A Framework for the Formal Verification of Infinite Systems

Ludovic Apvrille, Sophie Coudert

Institut TELECOM / TELECOM ParisTech

Chan Leduc

INRIALPES

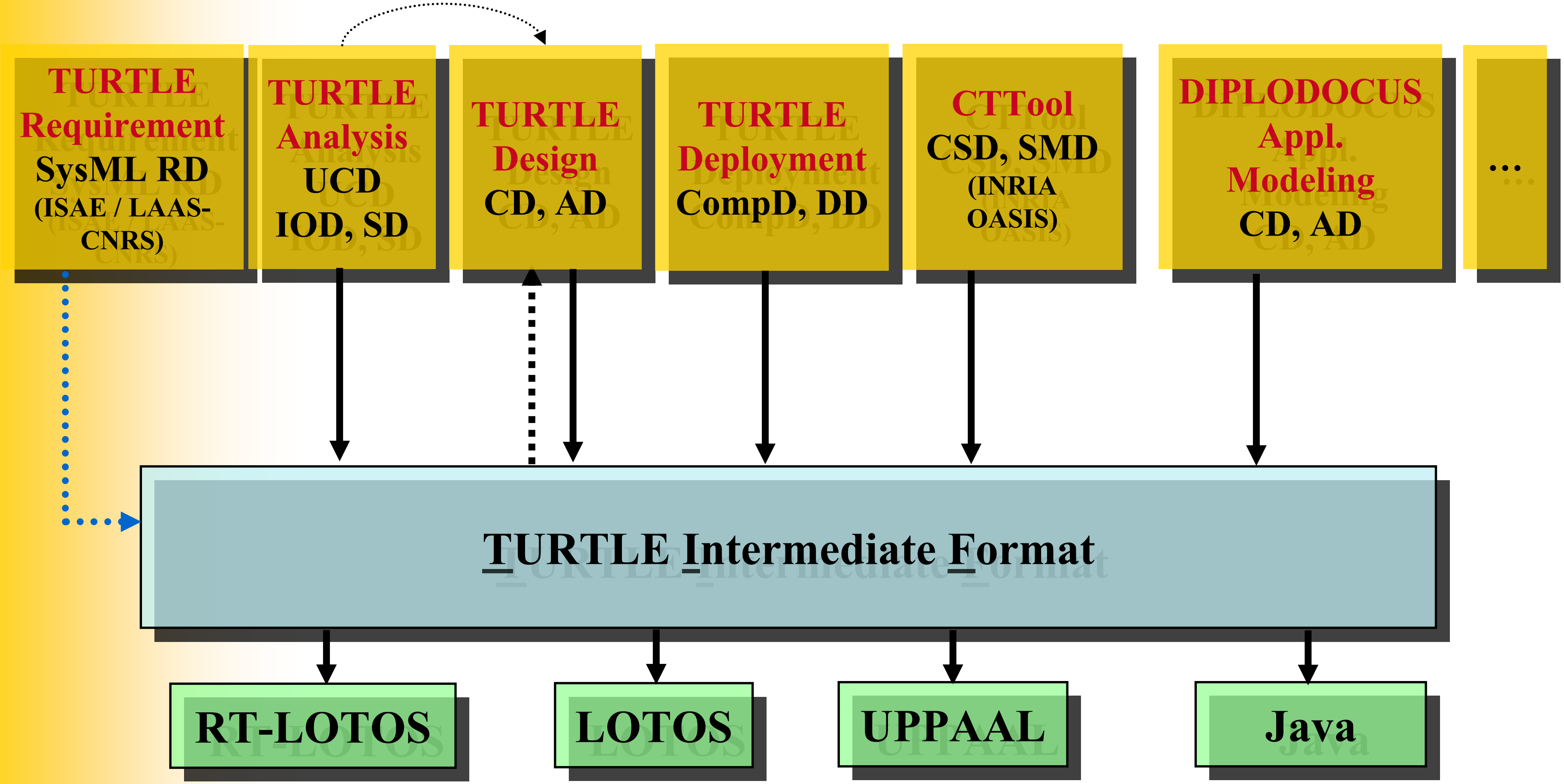
06/03/2008

Outline

- Context and problematic
- LOTOS, DL
- LOTOS to DL translation
- Example
- Conclusion

Context and Problematic

The TURTLE Environment



Implemented in TTool, an open-source UML toolkit
http://labsoc.comelec.enst.fr/turtle/ or simply type “UML TURTLE” under google

