CADP'97
Status, Applications, and Perspectives

**H. Garavel, M. Jorgensen, Ch. Pecheur,
R. Mateescu, M. Sighireanu, B. Vivien**

INRIA Rhône-Alpes and Dyade / VASY
655, avenue de l'Europe
F-38330 Montbonnot Saint-Martin

CADP (Caesar/Aldebaran) toolbox

- **compilers:**
  - LOTOS (Caesar and Caesar.adt)
  - networks of finite-state machines
- **verification tools:**
  - bisimulations (Aldebaran)
  - temporal logics (Evaluator and XTL)
- **many other tools:**
  - simulation
  - partial verification
- **open** and **extensible** (set of APIs)

Recent papers about CADP

In 1996: two overview papers

- COST 247 Maribor workshop (June 97)
- CAV’97 Conference (July 97)

Since then:

- Two new releases: Dec. 96 and June 97
- Many improvements and new features
- New applications and case-studies

The *visible* changes

- The **Eucalyptus 2.2** Graphical User Interface
- The new **Xsimulator** tool (rewritten in Tcl/Tk)
- The new **Monitor** tool
- A LOTOS-mode for Emacs and Xemacs
- A Web site (distribution, release notes, FAQ)

http://www.inrialpes.fr/vasy/cadp.html

The *invisible* changes

- **CAESAR** is faster (2-160 times)
- **OPEN/CAESAR** is also faster

The **Exec/Caesar** functionality

(1) "standard" Caesar:

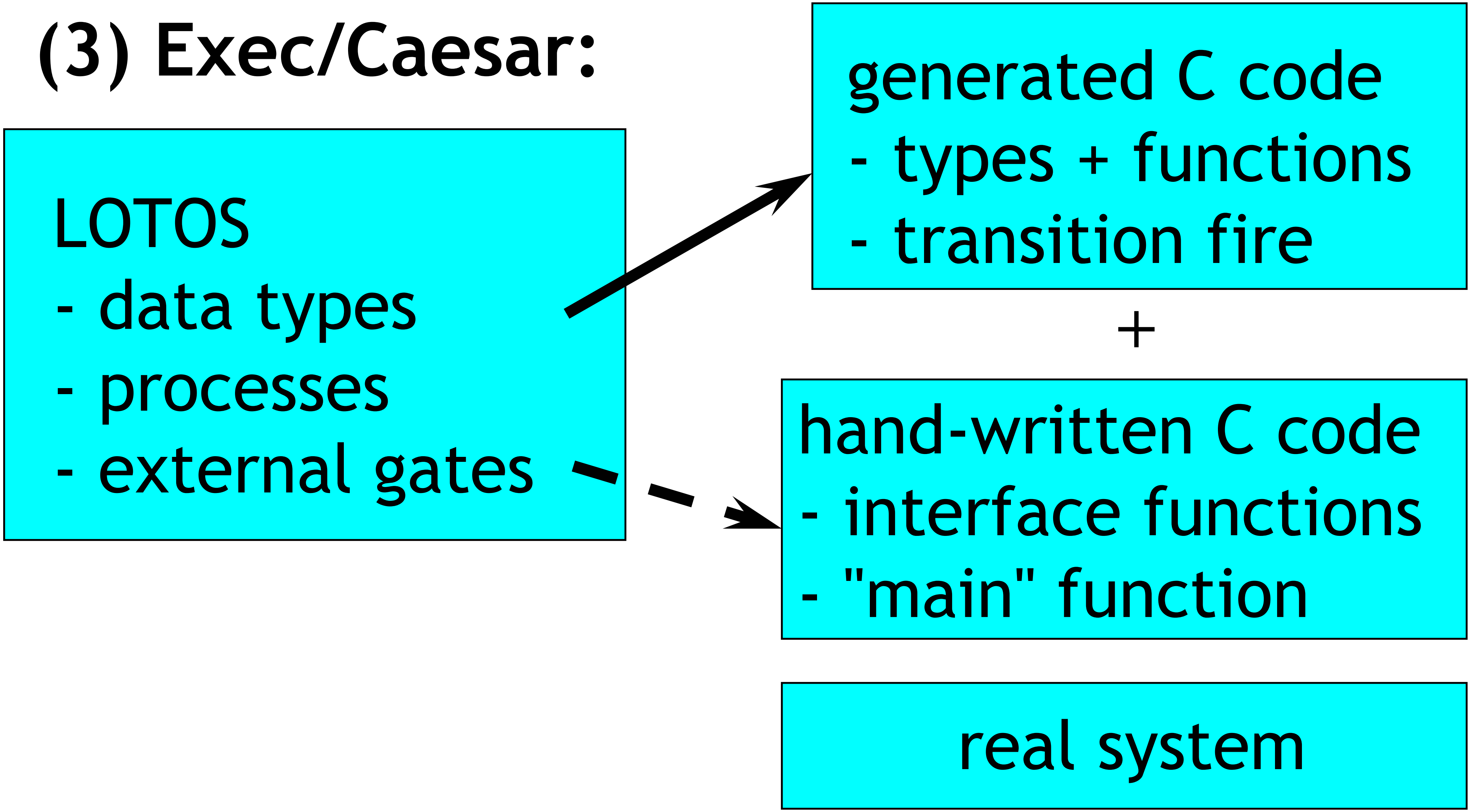
- model generation
- LOTOS => LTS (exhaustive simulation)
- distinctive feature: data types, even of unbounded size (lists, sets...)

