

Sistemas Distribuidos de Tiempo Real

Apuntes: TEMA 8

Por: Héctor Pérez Tijero perezh@unican.es
<http://www.ctr.unican.es/>

Grupo de Computadores y Tiempo Real, Universidad de Cantabria

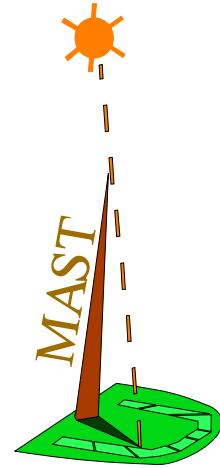
Sistemas distribuidos de tiempo real

PARTE IV: Análisis de Sistemas de Tiempo Real Distribuidos

- **TEMA 8. Modelo transaccional de sistema distribuido**
- TEMA 9. Análisis de sistemas distribuidos

8. Modelling and Analyzing with MAST Tools

- 8.1 Overview of the real-time model
- 8.2 The platform model
- 8.3 Modelling the software modules
- 8.4 Modelling the activities
- 8.5 Analysis tools
- 8.6 Distributed Systems



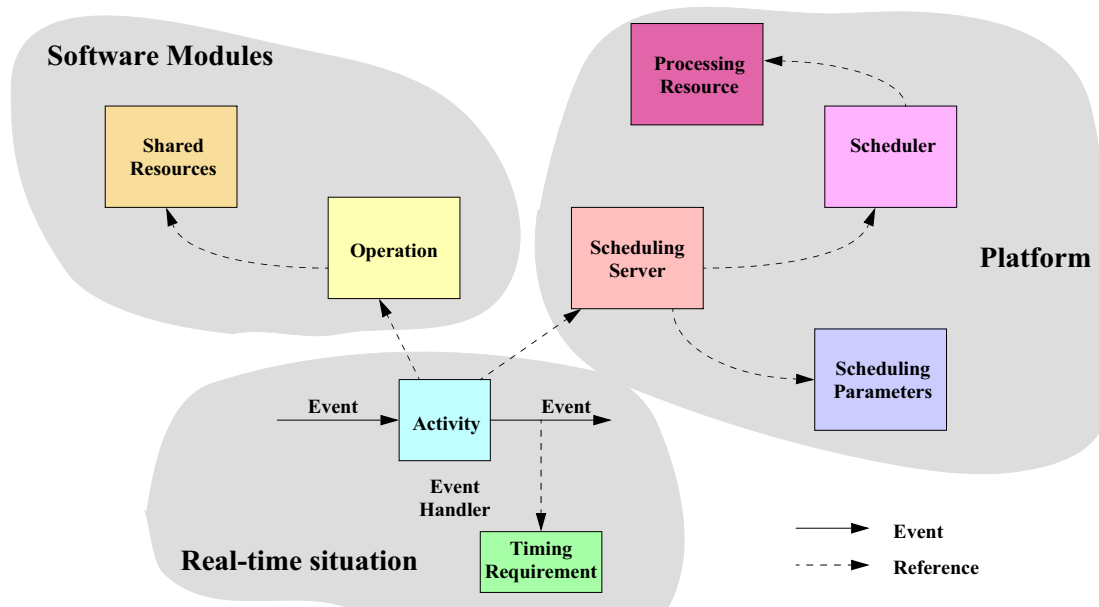
8.1 Overview of the Real-Time Model

The real-time system model contains three independent parts:

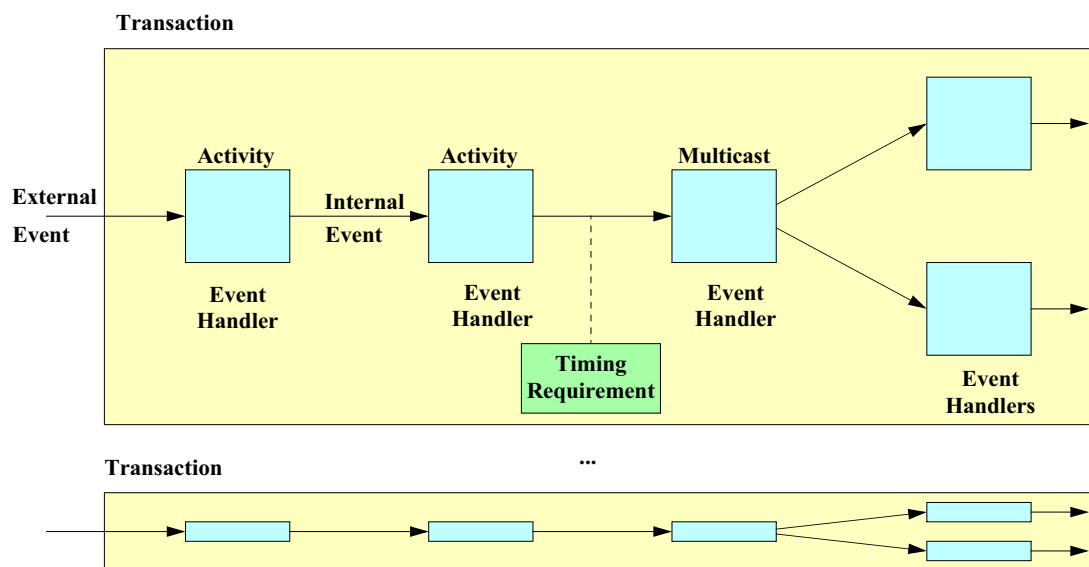
- Platform
- Software modules
- Real-time situation
 - representing a particular mode of operation of the system
 - composed of a set of *transactions*

A transaction contains a set of *activities* that will be executed by the system in response to events

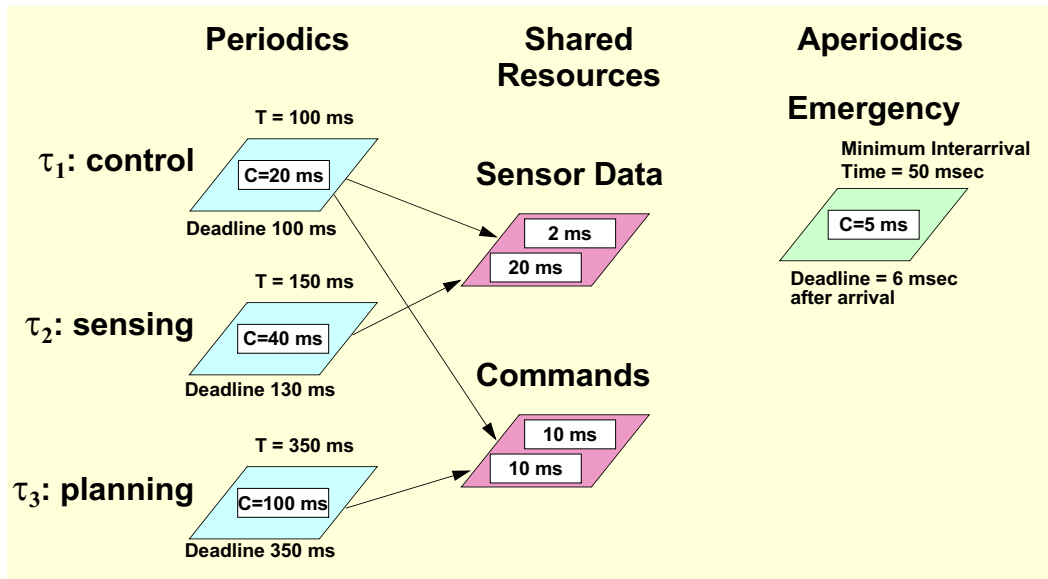
Real-Time Model: Overview



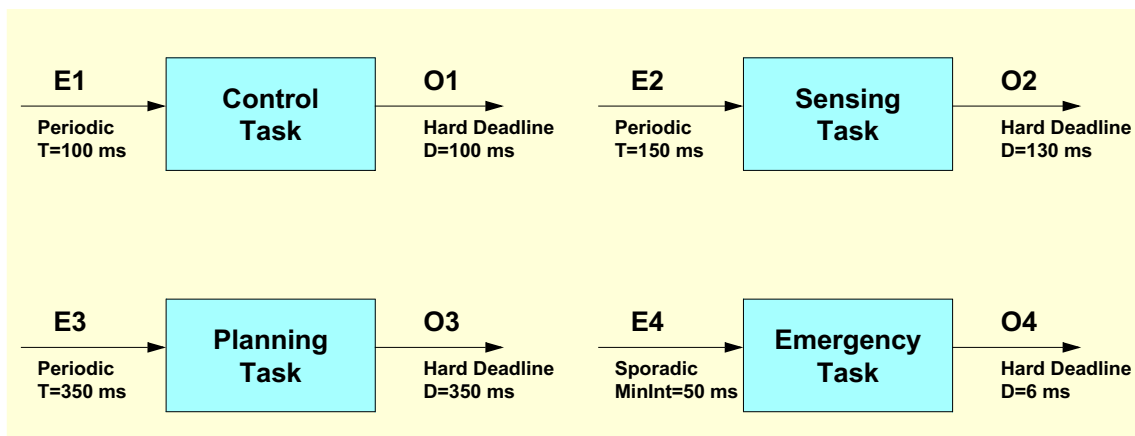
Real-Time Situation



An example



Transactions in this example



Elements of the MAST Model



Platform

1. Processing Resources
2. System Timers
3. Network Drivers
4. Schedulers (primary, secondary)
5. Scheduling policies (fixed priorities, EDF, ...)
6. Scheduling Parameters (priorities, deadlines)
7. Synchronization parameters (preemption levels, ...)
8. Scheduling servers (tasks, processes, threads,...)

