UNIVERSIDAD DE CANTABRIA

MAST: a model and tools to predict response times in event-driven real-time systems

Tutor'2016

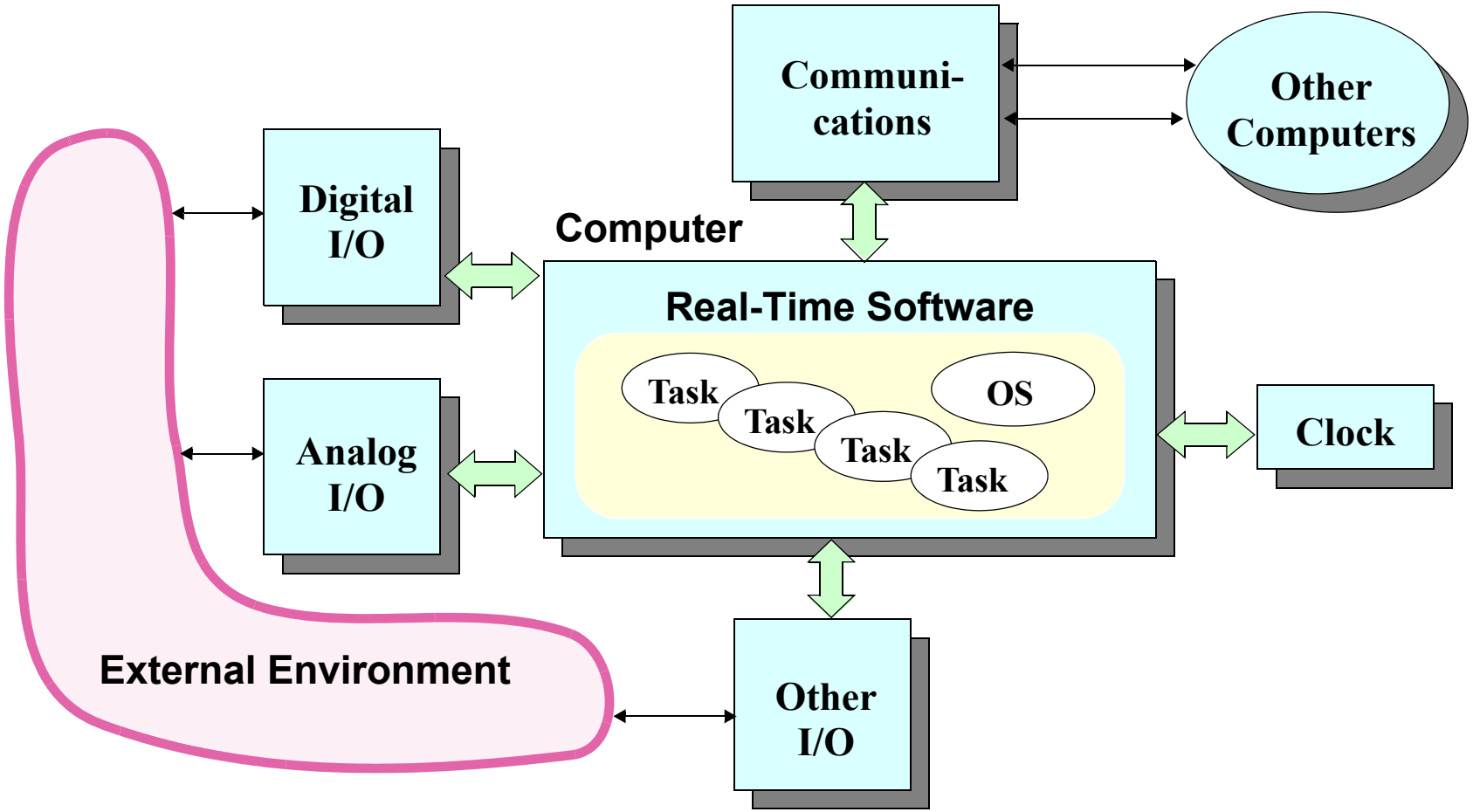
April 2016, Vienna, Austria

Michael González Harbour & Julio Medina
mast@unican.es
www.istr.unican.es

MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

Elements of a real-time system



Real-time systems

UNIVERSIDAD DE CANTABRIA

A Real-time system is a combination of a computer, hardware I/O devices, and special-purpose software, in which:

- there is a strong interaction with the environment
- the environment changes with time
- the system simultaneously controls and/or reacts to different aspects of the environment

As a result:

- timing requirements are imposed on software
- software is naturally concurrent

To ensure that timing requirements are met, the system's timing behavior must be *predictable*

What's important in real-time

UNIVERSIDAD DE CANTABRIA

Predictability **of the response time**

Criteria **for real-time systems differ from that for time-sharing systems.**

	Time-Share Systems	Real-Time Systems
Capacity	High throughput	Ability to meet timing requirements: Schedulability
Responsiveness	Fast average response	Ensured worst-case latency bound
Overload	Fairness	Stability of critical part

Worst case ***cannot*** **be checked by** ***testing***

MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. *The search for predictability: Options for Managing Time*
3. Modeling and Analyzing real-time systems: tasking models
4. The MAST real-time model and tools environment.
5. Elements of the MAST model.
6. Modeling and integration into the design process
7. Current developments and future work in MAST
8. Conclusions

Options for managing time

UNIVERSIDAD DE CANTABRIA

Compile-time schedules:

- time triggered or cyclic executives
- predictability through static schedule
- logical integrity often compromised by timing structure
- difficult to handle aperiodic events & dynamic changes
- difficult to maintain

Run-time schedules:

- priority-based schedulers
- preemptive or non preemptive
- analytical methods needed for predictability
- separates logical structure from timing
- more flexibility

Fixed-priority scheduling (FPS)

UNIVERSIDAD DE CANTABRIA

Fixed-priority preemptive scheduling is very popular for practical applications, because:

- Timing behavior is simpler to understand
- Behavior under transient overload is easy to manage
- A complete analytical technique exists
- High utilization levels may be achieved (typically 70% to 95% of CPU)
- Supported in standard concurrent languages or operating systems:
  - Ada's RT-annex, Java RTSJ
  - Real-time POSIX

Dynamic priority scheduling

UNIVERSIDAD DE CANTABRIA

In dynamic priority scheduling it is possible to make better use of the available resources

We can find two kinds of dynamic priority policies

- *static job priority*: a static priority is assigned to each task job
  - e.g., *EDF* (earliest deadline first): each job is assigned a priority equal to its absolute deadline
  - the absolute deadline of a job does not change
- *dynamic job priority*: the priority of a task job may change before the job is finished
  - e.g., LLF (Least Laxity First): the priority of a job is inverse to the laxity: time left until the end of the job's absolute deadline, minus the remaining computation time
  - the laxity changes with time

Dynamic priority scheduling

UNIVERSIDAD DE CANTABRIA

Static job priority policies are more common, because they are simpler

- **EDF and its compatible variants are the most common**
- **treatment of transient overload is more complex**
- **not supported by standard operating systems**
  - **supported in Java RTSJ**
  - **added to Ada 2005**

Mixed EDF/FPS schemes are possibly the best approach

MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. The search for predictability: Options for Managing Time
3. ***Modeling and Analyzing real-time systems: tasking models***
4. The MAST real-time model and tools environment.
5. Elements of the MAST model.
6. Modeling and integration into the design process
7. Current developments and future work in MAST
8. Conclusions

Modeling and Analyzing Real-Time Systems: A simple tasking Model

UNIVERSIDAD DE CANTABRIA

Periodic task

- initiated at fixed intervals
- must finish before start of next cycle

Task's CPU utilization: $U_i = C_i / T_i$

- C_i = compute time (execution time) for task τ_i
- T_i = period of task τ_i
- P_i = priority of task τ_i
- D_i = deadline of task τ_i
- ϕ_i = phase of task τ_i
- R_i = response time of task τ_i

CPU utilization for a set of tasks: $U = U_1 + U_2 + \ldots + U_n$

Analysis techniques: (UT, RMA...)

$$\frac{C_1}{T_1} + \ldots + \frac{C_n}{T_n} \leq U(n) = n(2^{1/n} - 1)$$

$$a_0 = C_1 + C_2 + \ldots + C_i$$

$$a_{k+1} = W_i(a_k) = \left\lceil \frac{a_k}{T_1} \right\rceil C_1 + \ldots + \left\lceil \frac{a_k}{T_{i-1}} \right\rceil C_{i-1} + C_i$$

$$a_{k+1} = a_k = R_i$$

And far many enhancements...

UNIVERSIDAD DE CANTABRIA

... deadlines before the period, interrupts, protection protocols, blocking times, preemption context switches, OS timer overheads, activation jitter, varying priorities, deadlines after period, self-suspensions, sporadic events and servers, ... interacting tasks, precedence relationship, edf and hierarchical scheduling, distributed systems, holistic analysis, offset based activations, multiprocessor shared resources,.... you say yours...

Each technique, platform, and situation requires variants of the algorithms & equations + additional elements into the model.

e.g. distributed systems

Linear end-to-end flow

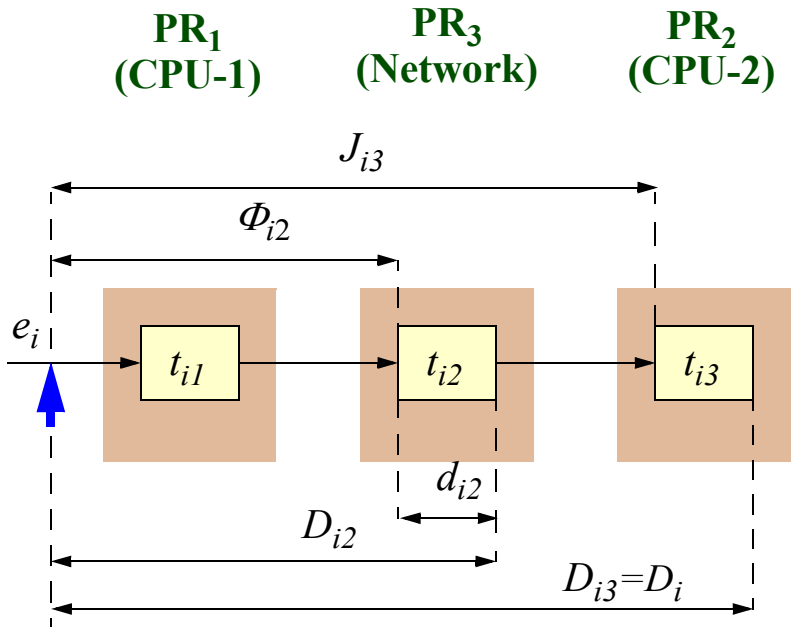
C: WCET

C^b: BCET

T: period

U=C/T

D>T allowed



- Periodic activation or sporadic with a minimum inter-arrival time
- All the steps after the first one have activation jitter
  - best-case execution times enable the reduction of jitter
- Offsets allow more accurate response times to be obtained

Knowledge transfer: From research to practitioners

UNIVERSIDAD DE CANTABRIA

Real-time analysis is an engineering basis for *analyzing* and *designing* real-time systems

- **Provides an analytical framework for verifying timing requirements**
- **Provides guidelines for optimum priority or deadlines assignment**
- **Helps to identify timing bottlenecks and errors**
- **Enable the optimization of allocations via sensitivity analysis**

Motivation for MAST

UNIVERSIDAD DE CANTABRIA

Need to do schedulability analysis in real-time applications

- formal model of the timing behavior
- analysis tools that can operate on such model

