CosyVerif: An Open Source Extensible Verification Environment

Étienne André, Lom Hillah, Francis Hulin-Hubard, Fabrice Kordon, Yousra Lembachar, Alban Linard, Laure Petrucci

ENS Cachan, Univ. Paris 6, Univ. Paris 13
17th July 2013
Many tools for distributed systems verification

- Relying on different formalisms
- Solving different problems
- Running on different OS
- Requiring some difficult installation procedures

⇝ Needs for:

- A unified representation
- Interoperability and tool integration
- For users: Easy installation and use
- For developers: Easy integration of tools
Outline

1 The CosyVerif Environment
2 Integrated Tools
3 Summary and Evolutions
1. The CosyVerif Environment
2. Integrated Tools
3. Summary and Evolutions
CosyVerif relies on reusable and extensible formalisms

[André et al., 2013]

FML (Formalism Markup Language)
  ▶ Describes formalisms (meta-models)

GrML (Graph Markup Language)
  ▶ Describes models

Advantages
  ▶ Unified model representation
  ▶ Easy addition of new formalisms

Applications: set of formalisms
  ▶ Large family of (timed) automata and Petri nets
Easy to install: simple and light multi-platform client connecting to servers
- Tool invocation through Web services transparent to the end-user
- Cloud-based architecture

- Authentication Server
- Formalisms & Models Repository
- BenchKit, a benchmark tool
- SOAP Web Service Bus
- Cluster 1
  - Server
  - Super server
- Cluster 2
  - Super server
- Cluster N
  - Super server
- Web Client
- Tool and command-line library
- Graphical User Interface
Outline

1. The CosyVerif Environment
2. Integrated Tools
3. Summary and Evolutions
Numerous tools integrated to CosyVerif

- One official client: **Coloane** (platform-independent)
- 6 tools and 9 integrated services
  - Integration via Web services: easy to use and compose
- Multiple formalisms supported (Petri nets and extensions, hybrid automata, timed automata... and more!)
**Statistical model checker** [Ballarini et al., 2011]

- **Input:** Generalised Stochastic Petri Nets with general distribution (GSPN) and a Hybrid Automaton Stochastic Logic (HASL) formula
- **Output:** Statistical estimation of the formula with a confidence interval
State space generation & CTL verification [Colange et al., 2011]

- Symbolic/symbolic approach based on Symmetric Nets with Bags [Haddad et al., 2009]
- Two symbolic techniques to counter state space explosion
  1. symmetries to reduce the reachability graph [Chiola et al., 1991]
  2. hierarchical Set Decision Diagrams to store the reachability graph [Couvreur and Thierry-Mieg, 2005]
Unfolding-based verification of Petri nets with read arcs (contextual nets) [Baldan et al., 2012]

- **Features**
  - Unfolding construction tool [Rodríguez et al., 2011]
  - Reachability and deadlock checking tool [Rodríguez and Schwoon, 2012]

- **Characteristics**
  - Unfoldings fully represent the state space of a c-net by a partial order rather than by a set of interleavings
    - Often exponentially smaller than the state space, and never larger
  - c-net unfoldings can be exponentially more compact than those of corresponding Petri nets [Baldan et al., 2012]
Parameter synthesis for real-time systems [André et al., 2012]

- Quantitative robustness analysis
  - “Can we increase some of the timing delays such that the system still behaves well?”
- Schedulability analysis
- Hybrid system verification
ModGraph

Construction and analysis of modular state spaces

[Lakos and Petrucci, 2004]

- Modular State Spaces for Synchronised Automata
  - synchronisation structure
  - only reachable parts of the automata

- Analysis
  - forward and backward reachability
  - deadlock-checking
  - liveness
State space generation and CTL formulæ evaluation on P/T nets [Hong et al., 2012]

- Handles Symmetric Nets through their unfolding into an equivalent P/T net
- Exploits hierarchy: a state is seen as a tree, where the leaves correspond to place markings
- Relies on Set Decision Diagrams [Couvreur and Thierry-Mieg, 2005]
An Open Environment

- Entirely open source

- Open to contributions
  - Tool integration
  - Alternative clients
  - New formalisms

- A repository of models using a common syntax
  - Coming from the integrated tools, and the model checking contests
    [Kordon et al., 2013]
Recent and Ongoing Evolutions

- **Asynchronous tool invocation**
  - Get the result later (e.g., by email)

- **Federation of servers and use of clusters**
  - Enable load balancing

- **Repository of formalisms and models**

- **Command-line version of the underlying platform**
Future Evolutions

- **Enhanced interaction** between tools
  - Output of a tool as input of another one

- Handling **semantics** (bridges between formalisms)
  - Also allows system simulation

- Handling **heterogeneous models** (mixing different formalisms)

Try it!
http://cosyverif.org/
Future Evolutions

- **Enhanced interaction** between tools
  - Output of a tool as input of another one
- Handling **semantics** (bridges between formalisms)
  - Also allows system simulation
- Handling **heterogeneous models** (mixing different formalisms)

Try it!

http://cosyverif.org/
Bibliography
A modular approach for reusing formalisms in verification tools of concurrent systems.
In ICFEM, Lecture Notes in Computer Science. Springer.
To appear.

IMITATOR 2.5: A tool for analyzing robustness in scheduling problems.
In Formal Methods, volume 7436 of Lecture Notes in Computer Science, pages 33–36. Springer.

Efficient unfolding of contextual Petri nets.
Theoretical Computer Science, 449:2–22.

HASL: An expressive language for statistical verification of stochastic models.
In VALUETOOLS, pages 306–315.
On well-formed coloured nets and their symbolic reachability graph.
In ICATPN. Springer-Verlag.

Crocodile: A symbolic/symbolic tool for the analysis of symmetric nets with bags.
In ICATPN, volume 6709 of Lecture Notes in Computer Science, pages 338–347. Springer.

Hierarchical decision diagrams to exploit model structure.
In FORTE, volume 3731 of Lecture Notes in Computer Science, pages 443–457. Springer.

Efficient state-based analysis by introducing bags in Petri net color domains.
In ACC, pages 5018–5025. Omnipress IEEE.

Computing a hierarchical static order for decision diagram-based representation from P/T nets.
Transactions on Petri Nets and Other Models of Concurrency, V:121–140.


Licensing
Title: Scottish Kitten
Author: RN3DLL
Source: https://commons.wikimedia.org/wiki/File:Scottish_Kitten.png
License: CC BY-SA 3.0
This presentation can be published, reused and modified under the terms of the Creative Commons license Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

Author: The CosyVerif team

https://creativecommons.org/licenses/by-sa/3.0/