Elements of the MAST Model (cont'd)



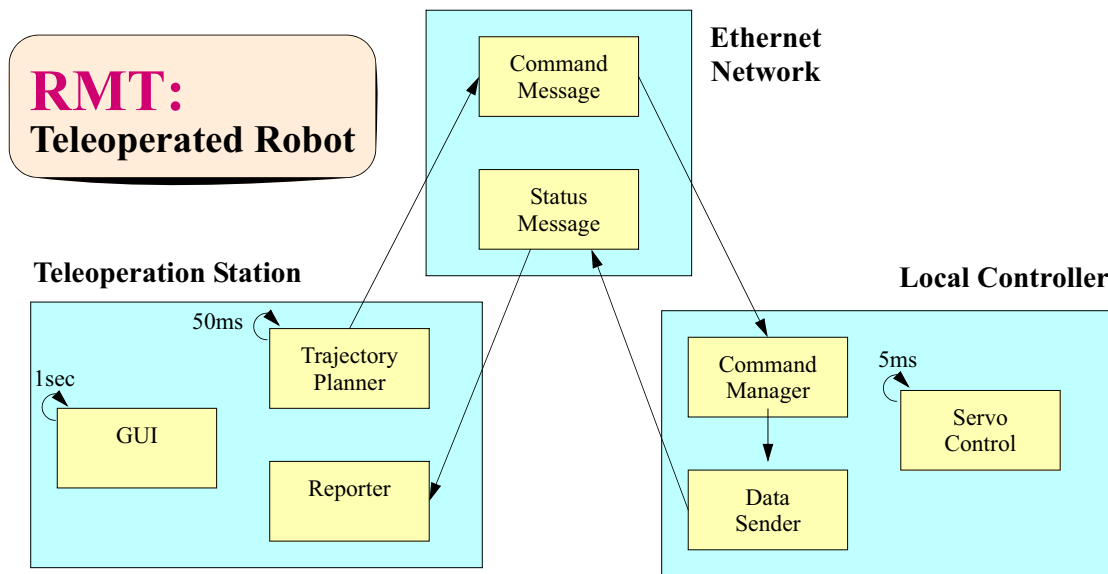
Software modules

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

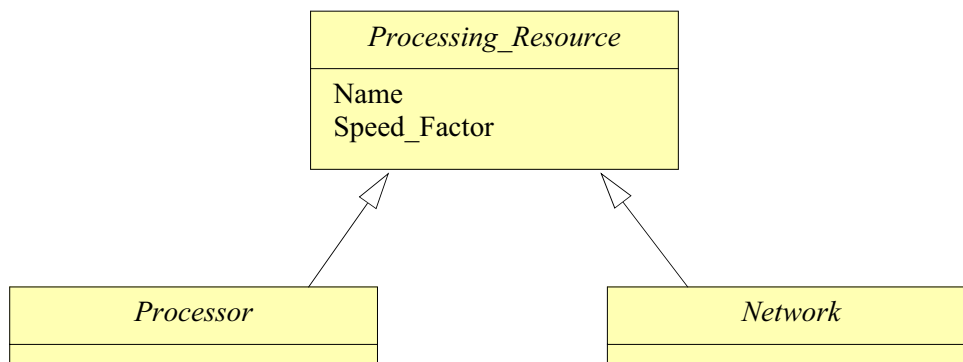
Real-time situation

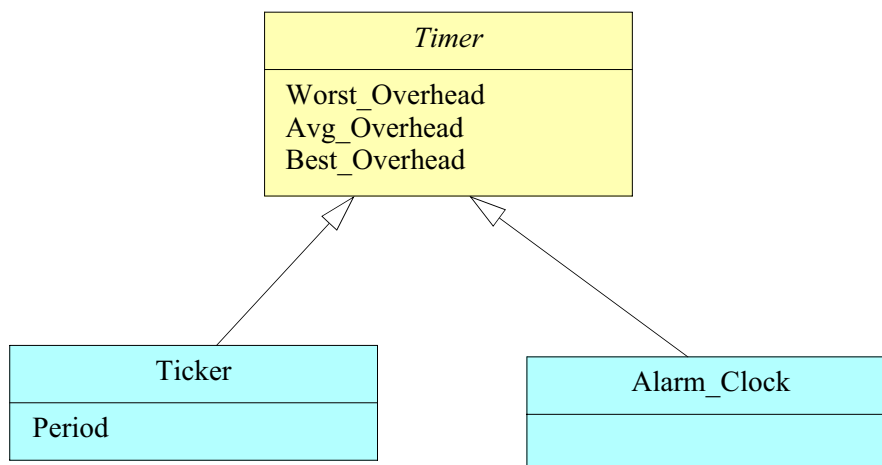
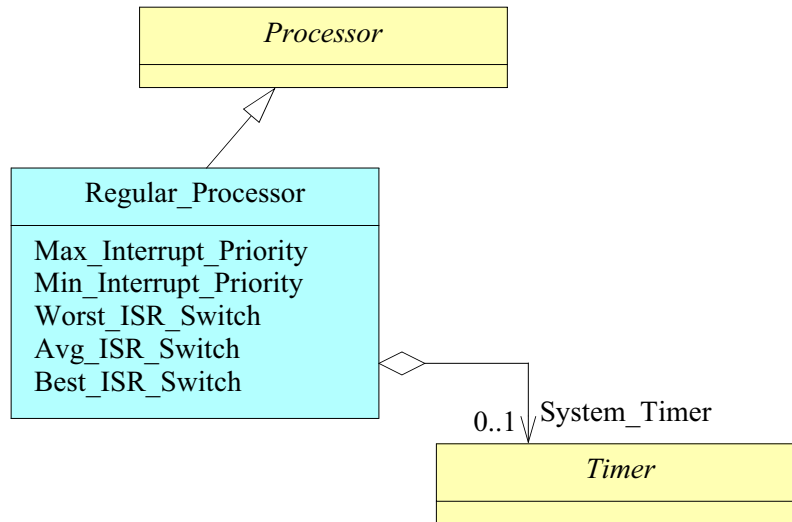
11. Events
12. Timing Requirements
13. Event Handlers
14. Transactions
15. Overall system model