Few free-software schedulability analysis tools

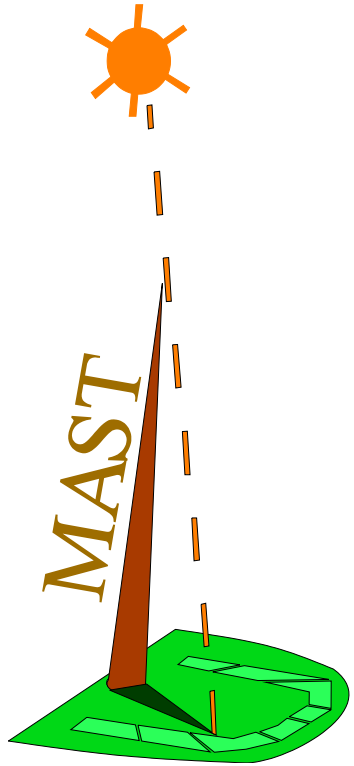
Need to integrate soft & hard real-time analysis

Need for an open tool that can be used as a plug-in module for other development tools

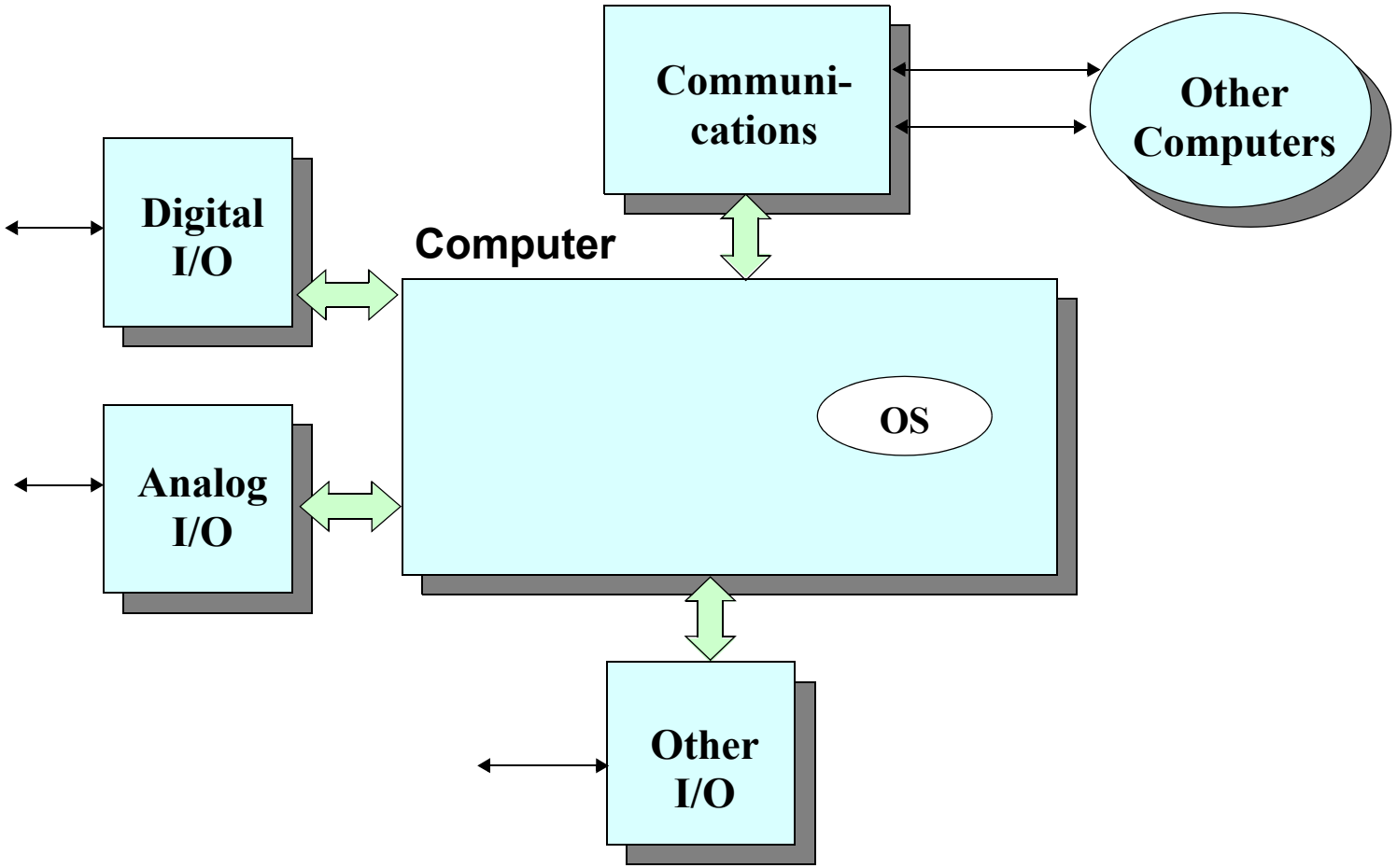
- e.g., MARTE / UML tools, model-driven frameworks, ...

Objectives of MAST

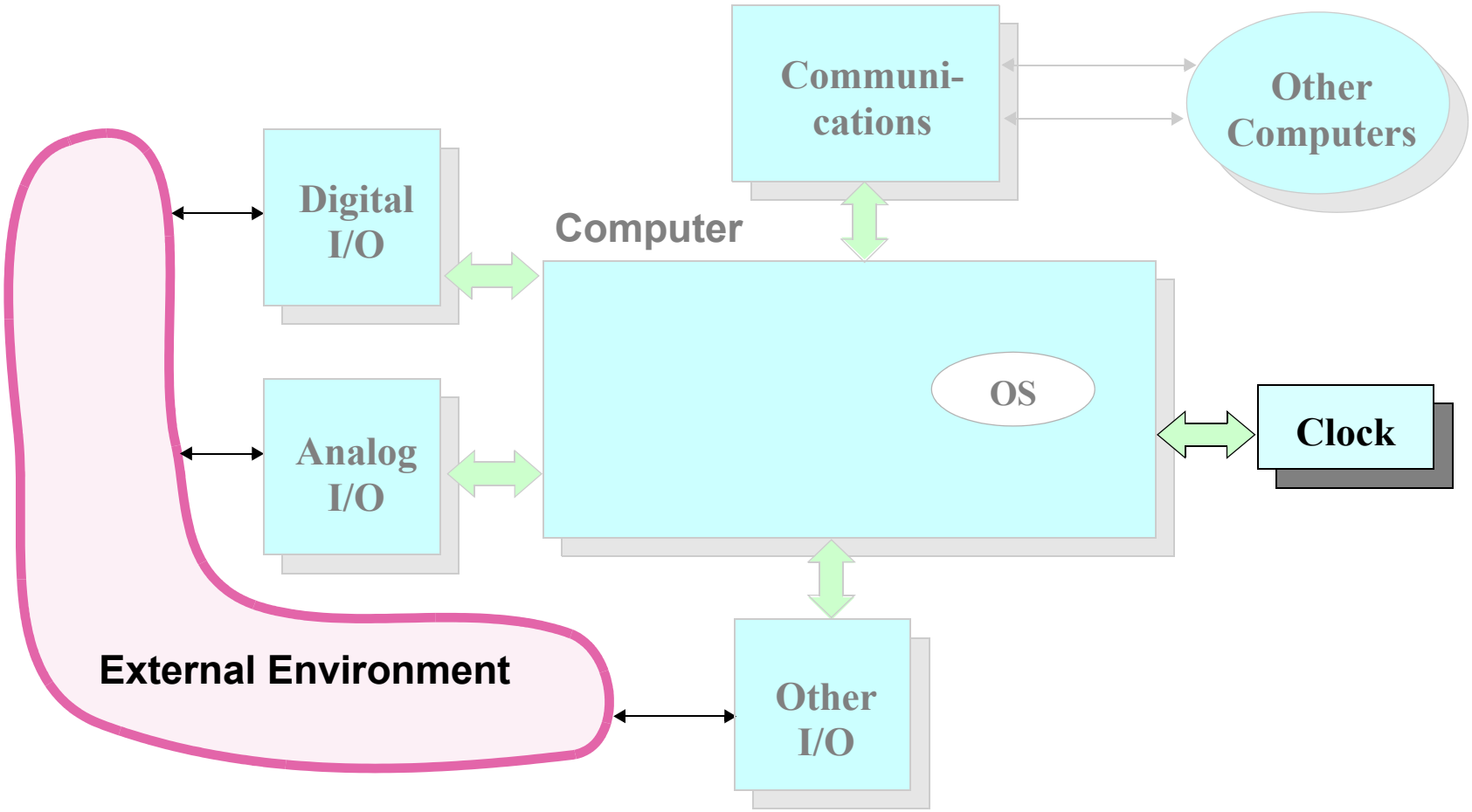
- Develop a *model* for describing the timing behavior of event-driven distributed real-time systems
  - composable software modules
  - separation of architecture, platform, and software modules
- *Open model* that may evolve to include new characteristics or points of view of the system
- Develop a set of *tools* for analyzing the timing behavior of the application



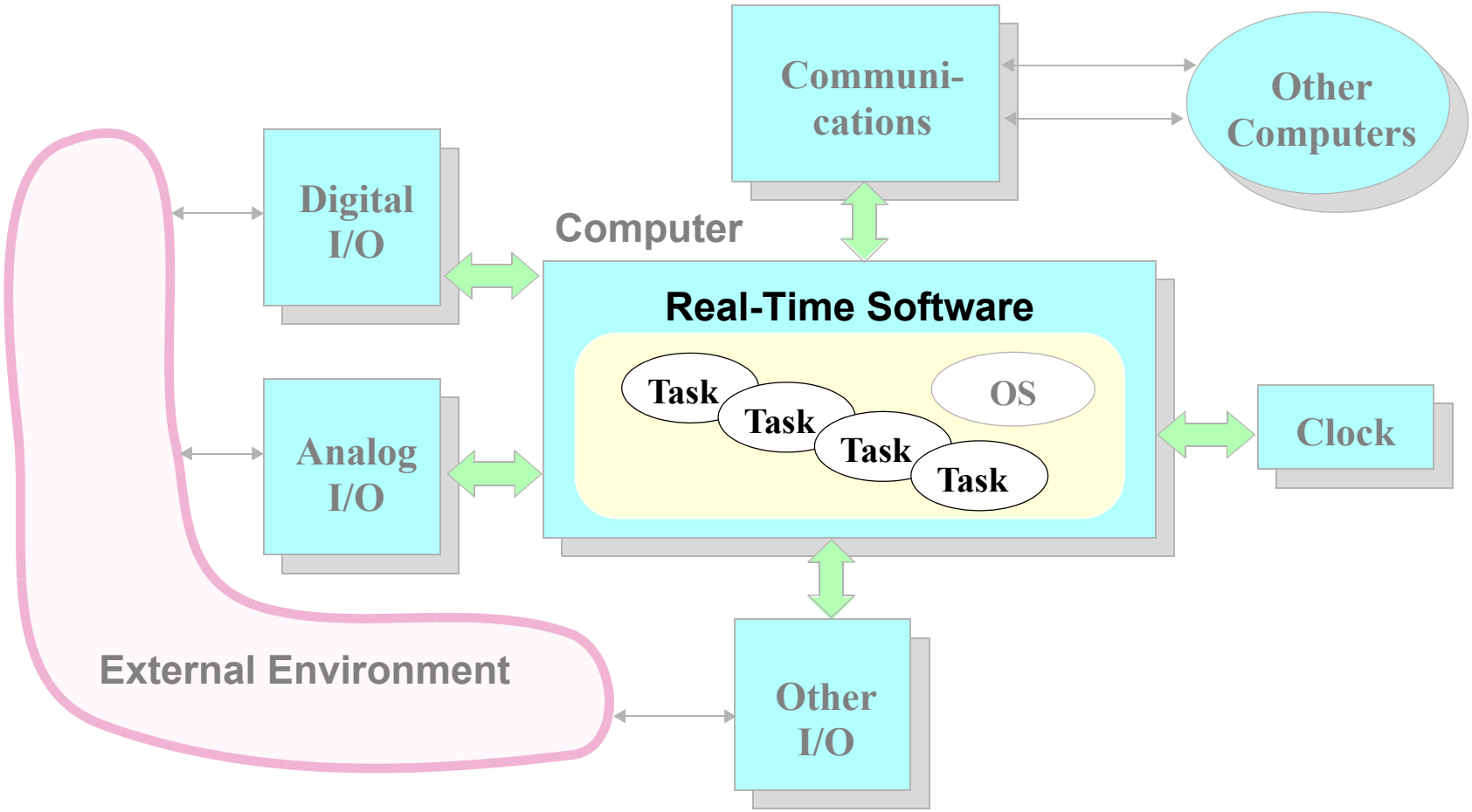
Elements involved in a real-time model: Platform



