Valentin Perrelle

Synchronous Programming of Tasks that can miss Deadlines

4 December 2014

Projet porté par
Campus Paris Saclay
Fondation de Cooperation Scientifique

Labellisation principale
SYSTEMATIC
Paris Region Systems & ICT Cluster

Labellisations secondaires
advancity
Ville & Mobilité Durables
ASTech
Paris Region
mov'eo

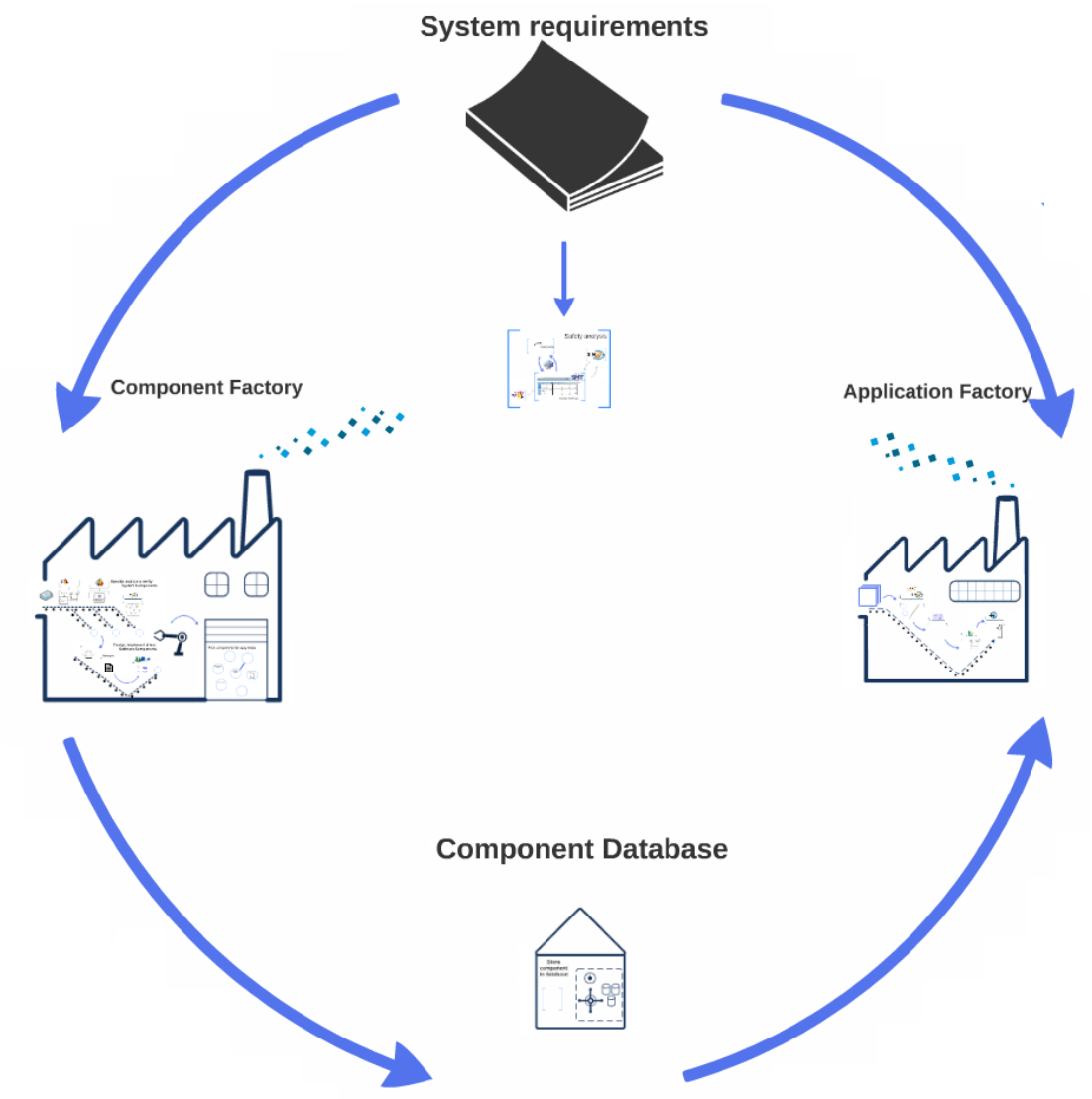
Soutien de collectivités territoriales
îledeFrance
Essonne
Le Conseil Général
CAPS