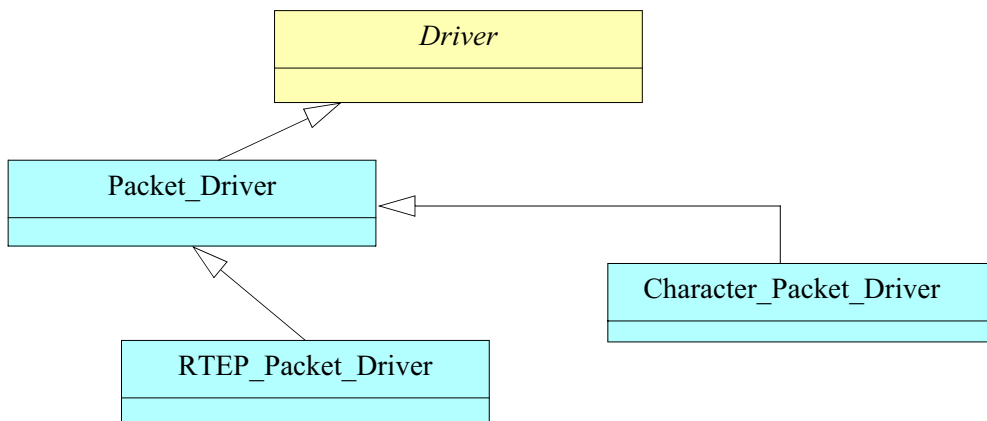
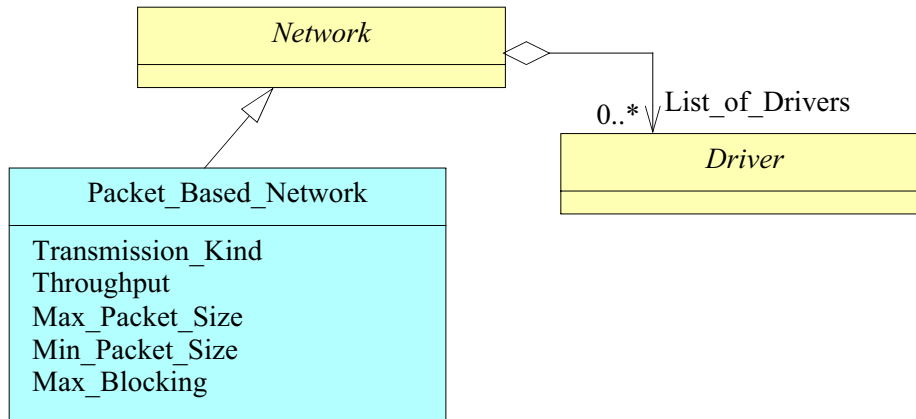
Example of modeling and analyzing with MAST



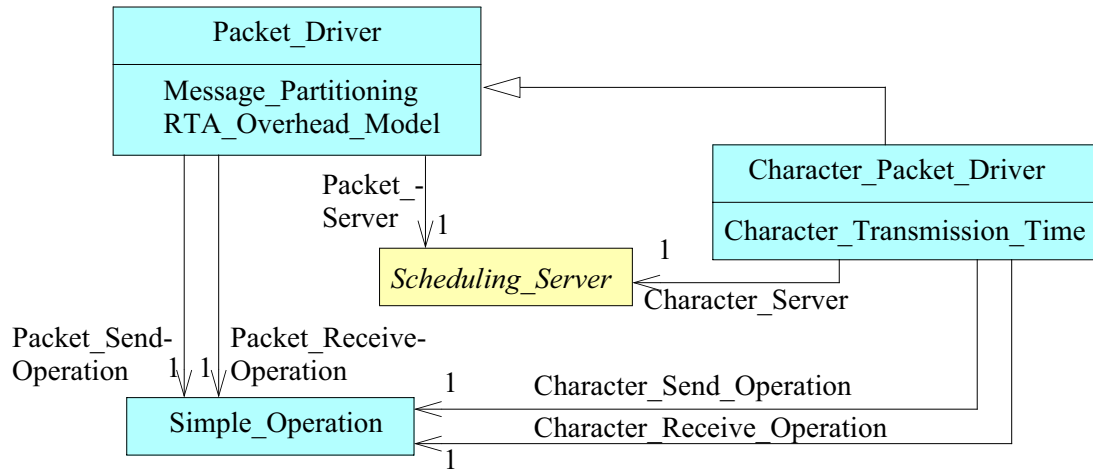
8.2 The Platform Model: Processing Resources



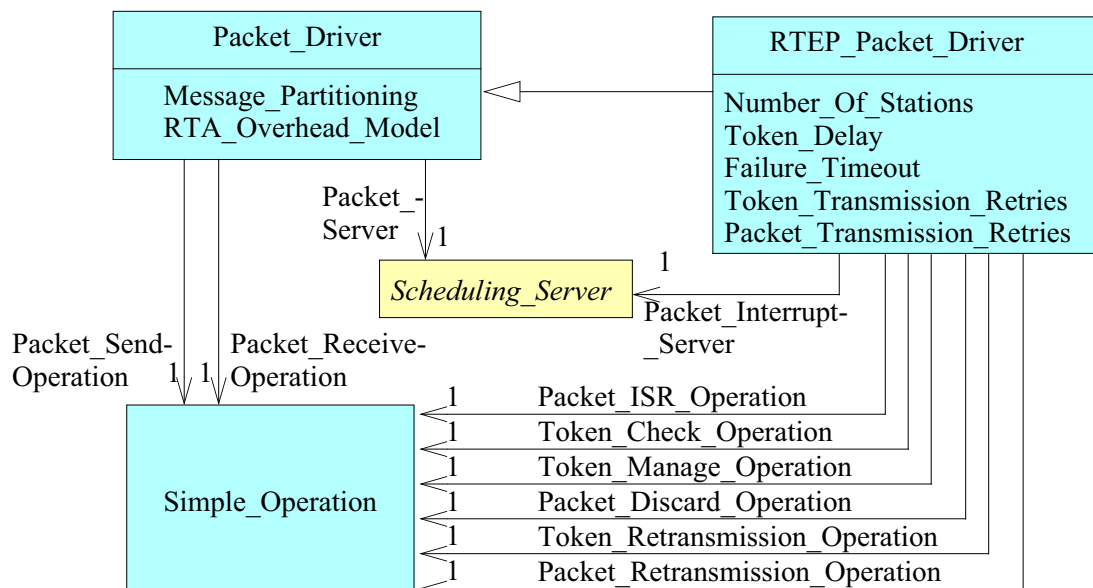




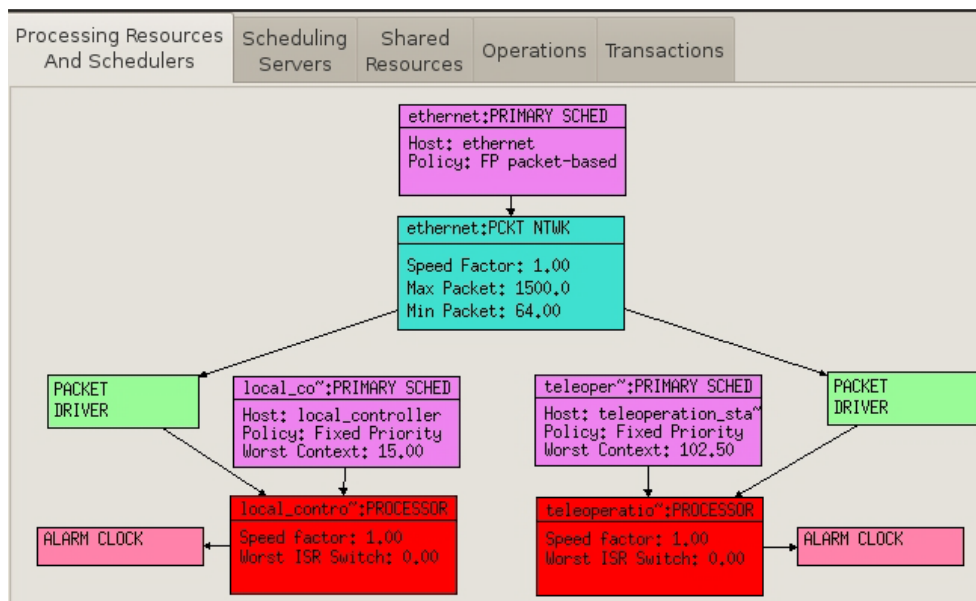
Network drivers (cont'd)



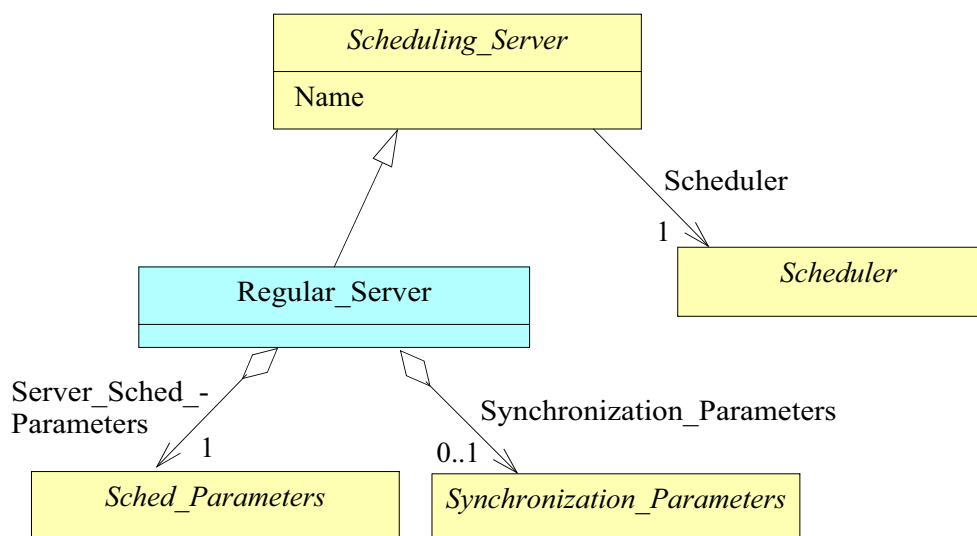
Network drivers (cont'd)

