Logical Time @ work: the RT-Simex project

Julien DeAntoni, Frédéric Mallet, Charles André
I3S/UNS/INRIA

Frédéric Thomas
Obeo

SAFA 2010

Logical Time @ work: the RT-Simex project

Julien DeAntoni, Frédéric Mallet,
I3S/UNS/INRIA

Frédéric Thomas
Obeo

SAFA 2010

Logical Time…

- focuses on causal relations between events
- Is independent of the abstraction level
- Is multi-clock (polychronous)
- Provides a partial order between events

After 23 starts of the computer, a disk check is done

The computation duration is 156 processor ticks

Logical Time…

- focuses on causal relations between events
- Is independent of the abstraction level
- Is multi-clock (polychronous)
- Provides a partial order between events

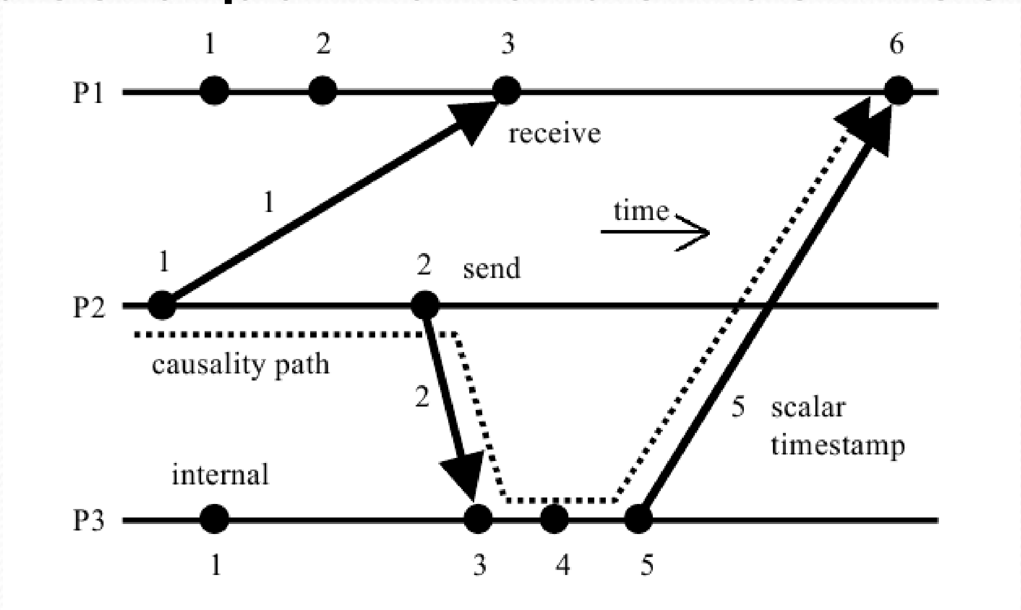
After 23 starts of the computer, a disk check is done

The computation duration is 156 processor ticks

In modern laptop, tick is logical since the processor speed depend on the battery level

Logical Time…

- focuses on causal relations between events
- Is independent of the abstraction level
- Is multi-clock (polychronous)
- Provides a partial order between events



LOGICAL TIME
Friedemann Mattern
Darmstadt University of Technology

Physical time Time...

- Can be seen as a special case of logical time

After 5 secondes, it stops...

≅

After 5 events of the “second” clock, it stops

Logical Time @ work: the RT-Simex project

Julien DeAntoni, Frédéric Mallet,
I3S/UNS/INRIA

Frédéric Thomas
Obeo

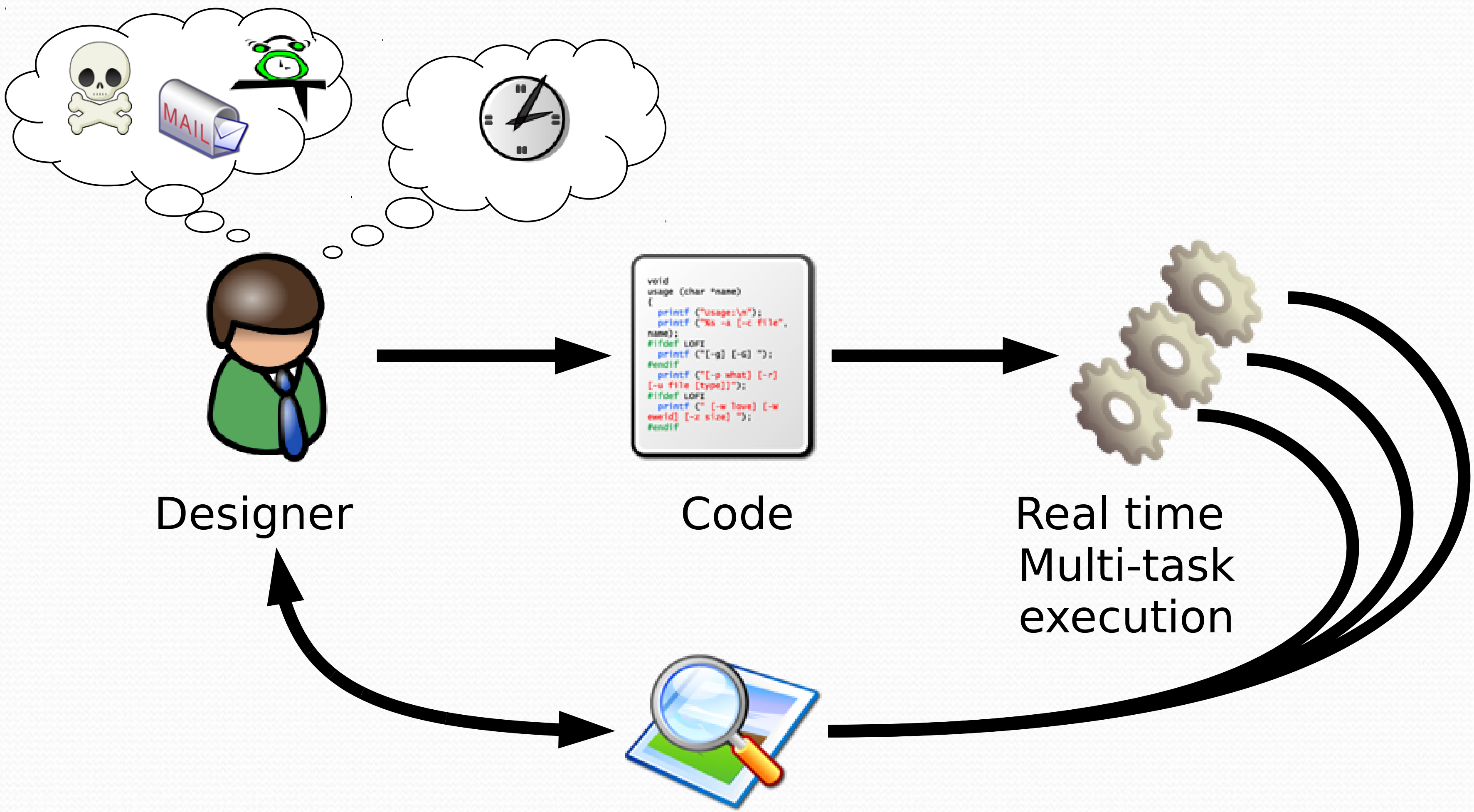
SAFA 2010

Retro-ingénierie de Traces d'analyse de SIMulation et d'EXécution de systèmes temps-réel