(2) Open/Caesar:

- generic API for model exploration
- support for **on-the-fly** verification, random execution, interactive simulation, testing...

The **Exec/Caesar** functionality

(3) Exec/Caesar:



LOTOS
- data types
- processes
- external gates

generated C code
- types + functions
- transition fire

+

hand-written C code
- interface functions
- "main" function

real system

The Evaluator tool (V2)

Evaluator: evaluation of mu-calculus formulas

Improvements in Evaluator:

- richer formula language (label sets, *not*, *or*)
- more efficient data structures
- two different evaluation algorithms:
  - global
  - local (on the fly)

Marius Bozga (Verimag)

The Exhibitor tool (V2)

Exhibitor: search of execution sequence defined by a pattern of visible actions

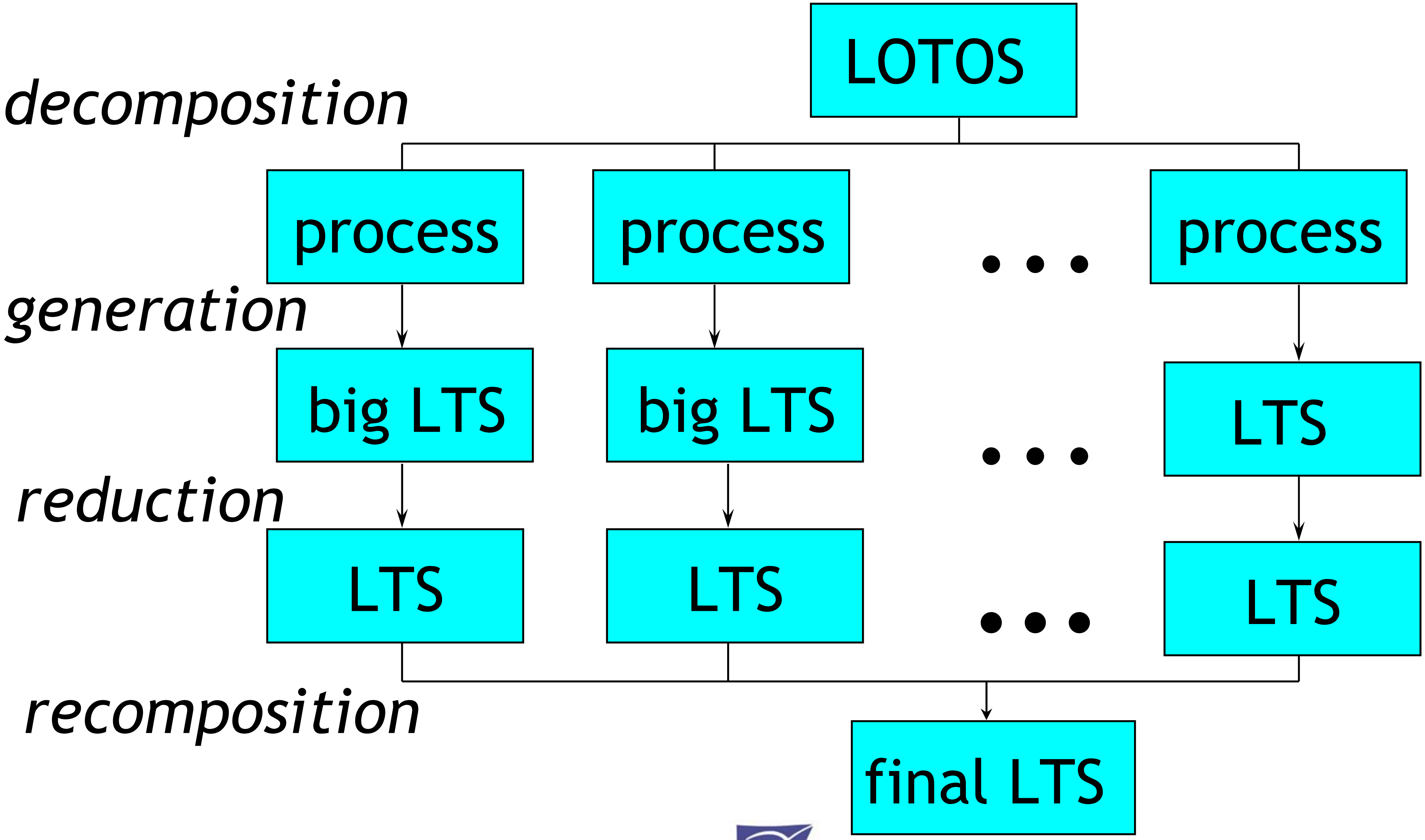
Improvements in Evaluator:

- more expressive pattern language:
  - regular expressions
  - boolean connectives
  - deadlock characterization
- two different search algorithms: DFS and BFS

X. Etchevers and H. Garavel (INRIA/VASY)

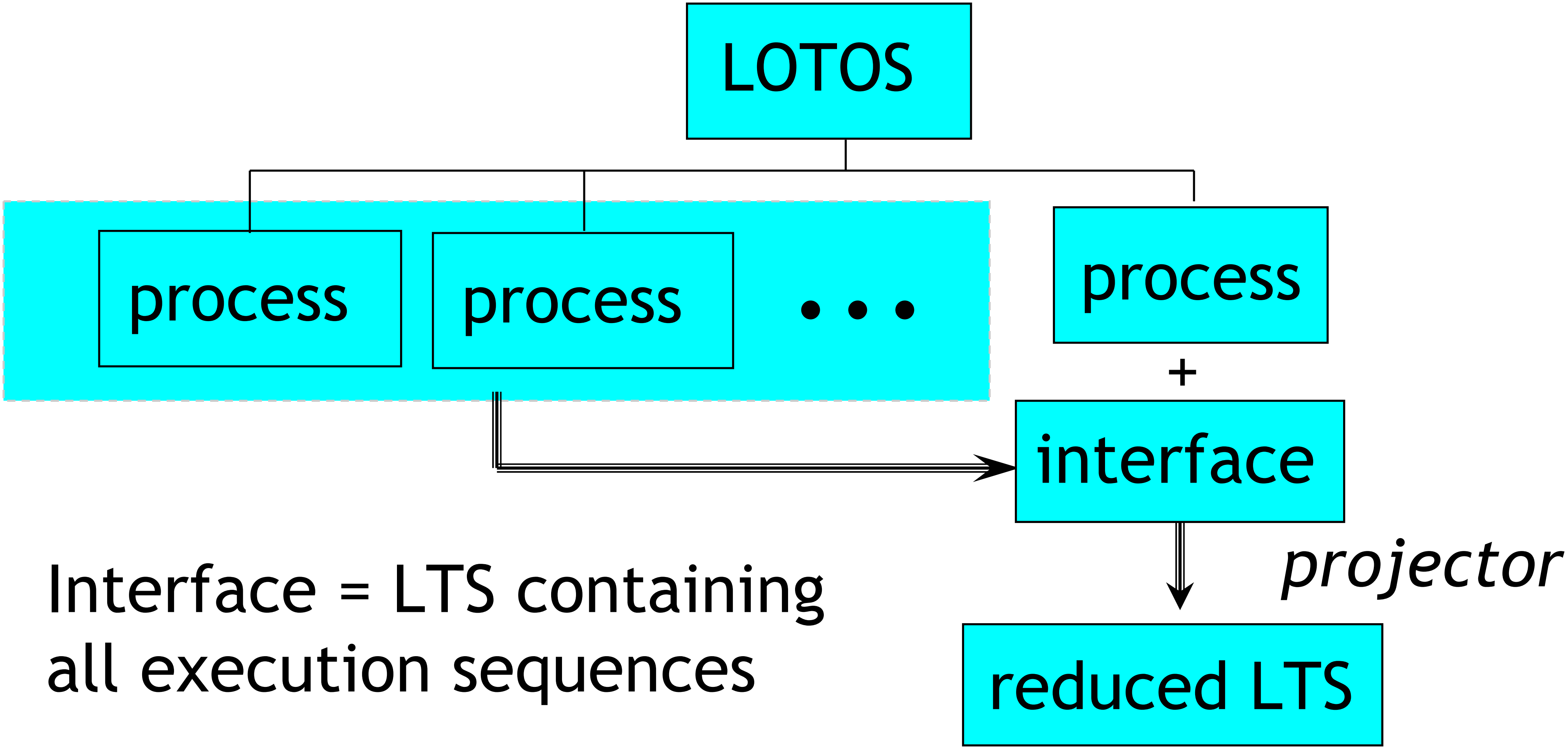
The **Projector** tool

"simple" approach to compositional verification



The **Projector** tool

"refined" approach [Graf-Steffen]



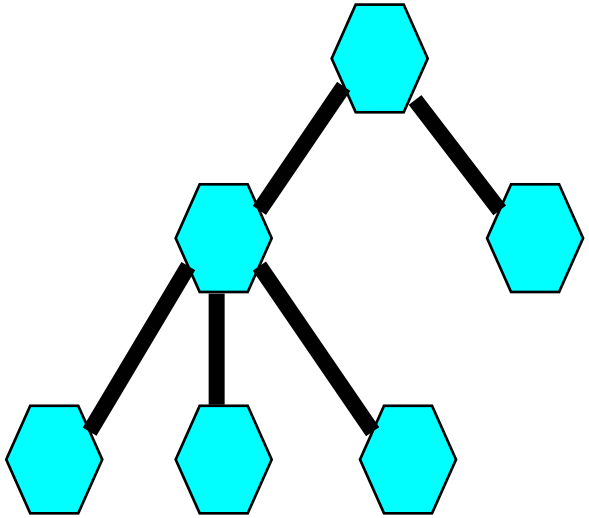
Interface = LTS containing all execution sequences

J.P. Krimm and L. Mounier (Verimag)

Application 1: CO4

C04: a Distributed Knowledge Data Base

- hierarchy of bases (with dynamic creation)
- consensus protocol (peer-reviewing policy)



Formal Specification

- 1,200 lines of LOTOS
- APERO notations for data types
- many errors detected

Verification

- Finite scenarios
- use of Exhibitor (on-the-fly)
- 4 unexpected message receptions
- 2 violations of invariants

Charles Pecheur (INRIA/VASY)

Application 2: **IEEE 1394**

IEEE high performance serial bus (FireWire)

Formal Specification

- Base: description written in mu-CRL [Luttik]
- 800 lines of Extended-LOTOS (hand-writing)
- 1,000 lines of LOTOS (TRAIAN translator)

Verification

- Finite state scenarios (CAESAR)
- ACTL formulas (XTL model-checker)
- 1 unexpected message reception detected

M. Sighireanu and R. Mateescu (INRIA/VASY)

Application 3: **Equicrypt**

- Equicrypt: a Trusted Third-Party Protocol defined in the ACTS 051 project (OKAPI)