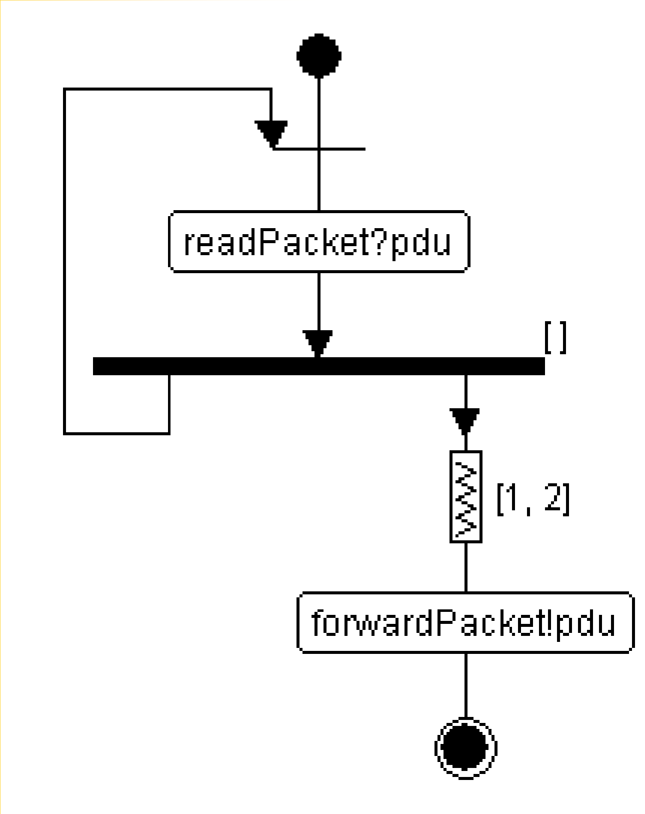
LOTOS

- Formal Description Technique
- Based on process algebra
  - ⇨ Data part
  - ⇨ Process part
    - Variables, gates
    - |||, |[g0, .., gn]|, [], >>, [>
- Temporal extensions: RT-LOTOS
  - ⇨ Delay, non deterministic delay, time-limited offer, time capture

Example TURTLE -> RT-LOTOS

Design TURTLE

RT-LOTOS



```
P[readPacket, forwardPacket](pdu) =
    readPacket?pdu;
    P[...](pdu) ||| P1[...](pdu)
endProc


P1[readPacket, forwardPacket](pdu) =
    Delay(1,2) forwardPacket!pdu;
        exit
endProc
```