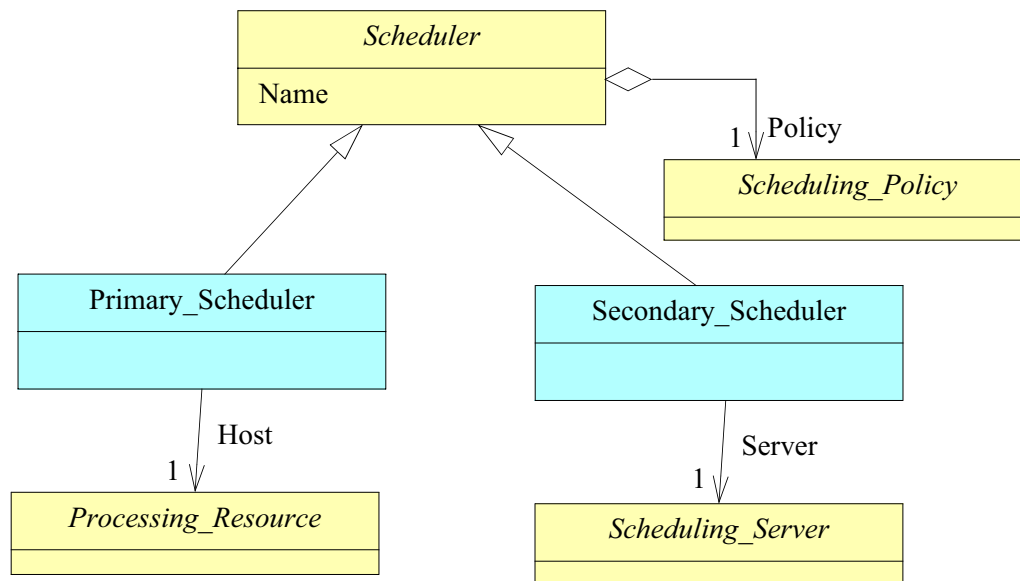


RMT: Processing resources, schedulers, drivers, and timers

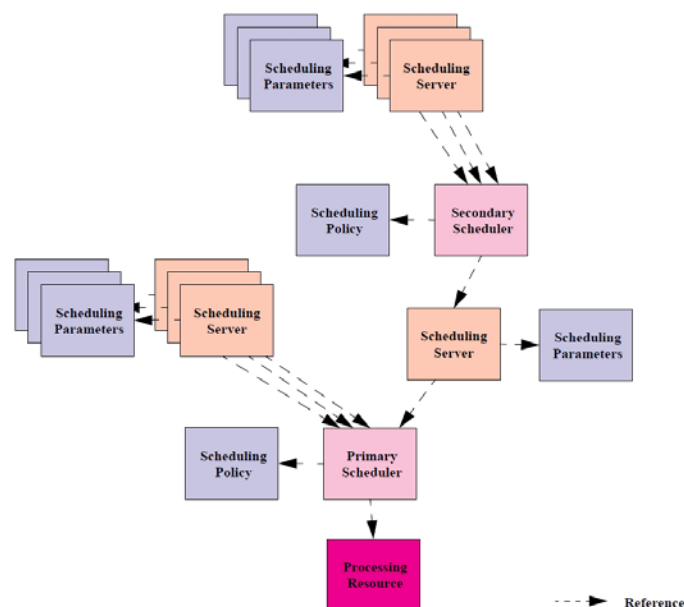


Scheduling Servers (tasks, processes, threads, ...)