- Authentication between customers and providers

Formal specification

- subscription and registration protocols (1,000 lines)

Verification

- use of a ***generic intruder*** process
- model-checking (Caesar, Aldebaran and Exhibitor)
- several unexpected attacks discovered
- model-checking diagnostic gives the attack

Guy Leduc et al. (University of Liege, RUN)

Application 4: **DCL**

- DCL: Departure Clearance Protocol
- air-trafic control protocol (Eurocontrol)

Formal specification

- 300 lines of LOTOS

Verification

- compositional verification (3 sub-processes)
- Caesar, Aldebaran and Exhibitor
- bad execution sequences discovered
- => the use of DCL will be limited

Ch. Hernalsteen and Th. Massart (Univ. Brussels)

Application 5: Production Cell

- a real automated metal plant
- challenge by K. Lewerentz and Th. Lindner (FZI Karlsruhe)

Formal specification

- 1,000 lines of LOTOS
- one process per device or motion

Execution

- use of Exec/Caesar functionalities
- a small driver to interface the Tcl/Tk simulator

H. Garavel and M. Jorgensen (INRIA/VASY)

Current and future work

Making formal methods applicable in the industry

- improve CAESAR to generate smaller LTSs
- develop the TRAIAN compiler for E-LOTOS
- develop the XTL (V2) model-checker

- connect CADP and **Fc2Tools** (INRIA Sophia)
- connect CADP and **TGV** (INRIA Rennes)