RT-Simex

Obeo INRIA i3S sophia antipolis Université Nice SOPHIA ANTIPOLIS

Context : help the designer



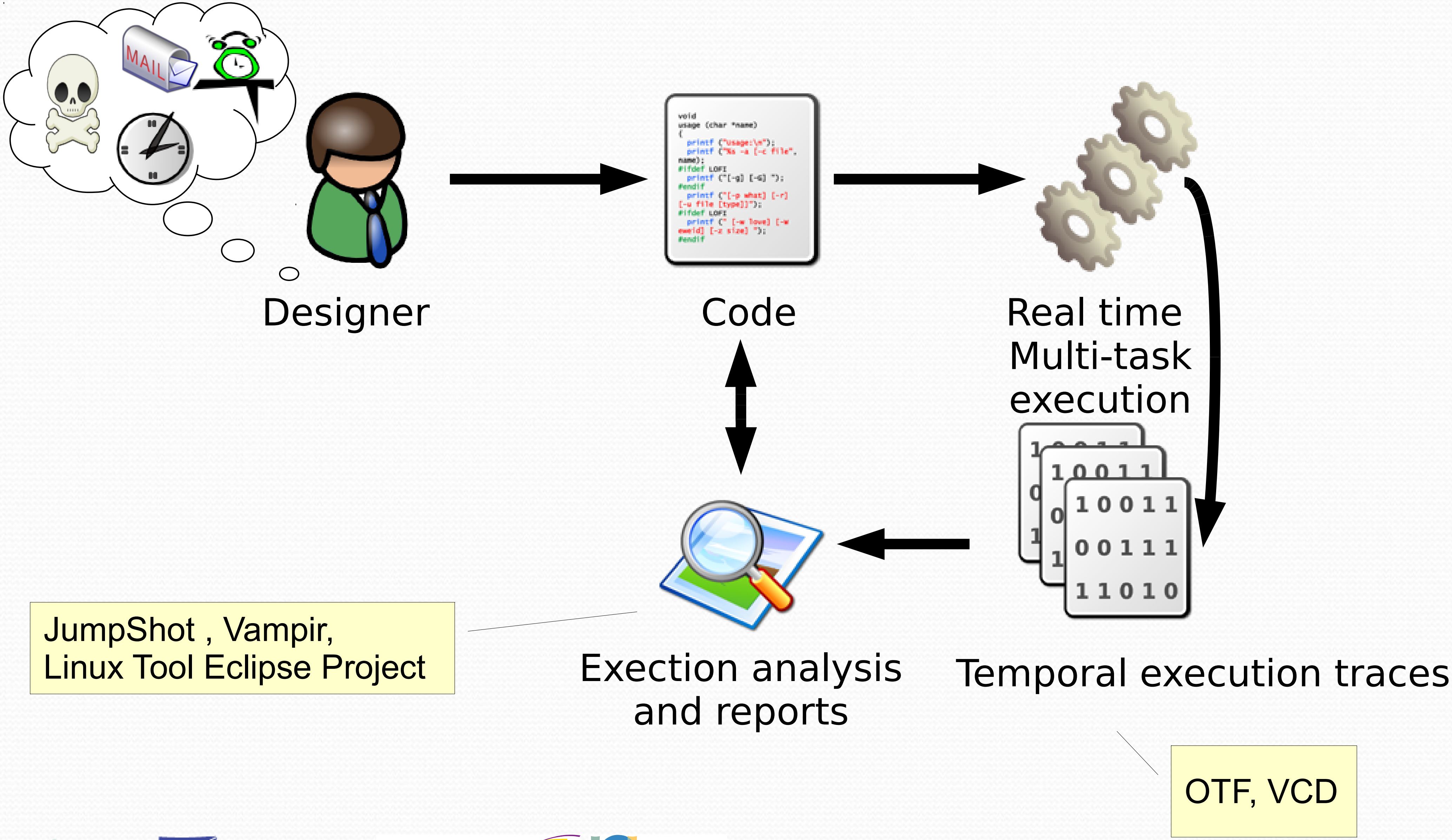
Designer

Code

Real time
Multi-task
execution

Analysis of the
execution

Context: execution traces



Designer

Code

Real time Multi-task execution

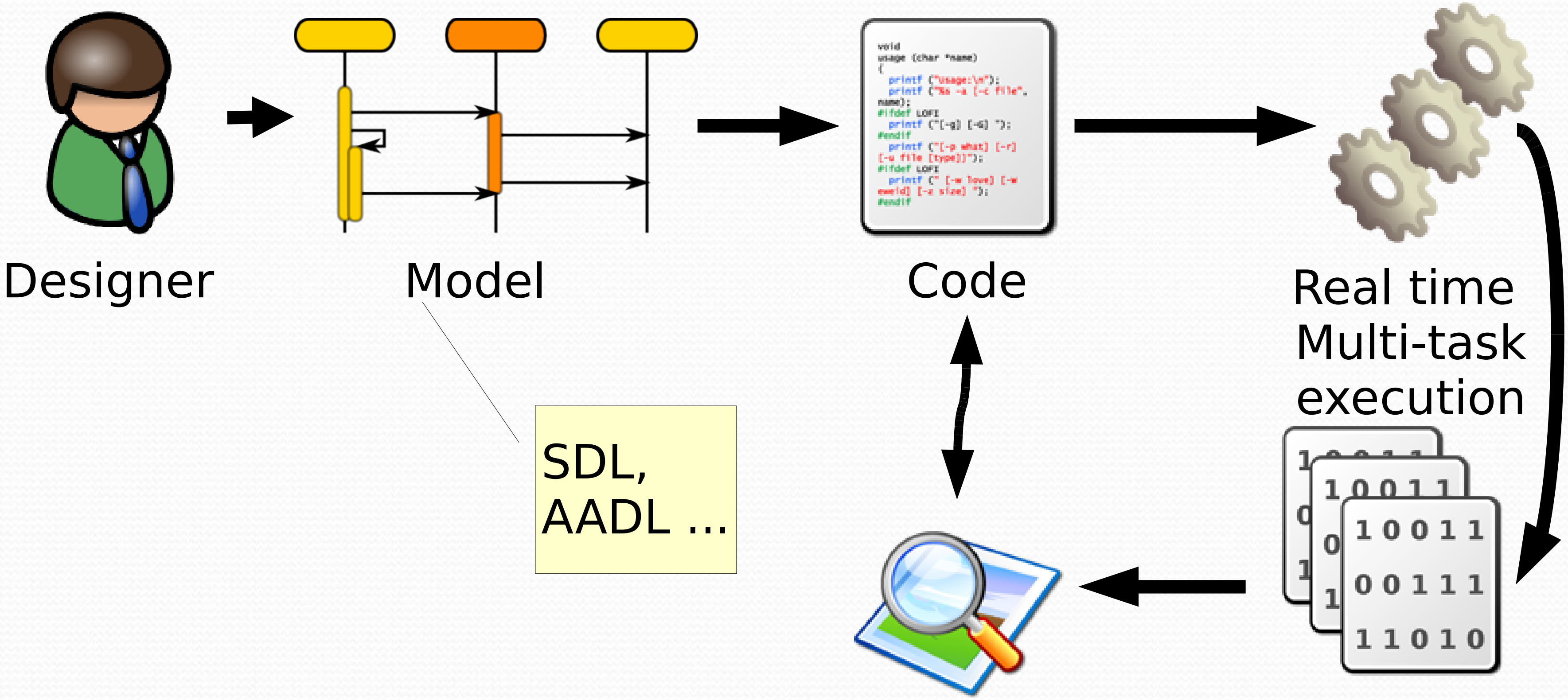
Temporal execution traces

Exection analysis and reports

JumpShot , Vampir,
Linux Tool Eclipse Project

OTF, VCD

observations



Designer

Model

SDL, AADL ...

Code

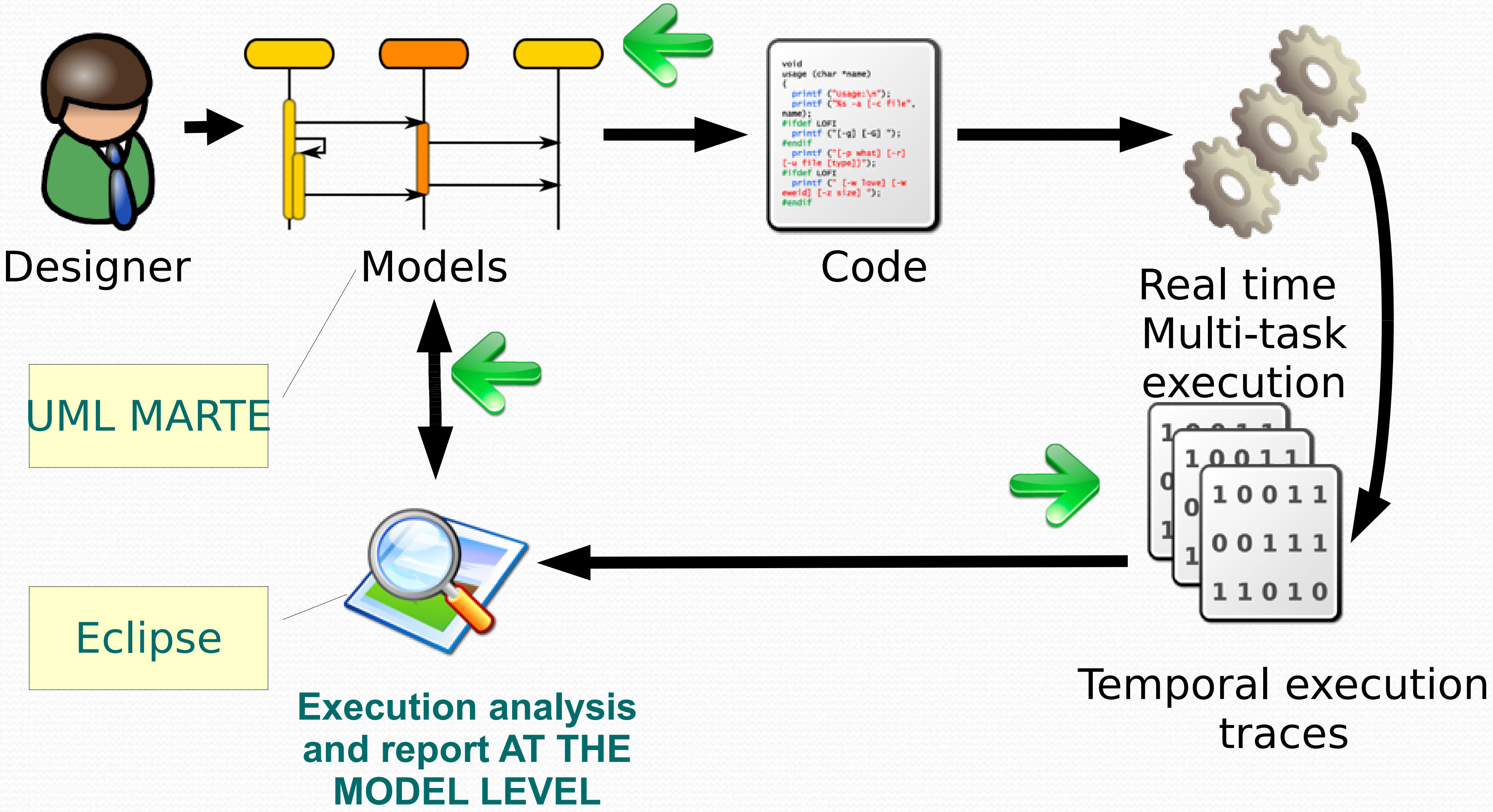
```
void
usage (char *name)
{
   printf ("usage:\n");
   printf ("%s -a [-c file",
name);
#ifdef LOFI
   printf ("[-g] [-G] ");
#endif
   printf ("[-p what] [-r]
[-u file [type]]");
#ifdef LOFI
   printf (" [-w love] [-w
eweird] [-z size] ");
#endif
```

Real time Multi-task execution

Temporal execution traces

Execution analysis and report

RT-Simex objectives



Designer

Models

Code

```
void
usage (char *name)
{
   printf ("usage:\n");
   printf ("%s -a [-c file",
name);
#ifdef LOFI
   printf ("[-g] [-G] ");
#endif
   printf ("[-p what] [-r]
[-u file [type]]");
#ifdef LOFI
   printf (" [-w love] [-w
eweld] [-z size] ");
#endif
```

Real time Multi-task execution

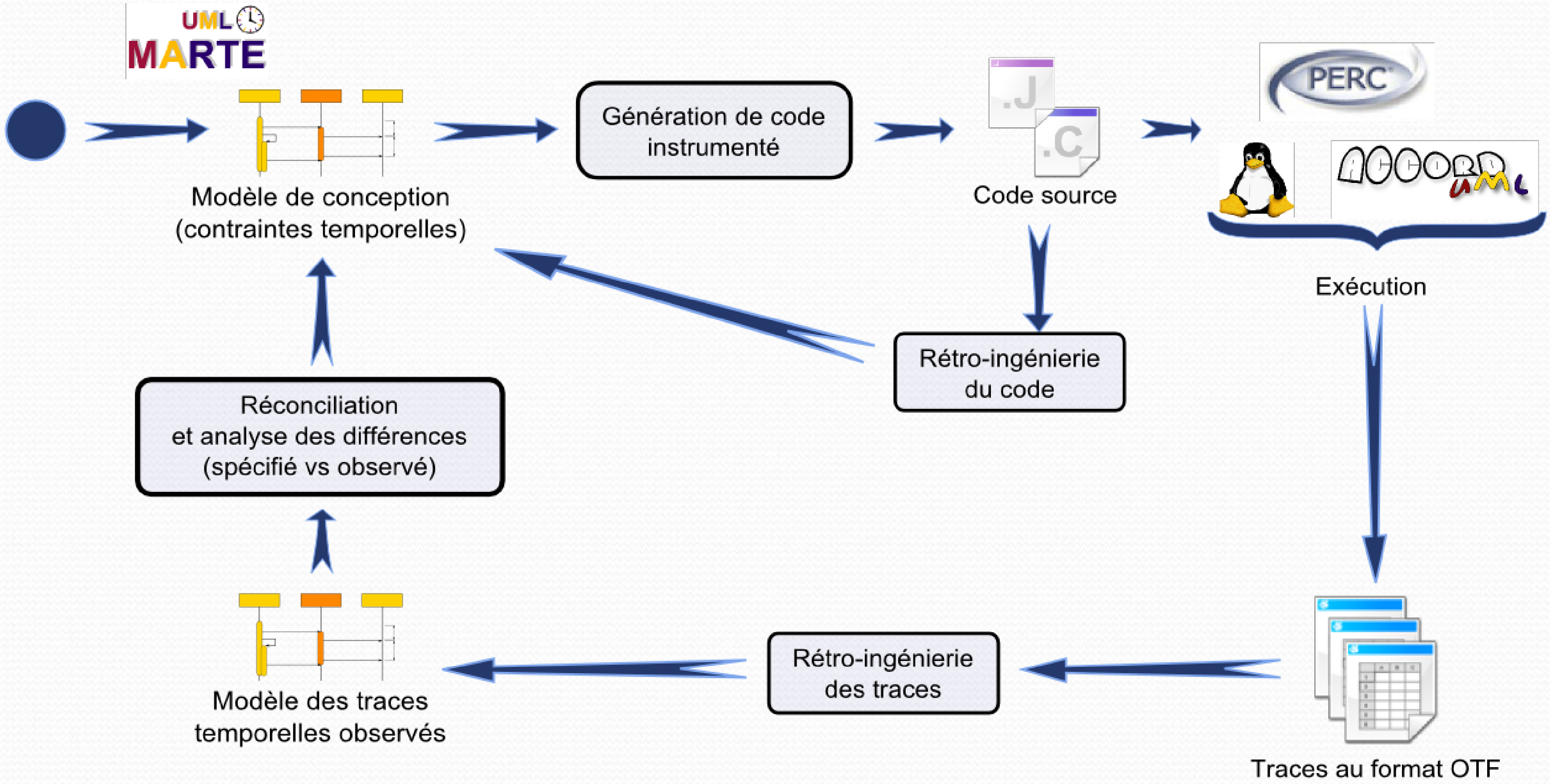
Temporal execution traces

UML MARTE

Eclipse

Execution analysis and report AT THE MODEL LEVEL

Processus RT-Simex



Modèle de conception (contraintes temporelles)

Génération de code instrumenté

Code source

Exécution

Rétro-ingénierie du code

Réconciliation et analyse des différences (spécifié vs observé)

Traces au format OTF

Rétro-ingénierie des traces

Modèle des traces temporelles observés

Logical Time @ work: the RT-Simex project

Julien DeAntoni, Frédéric Mallet,
I3S/UNS/INRIA

Frédéric Thomas
Obeo

SAFA 2010

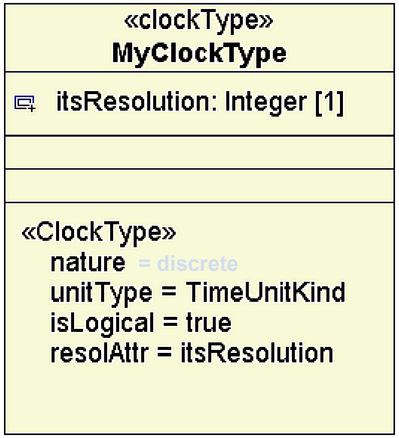
MARTE TIME MODEL

- SubProfile of the MARTE UML profile standardized by the OMG (Object Management Group)
  - Reviewed and accepted by the community
  - Implemented in Papyrus, magic draw, etc
  - Under Implementation in other UML tools