Elements involved in a real-time model: Stimuli with external events



Elements involved in a real-time model: sequential software & tasks

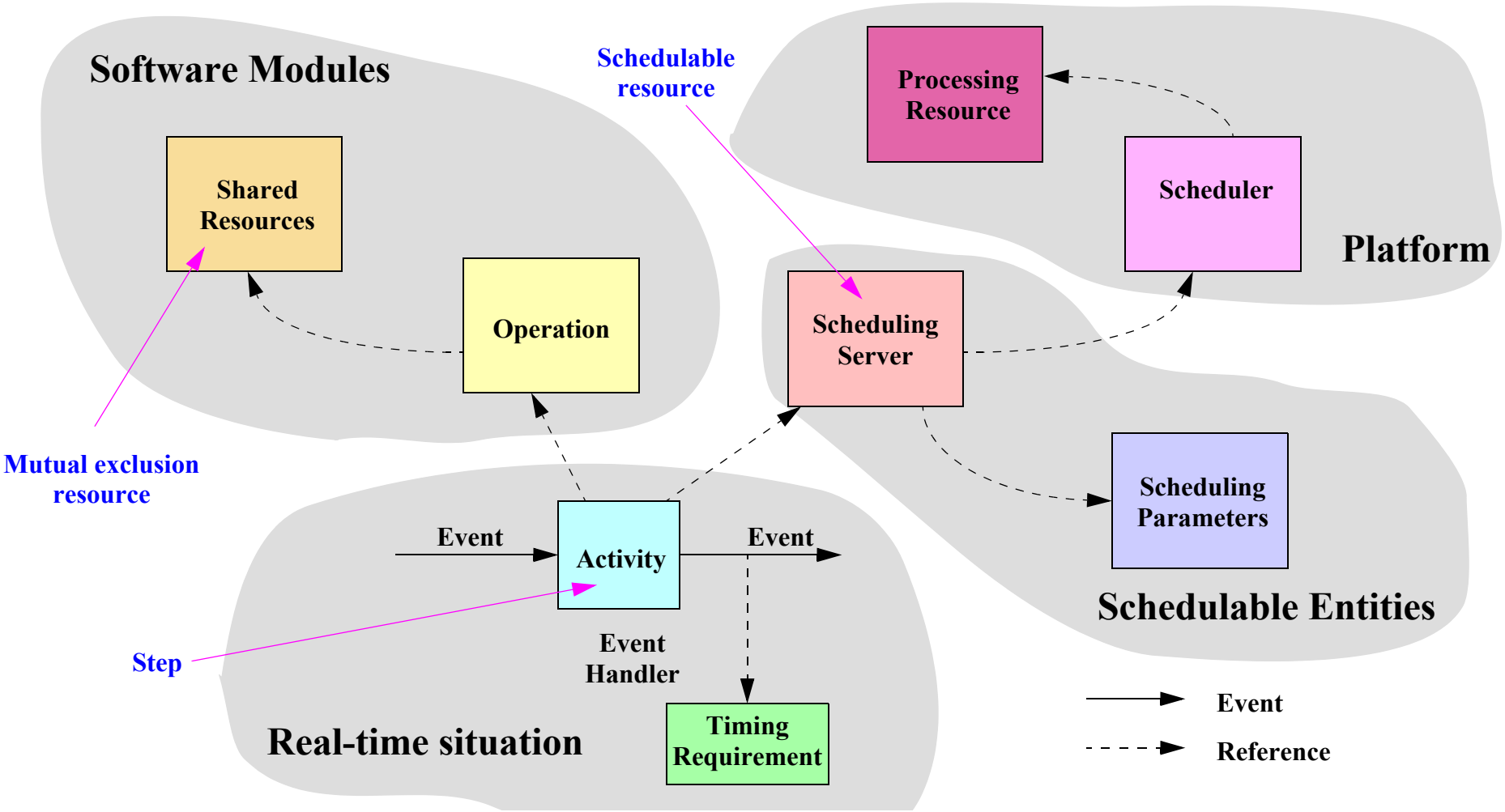


MAST: predicting response times in event-driven real-time systems

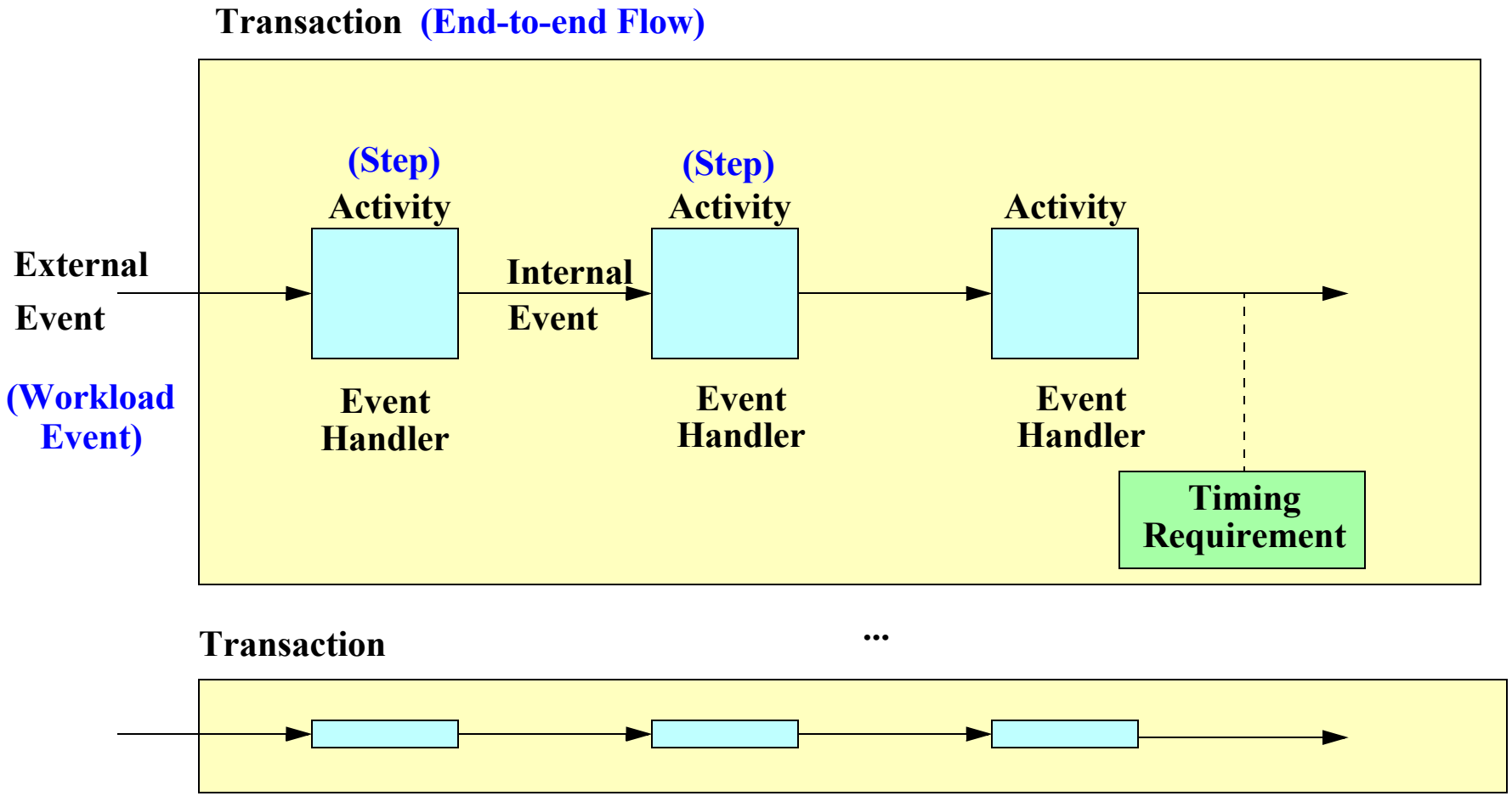
UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. The search for predictability: Options for Managing Time
3. Modeling and Analyzing real-time systems: tasking models
4. ***The MAST real-time model and tools environment.***
5. Elements of the MAST model.
6. Modeling and integration into the design process
7. Current developments and future work in MAST
8. Conclusions

The MAST Real-Time Model: Overview

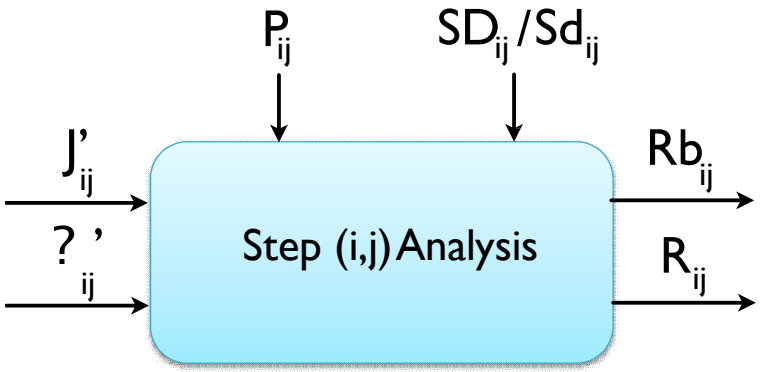


Real-Time Situation



The Holistic (offset-based) RTA algorithm

```
begin
    Jitters and Offsets Initialization
    loop
        for each end-to-end flow (i)
            for each step (j)
```