(RT-)LOTOS: Formal Verification

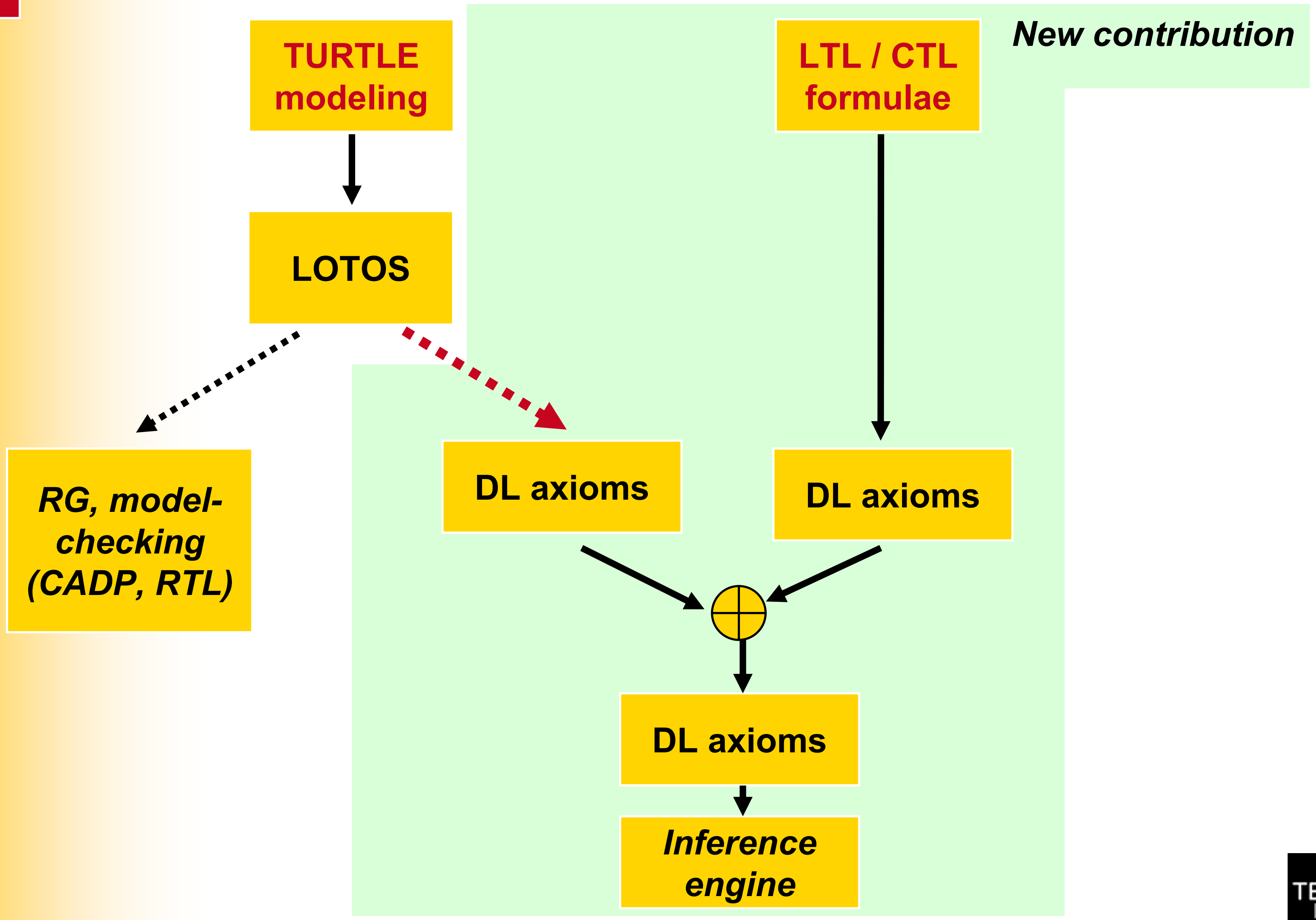
- Toolkits
  - For LOTOS: *CADP* (INRIA)
    - Based on Petri nets
    - Model-checking, reachability graph
  - For RT-LOTOS : *RTL* (LAAS-CNRS)
    - Construct a reachability graph
    - DTA (Dynamic Timed Automata)
- Current strong limitations
  - Only “regular” LOTOS description
    - For example, no recursivity over parallel operator
      - Useful in many schemes: modeling a web server, etc.
  - Time capture operator is not taken into account
    - Operator quite useful for modeling task schedulers
  - From TURTLE diagrams, very hard in code generators to avoid the two above mentioned schemes
  - Combinatory explosion

New Formal Verification Framework

- Idea: Rely on First Order Logics rather than on automata
  - ⇨ Translate (RT-)LOTOS specifications into First Order Logic specifications
- And more precisely LOTOS to DL
  - ⇨ DL = Description Logic
  - ⇨ Fragment of First Order Logic
  - ⇨ DL is decidable … but reduced expression power

LOTOS, DL

Main Principles



TURTLE modeling

LOTOS

RG, model-checking (CADP, RTL)

DL axioms

LTL / CTL formulae

DL axioms

DL axioms

Inference engine

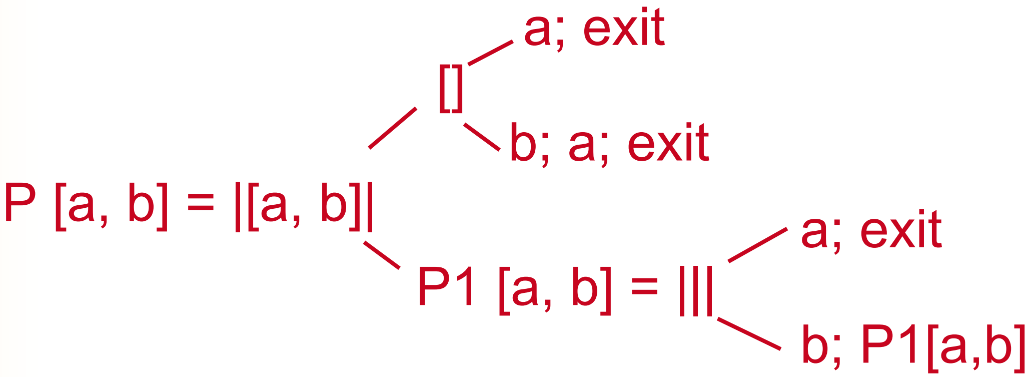
New contribution

LOTOS: Current Hypothesis / Limitations

- Basic LOTOS
  - ⇨ No variable
  - ⇨ RT-LOTOS operators are not taken into account
- No infinite recursion over synchronization contexts
- Infinite generation of gates (infinite recursion with hide operator)
- Preemption is not taken into account

LOTOS

- Gates
- Binary operator
    - |||, |[g0, .., gn]|, [], >>
- Termination processes
    - ⇨ Stop, exit



P [a, b] = |[a, b]|

[]

a; exit

b; a; exit

P1 [a, b] = |||

a; exit

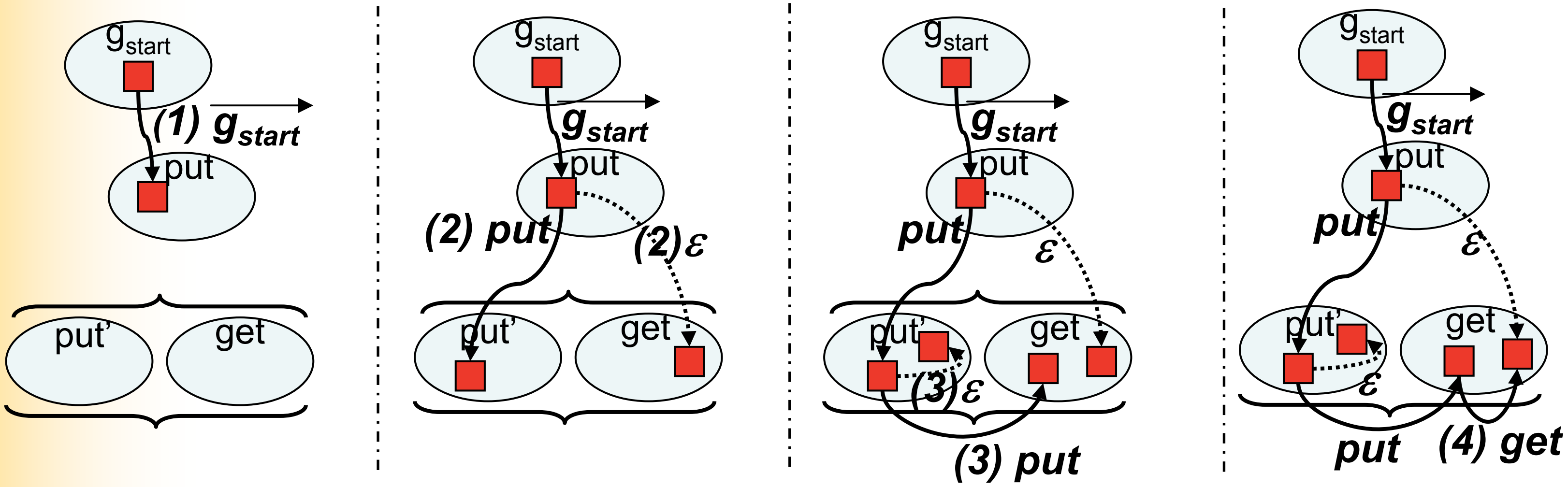
b; P1[a,b]

Example

Producer / Consumer System (1)

- Note: this example cannot be formally verified using usual LOTOS-based verification schemes
- P [put, get] = put; ((get; stop) ||| P[put, get])
- We have proved that the number of put is greater or equal to the number of get

Producer / Consumer System (2)



Conclusions

Conclusions and Future Work

- New formal verification scheme
  - ⇨ Very promising
- Future work
  - ⇨ Address limitations!
    - Some are weak (preemption)
    - Some are strong (variables, temporal operators)
  - ⇨ Probably another more “powerful” FOL shall be used
    - But decidability issue
  - ⇨ Implementation of an inference engine

References

- C. Leduc and S. Coudert and L. Apvrille: Formal Verification of LOTOS Specifications Using Description Logics, Technical Report, Institut TELECOM / TELECOM ParisTech, number FR-0451
  - http://www.comelec.enst.fr/recherche/labsoc/projets/AMIGOS.en
- L. Apvrille, S. Coudert and C. Leduc , A Framework for the Formal Verification of infinite Systems, The 18th IEEE International Symposium on Software Reliability Engineering (ISSRE 2007), Trollhättan, Sweden, November 2007