A comparative study of FIACRE and TASM to define AADL real time concepts

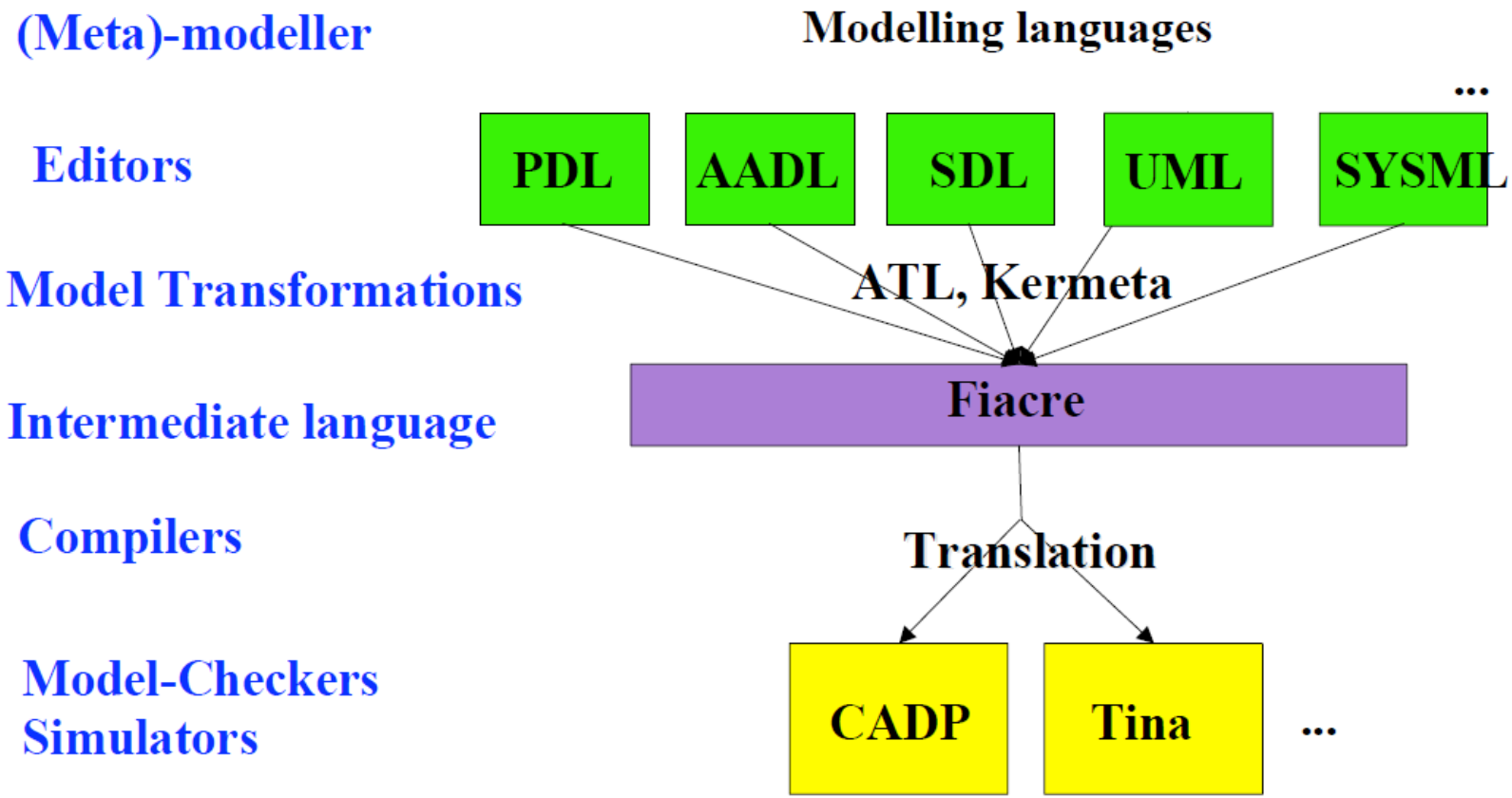
Lei Pi[1], **Jean-Paul Bodeveix**[1], Mamoun Filali[1] and Zhibin Yang[2], Kai Hu[2], Dianfu Ma[2]

[1]Institut de Recherche en Informatique de Toulouse(IRIT), France
[2]BeiJing University of Aeronautics & Astronautics, China