- A Domain Model integrated with eclipse and usable with Domain Specific Language

MARTE TIME MODEL

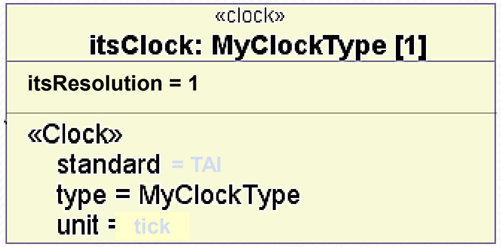
- The main concept is the **Clock**.
  - It is a way to specify a, possibly infinite, ordered set of instants
  - It can be logical or chronometric, discrete or dense
  - Its type is a ClockType



«clockType»
MyClockType

itsResolution: Integer [1]

«ClockType»
nature = discrete
unitType = TimeUnitKind
isLogical = true
resolAttr = itsResolution

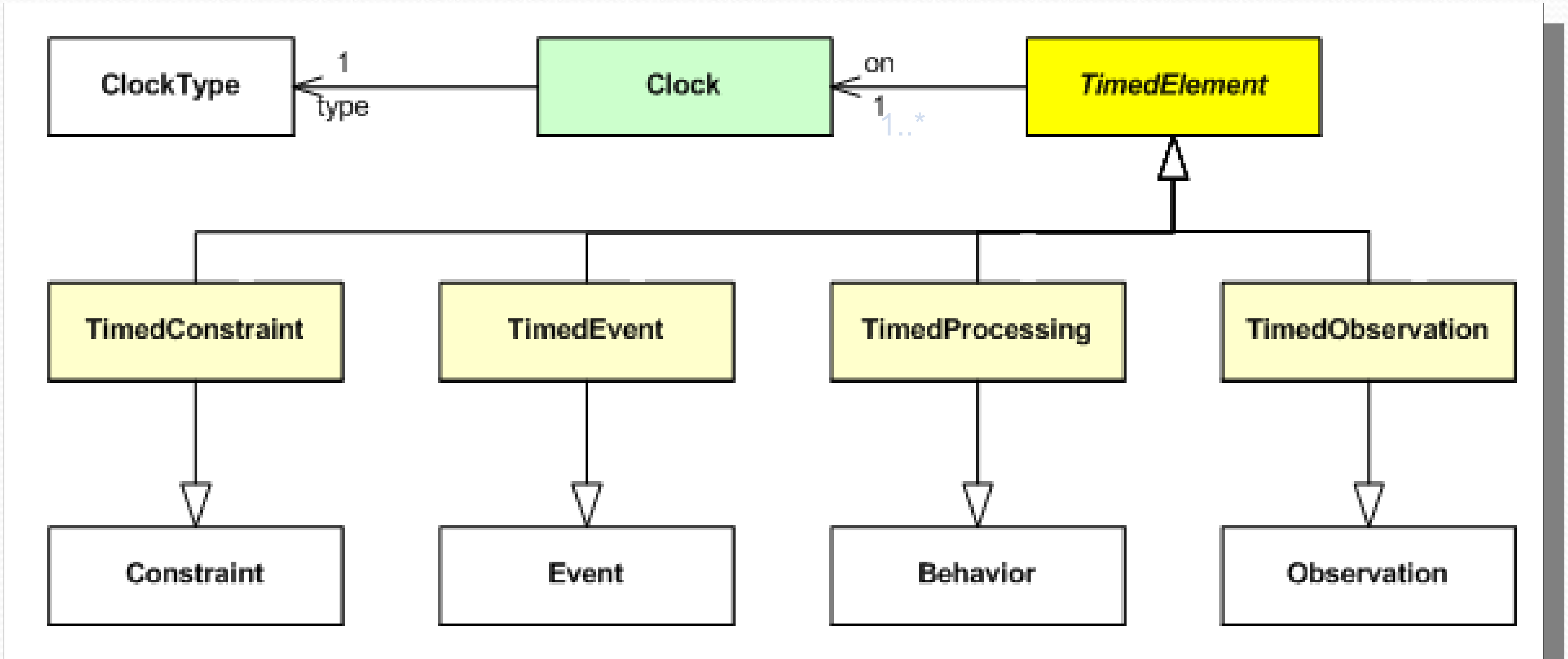


«clock»
itsClock: MyClockType [1]

itsResolution = 1

«Clock»
standard = TAI
type = MyClockType
unit = tick

MARTE TIME MODEL

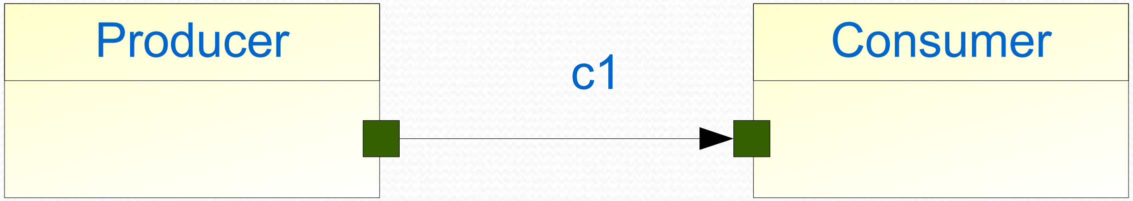


Simplified view of the MARTE Time meta-Model

MARTE TIME MODEL

- Sketchy example of its use

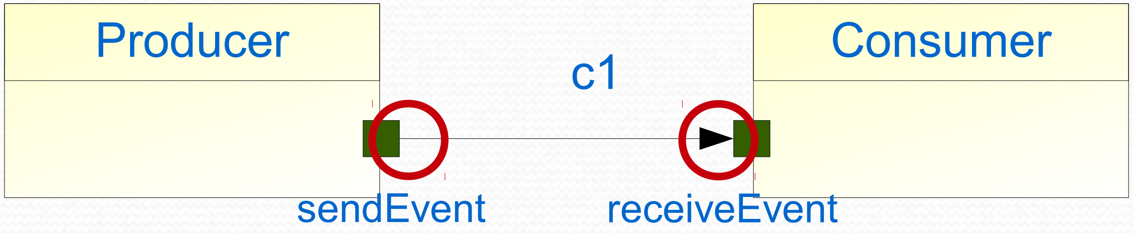
User model



MARTE TIME MODEL

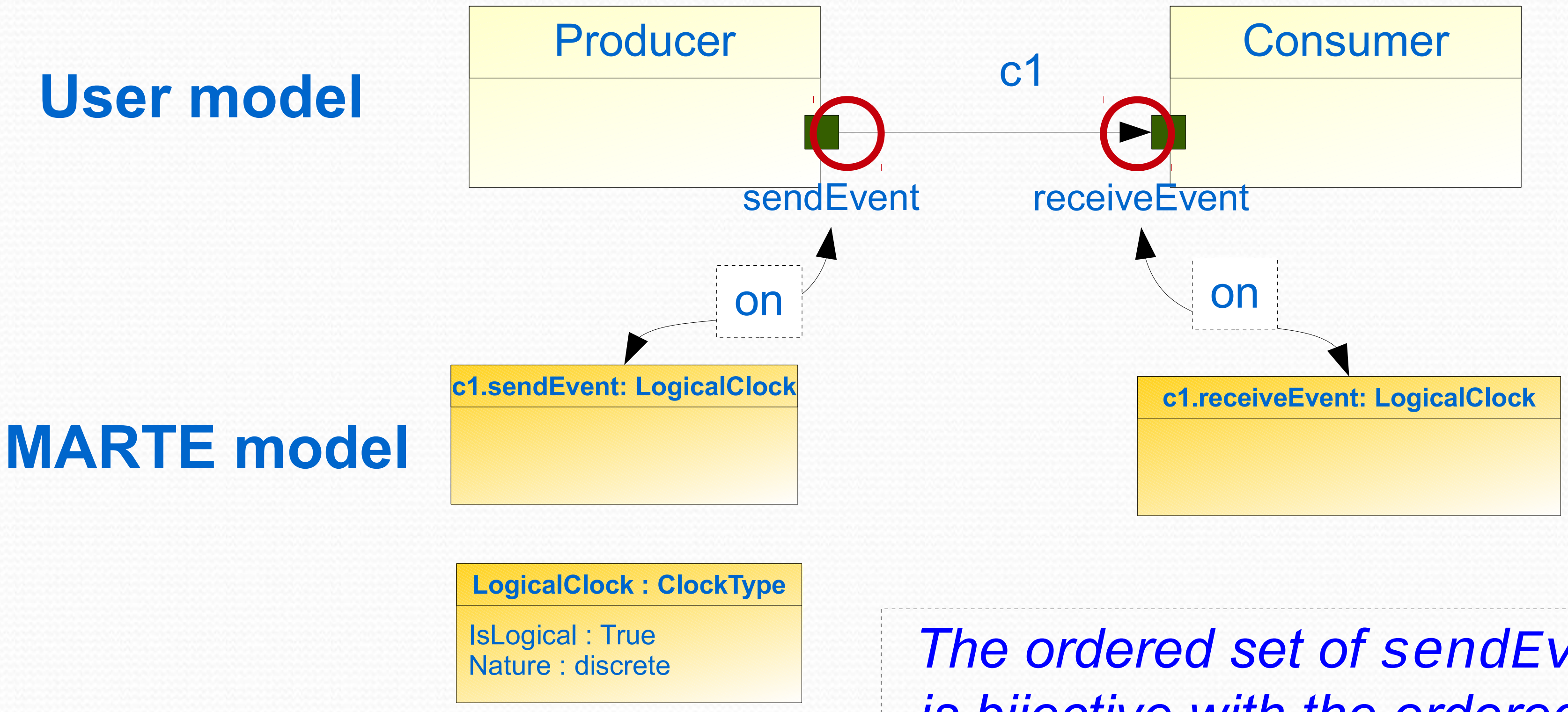
- Sketchy example of its use

User model



MARTE TIME MODEL

- Sketchy example of its use



The ordered set of `sendEvent` is bijective with the ordered set of instants of `c1.sendEvent`

CCSL

- **C**lock **C**onstraint **S**pecification **L**anguage
  - Firstly introduced in the MARTE TIME profile
  - Declarative model-based language integrated with Eclipse
  - Formal semantics (both denotational and operational)
  - Tooled (TimeSquare)

→ **Explicitly represents / specifies relations between clocks**

CCSL (**C**lock **C**onstraint **S**pecification **L**anguage)

– Relations: dependencies between clocks

- Coincidence → **=**
- Exclusion → **#**
- Precedence → **<**
- Alternance → **~**