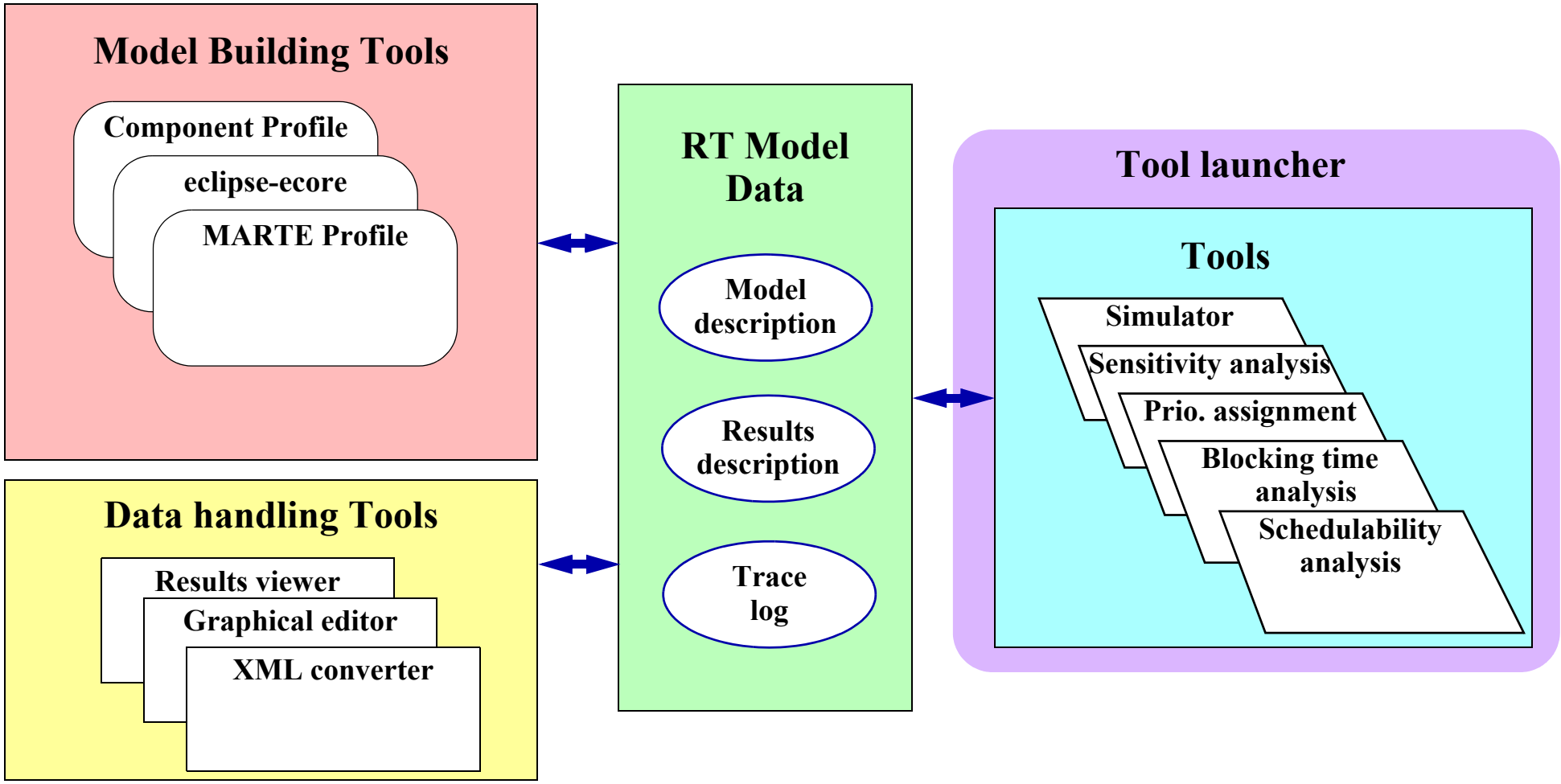



```
            end loop
        end loop
        exit when No Changes in Rij,Rbij
    end loop;
end RTA_Holistic
```

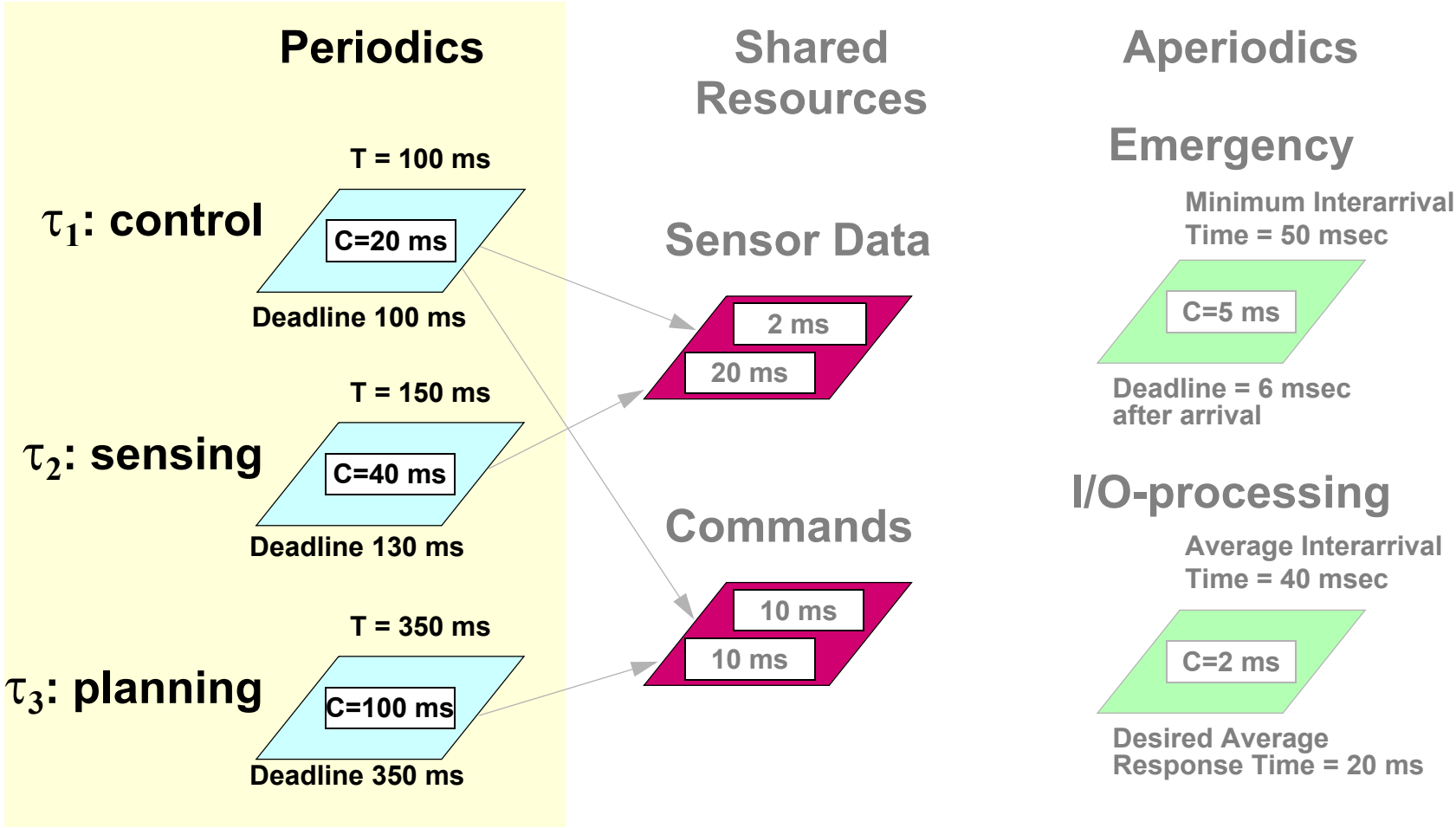
$$J'_{ij} = J_{ij} + \max\left(R_{ij-1}, \Phi_{ij}\right) - \max\left(R^{b}_{ij-1}, \Phi_{ij}\right)$$

$$\Phi'_{ij} = \max\left(R^{b}_{ij-1}, \Phi_{ij}\right)$$

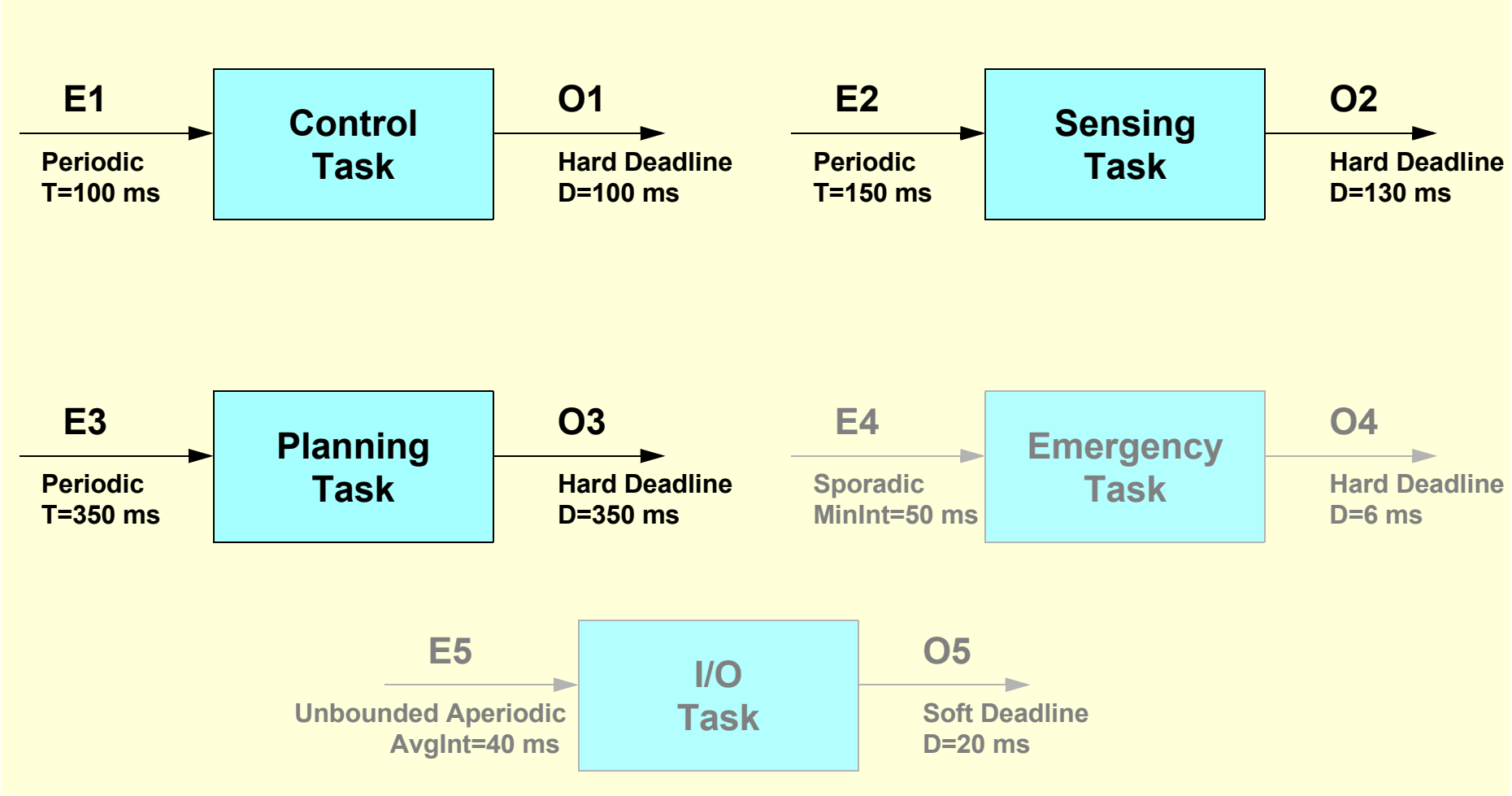
MAST tooling Environment



A basic real-time system example



Transactions in this example



MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. The search for predictability: Options for Managing Time
3. Modeling and Analyzing real-time systems: tasking models
4. The MAST real-time model and tools environment.
5. ***Elements of the MAST model.***
6. Modeling and integration into the design process
7. Current developments and future work in MAST
8. Conclusions

Elements of the MAST Model

UNIVERSIDAD DE CANTABRIA

Platform

1. Processing Resources
2. Schedulers (primary, secondary)
3. Scheduling policies (fixed priorities, EDF,...)
4. System Timers (overhead)
5. Network Drivers (overhead)

Schedulable entities

6. Scheduling Parameters (priorities, deadlines)
7. Scheduling servers or schedulable resources (tasks, processes, threads, message streams...)
8. Synchronization parameters (preemption levels,...)