UML&AADL'09
02/06/2009

Outline

TOPCASED



ACADIE

Motivation

- Formal expression of Architecture Analysis & Design Language (AADL) semantics
- Analysis of AADL models
  - end-to-end flow latency
  - schedulability
  - buffer overflows
  - users properties

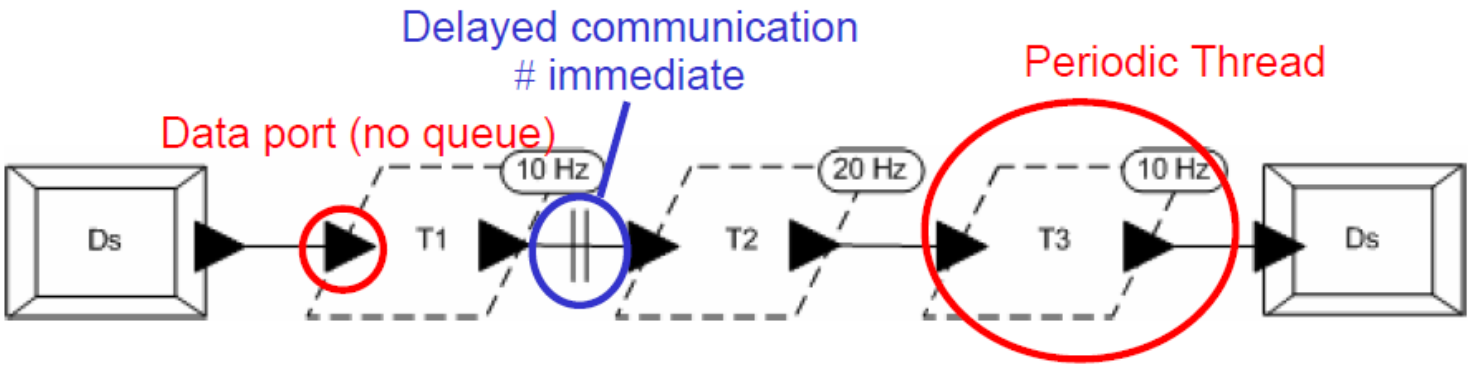
ACADIE

Outline

AADL

- Hierarchical definition of hardware / software systems
- Real-time models
- communicating threads : port based or shared memory based communications
  - Periodic or sporadic threads
  - Immediate and delayed communications
  - Execution time of tasks (WCET)