– Expressions: a mean to create new clocks from others

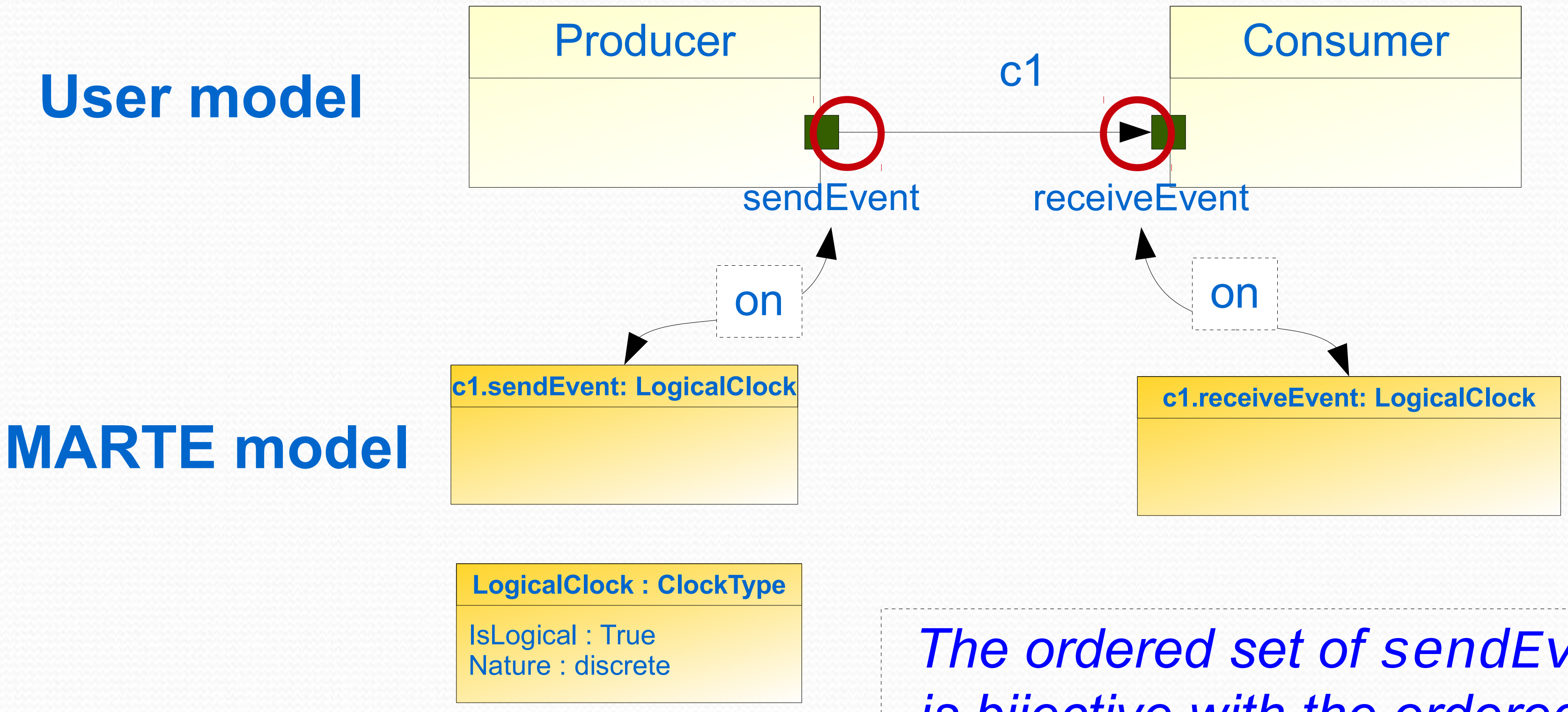
- Delay → **delayedFor** *X* **on** *aClock*
- Filtering → *aClock* **filteredBy** *aBinaryWord*
- Union → *aClock* **union** *anotherClock*
- Intersection → *aClock* **inter** *anotherClock*
- Periodicity → **periodicOn** aClock **period** X **offset** Y
- …

CCSL (**C**lock **C**onstraint **S**pecification **L**anguage)

- Relations: dependencies between clocks
  - Coincidence → **=**
  - Exclusion → **#**
  - Precedence → **<**
  - Alternance → **~**
- Expressions: a mean to create new clocks from others
  - Delay → **delayedFor** *X* **on** *aClock*
  - Filtering → *aClock* **filteredBy** *aBinaryWord*
  - Union → *aClock* **union** *anotherClock*
  - Intersection → *aClock* **inter** *anotherClock*
  - Periodicity → **periodicOn** aClock **period** X **offset** Y
- Libraries: user-defined relations and expressions
  - ...

CCSL

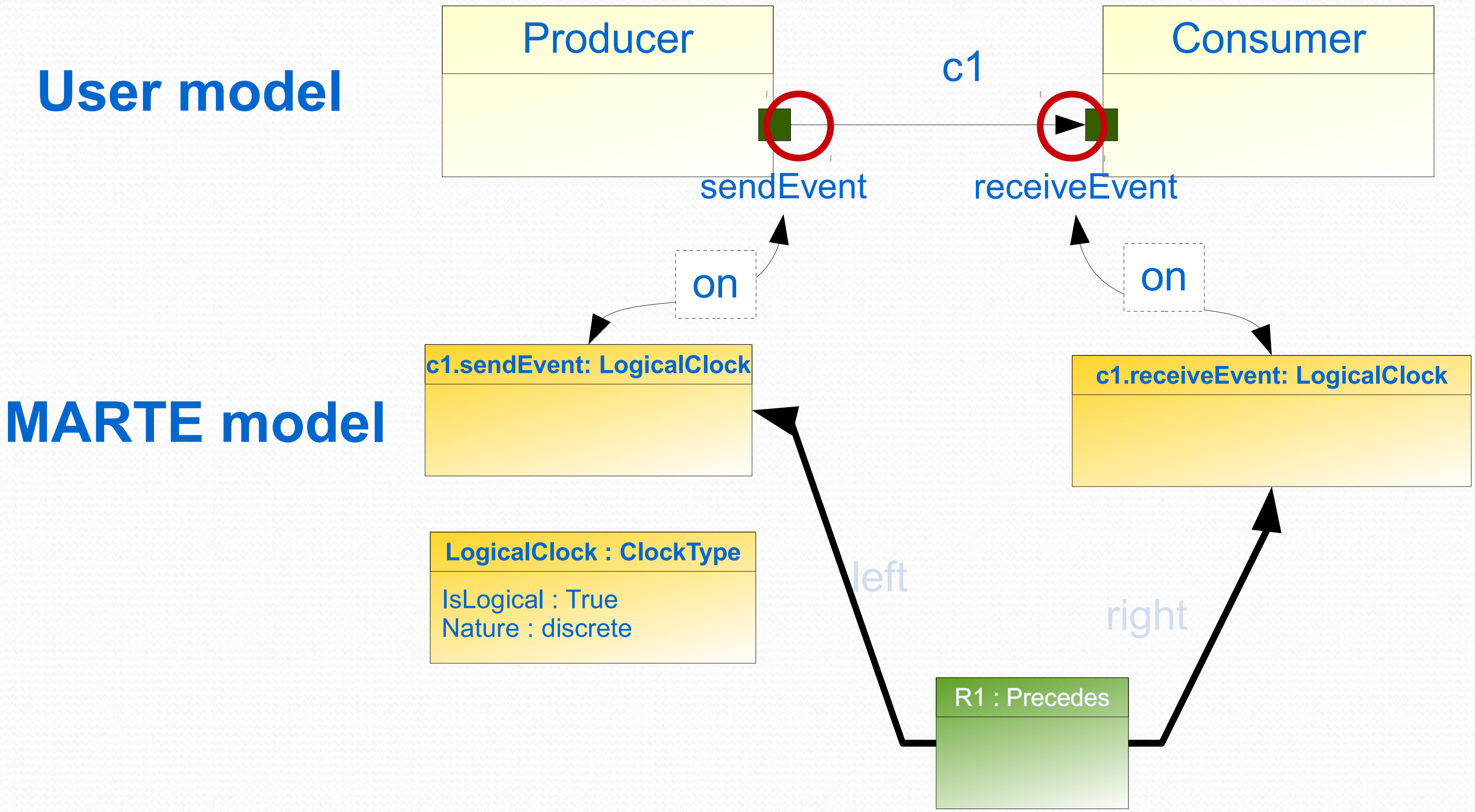
- Sketchy example of its use



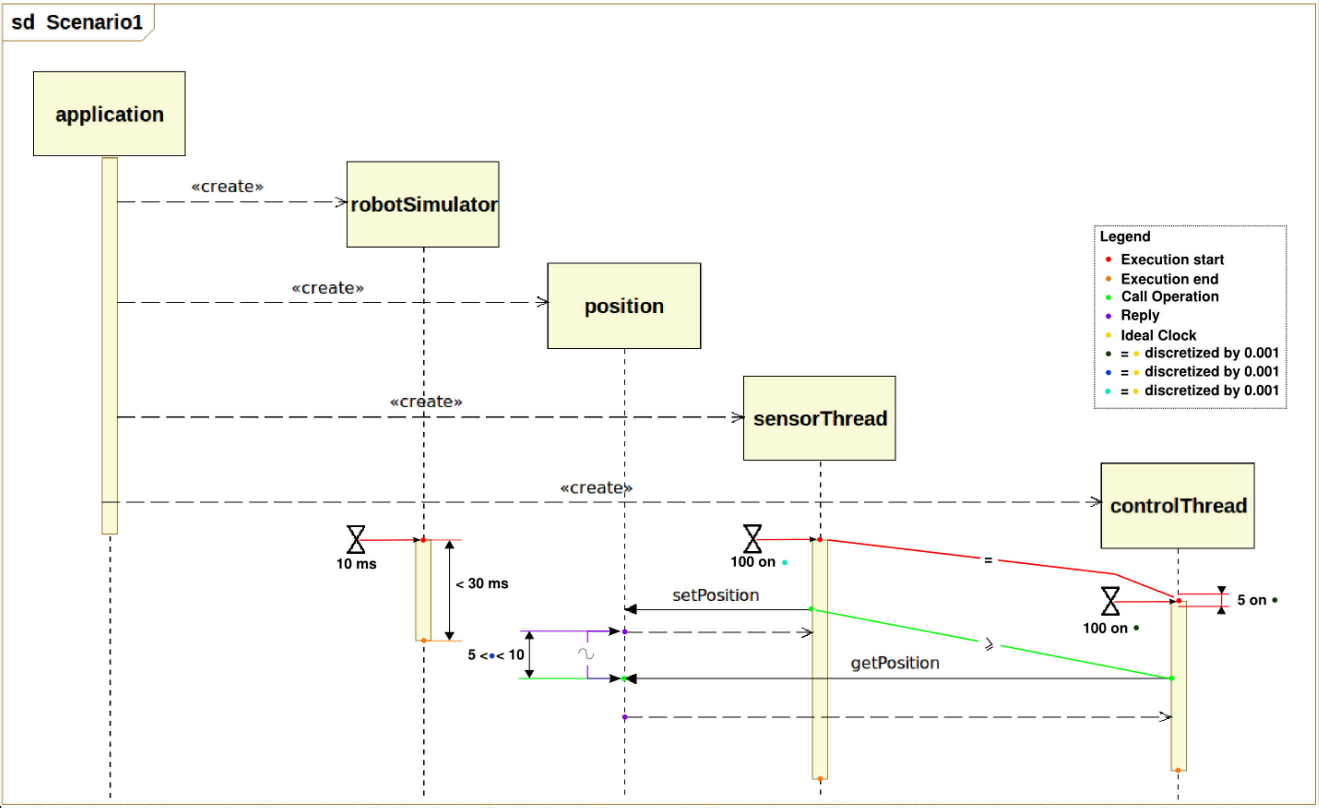
The ordered set of `sendEvent` is bijective with the ordered set of instants of `c1.sendEvent`