01 THE FSF PROJECT

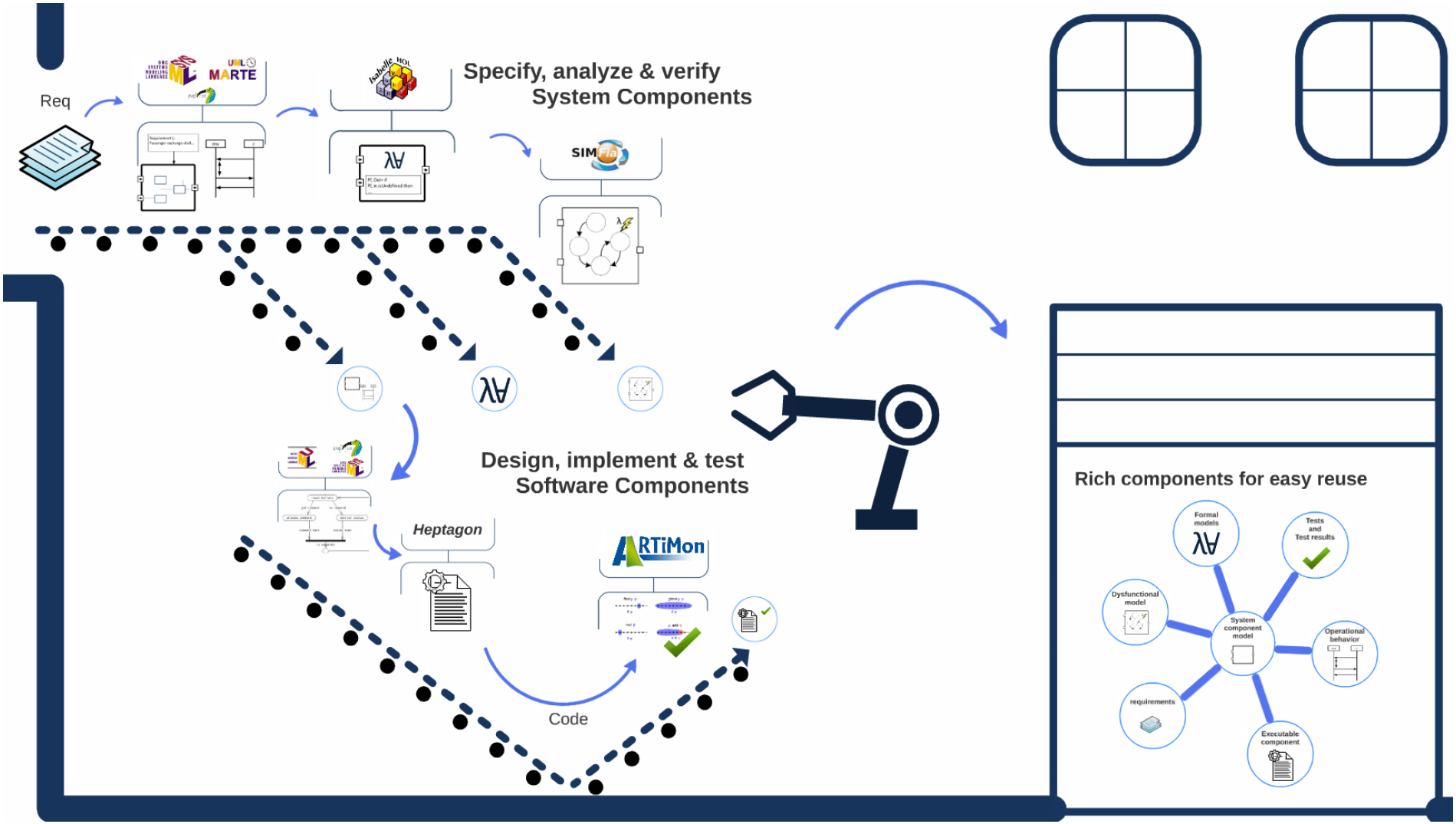
Projet FSF


ALSTOM
Inria
INVENTEURS DU MONDE NUMÉRIQUE
APSYS
LA MAITRISE DES RISQUES
KRONO-SAFE
MINES
TELECOM
INSTITUT
Mines-Télécom
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
ESTEREL
Technologies
Scaleo
chip
Trusted Labs
DE LA RECHERCHE À L'INDUSTRIE
cea
Partenaires
industriels
Partenaires
académiques