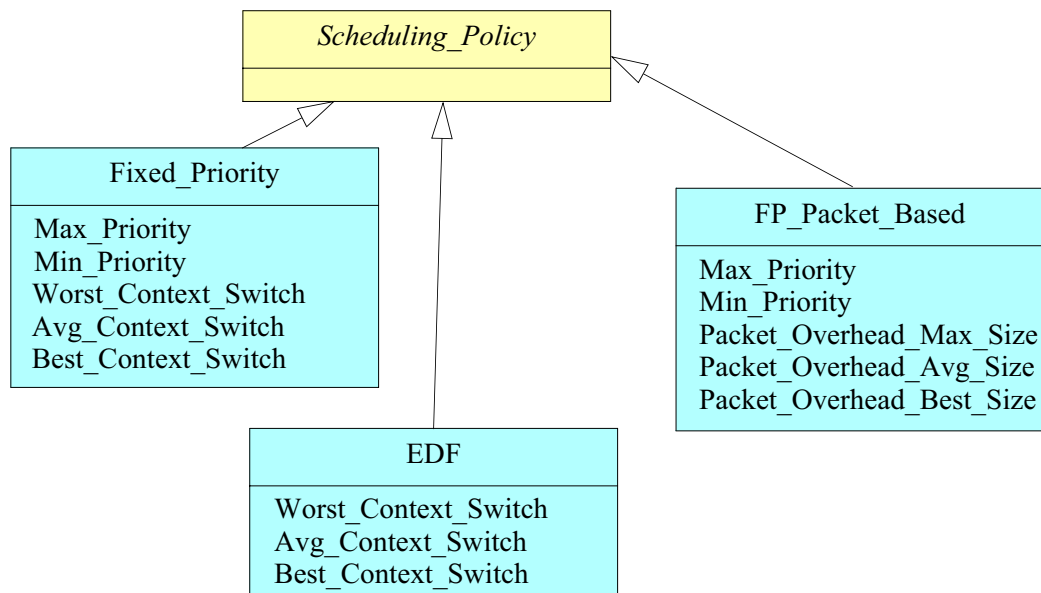




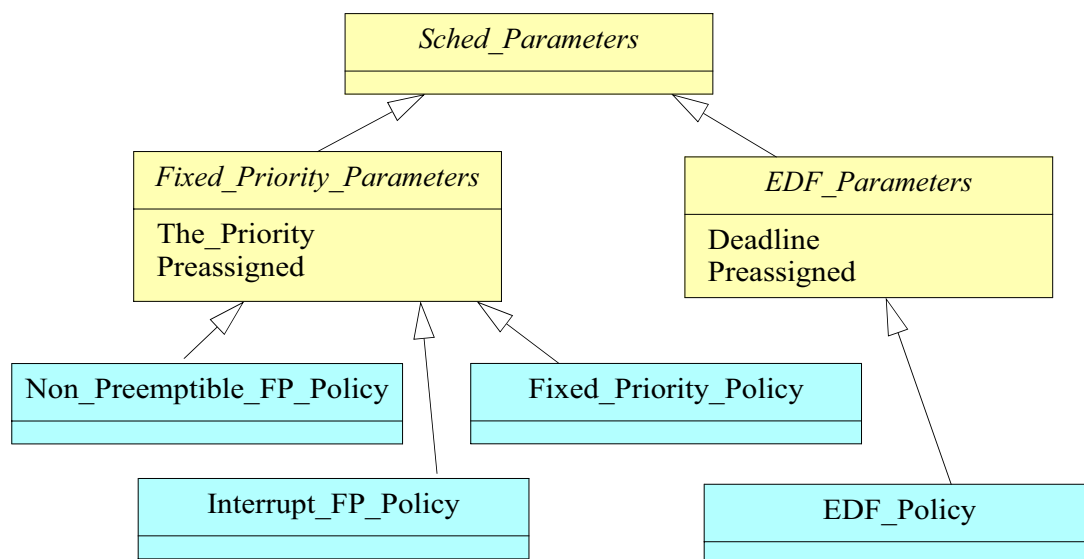
Hierarchical scheduling



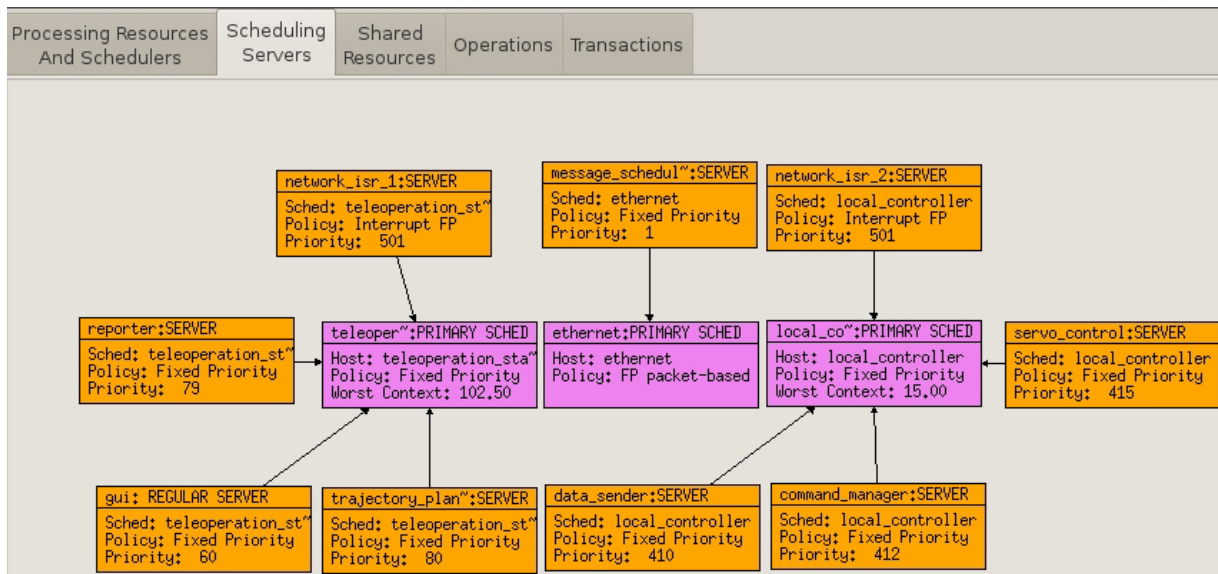
Scheduling Policies



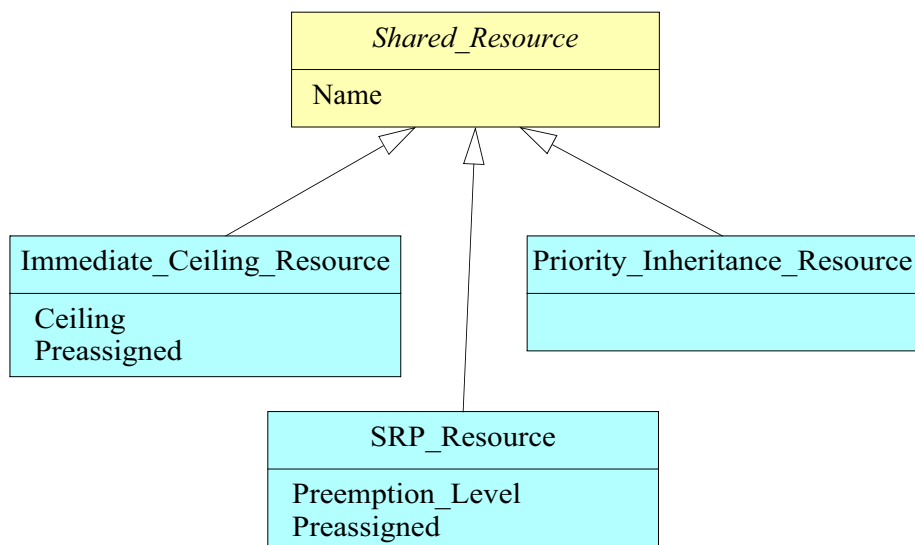
Scheduling Parameters

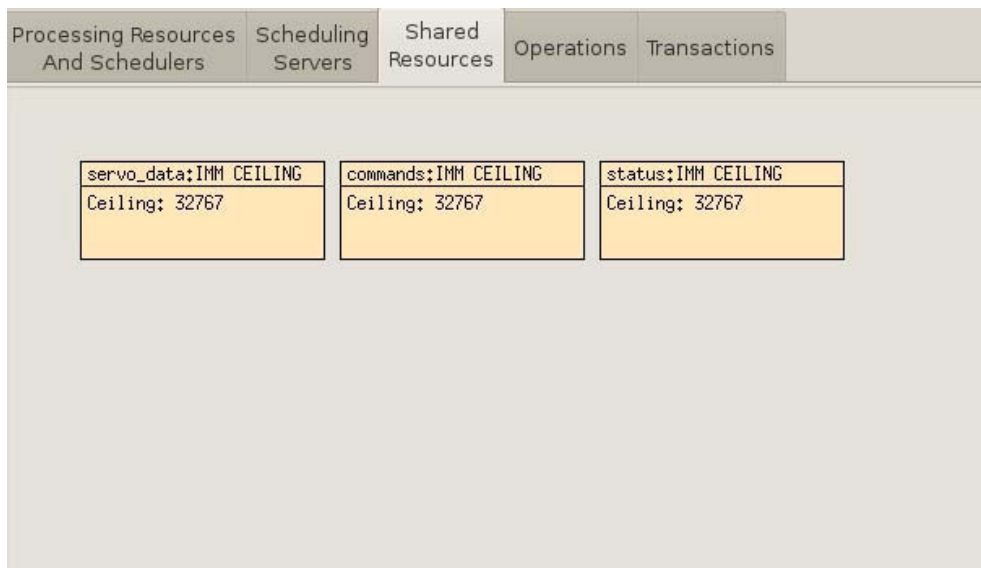


RMT: Scheduling servers