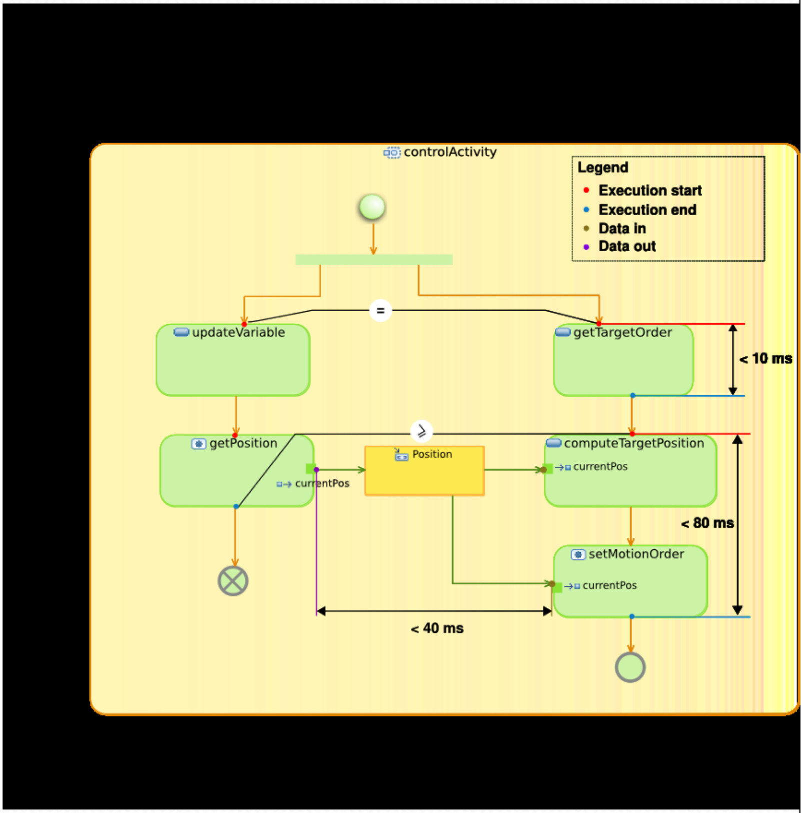
CCSL

- Sketchy example of its use



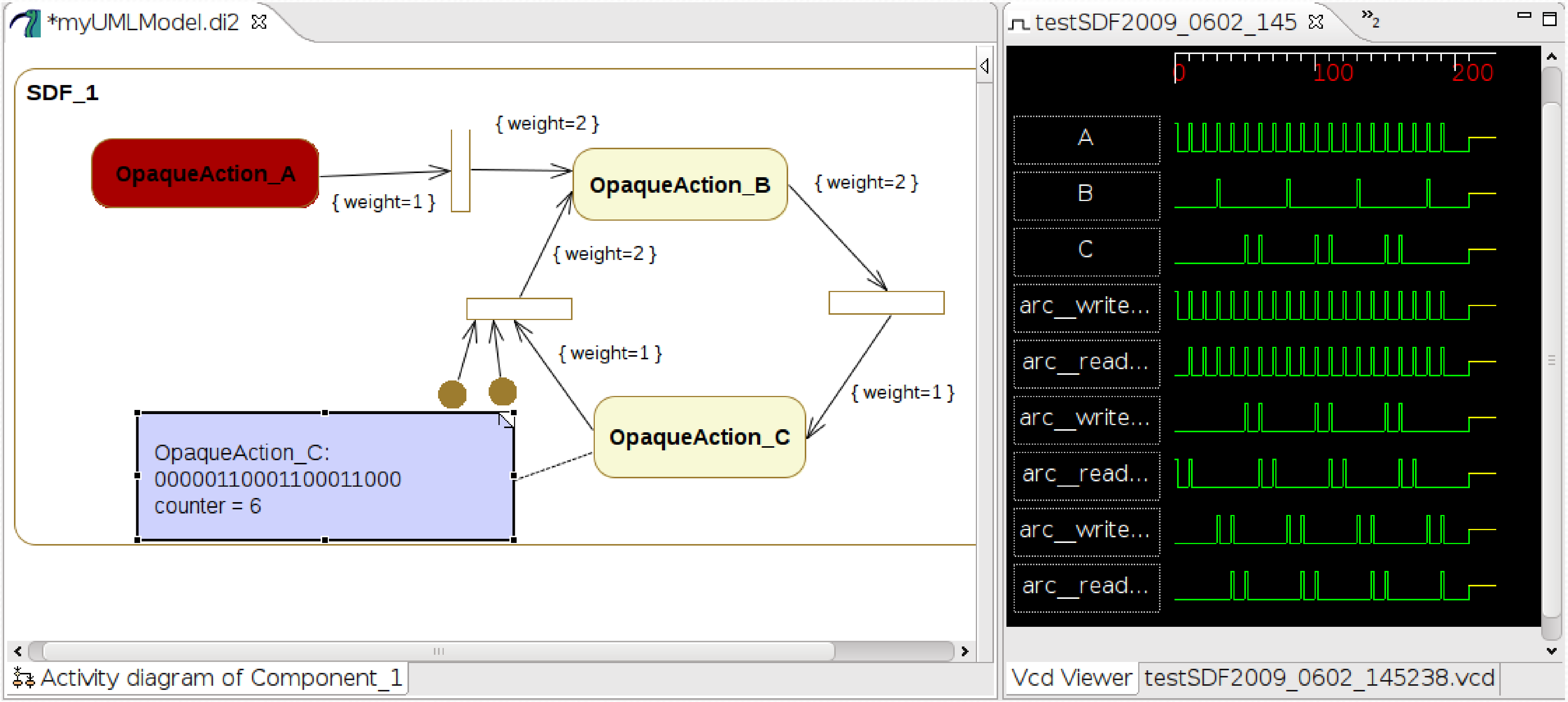
Graphical formal annotation over a UML model **for RT-Simex**…





Simulate and animate the UML/MARTE model **in TimeSquare**

(http://www-sop.inria.fr/aoste/dev/time_square/)

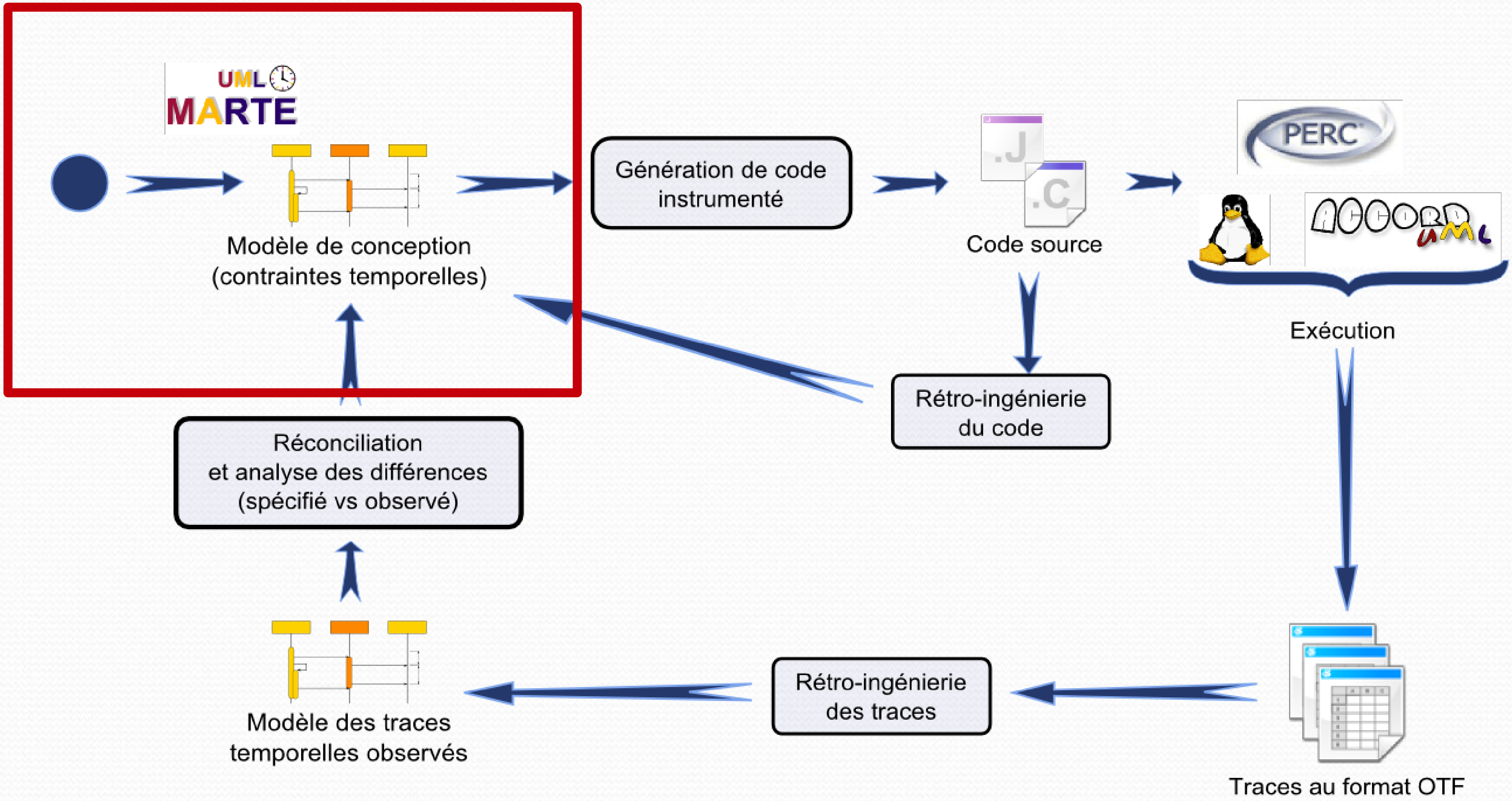


OpaqueAction_C:
00000110001100011000
counter = 6

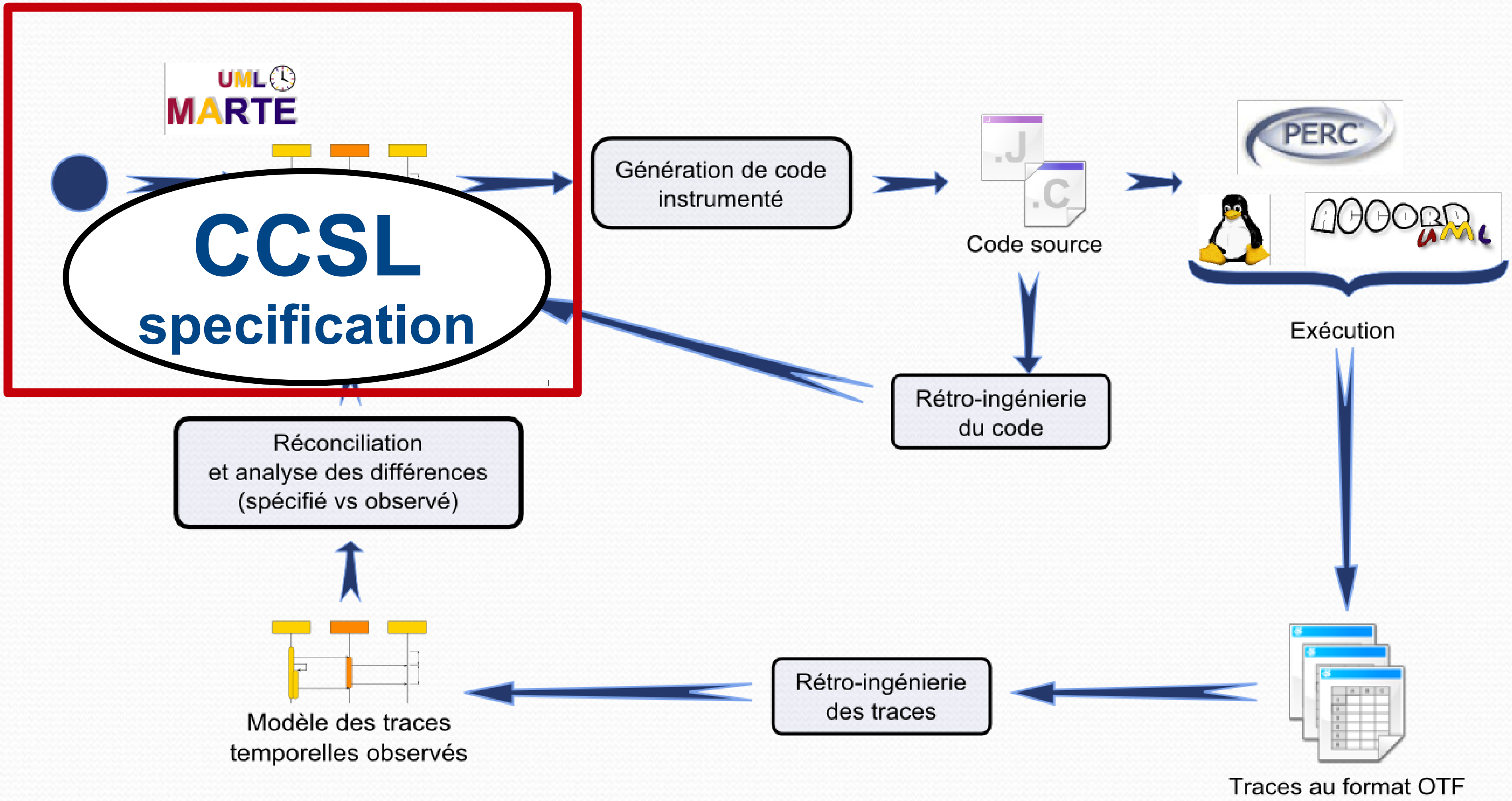
Simulate and animate the UML/MARTE model **in TimeSquare**

(http://www-sop.inria.fr/aoste/dev/time_square/)