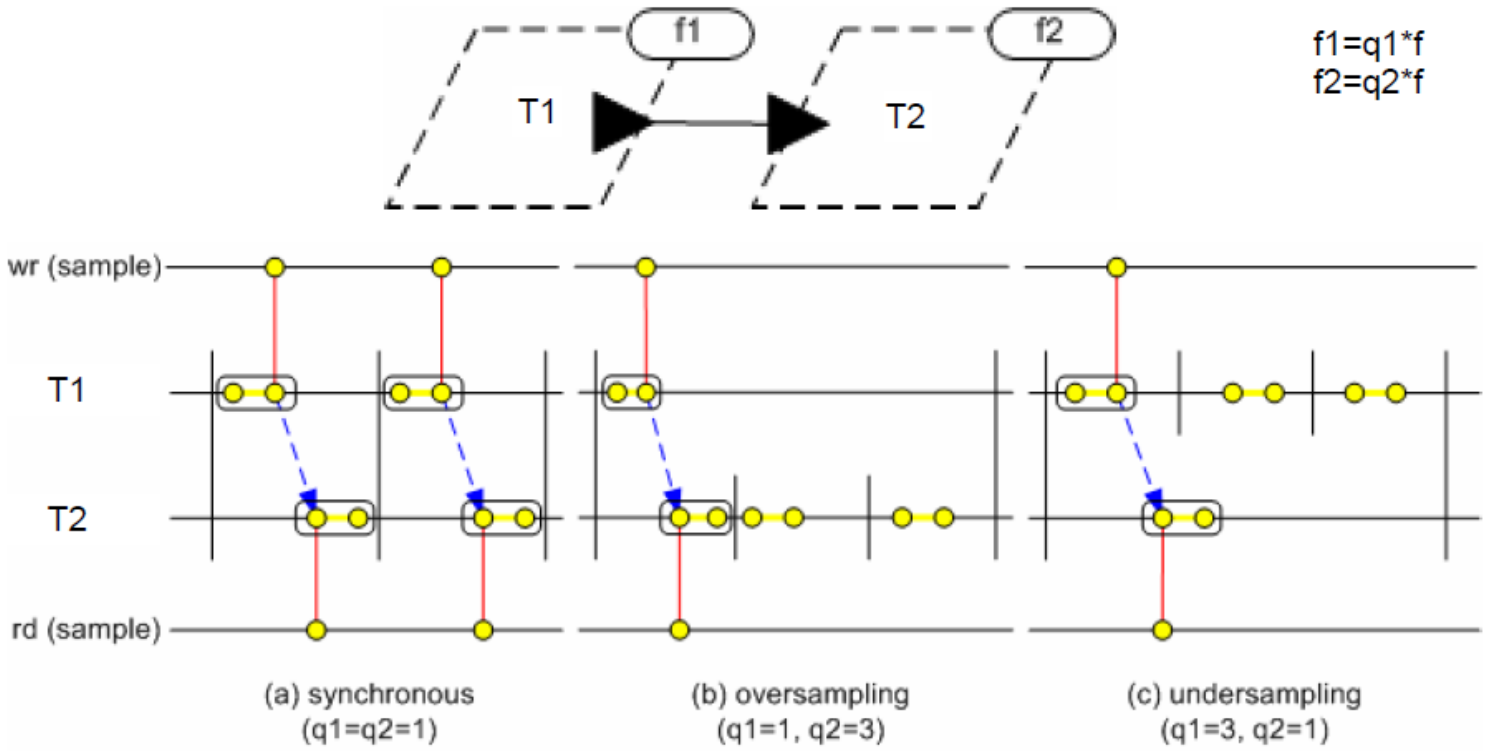
Synchronous subset of AADL



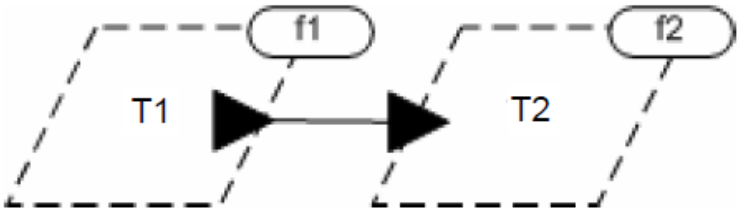
- Periodic threads
- data port communication
- immediate and delayed communication

Immediate Communications

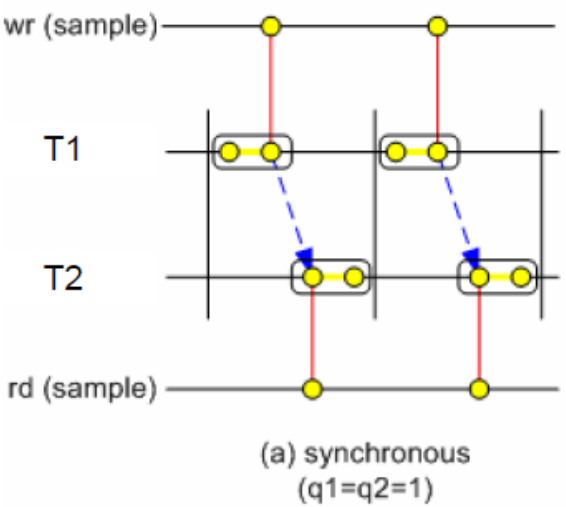
AADL implementation of instantaneous communications of the synchronous model