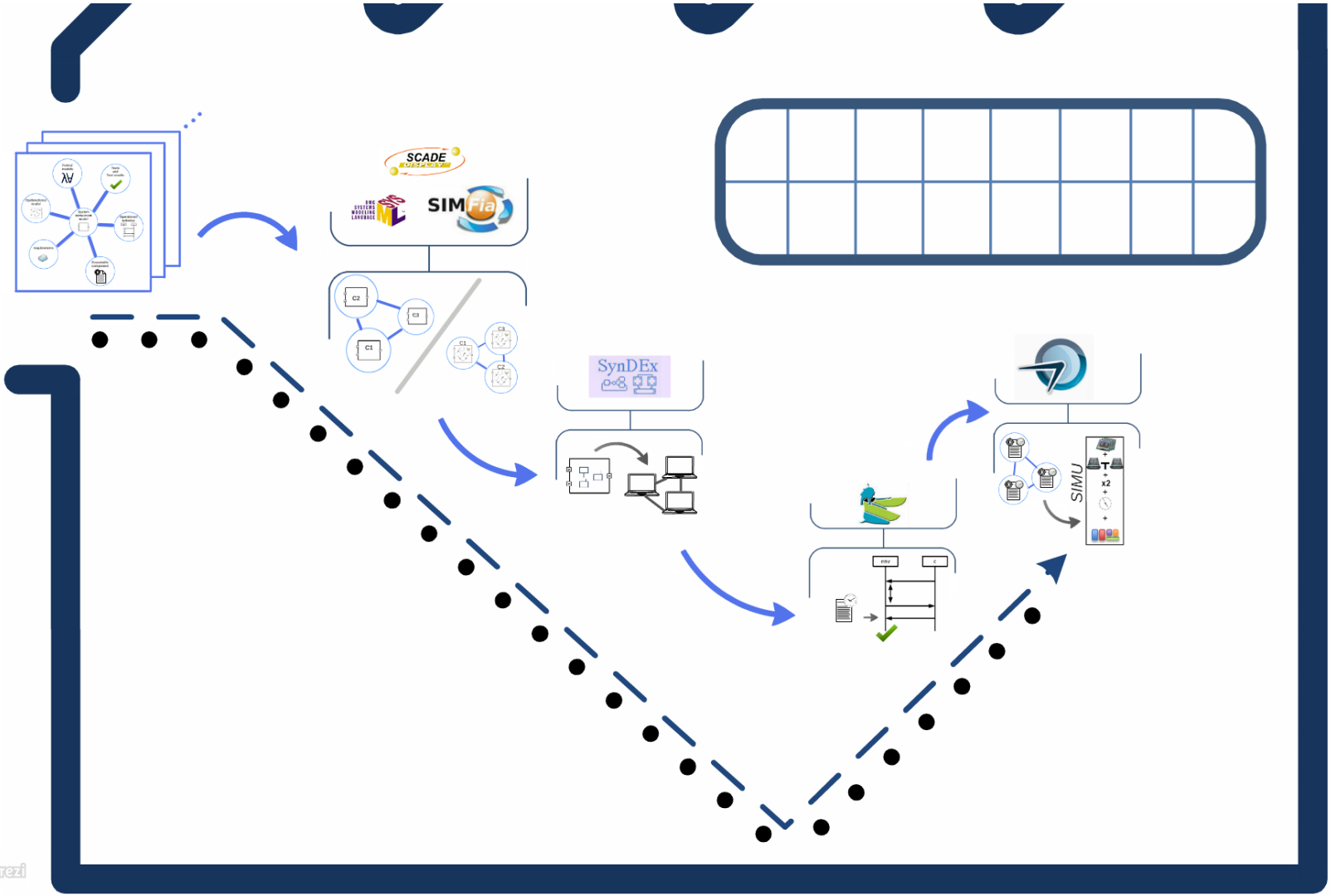
FSF Project Overview



The component factory



The application factory



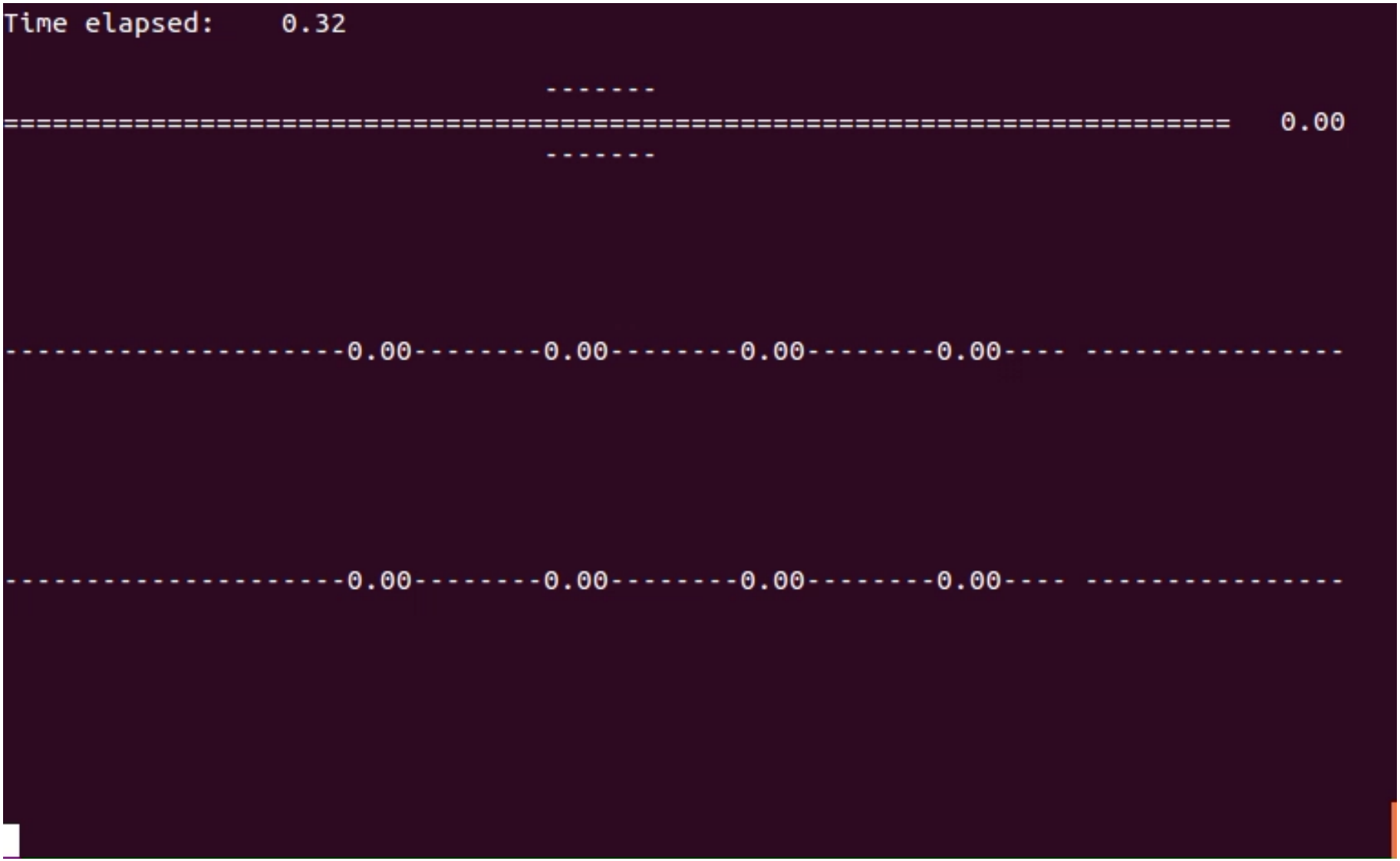
02 A CASE STUDY

- Passenger exchange
- Simulation
- Metrics
- Partitionning and scheduling [Zhang 2014]

PASSENGER EXCHANGE

- Mission
  - Issue commands to open or close doors according to a given mission
  - Issue announcements to inform the passenger of an imminent opening/closing
  - Send warnings to the traffic supervision when the passenger exchange cannot be completed
- Safety
  - If the train is not immobilized, the doors can't be opened
  - Only properly aligned doors can be opened
  - The train is not allowed to leave as long as all the doors are not closed