Processus RT-Simex

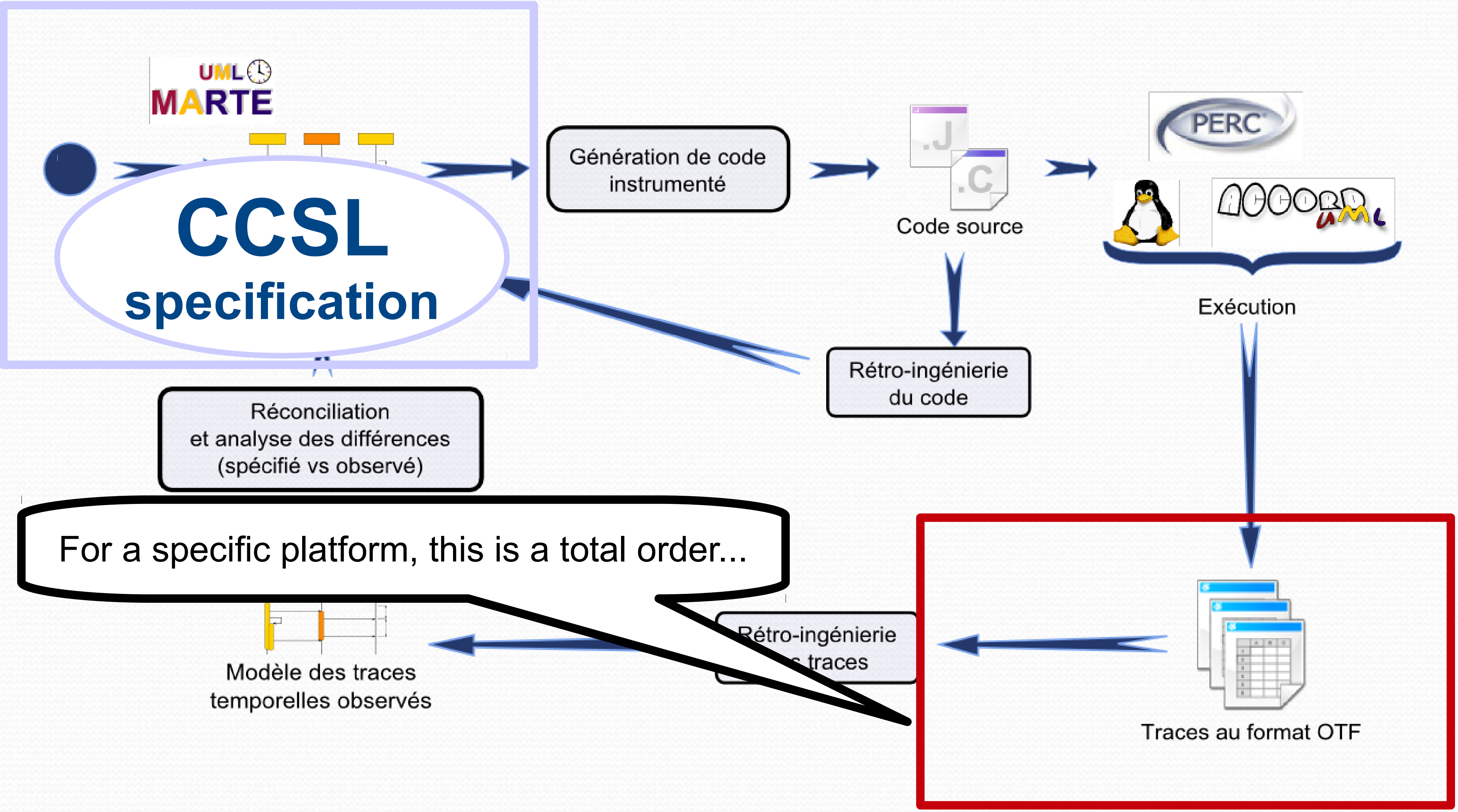


Processus RT-Simex



UML MARTE
CCSL
specification
Génération de code instrumenté
.J
.C
Code source
PERC
ACCORD UML
Exécution
Rétro-ingénierie du code
Réconciliation et analyse des différences (spécifié vs observé)
Modèle des traces temporelles observés
Rétro-ingénierie des traces
Traces au format OTF

Processus RT-Simex



CCSL specification

Génération de code instrumenté

Code source

Exécution

Rétro-ingénierie du code

Réconciliation et analyse des différences (spécifié vs observé)

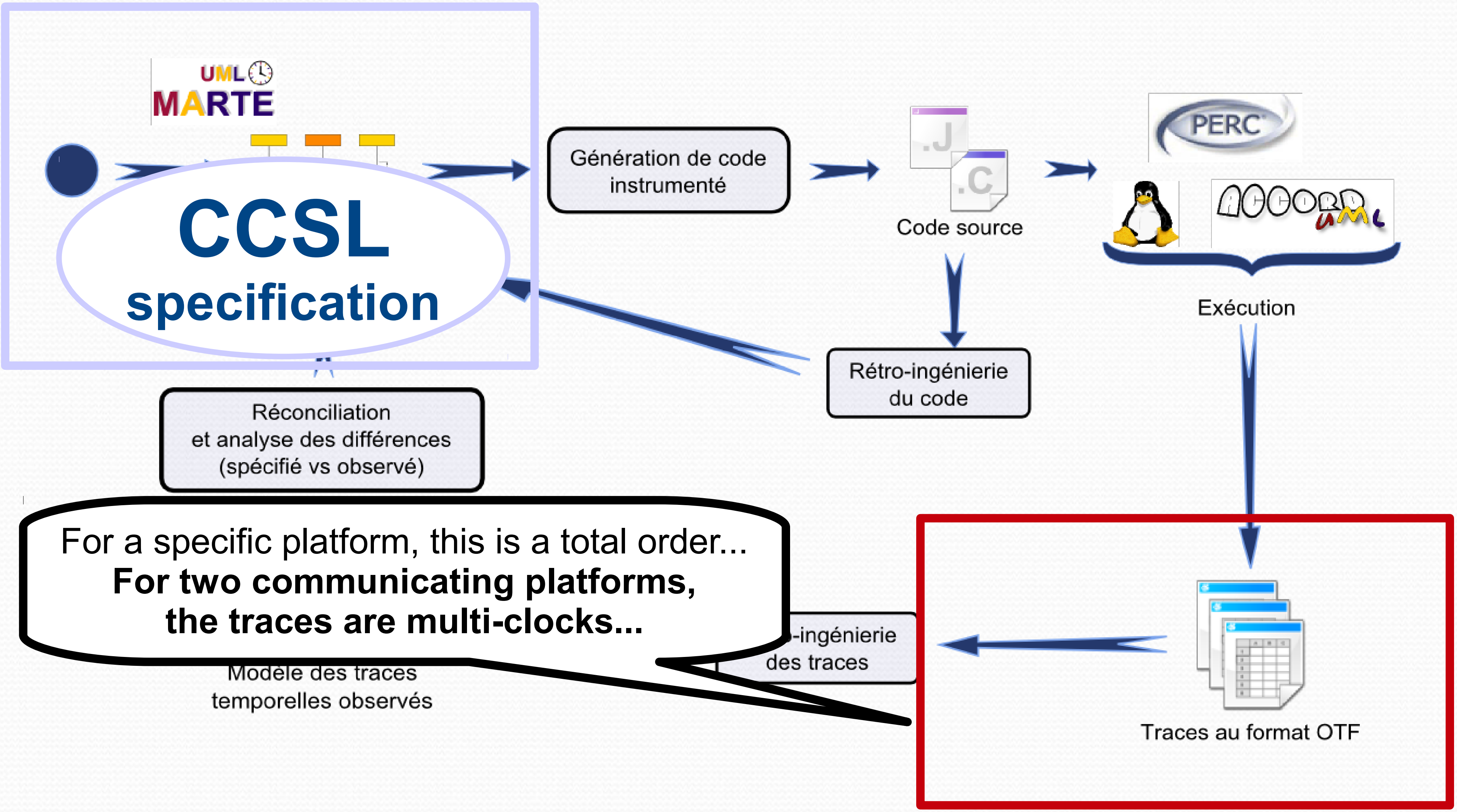
For a specific platform, this is a total order...

Rétro-ingénierie s traces

Modèle des traces temporelles observés

Traces au format OTF

Processus RT-Simex



CCSL specification

Génération de code instrumenté

Code source

Exécution

Rétro-ingénierie du code

Réconciliation et analyse des différences (spécifié vs observé)

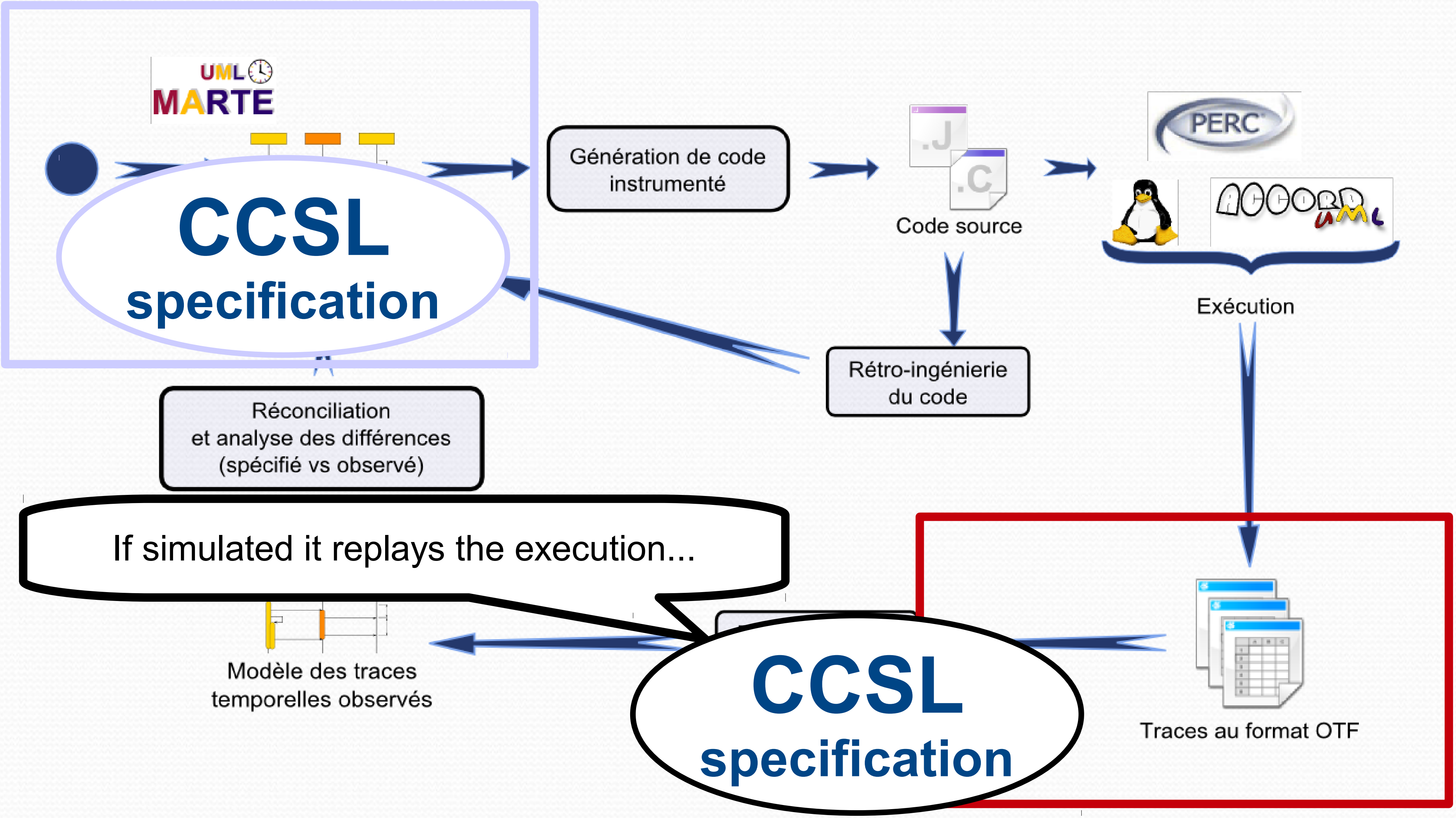
For a specific platform, this is a total order...
For two communicating platforms,
the traces are multi-clocks...

