

Real-time Software Systems

- Overview of concerns
 - Important UML notations and their role in R-T
 - Diagrams important for R-T system design
- UML behavioural notations, and main roles:
- Interaction diagrams vs. Statecharts
 - Statechart Notation

Real-time Systems

Terminology:

- Real-time
- Embedded
- Reactive

...

Systems are often real-time, embedded and reactive.

Real-time Systems

Real-time System: software system where the correct functioning of the system depends on the results produced by the system and the time at which these results are produced.

- Soft Real-time System:
 - a system whose operation is **degraded** if results are not produced according to the specified timing requirements.
- Hard Real-time System:
 - a system whose operation is **incorrect** if results are not produced according to the specified timing requirements.

Implementing Real-Time Software Systems

- A range of software of Real-Time systems from the very large, e.g. telecoms, to very small, e.g. a wireless medical sensor
- Previous, implementation directly in assembler and C (and associated concerns) being replaced by graphical notations, such as Statecharts notation of UML, and associated tools for simulation, debugging, code generation etc.

Implementing Real-time systems

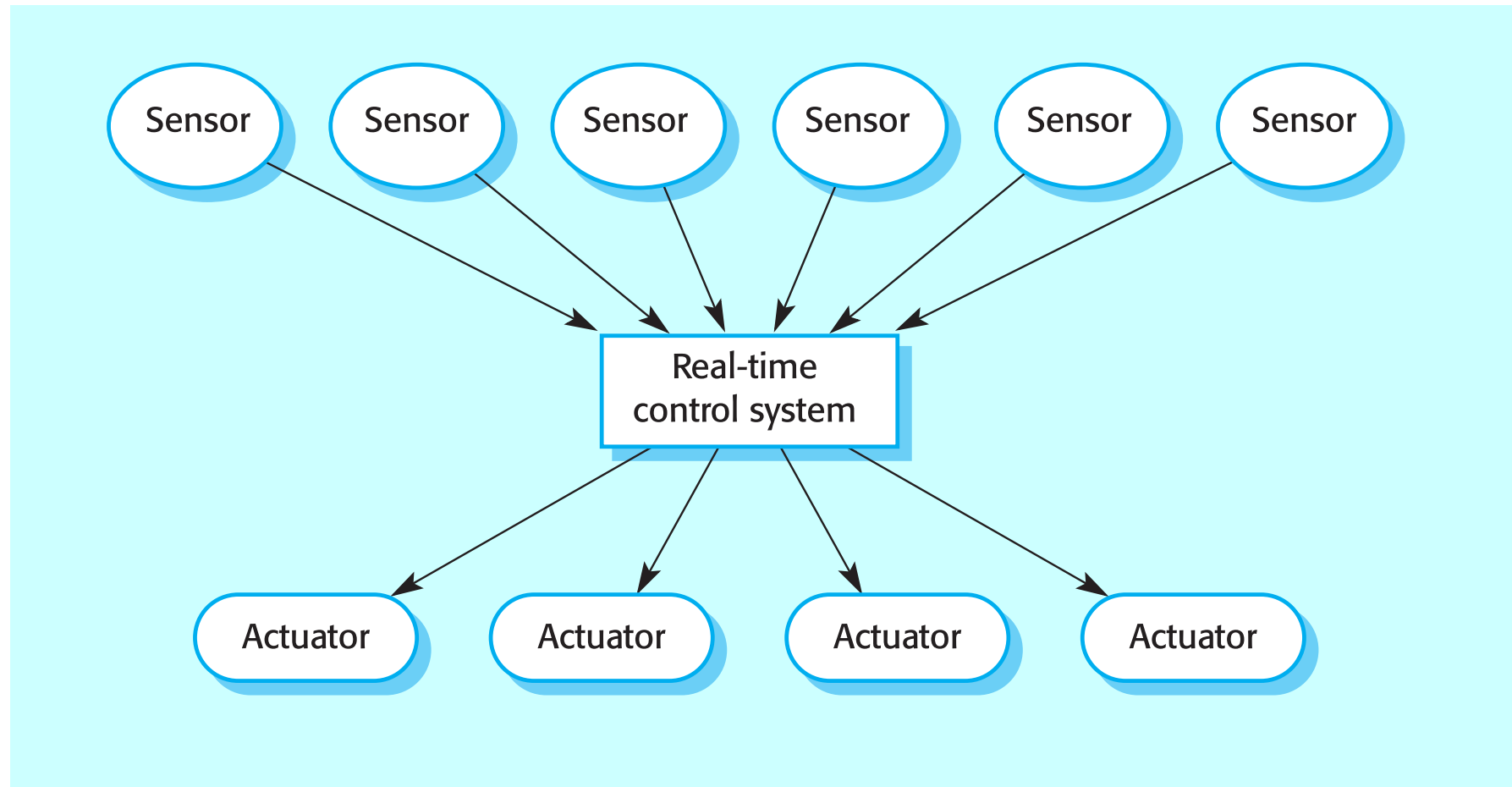
- R-T systems often monitor and control the environment
- Involves hardware devices:
 - Sensors: collect data from the environment
 - Actuators: activate a device, changing the environment