SIMULATION

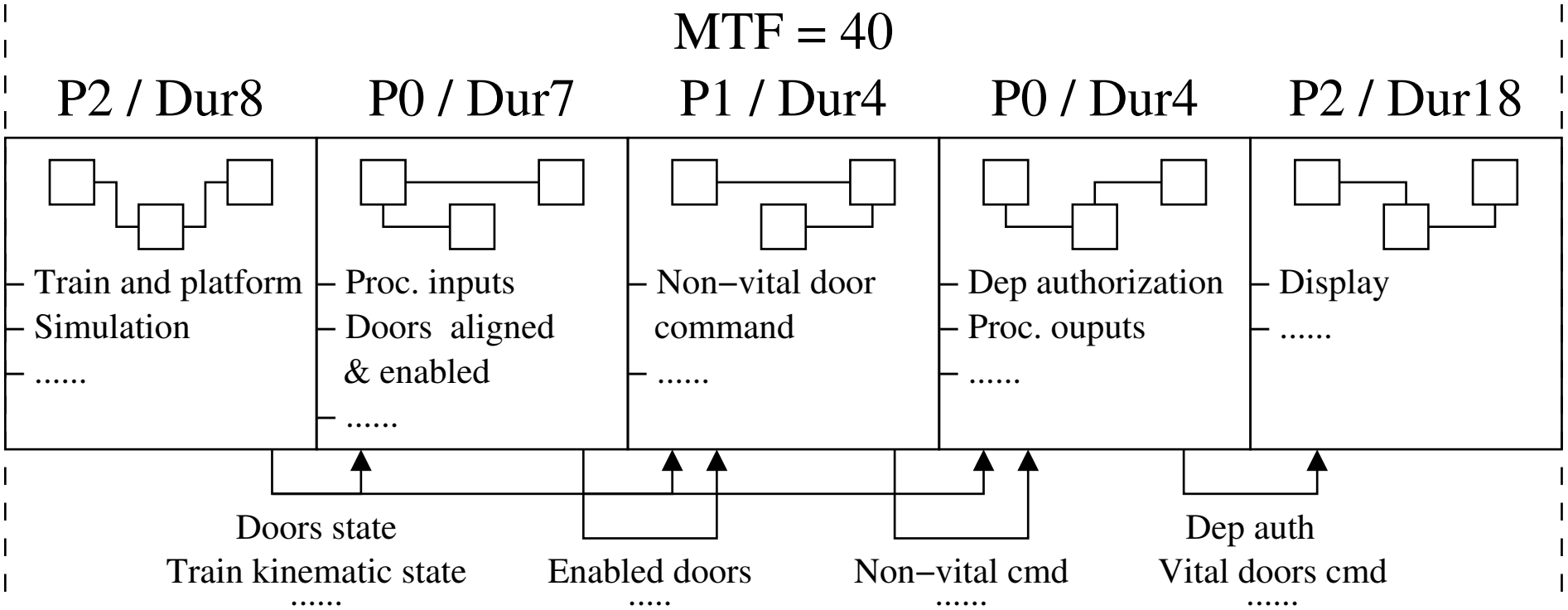


METRICS

Software specifications metrics	
Functions	≈ 30
Requirements	≥ 100

Code metrics	Files	LOC
Heptagon sources	27	2741
C generated from Heptagon	70	7014
Additionnal C code	11	611

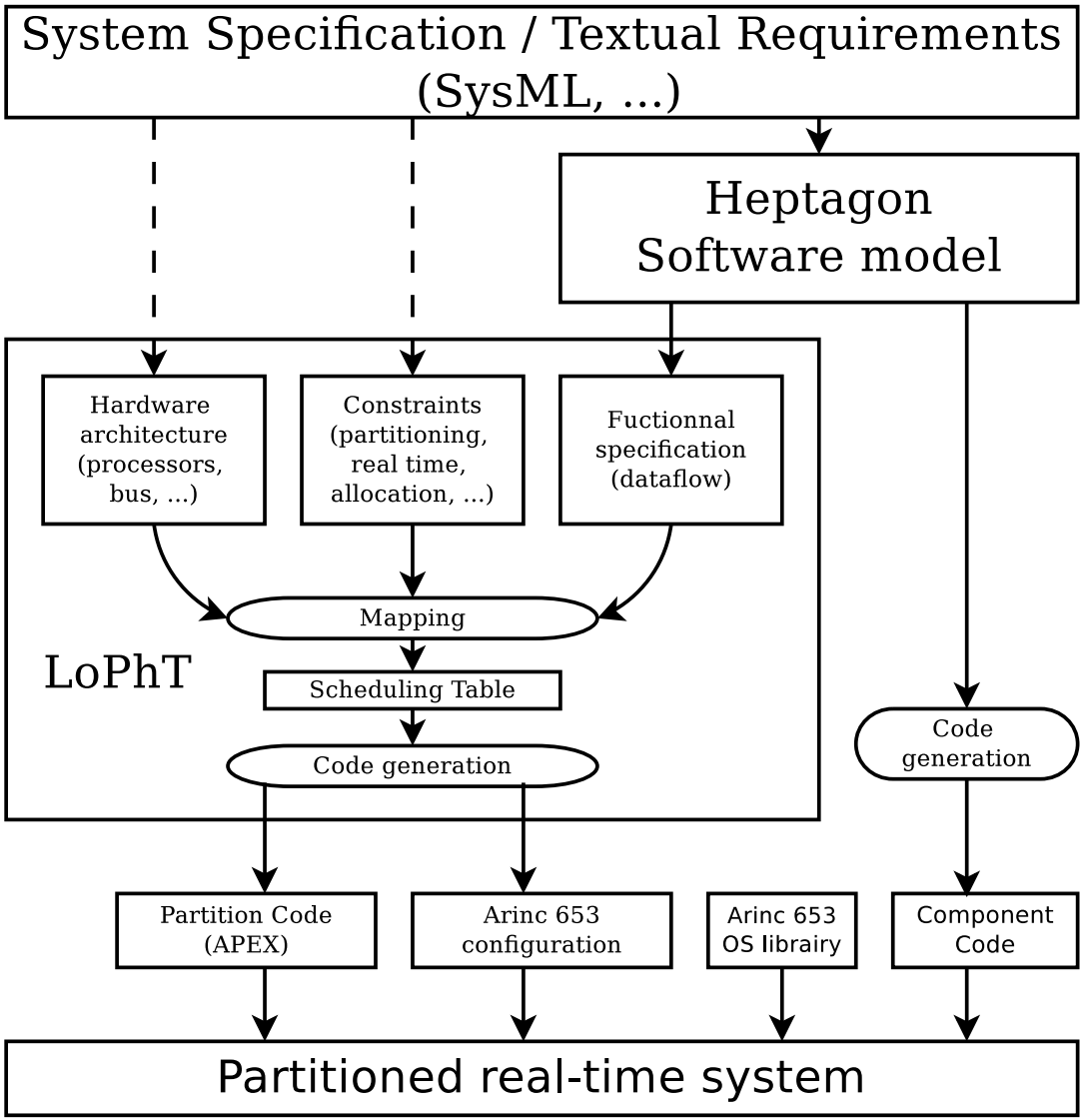
PARTITIONNING AND SCHEDULING [ZHANG 2014]