-ingénierie des traces

Modèle des traces temporelles observés

Traces au format OTF

Processus RT-Simex



CCSL specification

Génération de code instrumenté

Code source

Exécution

Rétro-ingénierie du code

Réconciliation et analyse des différences (spécifié vs observé)

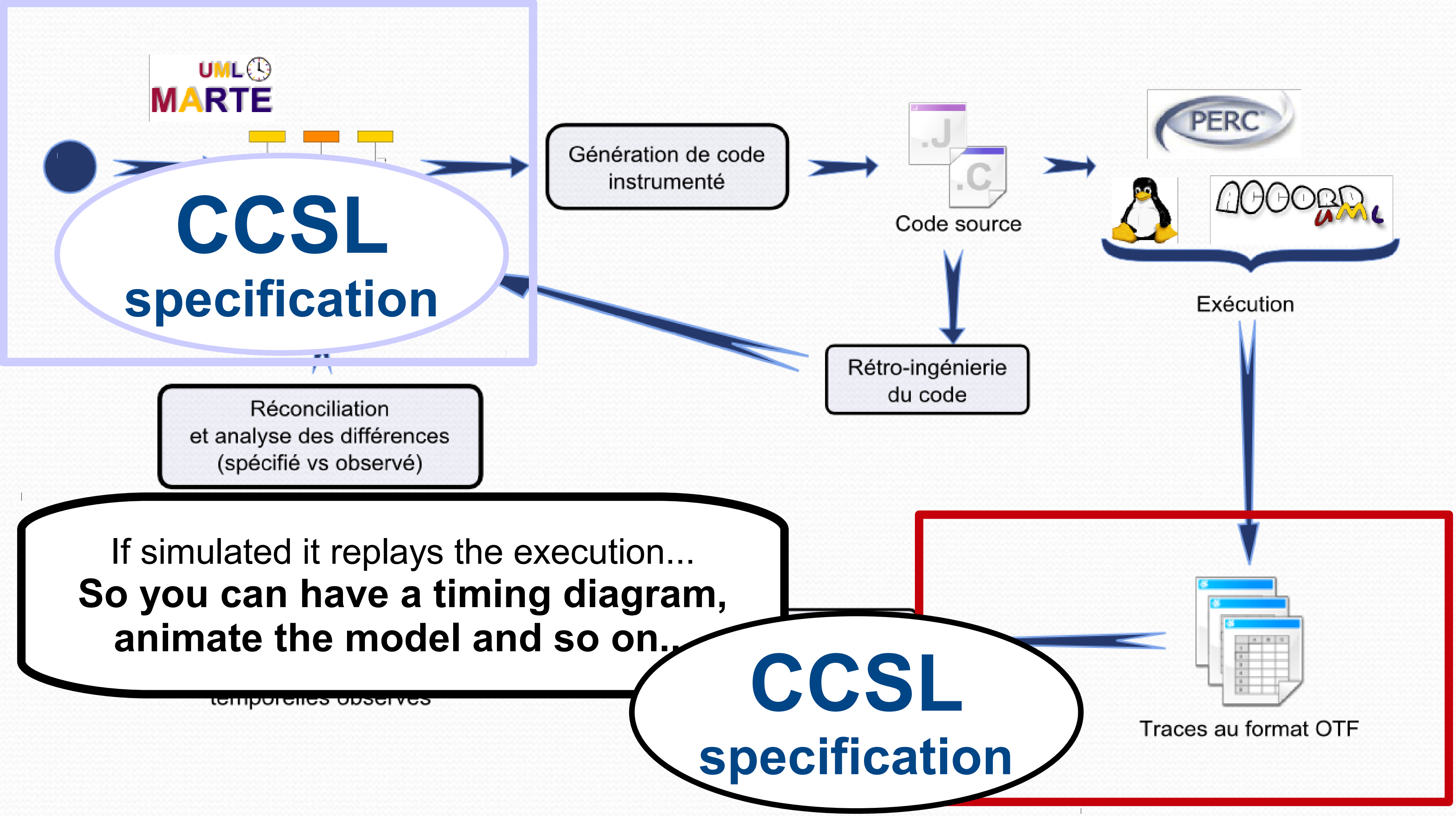
If simulated it replays the execution...

Modèle des traces temporelles observés

CCSL specification

Traces au format OTF

Processus RT-Simex



CCSL specification

Génération de code instrumenté

Code source

Exécution

Rétro-ingénierie du code

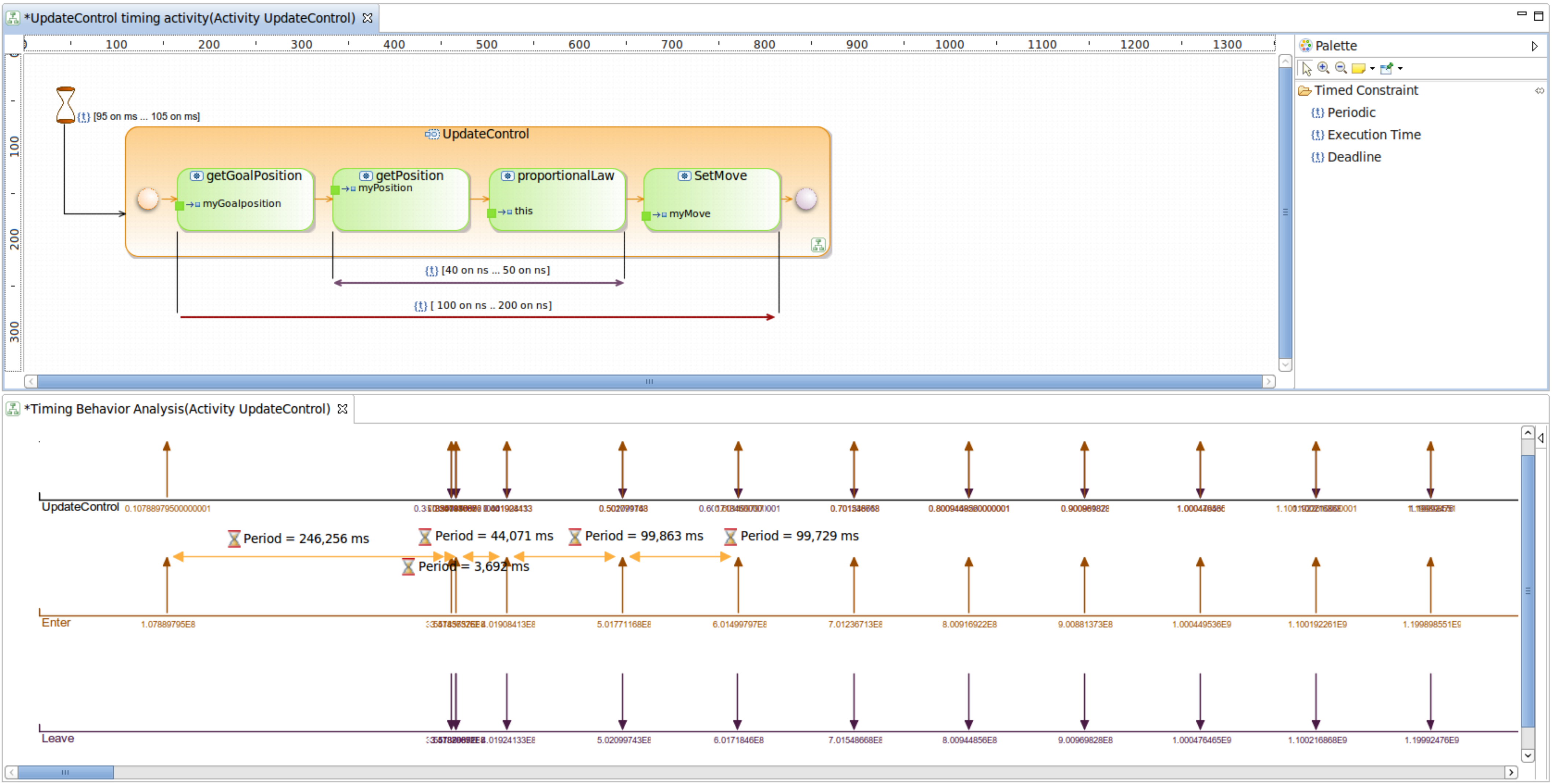
Réconciliation et analyse des différences (spécifié vs observé)

If simulated it replays the execution...
So you can have a timing diagram, animate the model and so on..