Immediate Communications

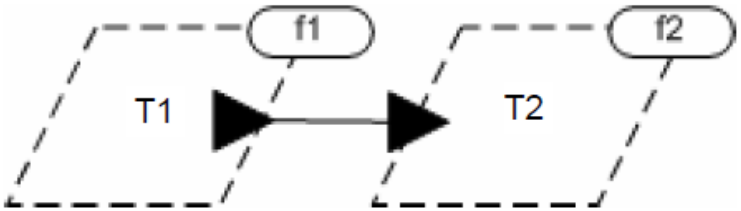


f1=q1*f
f2=q2*f

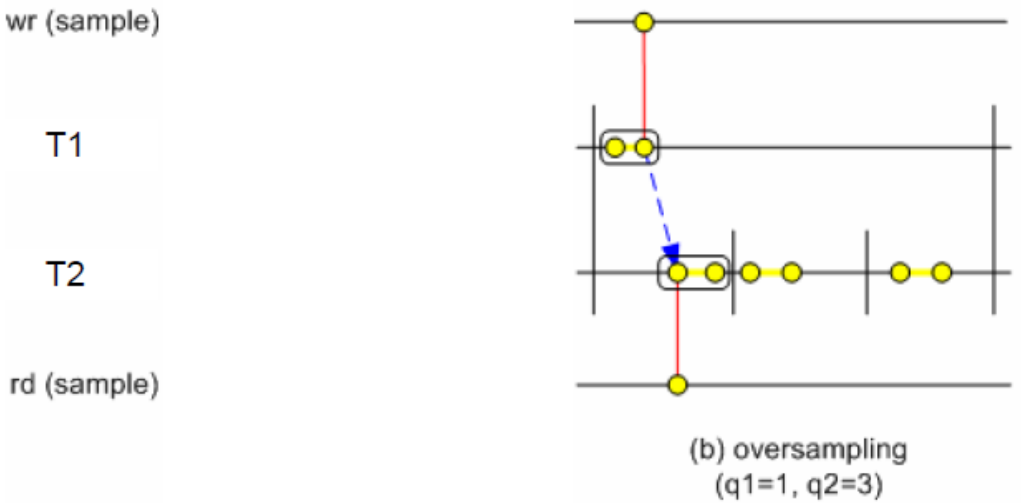


(a) synchronous
(q1=q2=1)

Immediate Communications

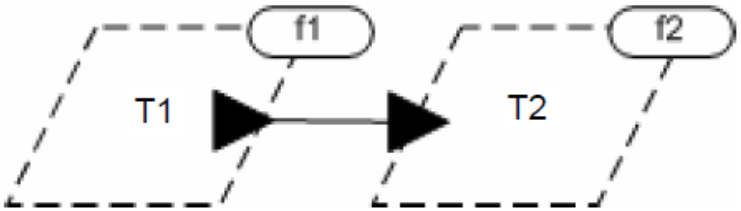


f1=q1*f
f2=q2*f



(b) oversampling
(q1=1, q2=3)