03 DESCRIBING FUNCTIONAL DEPENDENCIES

- Tool chain
- Some advantages of LoPhT and Heptagon
- Writting clocked graphs in Heptagon
- Clocked Graphs
- Heptagon architecture
- Clock translation 1/2
- Clock translation 2/2

Tool chain


System Specification / Textual Requirements
(SysML, ...)
Heptagon
Software model
Hardware architecture (processors, bus, ...)
Constraints (partitioning, real time, allocation, ...)
Fuctionnal specification (dataflow)
Mapping
LoPhT
Scheduling Table
Code generation
Code generation
Partition Code (APEX)
Arinc 653 configuration
Arinc 653 OS librairy
Component Code
Partitioned real-time system

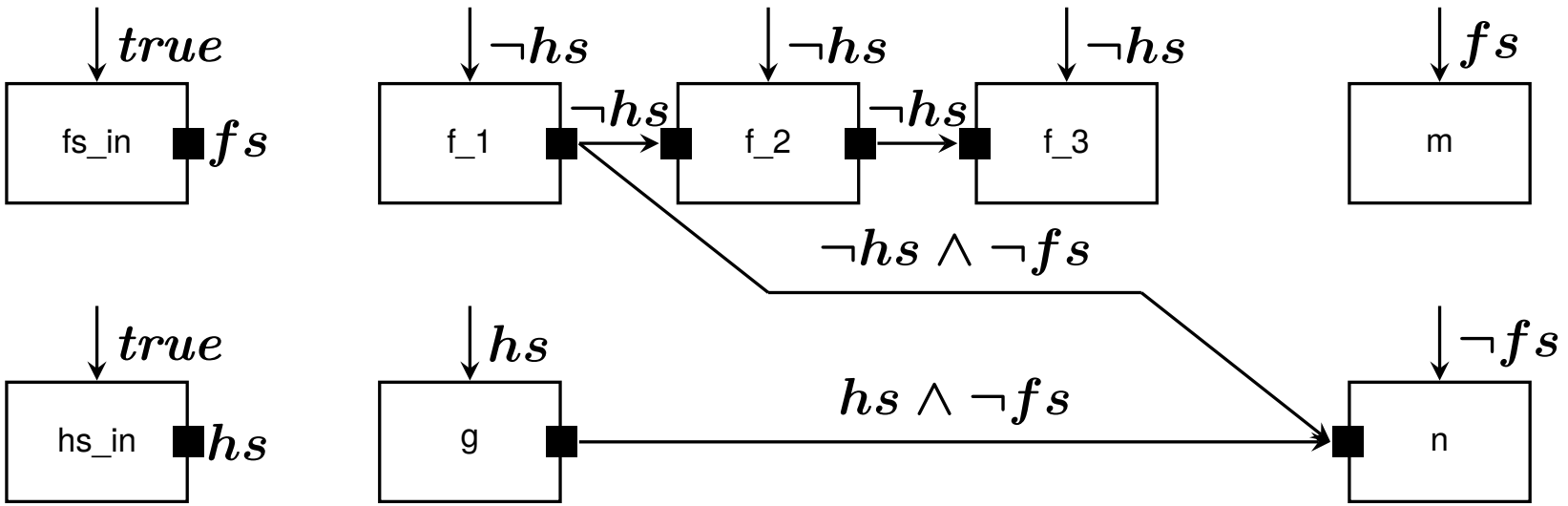
Some advantages of LoPhT and Heptagon

- Passenger Exchange software specifications are written in a equational and synchronous style
- Almost direct translation from Heptagon to Clocked Graphs
- Easy to implement experimental features in Heptagon
- Mutual exclusion in LoPhT