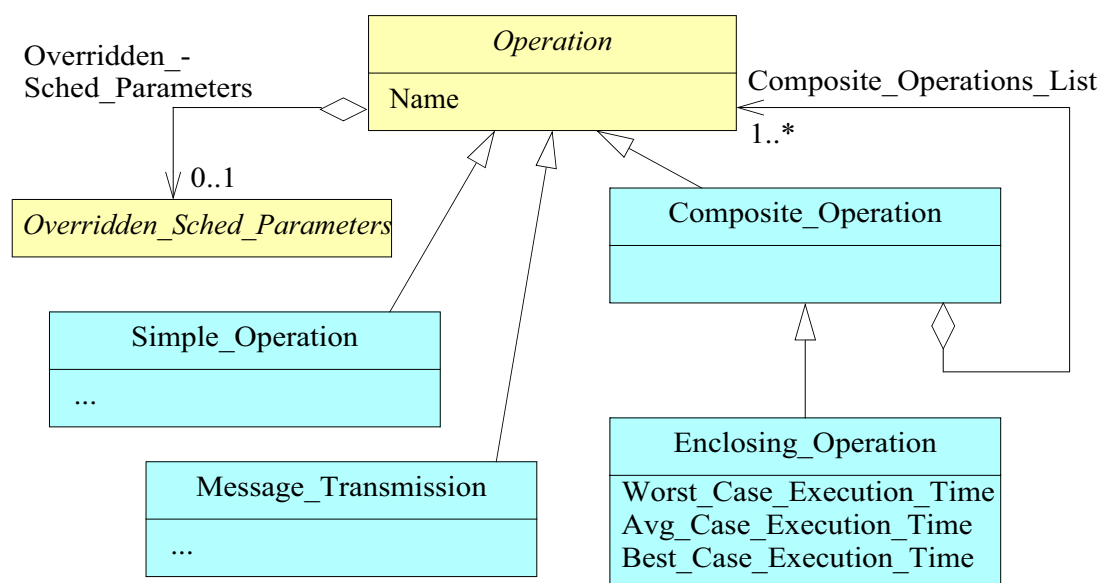


8.3 Modelling synchronization: Shared Resources (mutually exclusive)

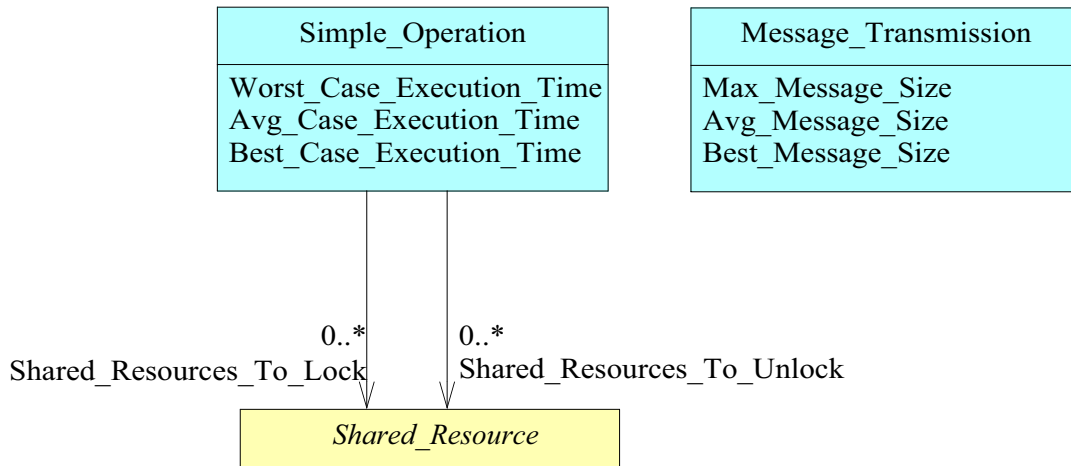




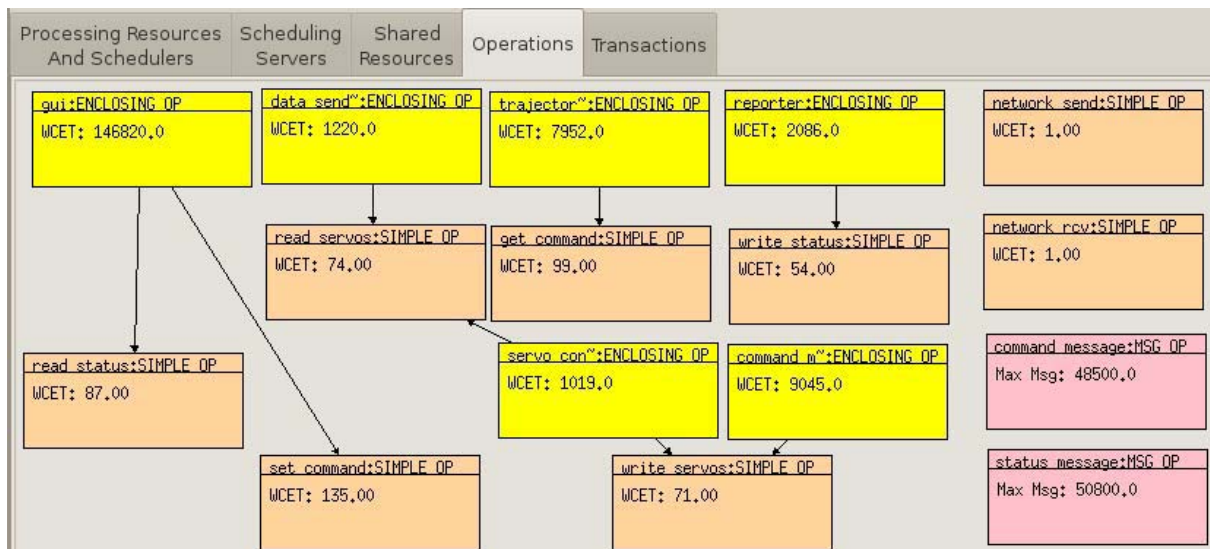
8.4 Modelling the Software Modules: Operations (functions, messages)



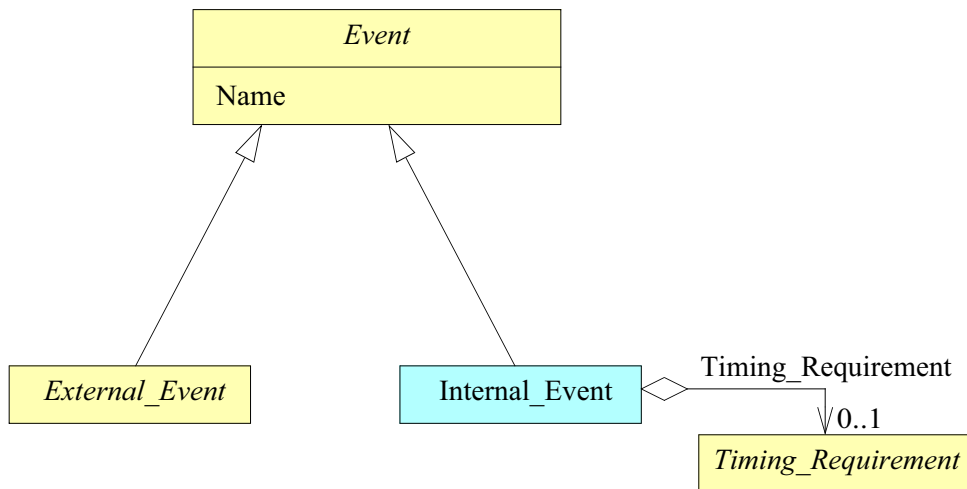
Operations (cont'd)