Immediate Communications

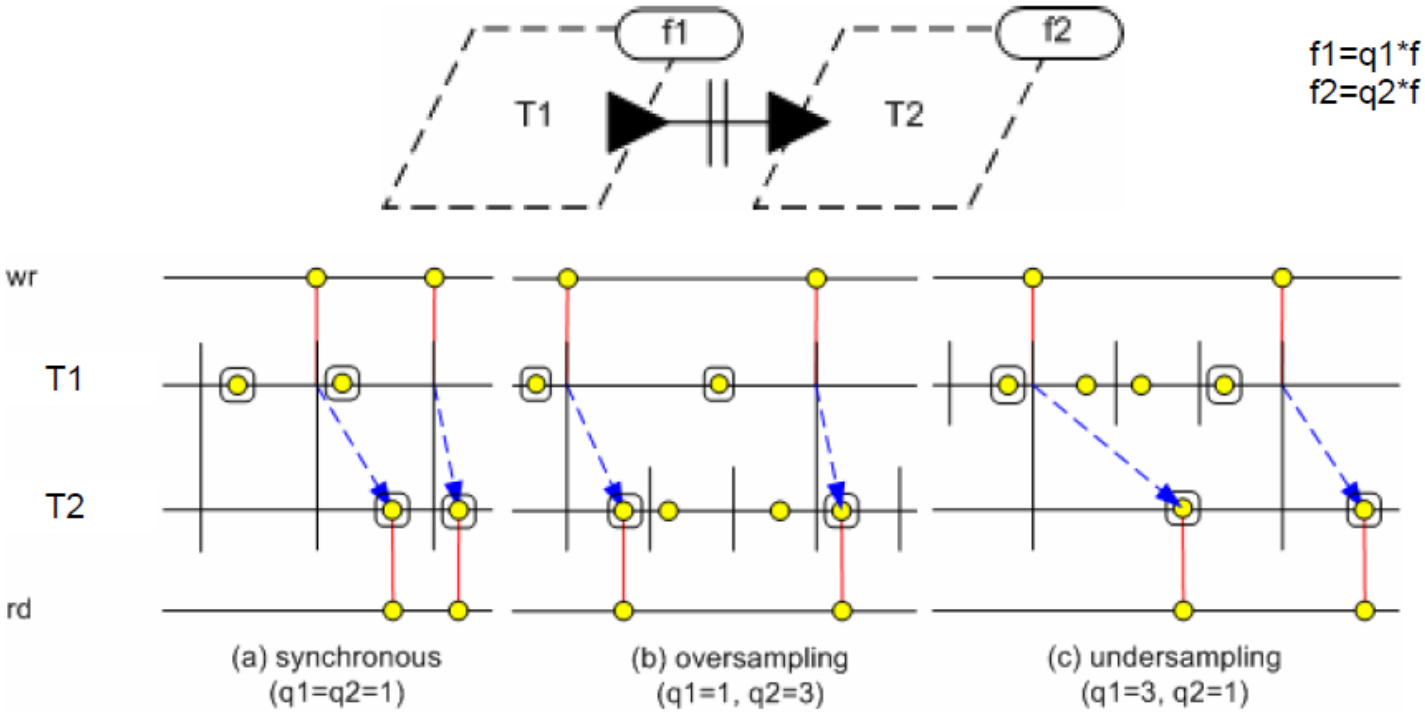


f1=q1*f
f2=q2*f

(c) undersampling
(q1=3, q2=1)

Delayed Communications

AADL implementation of the delay operator of the synchronous model



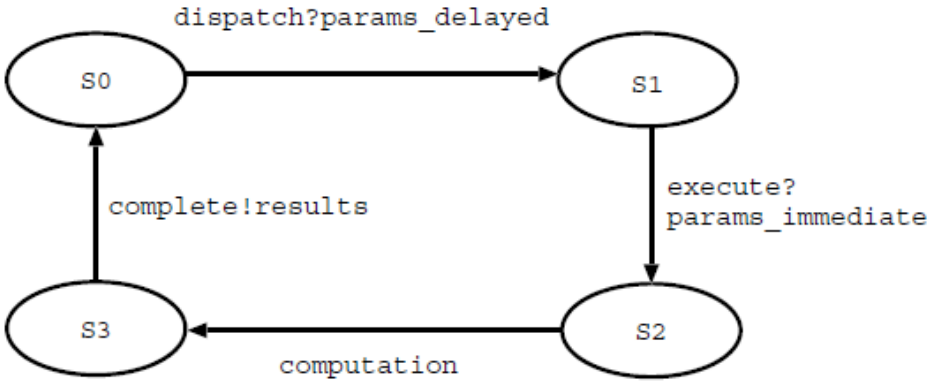
Outline

Comparison

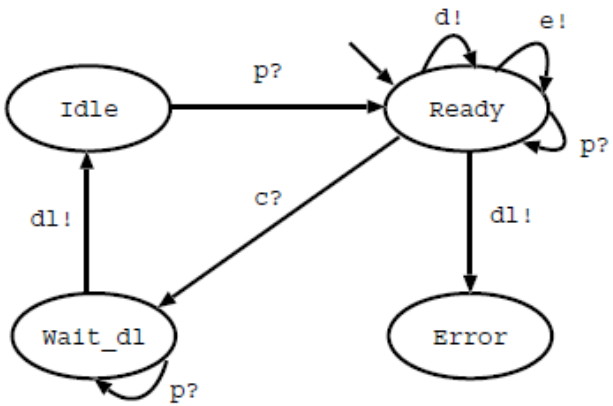
	Communication	**Synchronization model**	**Time model**	**Resource management**
FIACRE	port/shared data	**C**ommunicating **S**equential **P**rocesses	**T**imed **T**ransition **S**ystems implicit	no
TASM	shared data	**C**alculus **C**ommunication **S**ystems	continuous implicit	yes

Outline

Modeling periodic thread execution in FIACRE



Thread automaton



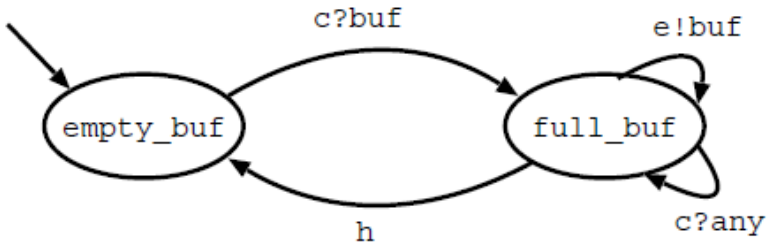
Scheduler automaton

```
process P1 [dispatch1 : in none,
            execute1: in none,
            computation1: in none,
            completion1: out int]is
 states s0, s1, s2, s3
 var x: data
 init to s0
 from s0  dispatch1; to s1
 from s1  execute1; to s2
 from s2  computation1; to s3
 from s3  completion1!x; to s0
```