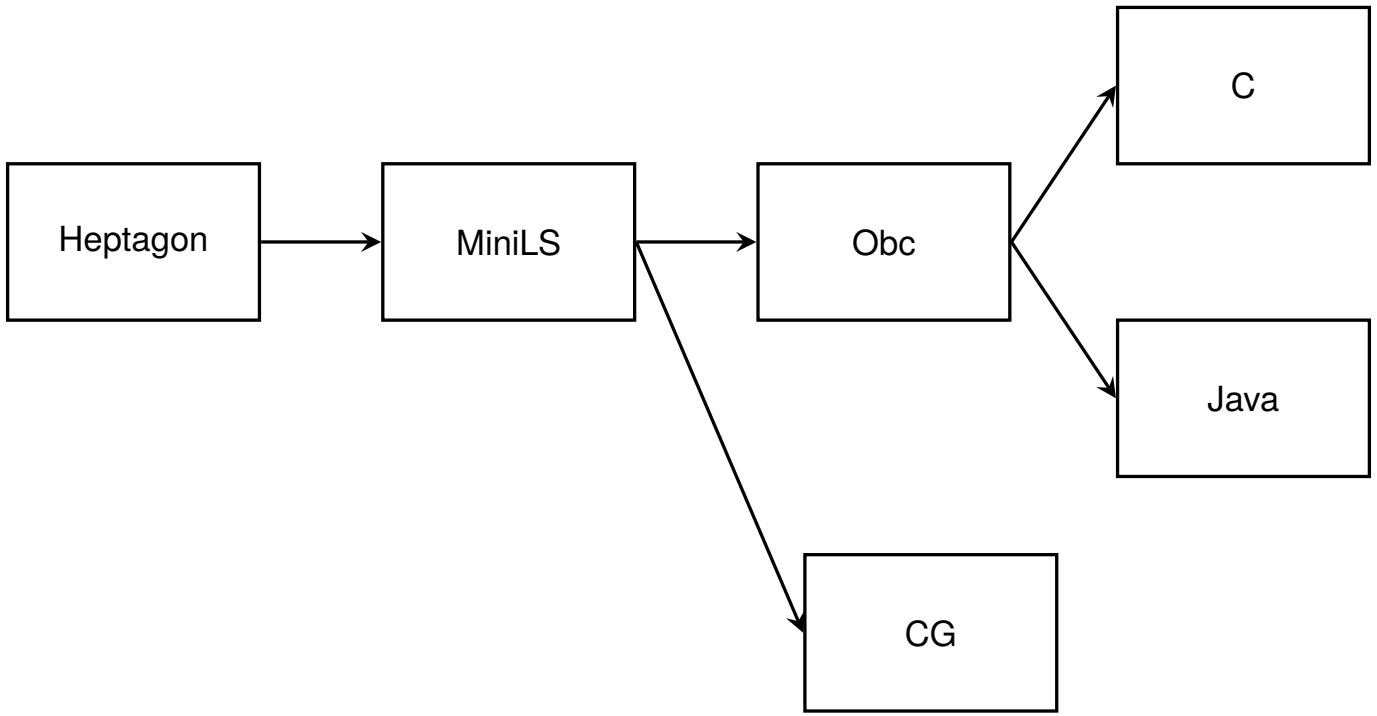
Writting clocked graphs in Heptagon

```
node fdc(hs : bool) returns (id : int)
let
  if hs then
      id = g();
  else
    var v : int; in
      id = f1();
      v = f2(id);
      () = f3(v);
  end
tel

node correction(fs : bool; id : int) returns ()
let
  if fs then
    () = n();
  else
    () = m(id);
  end
tel

node main() returns ()
var
  id : int;
  fs, hs : bool;
let
  fs = fs_in();
  hs = hs_in();
  id = fdc(hs);
  () = correction(fs, id);
tel
```

[Potop-Butucaru et al. 2009]