Stimulus/Response Systems

Given a stimulus, system must respond within specified time

- Periodic stimuli: occur at predictable time intervals
 - Example: a room temperature sensor polled every 10 seconds
- Aperiodic stimuli: occur at unpredictable times
 - Example: nuclear power plant overheating alert

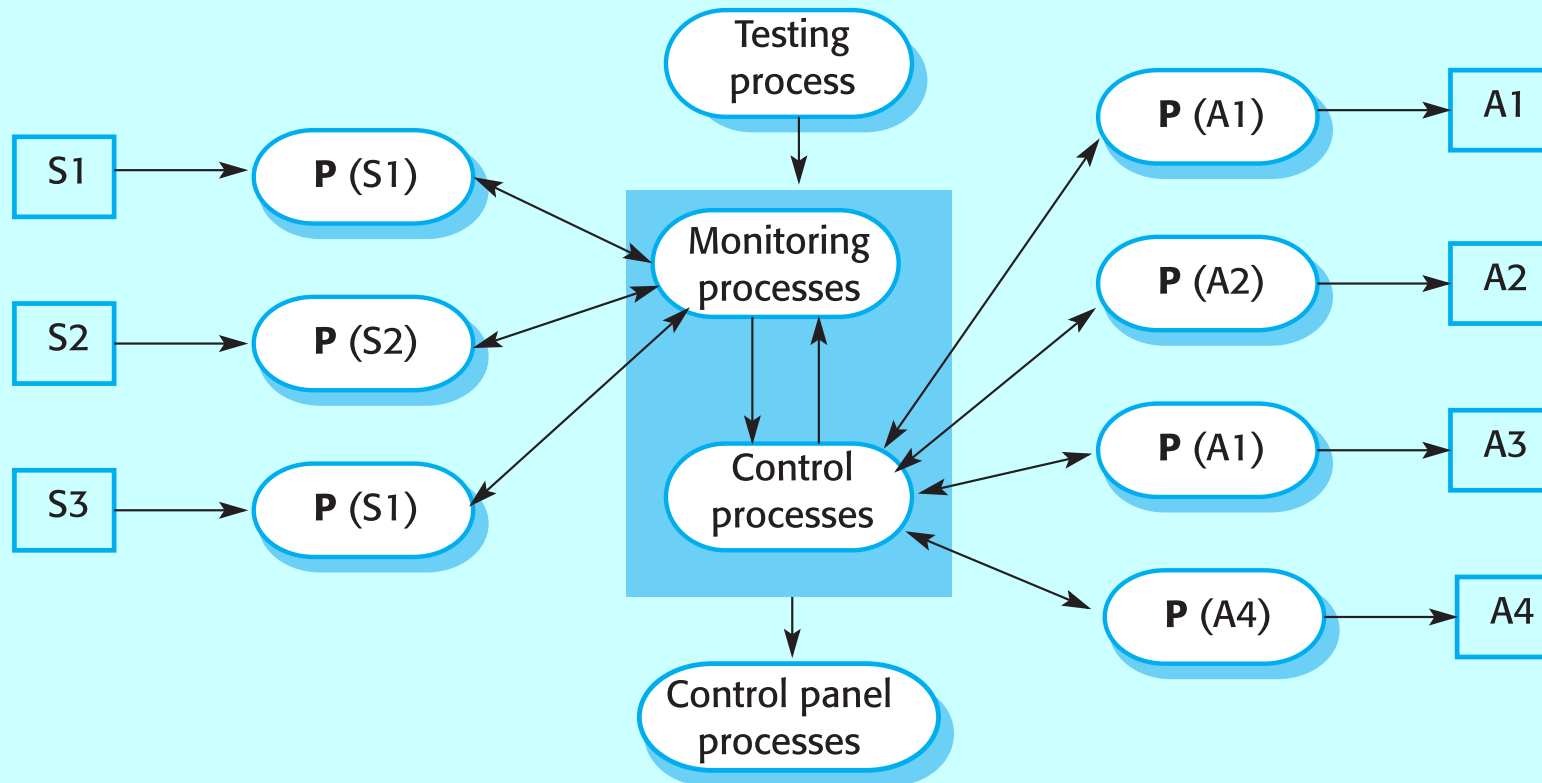
A real-time control system model (Sommerville)



Real-Time Software System

- Timing demands often result in programming in assembler or C
 - Need for concurrent processes
 - Need for scheduling of processes
 - Need for process priorities
 - Need to meet timing deadlines
- > Real-time operating systems underlying most real-time systems

Generic monitoring-control architecture (Sommerville)



System elements (Sommerville)

- Sensor control processes
 - Collect information from sensors. May buffer information collected in response to a sensor stimulus.
- Data processor
 - Carries out processing of collected information and computes the system response.
- Actuator control processes
 - Generates control signals for the actuators.

R-T Operating system components (Sommerville)

- Real-time clock
 - Provides information for process scheduling.
- Interrupt handler
 - Manages aperiodic requests for service.
- Scheduler
 - Chooses the next process to be run.
- Resource manager
 - Allocates memory and processor resources.
- Dispatcher
 - Starts process execution.

R-T OS components (Sommerville)