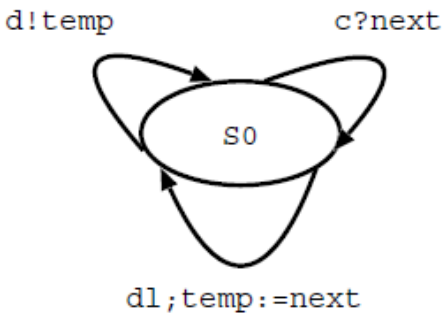
```
process P2 [dispatch2 : in int,
            execute2: in dataOption,
            computation2 :in none,
            completion2 : out none]is
 states s0, s1, s2, s3
 var x: data, o: dataOption
 init to s0
 from s0  dispatch2?x; to s1
 from s1
   select
      execute2?None; to s2
   [] execute2?Some x; to s2
   end
 from s2  computation2; to s3
 from s3  completion2; to s0
```

IRIT ADIE

Modeling communication semantics in FIACRE



Immediate communication



Delayed communication

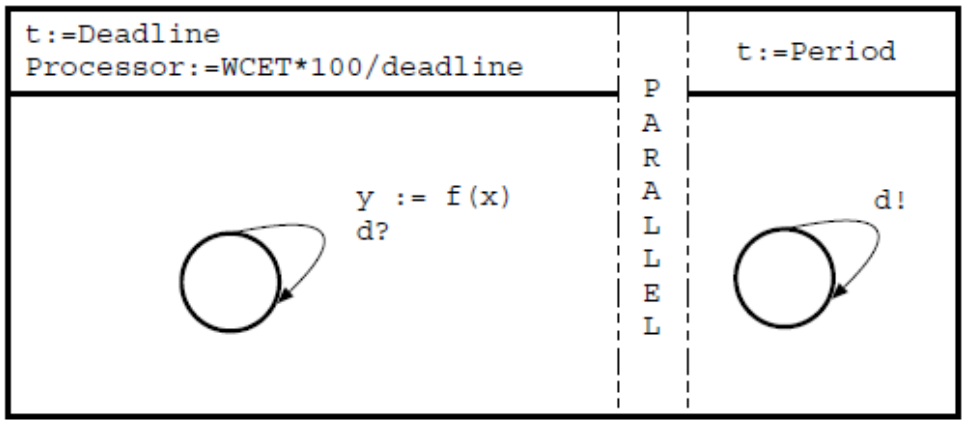
```
process Buf_immediate [e : out dataOption,
                       c : in int, h : none]is
 states empty_buf, full_buf, wait
 var buf : int := 0
 init to empty_buf
 from empty_buf
    select  c?buf; to full_buf
    end
 from full_buf
    select  c?any; to full_buf
         []  e!buf; to full_buf
         []  h; to empty_buf
    end

process Buf_delayed [d : out int,
         c : in int, dl : out none]is
  states s0
  var temp : int := 0,
      next : int := 0
  init to s0
  from s0
    select  c?next; to s0
       []   d!temp; to s0
       []   dl; temp := next; to s0
    end
```

Modeling communication semantics in FIACRE

```
component sys is
 port  t1 : none in [2,2], t2 : none in [3,3], t3 : none in [4,4], tmp : none in [6,6],
       e1 : none in [0,0], e2 : dataOption in [0,0], e3 : dataOption in [0,0],
       w1 : none in [0,1], w2 : none in [0,1], w3 : none in [0,1],
       d1 : none in [0,0], d2 : int in [0,0], d3 : int in [0,0],
       c1 : int in [0,0], c2 : none in [0,0], c3 : none in [0,0],
       dl1 : none in [2,2], dl2 : none in [3,3], dl3 : none in [4,4]
 priority
         c1>t1, c2>t2, c3>t3,
         t1|t2|t3>d1|d2|d3,
         d1>d2, d1>d3,
         d1|d2|d3>e1|e2|e3,
         e1>e2
  par
      P1[d1, e1, w1, c1] || P2[d2, e2, w2, c2] || P2 [d3, e3, w3, c3]
 ||   Buf_immediate[e2, c1, tmp]
 ||   Buf_delayed[d3, c1, dl1]
 ||   scheduler[t1, t2, t3, c1, c2, c3, d1, d2, d3, e1, e2, e3, dl1, dl2, dl3]
 end
```