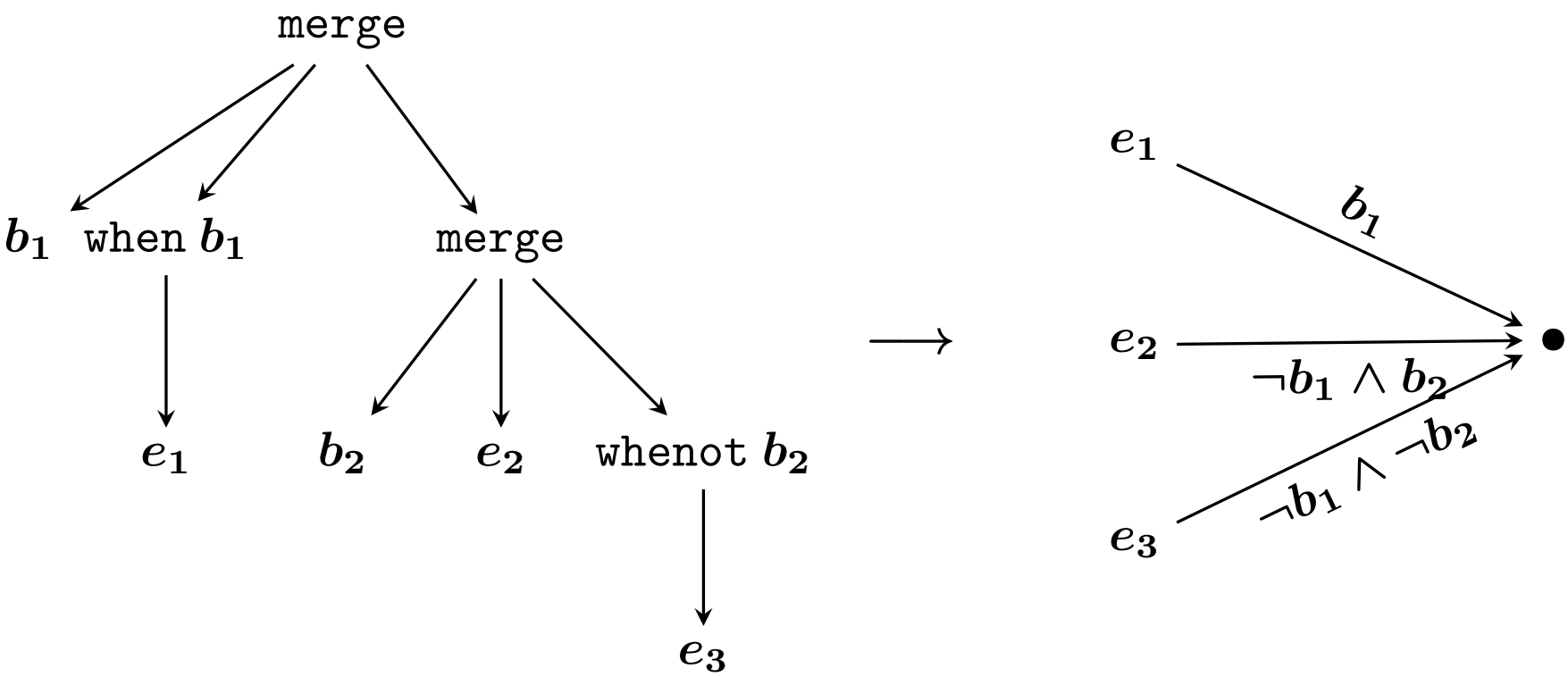


Heptagon architecture



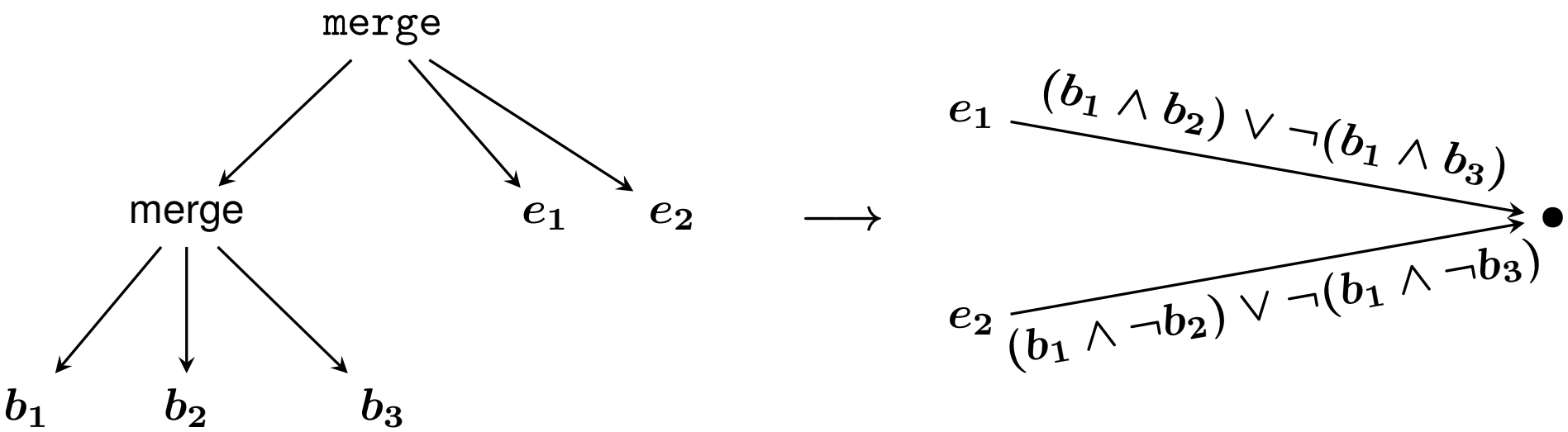
Clock translation 1/2

```
merge b_1 (e_1 when b_1) (merge b_2 e_2 e_3)
```



CLOCK TRANSLATION 2/2

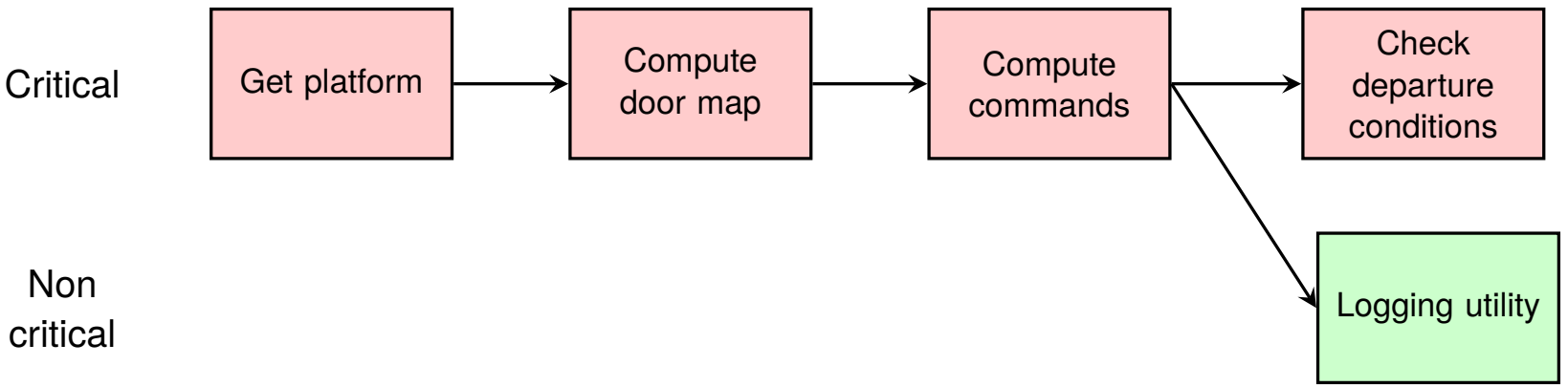
```
merge (merge b_1 b_2 b_3) e_1 e_2
```



