Intermediate Models for the Verification of Asynchronous Real-Time Embedded Systems

Definition and Application of the ATLANTIF language

Jan Stöcker

Advisor: Hubert Garavel / Co-Advisor: Frédéric Lang (VASY project team)

Funded by INRIA PhD fellowship

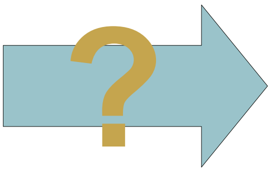
Abstract: To model real-life critical systems, one needs "high-level" languages to express three important concepts: complex data structures, concurrency, and real-time. So far, the verification of timed systems has been successfully applied to "low-level" models, such as timed extensions of automata or of Petri nets.
To bridge the gap between high-level languages, which allow a concise modeling of systems, and low-level models, for which efficient algorithms and tools have been designed, this work proposes an intermediate model named ATLANTIF. This model has a formally defined syntax and semantics covering a large set of high-level constructs. Furthermore, translations to low-level models have been implemented.

Keywords: concurrency, formal method, intermediate model, process algebra, real-time, time Petri net, timed automaton, verification

High-level languages:

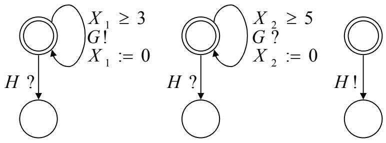
- Examples: timed process algebras such as E-LOTOS, LOTOS NT, TCSP, …
- Expressive and concise
- But not many tools

```
specification X is gates G, H
behaviour (loop wait 3; G endloop
               | [G] |
              loop wait 5; G endloop) [> H; null
endspec
```



Low-level models:

- Examples: timed automata (TA), time Petri nets (TPN)
- Conceived for verification tools: Uppaal, Kronos, Red, Tina, Roméo
- But in which it is difficult to specify large industrial examples



Problem: Bridge the gap between languages and models, to allow formal verification on complex specifications.

Solution: We developed the **ATLANTIF** intermediate format, which covers the following aspects:

- *Data* (simple types and complex types)
- *Control* (communication, synchronization between processes, process activation/deactivation)
- *Real-Time* (delays, temporal constraints on communications, urgency, latency)

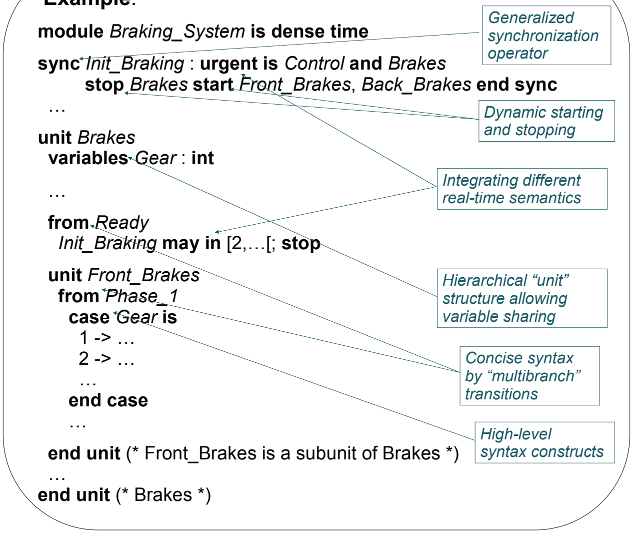
Related Work:

Other intermediate models exist.

- Fiacre,
- BIP,
- MoDeST, etc.

But ATLANTIF represents Data, Control, and Real-Time with several features unavailable in these models.

Example:

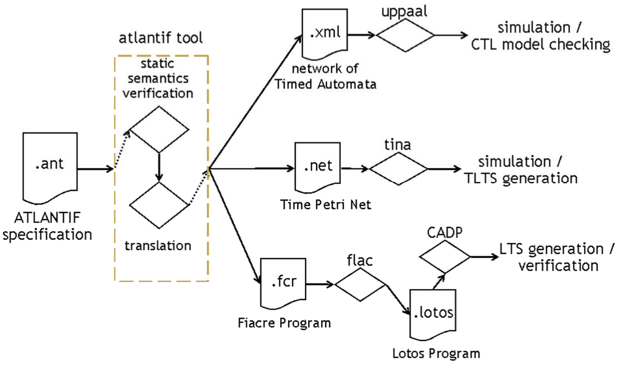


```
module Braking_System is dense time
sync Init_Braking : urgent is Control and Brakes
     stop Brakes start Front_Brakes, Back_Brakes end sync
  …
unit Brakes
  variables Gear : int

  …
  from Ready
    Init_Braking may in [2,…[; stop
  unit Front_Brakes
    from Phase_1
      case Gear is
        1 -> …
        2 -> …
        …
      end case
      …
  end unit (* Front_Brakes is a subunit of Brakes *)
  …
end unit (* Brakes *)
```

Generalized synchronization operator
Dynamic starting and stopping
Integrating different real-time semantics
Hierarchical "unit" structure allowing variable sharing
Concise syntax by "multibranch" transitions
High-level syntax constructs

Translator tool (~ 18,000 lines of code):


atlantif tool
static semantics verification
translation
.ant
ATLANTIF specification
.xml
network of Timed Automata
uppaal
simulation / CTL model checking
.net
Time Petri Net
tina
simulation / TLTS generation
.fcr
Fiacre Program
flac
.lotos
Lotos Program
CADP
LTS generation / verification

Conclusion: easier formal verification of complex systems

Publications:

- J. Stöcker, F. Lang, H. Garavel: *Parallel Processes with Real-Time and Data: The ATLANTIF Intermediate Format*, in M. Leuschel and H. Wehrheim (Eds.), Proc. of the 7th International Conference on integrated Formal Methods iFM 2009 (Düsseldorf, Germany), LNCS 5432, February 16-19 2009

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

Laboratoire d'Informatique de Grenoble — Janvier 2009

LIG