Elements of the MAST Model (cont'd)

UNIVERSIDAD DE CANTABRIA

Software modules

9. Operations (procedures, functions, messages)
10. Shared resources (mutually exclusive)

Real-time situation

11. Events (external stimuli and internal)
12. Timing Requirements
13. Event Handlers (steps, forks, joins, branch, barrier...)
14. Transactions
15. Overall system model

Textual Specification Language

UNIVERSIDAD DE CANTABRIA

Syntax rules:

- Object format: `Object_Name (Parameters);`
- Objects have a *type* and/or *name* (mandatory)
- Spaces, tabs and line breaks are not considered
- Names like in Ada:
  `letter+(letter | number | underline | period)`
- Names with or without "*quotes*" (mandatory for reserved words)
- Referenced names must have been defined previously
- Comments like in Ada: `"--"`
- Case insensitive

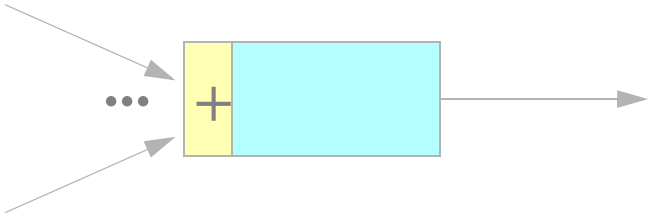
No need to define an identifier before it is used

An XML version for the input model also exists

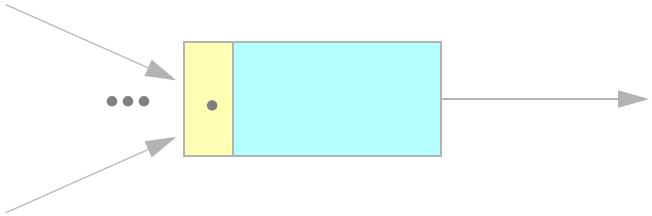
Event Handlers

Activity / Timed Activity / Rate Divisor / Delay / Offset

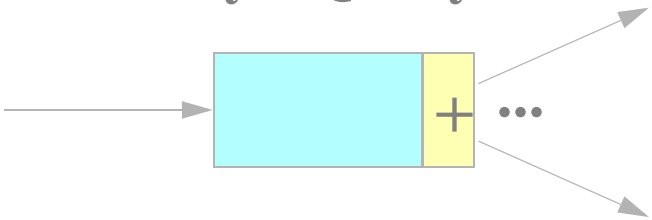
Concentrator



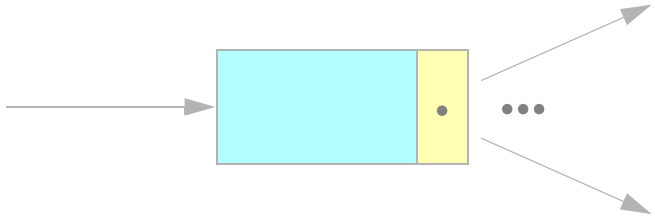
Barrier



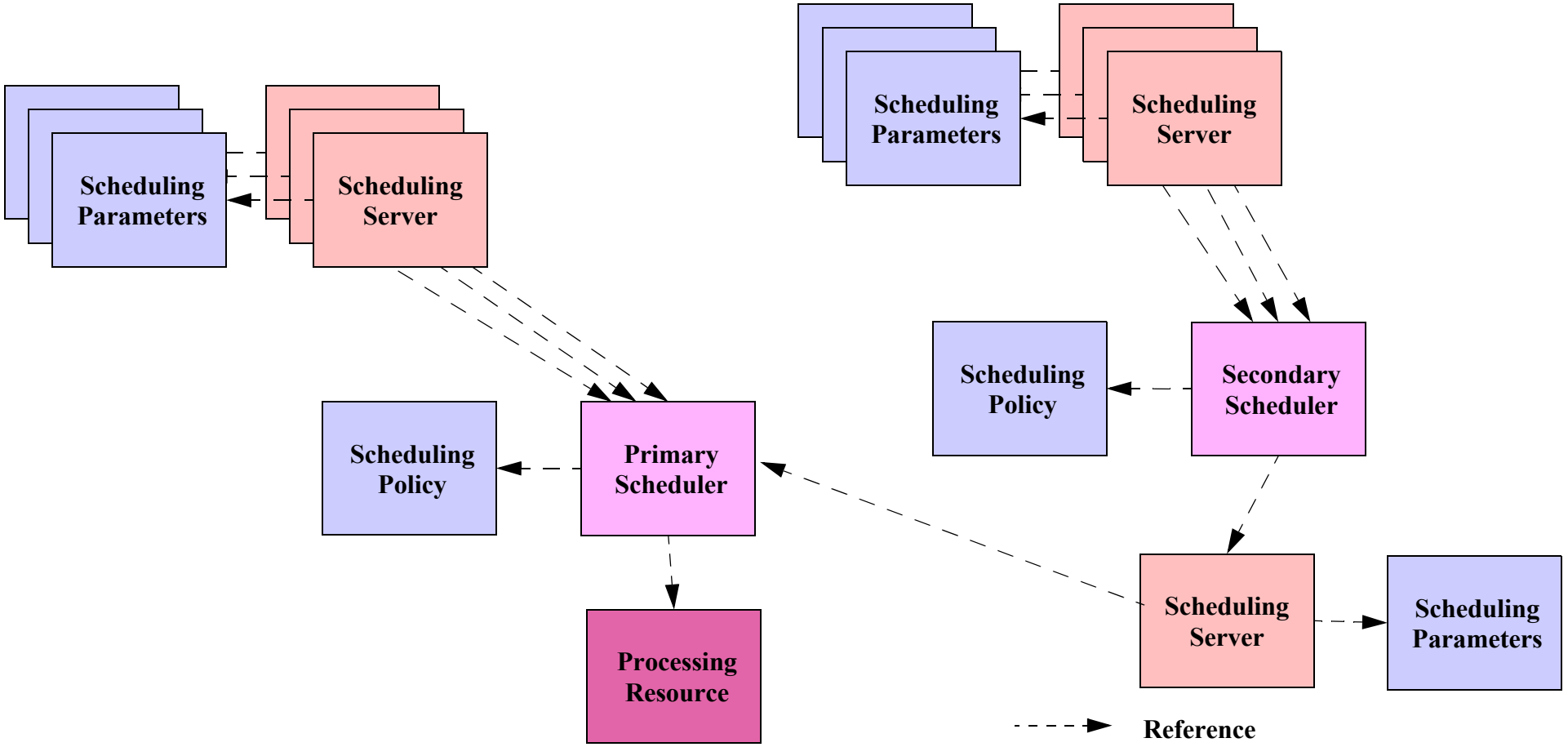
Delivery / Query Server



Multicast



Hierarchical schedulers in MAST



Real-time networks

UNIVERSIDAD DE CANTABRIA

Few networks guarantee real-time response

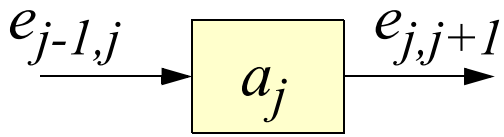
- many protocols allow collisions
- no priorities or deadlines in the protocols

Some solutions

- CAN bus and other field buses
- Priority-based token passing
- Time-division multiplexed access
- Point to point networks
- AFDX
- SpaceWire

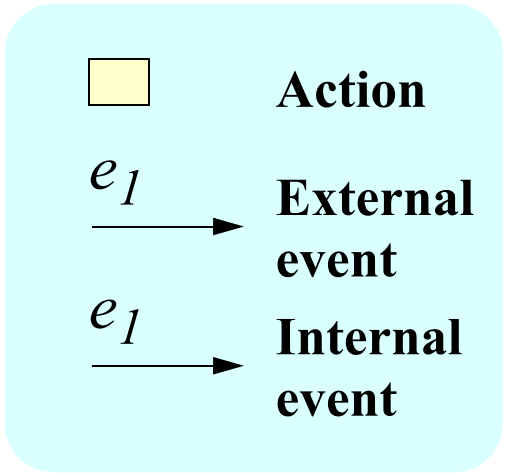
Distributed system model

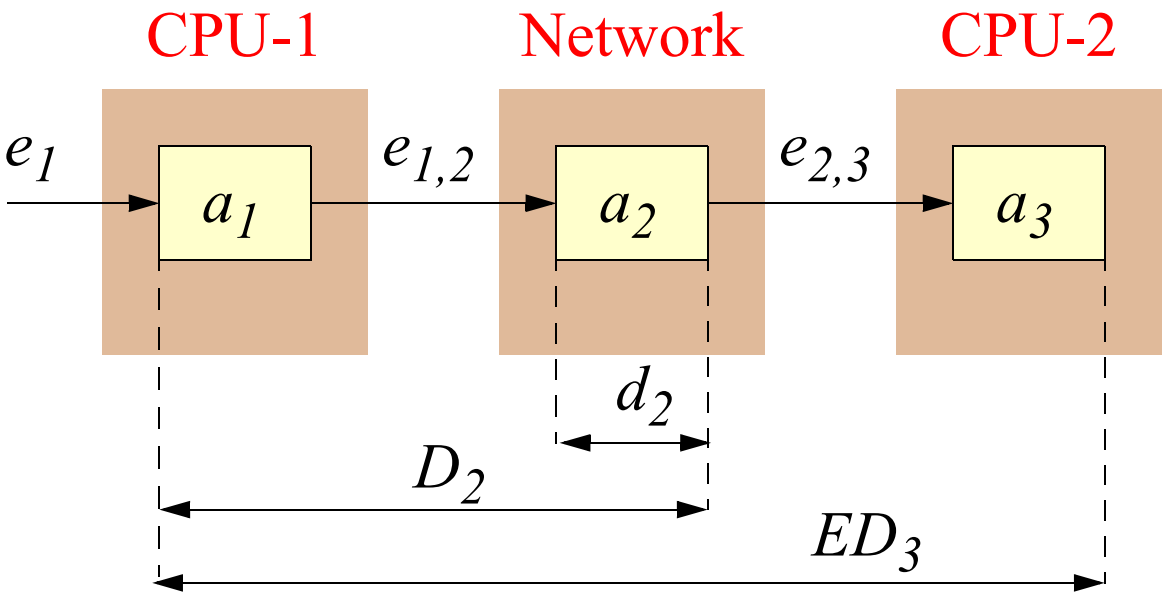
Linear Action:



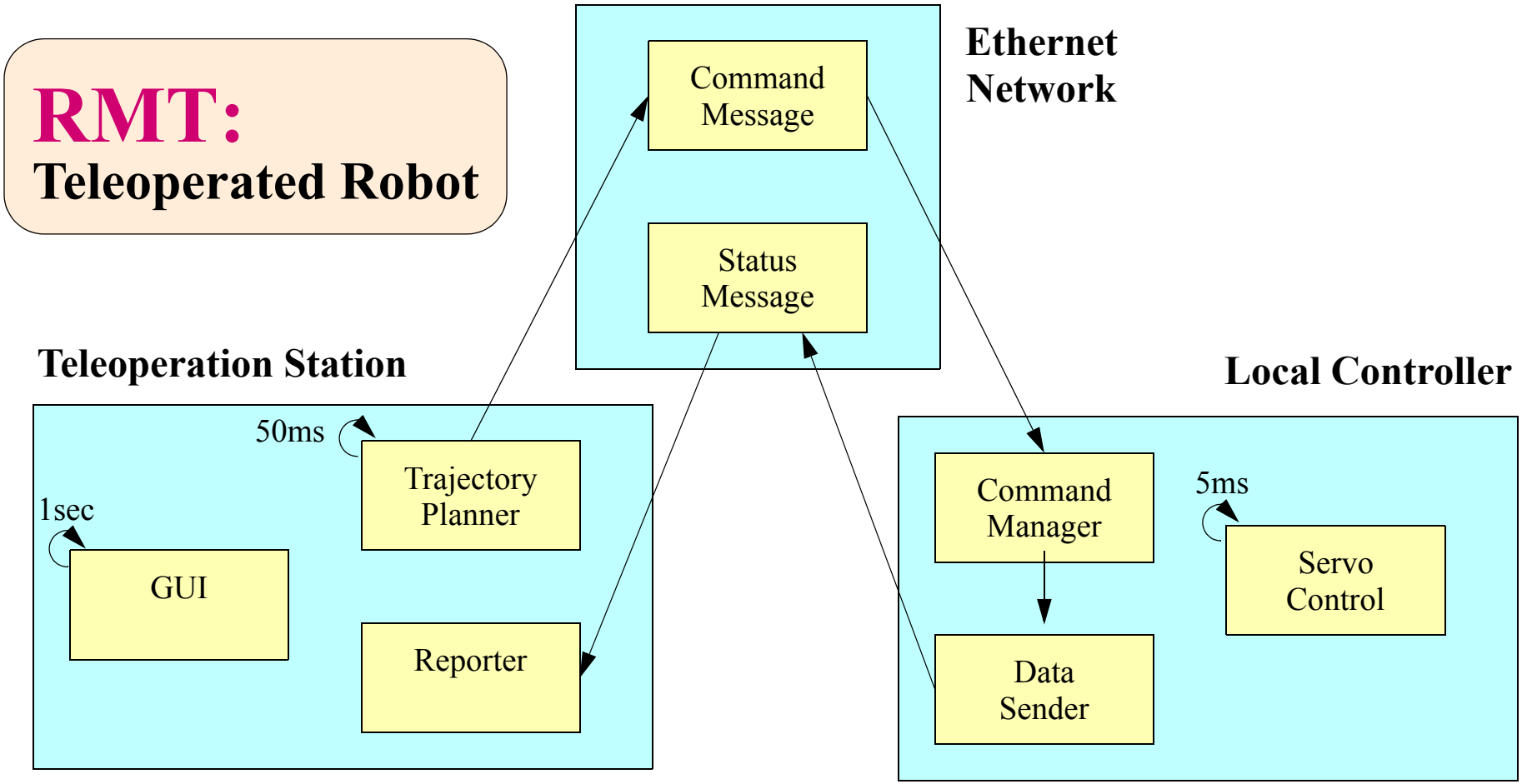
$$T_{j-1,j} = T_j = T_{j,j+1}$$

Linear Response to an Event:

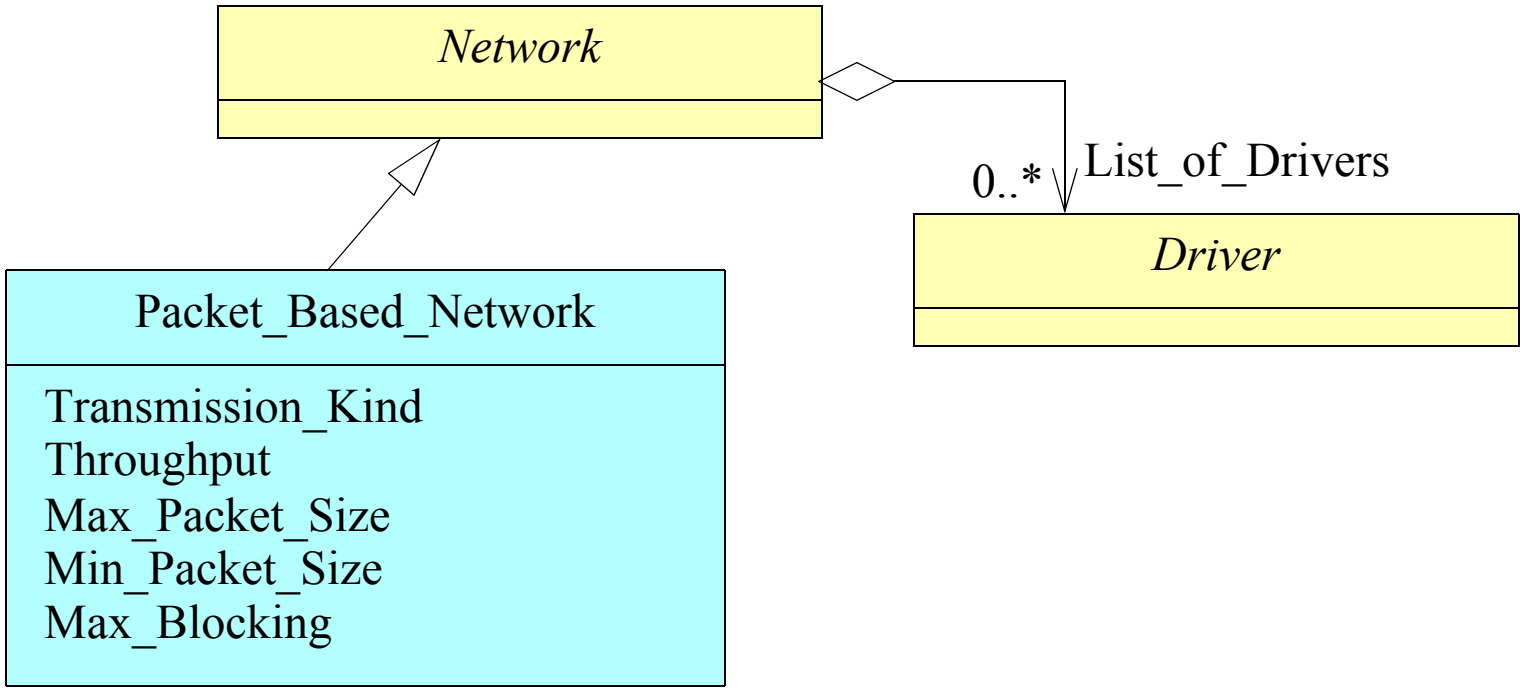




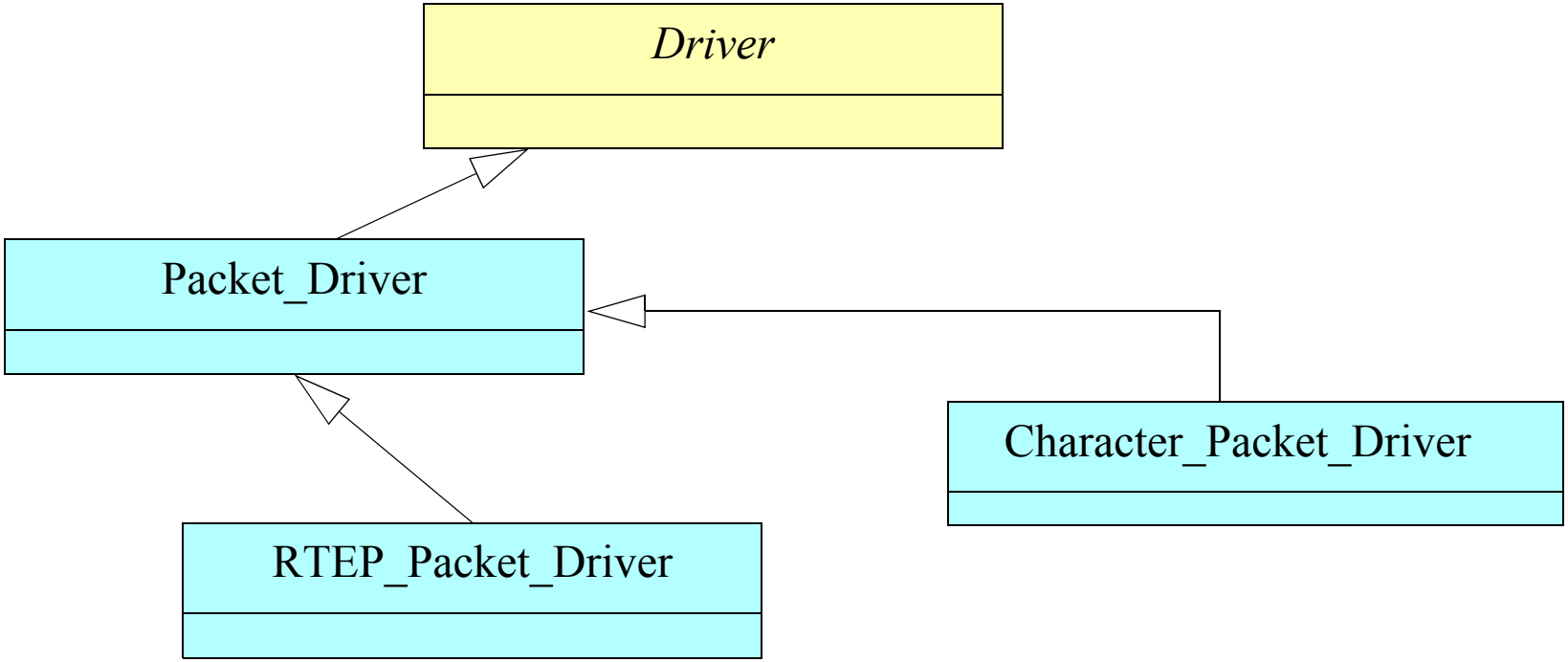
Example


RMT:
Teleoperated Robot
Ethernet
Network
Command
Message
Status
Message
Teleoperation Station
50ms
Trajectory
Planner
1sec
GUI
Reporter
Local Controller
Command
Manager
5ms
Servo
Control
Data
Sender