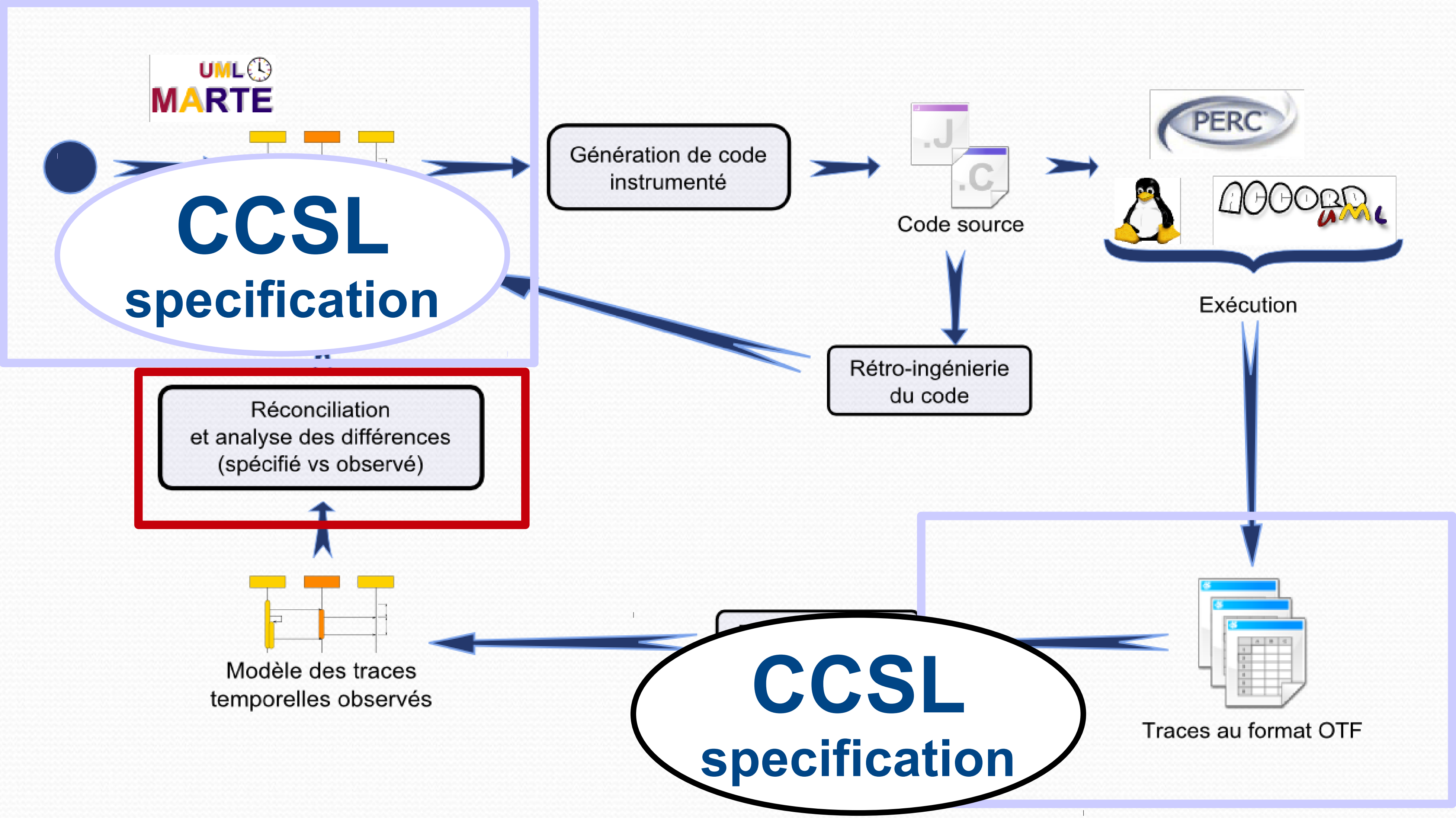
CCSL specification

Traces au format OTF

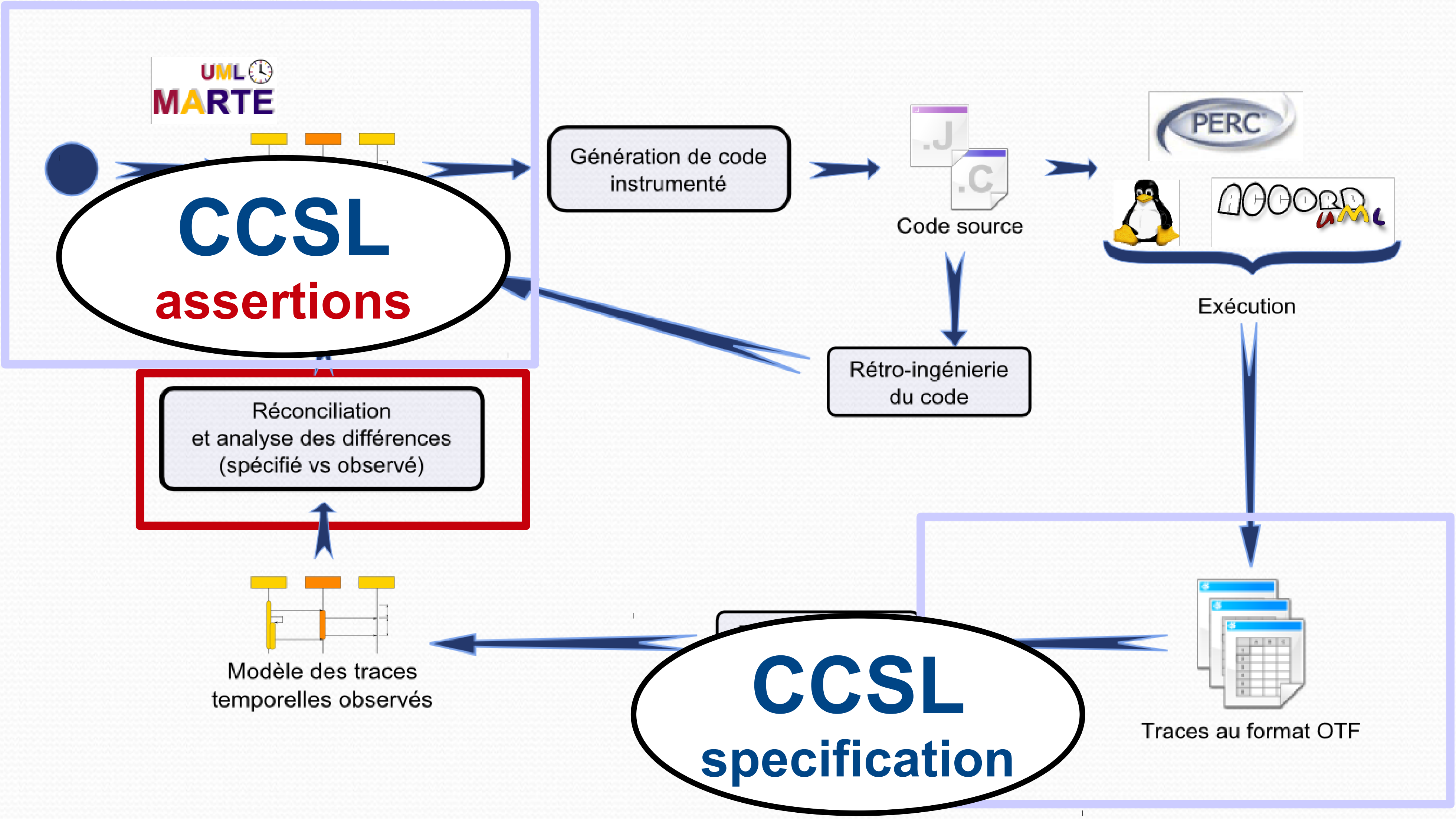
A specification and the actual execution



Processus RT-Simex


UML MARTE
CCSL specification
Génération de code instrumenté
.J
.C
Code source
PERC
ACCORD UML
Exécution
Rétro-ingénierie du code
Réconciliation et analyse des différences (spécifié vs observé)
Modèle des traces temporelles observés
CCSL specification
Traces au format OTF

Processus RT-Simex



CCSL assertions

Génération de code instrumenté

Code source

Exécution

Rétro-ingénierie du code

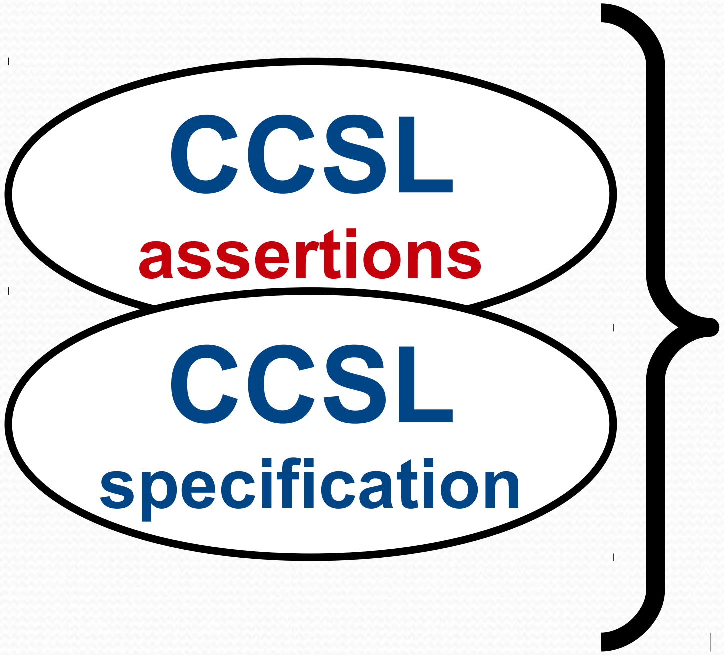
Réconciliation et analyse des différences (spécifié vs observé)

Modèle des traces temporelles observés

CCSL specification

Traces au format OTF

Processus RT-Simex



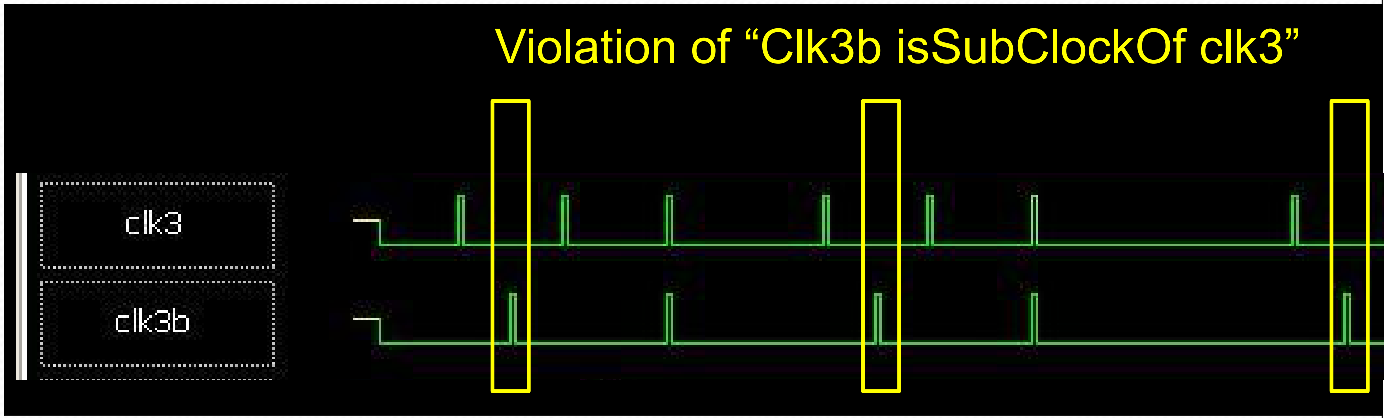
You can simulation both together to see if a violation occurs

Processus RT-Simex

CCSL
assertions

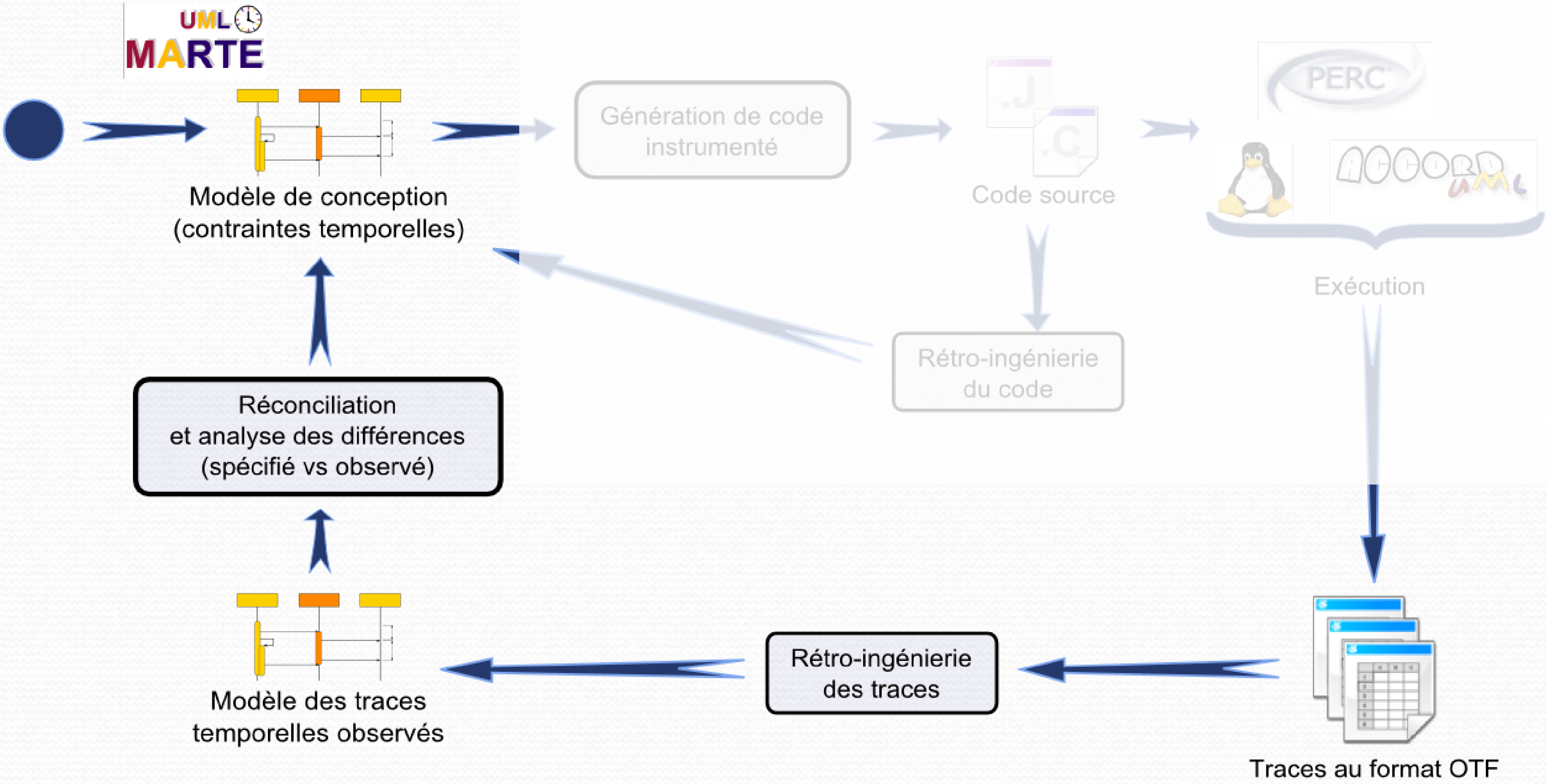
CCSL
specification

You can simulation both together to see if a violation occurs



Violation of “Clk3b isSubClockOf clk3”

Conclusion


UML MARTE
Modèle de conception
(contraintes temporelles)
Génération de code instrumenté
.J
.C
Code source
PERC
ACCORD UML
Exécution
Rétro-ingénierie du code
Réconciliation
et analyse des différences
(spécifié vs observé)
Modèle des traces
temporelles observés
Rétro-ingénierie
des traces
Traces au format OTF

Conclusion

- Logical Time, via Marte and CCSL is used for
  - The specification the expected behaviour
  - The simulation at the model level of this behaviour
  - The simulation at the model level of the actual behaviour (as monitored during the execution)
  - The comparison between the specified and the actual behaviour