Outline

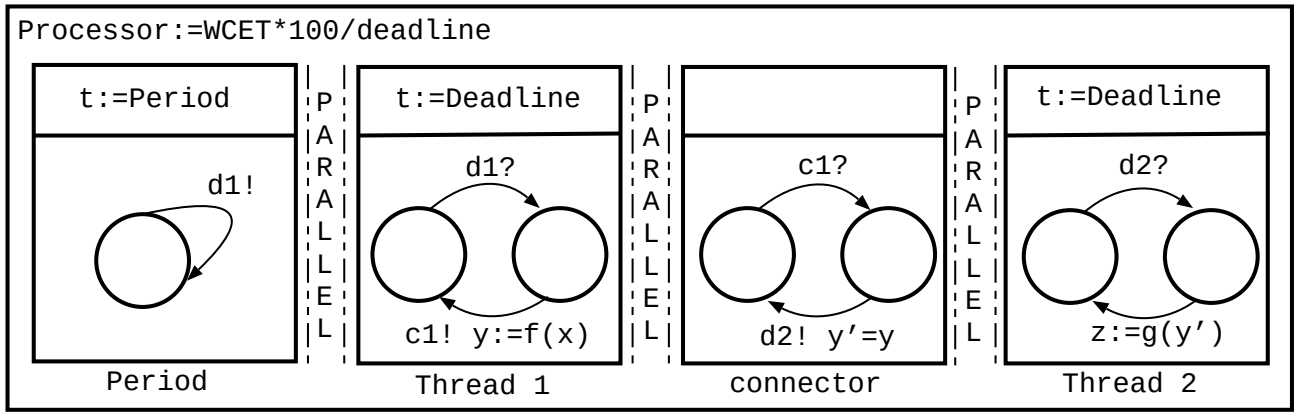
Modeling periodic thread execution in TASM



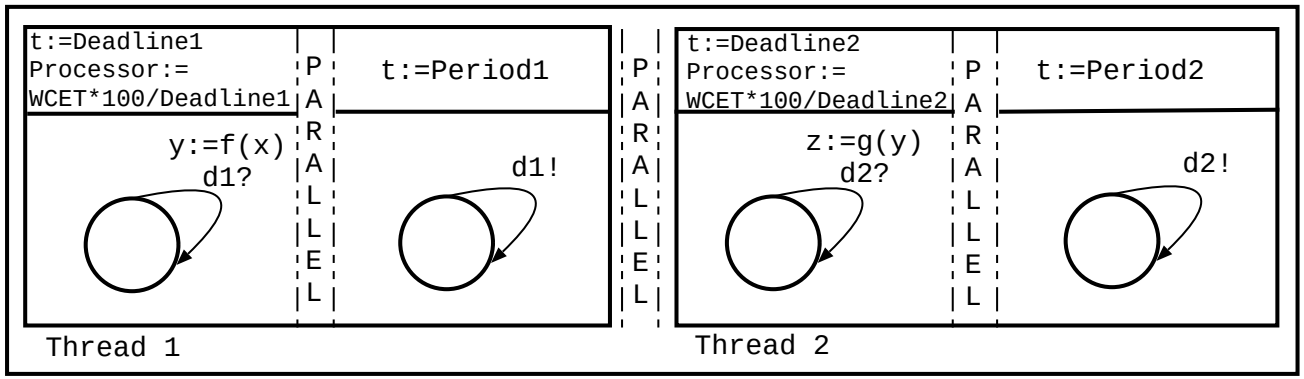
Periodic Thread

```
MAIN MACHINE: Period
Rule: period
{   t := period;
    if Nextdispatch = false then
       Nextdispatch := true;
       d!;
}
MAIN MACHINE: Execution
Rule: execution
{   t := [0,deadline];
    processor := WCET*100 / deadline;
    if Complete = false then
       Complete := true;
       d?;
}
```

Modeling communication semantics in TASM



Immediate communication



Delayed communication

Conclusion

- Schemas for expressing some real-time concepts in FIACRE and TASM.
  - AADL periodic threads.
  - AADL delayed and immediate communications
- FIACRE powerful for real time - verification through timed Petri nets.
- TASM powerful abstraction for resource management (including time) - verification through priced timed automata.

⇝ better understanding of AADL.

⇝ ideas for defining FIACRE extensions.