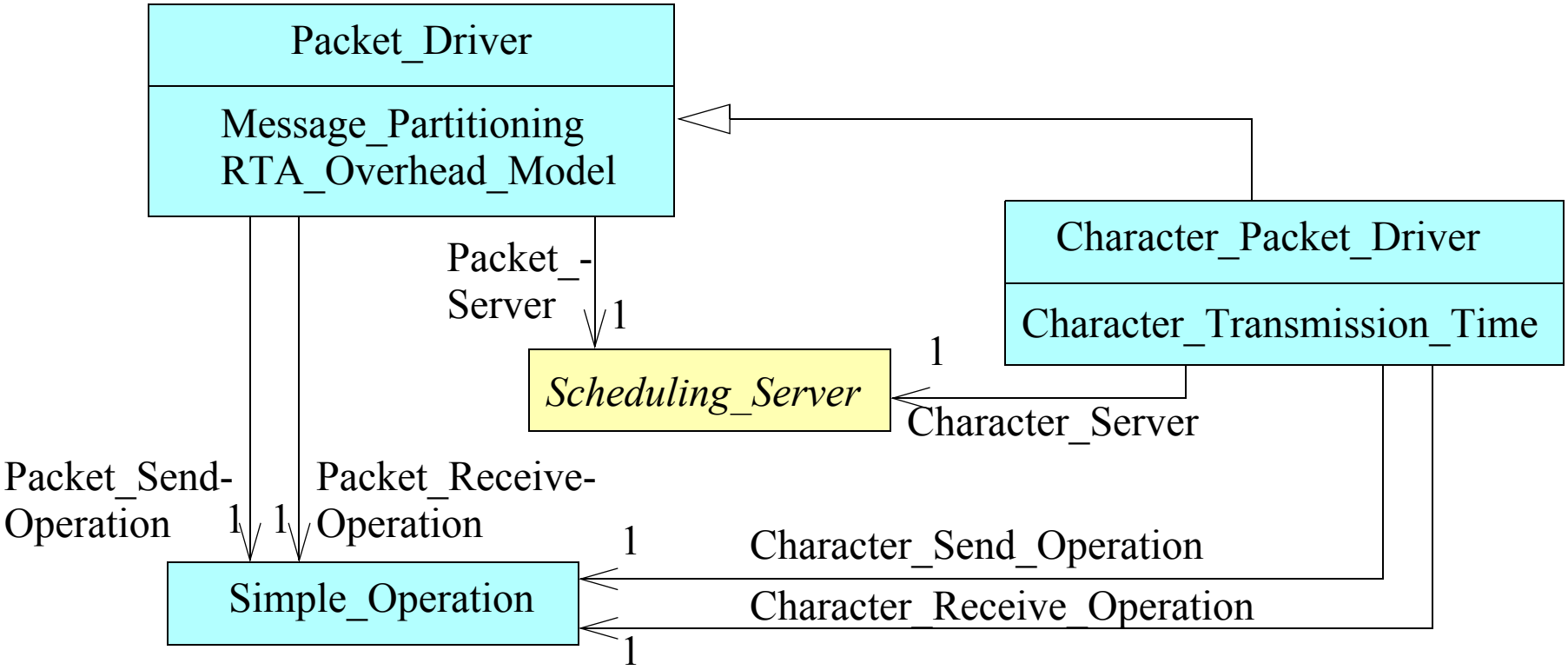
Platform: Networks



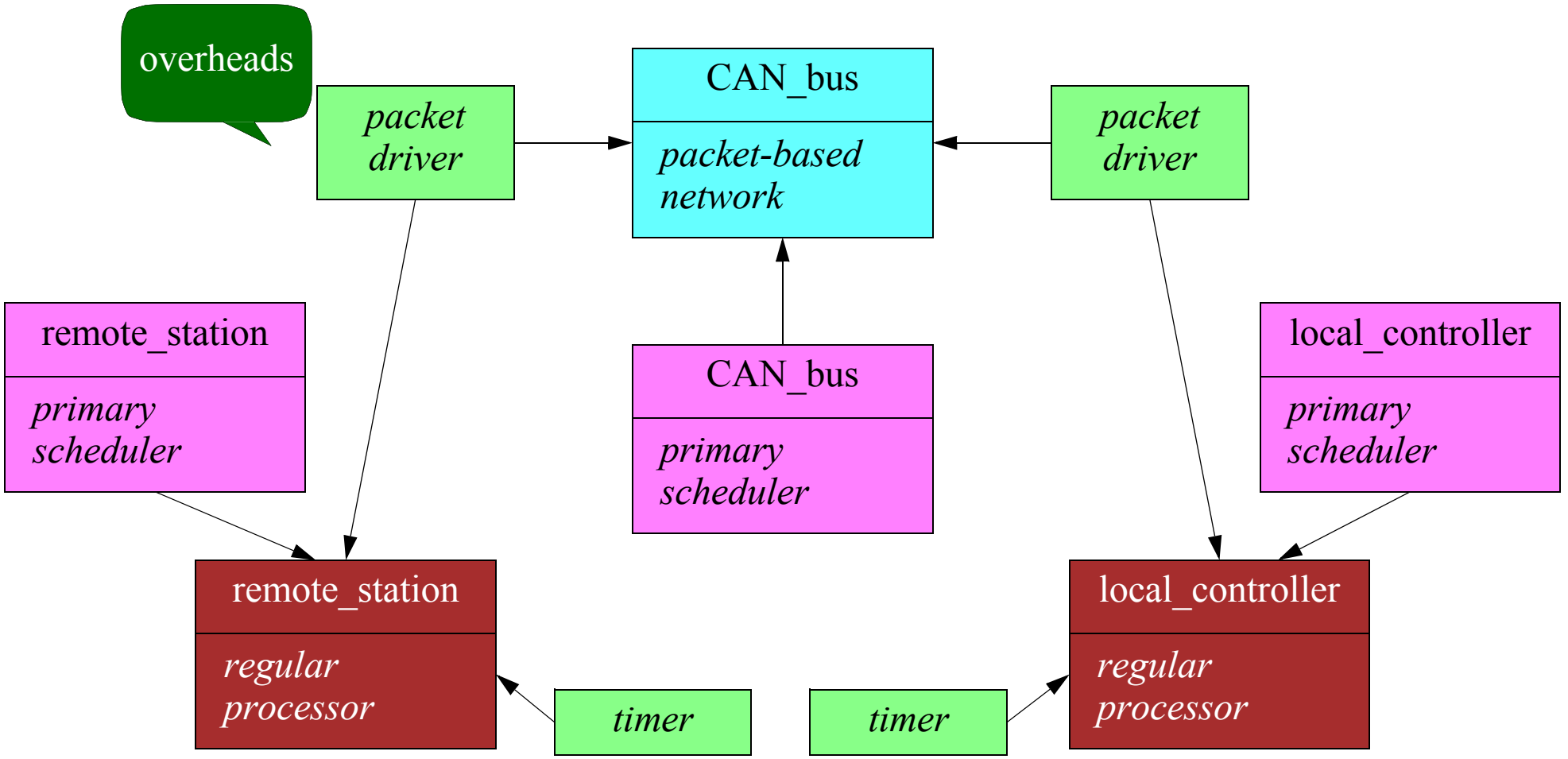
Processor overheads caused by network activity: network drivers



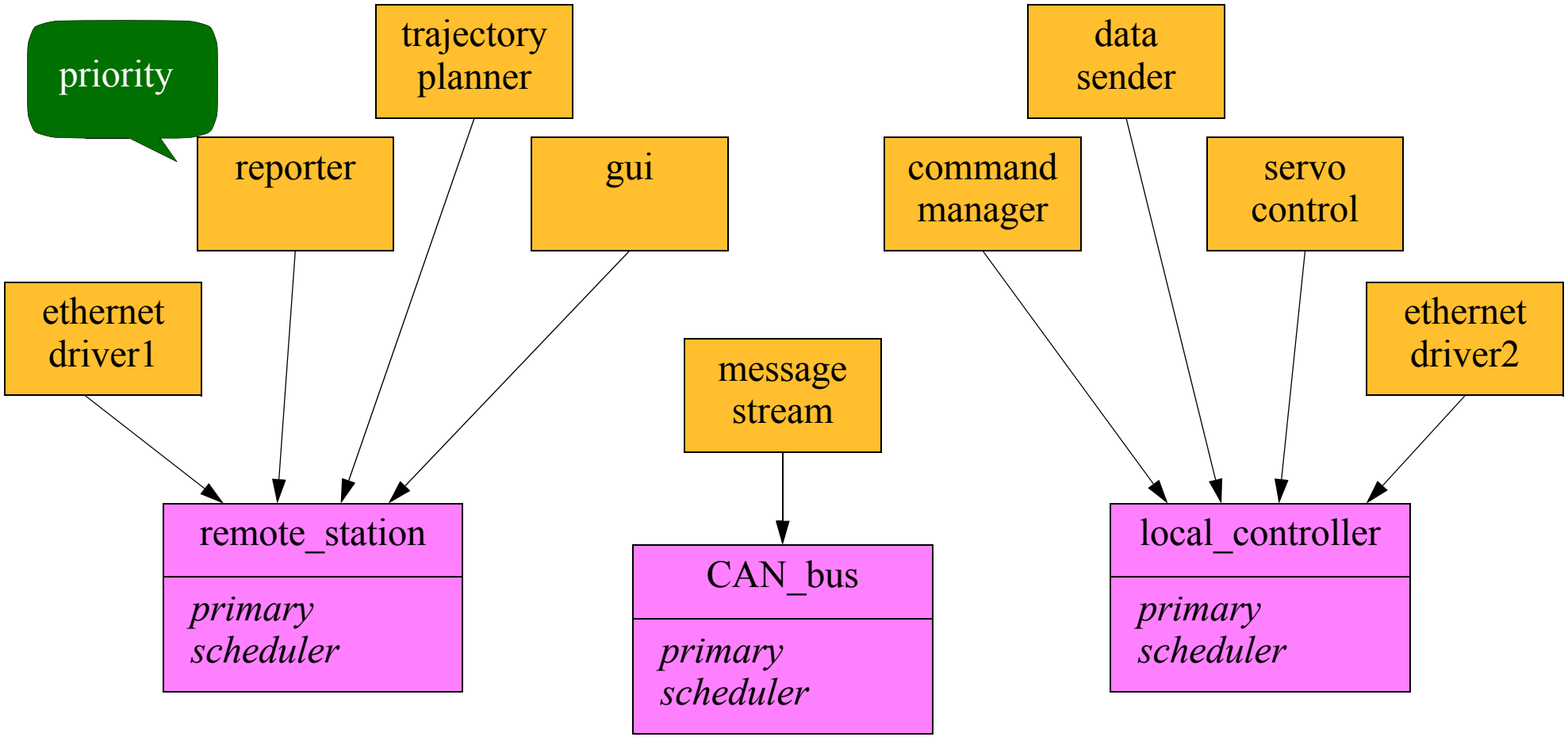
Network drivers (cont'd)



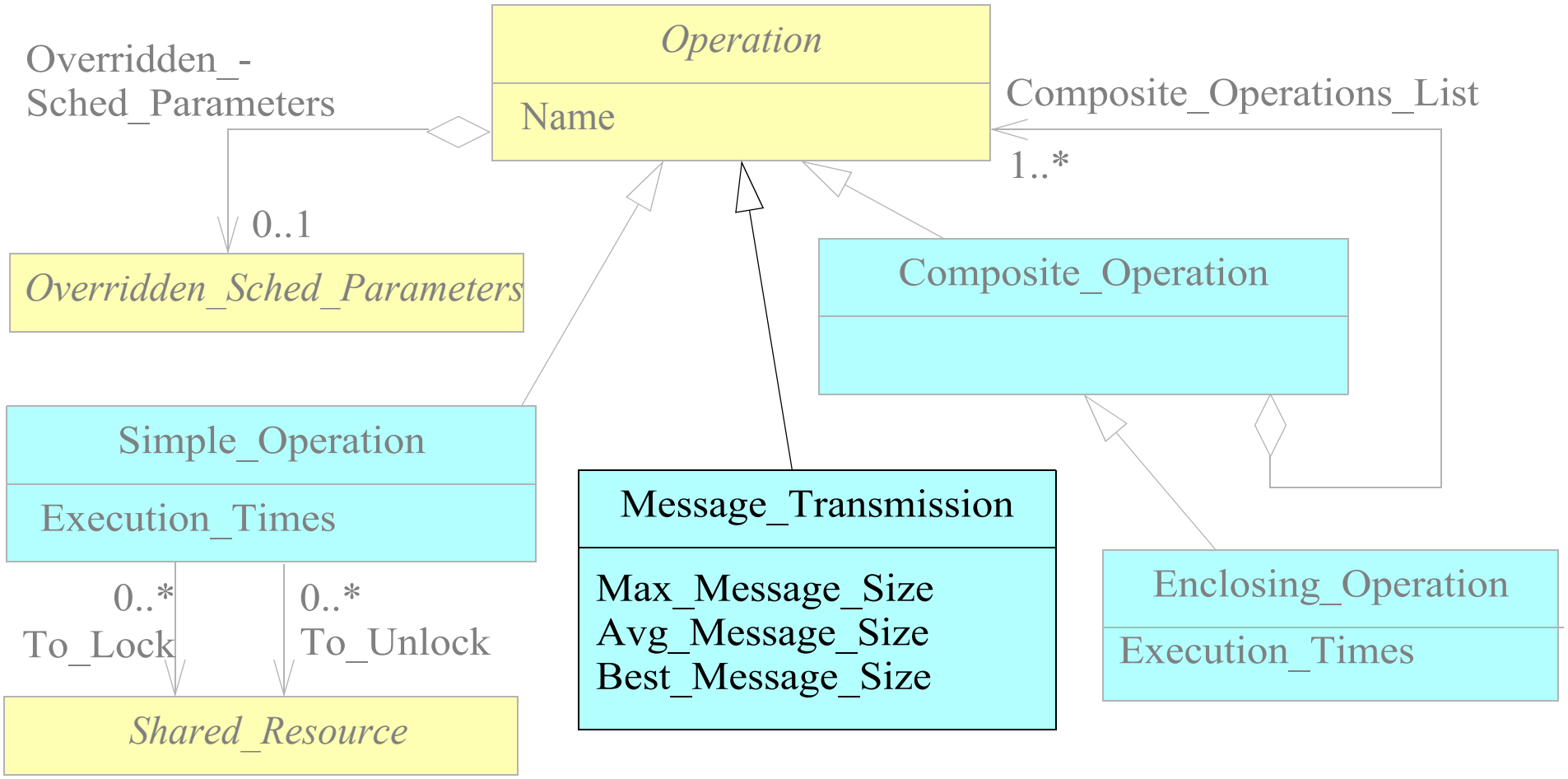
Processing resources, schedulers, drivers, and timers



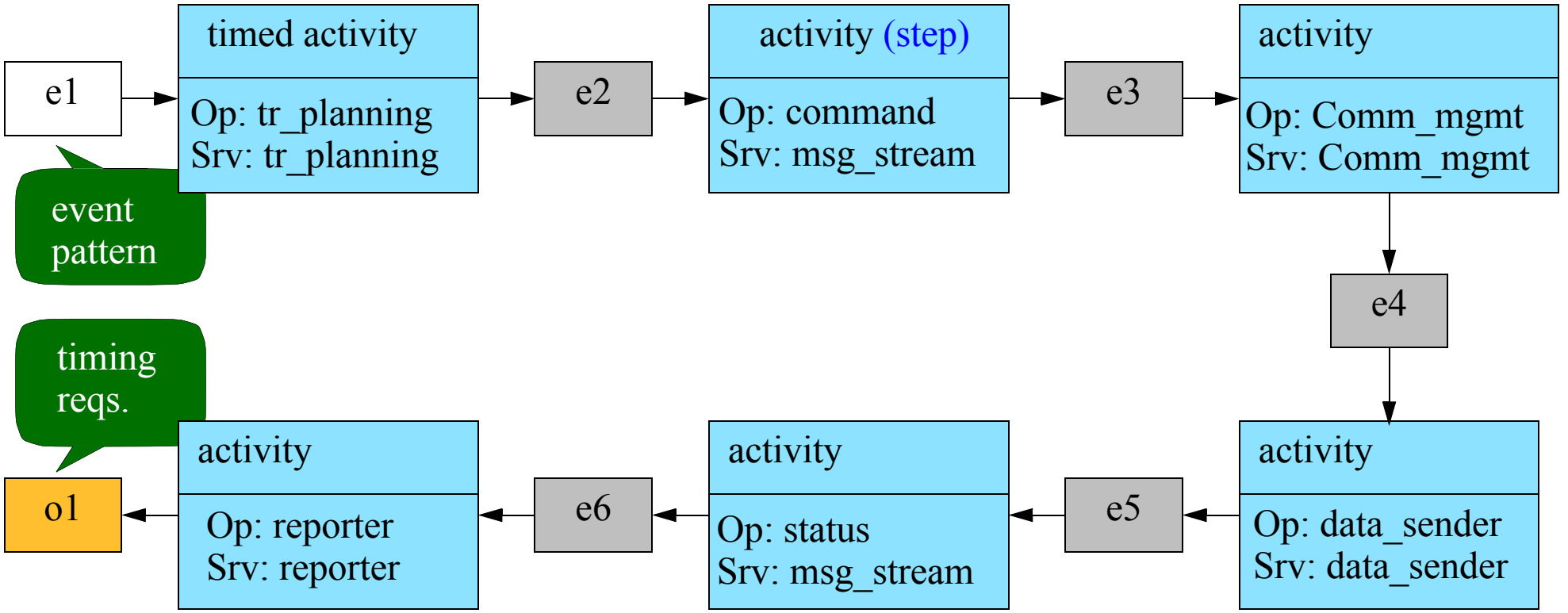
Scheduling servers (schedulable resources)



Operations


Operation
Name
Overridden_-
Sched_Parameters
0..1
Overridden_Sched_Parameters
Composite_Operations_List
1..*
Composite_Operation
Simple_Operation
Execution_Times
0..*
To_Lock
0..*
To_Unlock
Shared_Resource
Message_Transmission
Max_Message_Size
Avg_Message_Size
Best_Message_Size
Enclosing_Operation
Execution_Times

Transactions (end-to-end flows): Distributed transaction in the example



MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. The search for predictability: Options for Managing Time
3. Modeling and Analyzing real-time systems: tasking models
4. The MAST real-time model and tools environment.
5. Elements of the MAST model.
6. *Modeling and integration into the design process*
7. Current developments and future work in MAST
8. Conclusions

...in a design process...

UNIVERSIDAD DE CANTABRIA

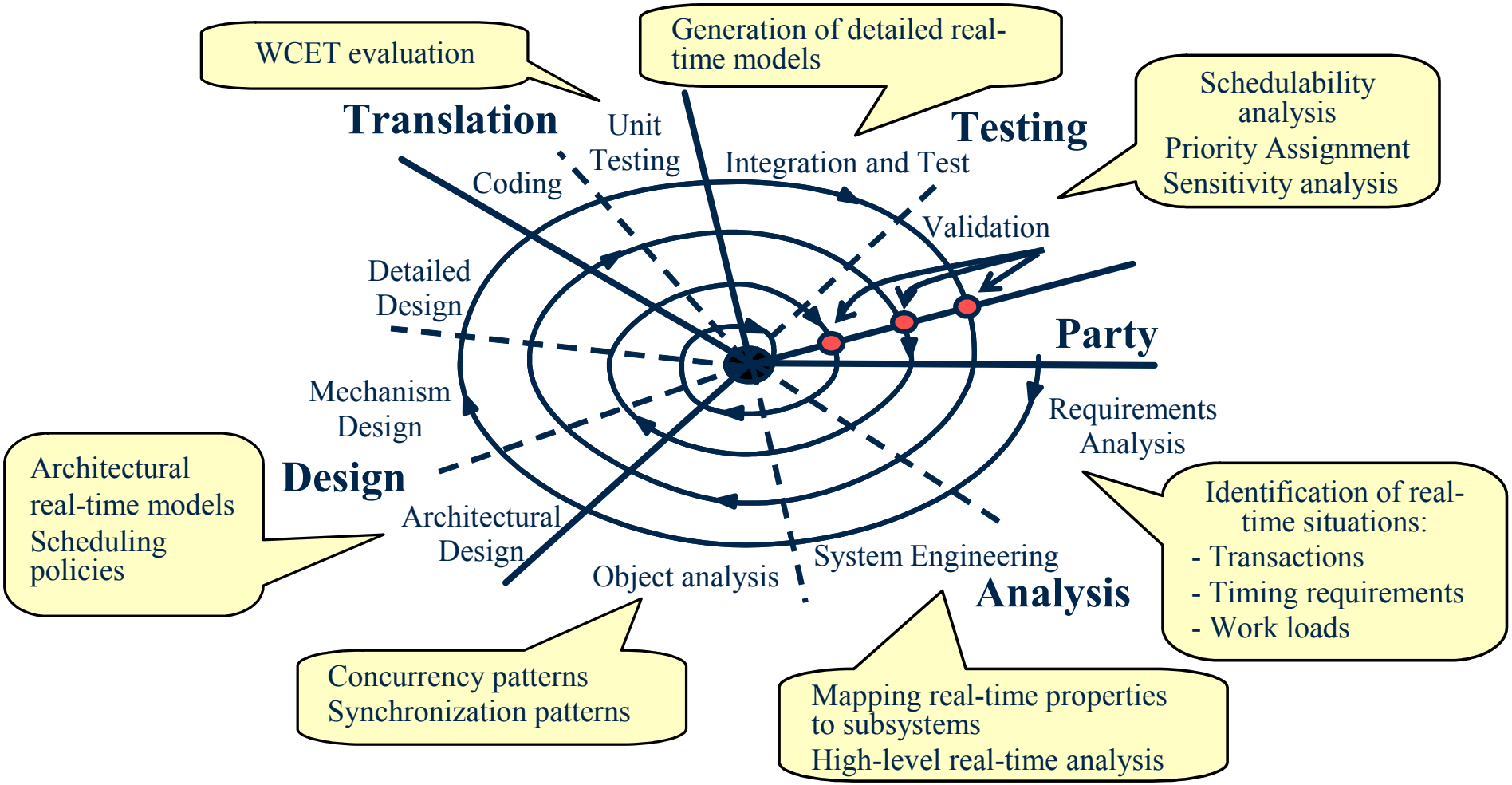
From a software engineering development process perspective, the latest schedulability analysis techniques are difficult to apply, more even by hand.

Need to integrate:

- schedulability analysis tools
- priority assignment
- sensitivity analysis
- design space exploration

A model of the real-time behavior at each relevant phase of the development process is necessary.

Integration into the design process



MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. The search for predictibility: Options for Managing Time
3. Modeling and Analyzing real-time systems: tasking models
4. The MAST real-time model and tools environment.
5. Elements of the MAST model.
6. Modeling and integration into the design process
7. *Current developments and future work in MAST*
8. Conclusions

Analysis Techniques Implemented

UNIVERSIDAD DE CANTABRIA

Technique	Single-Proc.	Multiple Proc.	Simple Trans.	Linear Trans.	Multi Path Trans.	FP	EDF
Classic RMA	☑		☑			☑	
EDF Monoprocesor	☑		☑				☑
Varying Priorities	☑		☑	☑		☑	
Holistic	☑	☑	☑	☑	☑	☑	☑
EDF within priorities	☑	☑	☑	☑		☑	☑
Offset Based aprox.	☑	☑	☑	☑		☑	☑
Offset based brute force	☑	☑	☑	☑		☑	☑
Offset Based aprox w precedence relations.	☑	☑	☑	☑		☑	☑
Offset Based Slanted	☑	☑	☑	☑		☑	☑

Scheduling parameters assignment

Technique	Fixed priorities	EDF	Single-Processor	Multiple Processors	Heteroge-neous
Monoprocessor	☑	☑	☑		
Simulated Annealing	☑		☑	☑	
HOSPA	☑	☑	☑	☑	☑
Proportional Deadline (PD)	☑	☑	☑	☑	☑
Normalized Proportional Deadline (NPD)	☑	☑	☑	☑	☑

Future Work: MAST-2

UNIVERSIDAD DE CANTABRIA

Align names with MARTE

Partitioned scheduling

- support for ARINC 653 systems with hierarchical scheduling
  - fixed priorities on top of timed partitions
- support for TTP networks

Network switches

- support for AFDX deterministic ethernet
- support for prioritized switches

Resource reservations

- virtual resources as a new primary scheduler

Future Work: MAST-2 (cont'd)

UNIVERSIDAD DE CANTABRIA

Enhance modelling capabilities

- support for thread locking from a transaction
  - enable modelling synchronous RPC
- enhanced modelling of timers
  - allow multiple timers

MAST: predicting response times in event-driven real-time systems

UNIVERSIDAD DE CANTABRIA

1. Introduction: Elements of a real-time system
2. The search for predictibility: Options for Managing Time
3. Modeling and Analyzing real-time systems: tasking models
4. The MAST real-time model and tools environment.
5. Elements of the MAST model.
6. Modeling and integration into the design process
7. Current developments and future work in MAST
8. *Conclusions*

Conclusions

UNIVERSIDAD DE CANTABRIA

Real-time theory is now capable of analyzing complex RT systems

MAST defines a model for describing real-time systems

- distributed and multiprocessor
- complex synchronization and event-driven schemes
- composable software modules
- independence of architecture, platform and modules

Conclusions (cont'd)

UNIVERSIDAD DE CANTABRIA

MAST provides an open set of tools

- hard and soft real-time analysis
- automatic blocking times
- priority assignment
- sensitivity analysis

XML and ecore input specification formalisms allow easy integration with other tools (e.g. UML and eclipse tools)

MAST is free software

Conclusions (cont'd)

UNIVERSIDAD DE CANTABRIA

MAST is evolving and will soon cover aspects such as:

- alignment with MARTE high level application modelling
- partitioned scheduling
- network switches
- additional modelling capabilities

URL

http://mast.unican.es