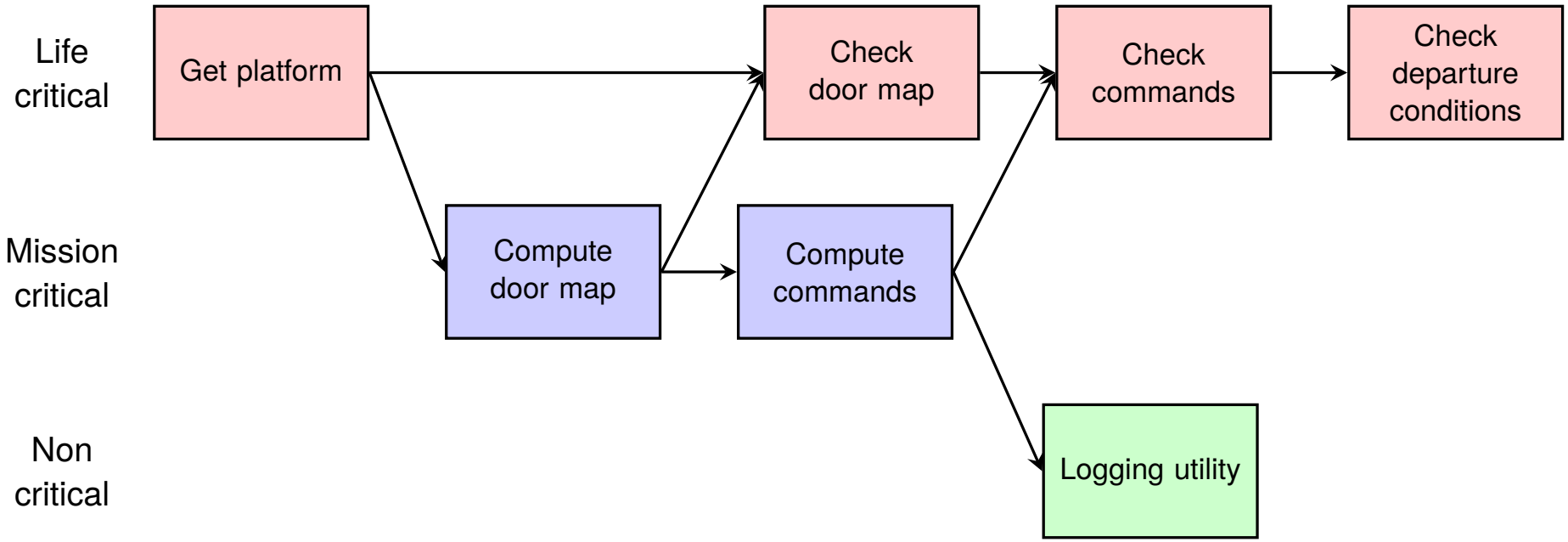
04 TASKS MISSING DEADLINES

- Mixed criticality
- The Idea
- Example
- Using clocks
- Ideas and Future work

Mixed criticality



Mixed criticality



The Idea

- Non time-critical tasks must not delay time-critical ones
- Allow non time-critical tasks to miss deadlines to reduce certification cost
- Handle data absence programmatically
- Safety vs Disponibility

EXAMPLE

```
task check_commands(unpunctual door_commands : command^n; door_map : int^n)
     returns (safe_commands : command^n)
let
   if ontime door_commands then
     safe_commands = map<<n>> check_command(door_commands, door_map);
   else
     safe_commands = None^n;
   end
tel

node check_command(door_command : command; door_map : int)
     returns (safe_command : command)
let
   safe_command = if door_map <> -1 then door_command else None;
tel
```

EXAMPLE

```
task check_commands(unpunctual door_commands : command^n = None^n;
                    door_map : int^n)
    returns (safe_commands : command^n)
let
  safe_commands = map<<n>> check_command(door_commands, door_map);
tel
```

Example

```
task check_commands(door_commands : command^n; door_map : int^n)
    returns (safe_commands : command^n)

task check_commands_degenerated(door_map : int^n)
    returns (safe_commands : command^n)

unpunctual task compute_commands(door_map : int^n)
    returns (commands : command^n)
```

EXAMPLE

```
node passenger_exchange() returns ()
var
  door_map : int^n;
  unpunctual door_commands : command^n;
  ...
let
  ...
  door_commands = compute_commands(door_map);
  if ontime door_commands then
    check_commands(door_commands, door_map);
  else
    check_commands_degenerated(door_map);
  end
  ...
tel
```

USING CLOCKS

```
node check_commands(
  door_commands_clock : bool;
  door_commands_value : command^n :: door_commands_clock;
  door_map : int^n)
    returns (safe_commands : command^n)
let
  if door_commands_clock then
    safe_commands = map<<n>> check_command(door_commands_value, door_map);
  else
    safe_commands = None^n;
  end
tel
```

USING CLOCKS

```
node check_command(
  door_commands_clock : bool;
  door_commands_value : command^n :: door_commands_clock;
  door_map : int^n)
    returns (safe_commands : command^n)
var
  door_commands : command^n;
let
  door_commands = merge door_commands_clock
    door_commands_value
    (None^n whenot door_commands_clock);
  safe_commands = map<<n>> check_command(door_commands, door_map);
tel
```

- Allow task killing
- Using futures [Gérard et al. 2012] in functionnal dependencies
- Relax synchronous hypothesis and use losseless buffering [Yip, Kuo, Roop, Broman 2014]
- Allow partial output

Valentin Perrelle

Synchronous Programming of Tasks that can miss Deadlines

4 December 2014

Projet porté par
Campus Paris Saclay
Fondation de Cooperation Scientifique

Labellisation principale
SYSTEMATIC
Paris Region Systems & ICT Cluster

Labellisations secondaires
advancity
Ville & Mobilité Durables
ASTech
Paris Region
mov'eo

Soutien de collectivités territoriales
îledeFrance
Essonne
Le Conseil Général
CAPS