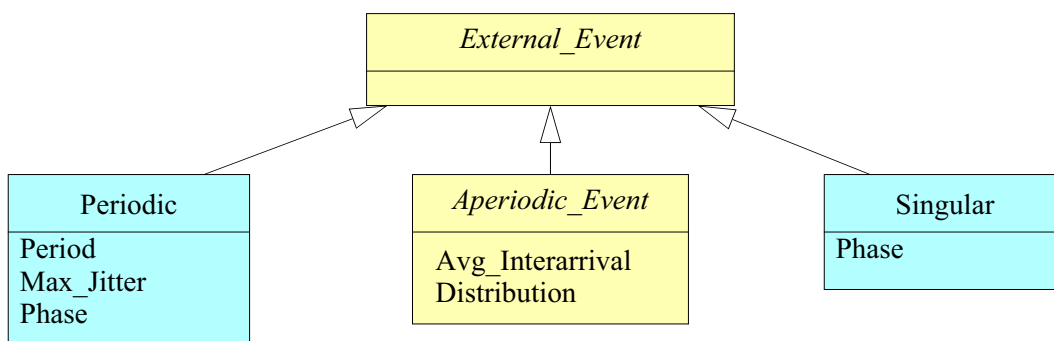
RMT: Operations



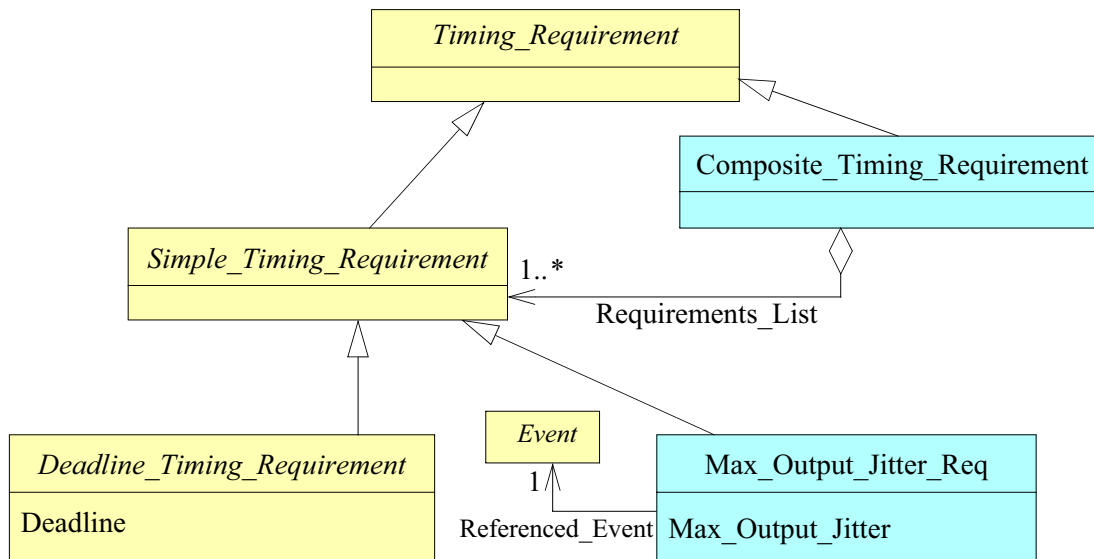
8.5 Modelling the Activities: Events



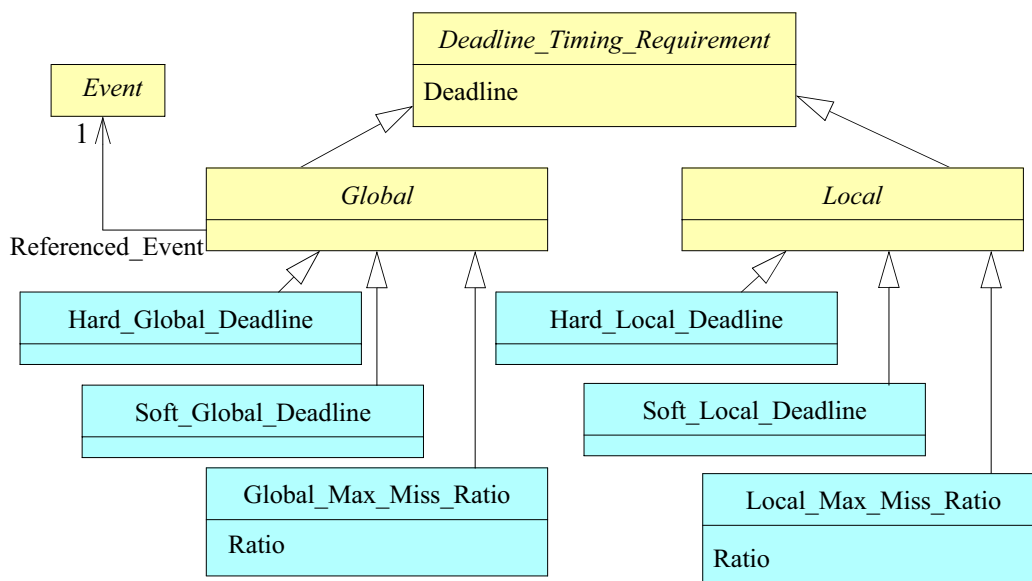
External Events



Timing Requirements



Deadline Timing Requirements

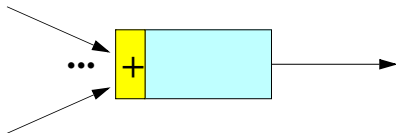


Event Handlers

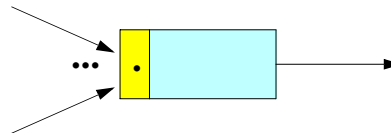
Activity / Rate Divisor / Delay / Offset



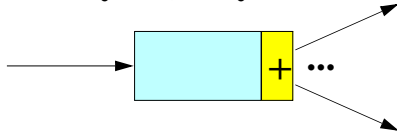
Concentrator



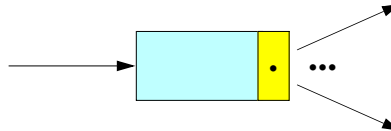
Barrier



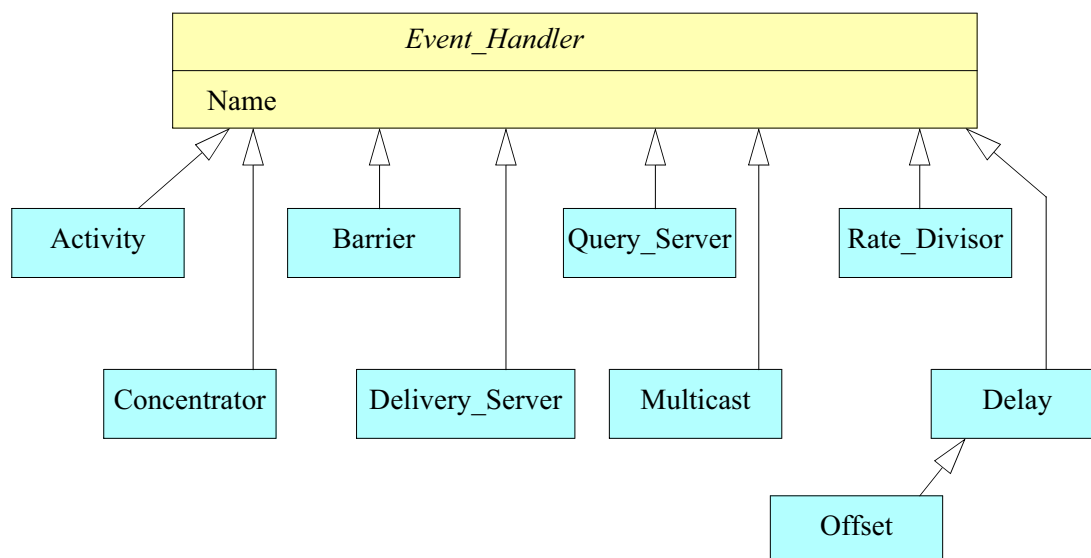
Delivery / Query Server



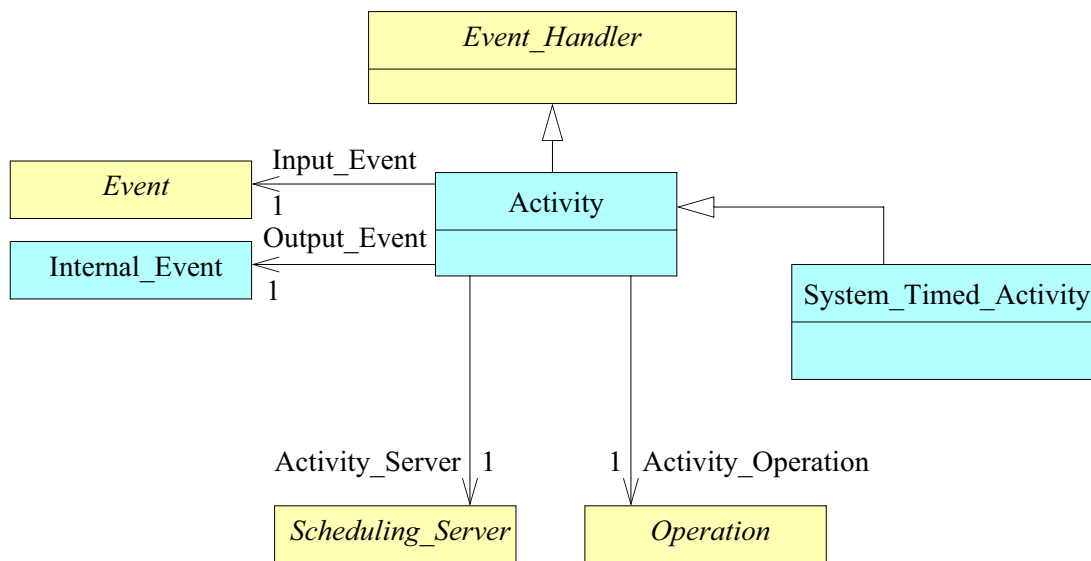
Multicast



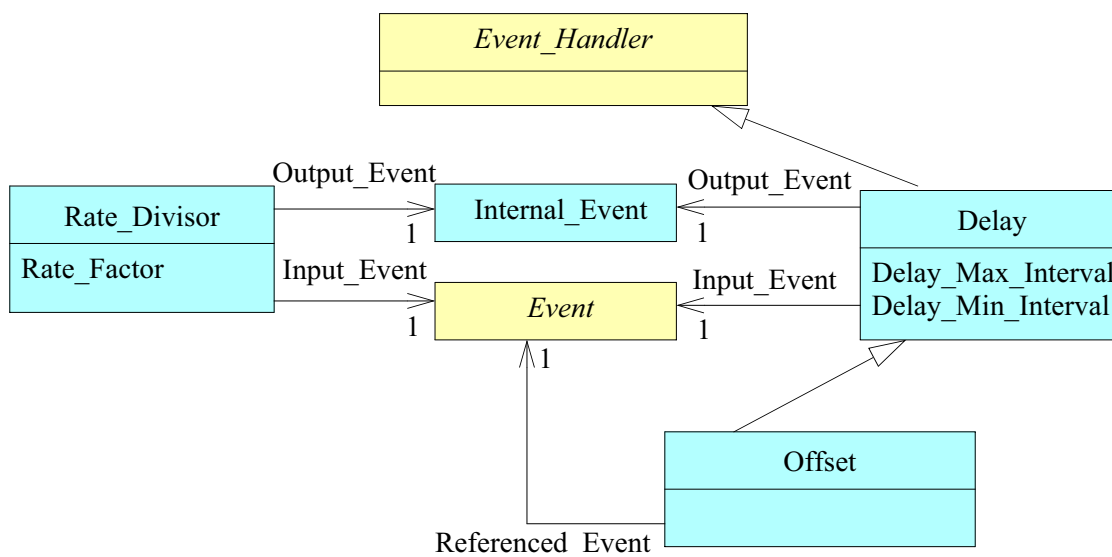
Event Handlers (cont'd)

