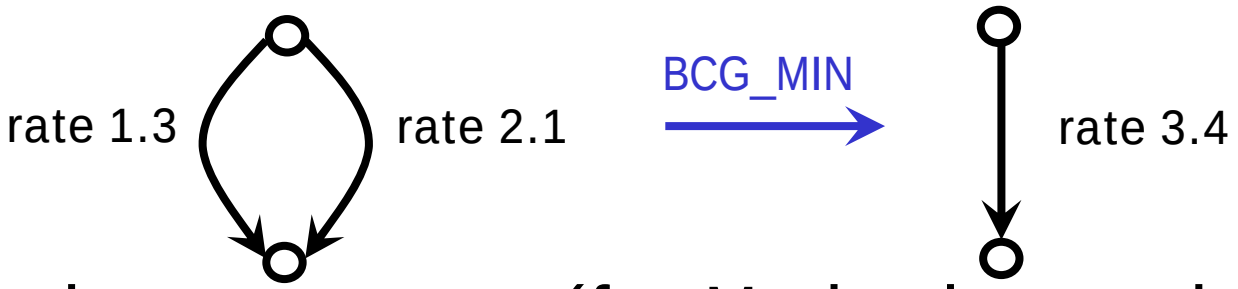
Blue = available in CADP 2010
Pink = separate prototypes

The EXP.OPEN 2.0 tool

- EXP: a small language for describing networks of communicating automata
  - parallel composition operators (LOTOS, CCS, CSP, mCRL, etc.) or synchronization vectors
  - label hiding, renaming, cutting using regexps
  - "priority" operator
- EXP.OPEN: a state space generator for EXP
  - on the fly partial order reductions (branching eq., weak trace eq., stochastic/probabilistic eq.)
- Generates parallel composition of extended Markovian models
- No synchronization on "**rate**"/"**prob**" transitions

The BCG_MIN 2.0 tool

- An efficient minimization tool
  - Inputs: BCG graph
    Chosen equivalence for minimization
  - Output: Minimized BCG graph
- Strong and branching bisimulations + lumpability



- Recent improvements (for Markovian models)
  - 500 times faster and 4 times less memory than previous version 1.0
  - Input graph up to 10^7 states and 10^8 transitions

The DETERMINATOR tool

- On-the-fly generation of a Markov Chain
  - Applies local transformations to remove stochastic nondeterminism
  - Implements a determinacy check ("well specified" stochastic process)
  - Algorithm is a variant of [Deavours-Sanders-99]
- Input: On-the-fly extended Markovian model
- Output: BCG graph (extended CTMC) or error message

CADP tools for numerical analysis of extended Markovian models

The BCG_TRANSIENT tool

- Numerical solver for Markov chains
- Transient analysis
- Inputs:
  - Extended Markovian model in the BCG format
  - List of time instants
- Outputs:
  - Numerical data usable by Excel, Gnuplot…
- Method:
  - BCG graph converted into a sparse matrix
  - Uniformisation method to compute Poisson probabilities
  - *Fox-Glynn* algorithm [Stewart94]

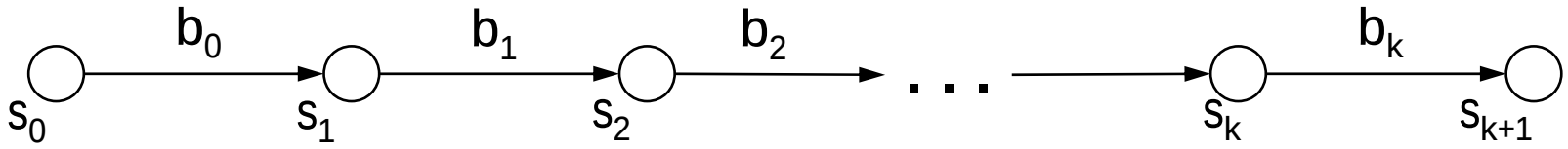
The BCG_STEADY tool

- Numerical solver for Markov chains
- Steady-state analysis (equilibrium)
- Inputs:
  - Extended Markovian model in the BCG format
  - No deadlock allowed
- Outputs:
  - Numerical data usable by Excel, Gnuplot…
- Method:
  - BCG graph converted into a sparse matrix
  - Computation of a probabilistic vector solution
  - Iterative algorithm using *Gauss-Seidel* [Stewart94]

CADP tool for on-the-fly Markovian model simulation

The CUNCTATOR tool

- A steady-state random simulator for IMCs
- On-the-fly label hiding and renaming to produce a (labeled) CTMC with internal actions
- On-the-fly exploration of a sequence:



- Compute the throughput of each stochastic action "*a*; rate *r*"
- Different scheduling strategies for internal acions
- Save/restore context of simulation
- Caching of internal sequences of transitions

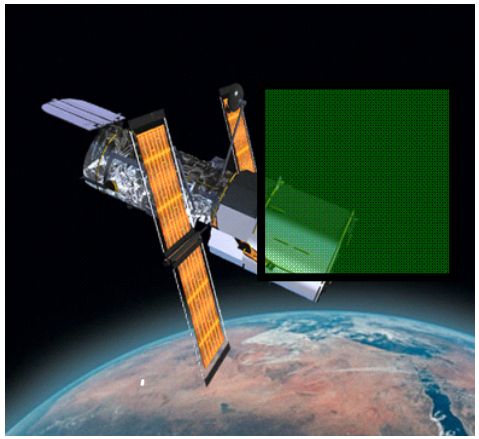
Additional tools for Interactive Probabilistic Chains

IPC (*Interactive Probabilistic Chains*)

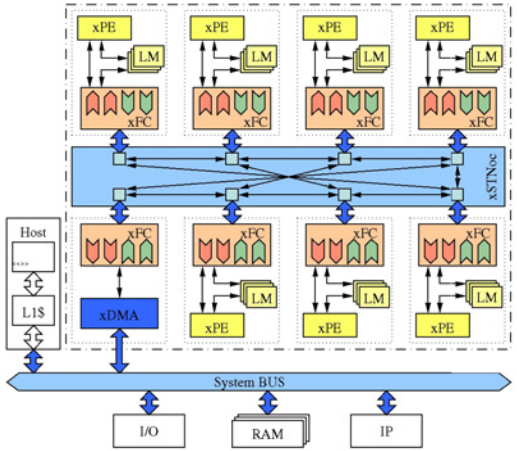
- Extended Markovian model [Coste, PhD 10]:
  - Ordinary transitions
  - Probabilistic transitions that take one time step
- Three prototype tools:
  - IPC_COMPOSE: parallel composition of IPC represented as BCG files

    Synchronized probabilistic transitions "**prob** p" and "**prob** q" yield "**prob** $p \times q$"
  - IPC_INSERT: insertion of delays in an IPC
  - IPC_DISTRIBUTION: computes the steady-state probability distribution of the latency between two actions *a* and *b*

Real-life applications



- Hubble telescope lifetime
  [Hermanns, EATCS 01]
- SCSI-2 bus arbitration protocol
  [Garavel & Hermanns, FM 02]
- MPI send/receive & barrier primitives
  [Chehaibar & Zidouni & Mateescu, Quest 09]
- XStream data-flow architecture
  [Coste et al., CAV 09]
- Mutual exclusion protocols
  [Mateescu & Serwe, FMICS 10]



Conclusions

- Since 2000, a significant development effort for performance evaluation in CADP
  - New tool development
  - Extensions of existing tools
  - Migration: 10 architectures (including 64-bit platforms) and 3 compilers supported (Gcc, Sun CC, Intel CC)
- Smooth integration with functional verification
  - The Xeuca GUI
  - The SVL scripting language
- Academic and industrial applications

Future work

- Enhance integration with LOTOS NT
- Generalize techniques and tools for models containing:
  - nondeterminism
  - both stochastic and probabilistic transitions
  - non-Markov distributions, e.g., constant delays (**wait** *r*)
- Replace QNAP2 in Bull's performance-evaluation process

For more information…

- CADP Web site:
  http://vasy.inria.fr/cadp

- CADP forum:
  http://cadp.forumotion.com

INRIA LIG