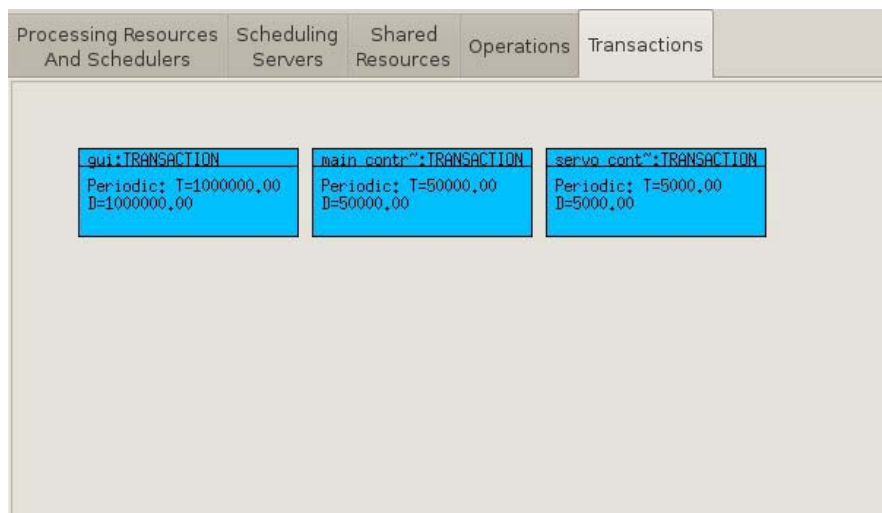
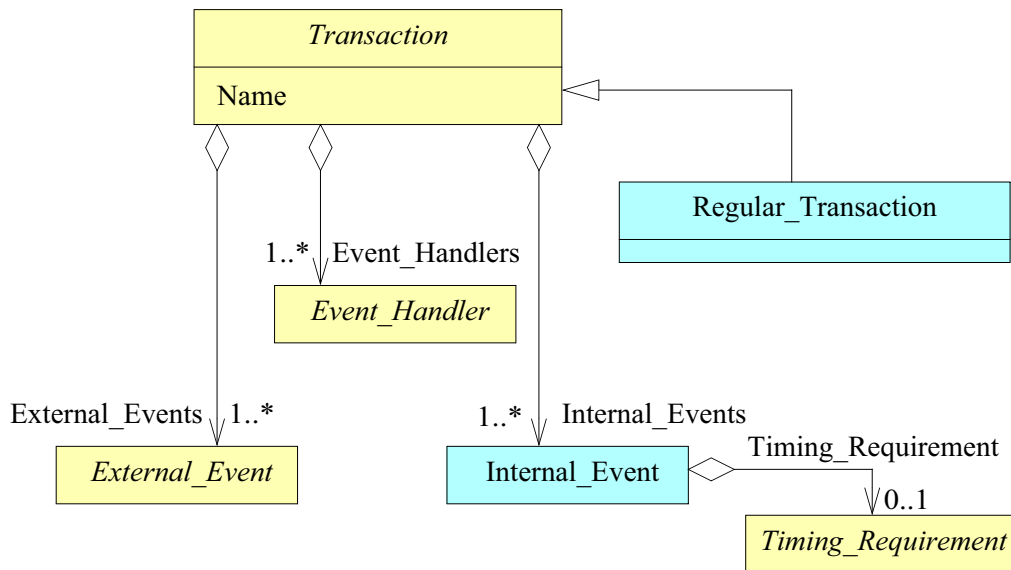


Activities

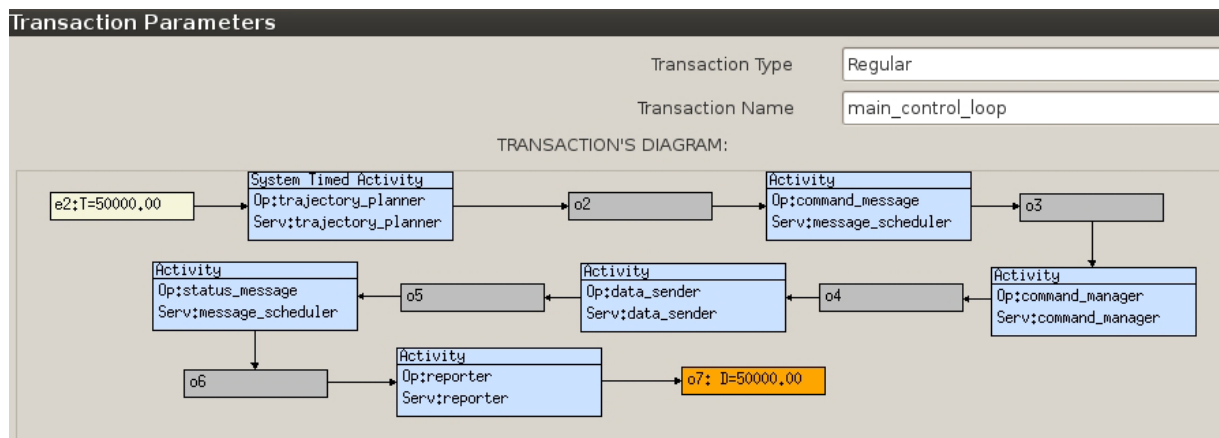


Delays and Rate Divisors





RMT: Transaction graphs



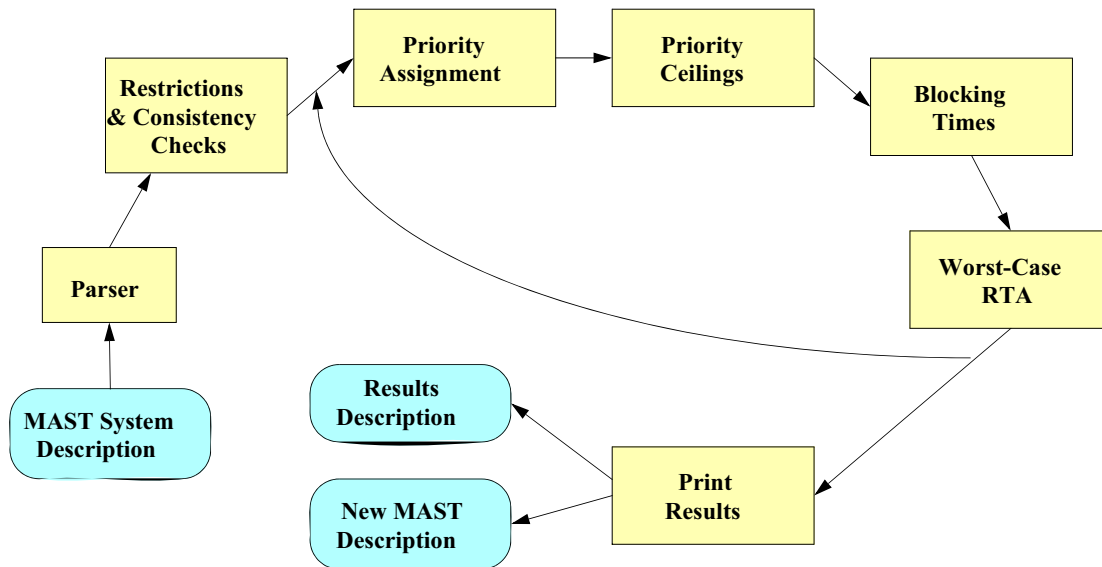
8.6 Analysis Tools: Specification Language

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

Schedulability Analysis Tools



Fixed Priority Response-Time Analysis

Technique	Single-Processor	Multi-Processor	Simple Transact.	Linear Transact.	Multiple Event T.
Classic Rate Monotonic	✓		✓		
Varying Priorities	✓		✓	✓	
Holistic	✓	✓	✓	✓	
Offset Based Unoptimized	✓	✓	✓	✓	
Offset Based	✓	✓	✓	✓	
Multiple Event	✓	✓	✓	✓	✓

8.7 Distributed Systems

Linear Action: $e_{j-1,j} \rightarrow a_j \rightarrow e_{j,j+1} \quad T_{j-1,j} = T_j = T_{j,j+1}$

Linear Response to an Event:

