

Chapter 7

Real-time Software Engineering

By Yohannes S.

Contents

❧ Introduction

- ❧ Definition, Characteristics of real-time system

❧ Embedded system design

- ❧ Real-time system modelling

- ❧ Real-time system Programming

❧ Architectural patterns for real-time software

- ❧ Observe and React

- ❧ Environmental Control

- ❧ Process Pipeline

❧ Timing analysis

❧ Real-time operating systems

Introduction

- ❧ Computers are used to control a wide range of systems from simple domestic machines, through games controllers, to entire manufacturing plants.
- ❧ Their software must react to events generated by the hardware and, often, issue control signals in response to these events.
- ❧ The software in these systems is embedded in system hardware, often in read-only memory, and usually responds, in real time, to events from the system's environment
- ❧ Embedded software is very important economically because almost every electrical device now includes software

Introduction...

- ✧ Responsiveness in real-time is the critical difference between embedded systems and other software systems, such as information systems, web-based systems or personal software systems
- ✧ For non-real-time systems, **correctness** can be defined by specifying how system inputs map to corresponding outputs that should be produced by the system.
- ✧ In a real-time system, the correctness depends both on the response to an input and the time taken to generate that response. If the system takes too long to respond, then the required response may be ineffective.

Definition

- ✧ A *real-time system* is a 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.
- ✧ A *soft real-time system* is a system whose operation is degraded if results are not produced according to the specified timing requirements.
 - ✧ are those in which the failure to meet a deadline reduces the utility of the service E. g. letter sorting machine
- ✧ A *hard real-time system* is a system whose operation is incorrect if results are not produced according to the timing specification.
 - ✧ are those in which the failure to meet a

Characteristics of embedded systems

- ✧ Embedded systems generally run continuously and do not terminate
 - ✧ Techniques for reliable software engineering may have to be used to ensure continuous operation
- ✧ Interactions with the system's environment are unpredictable
 - ✧ real-time embedded systems must be able to respond to unexpected events at any time.
- ✧ There may be physical limitations that affect the design of a system
 - ✧ conserve power, size, weight, ...
- ✧ Direct hardware interaction may be necessary
- ✧ Issues of safety and reliability may dominate the system design

Embedded system design

- ✧ The design process for embedded systems is a systems engineering process that has to consider, in detail, the design and performance of the system hardware.
- ✧ Part of the design process may involve deciding which system capabilities are to be implemented in software and which in hardware.
- ✧ Low-level decisions on hardware, support software and system timing must be considered early in the process.
- ✧ These may mean that additional software functionality, such as battery and power management, has to be included in the system.

Embedded system design...

- ❧ Real-time systems are often considered to *be reactive systems*
- ❧ Given a stimulus, the system must produce a reaction or response within a specified time.
- ❧ *Periodic stimuli* - Stimuli which occur at predictable time intervals
 - ❧ For example, a temperature sensor may be polled 10 times per second.
- ❧ *Aperiodic stimuli* - Stimuli which occur at unpredictable times
 - ❧ For example, a system power failure may trigger an interrupt which must be processed by the system
- ❧ Stimuli come from sensors in the systems

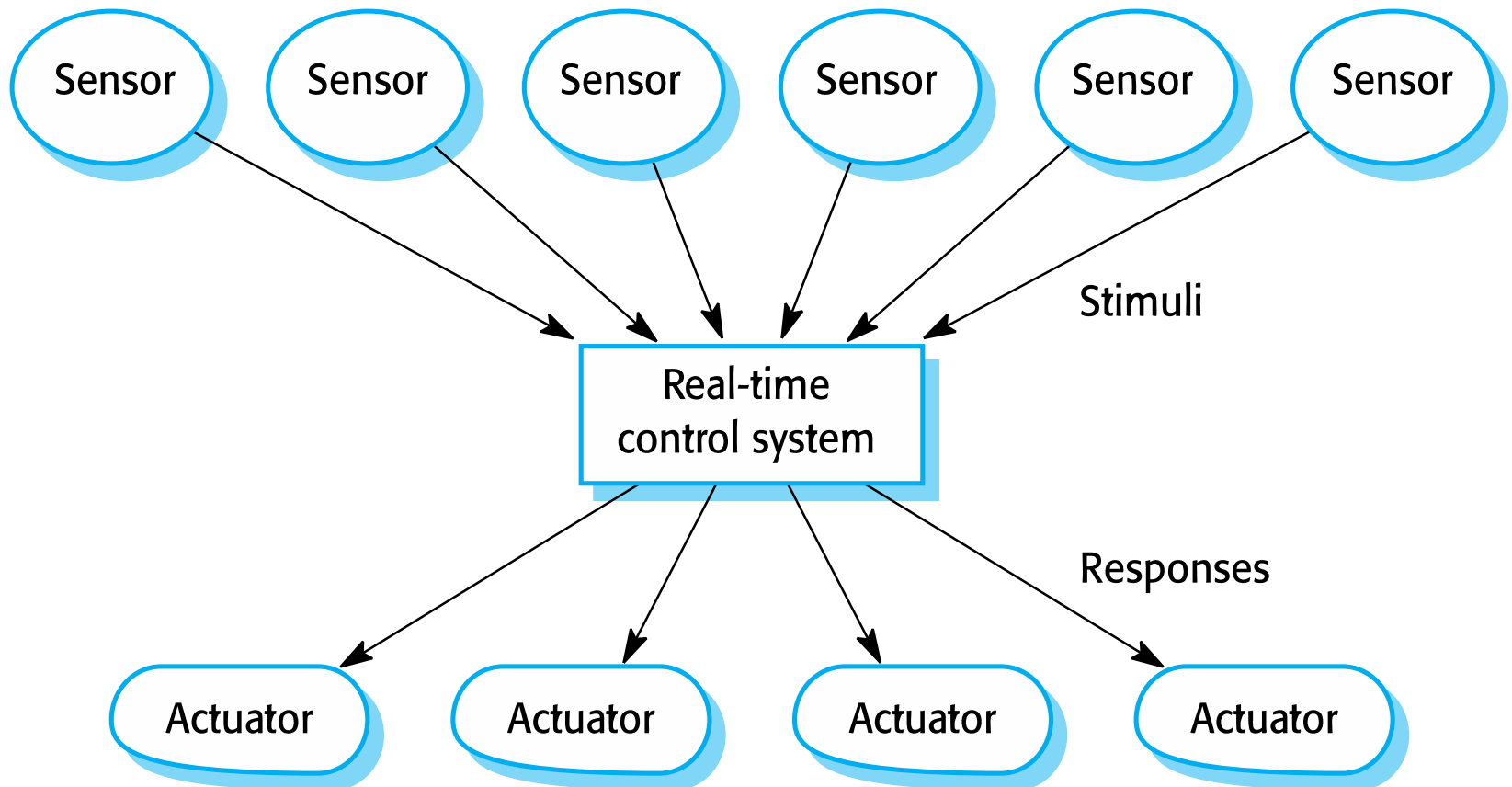
Embedded system design...

Stimuli and responses for a burglar alarm

Stimulus	Response
Clear alarms	Switch off all active alarms; switch off all lights that have been switched on.
Console panic button positive	Initiate alarm; turn on lights around console; call police.
Power supply failure	Call service technician.
Sensor failure	Call service technician.
Single sensor positive	Initiate alarm; turn on lights around site of positive sensor.
Two or more sensors positive	Initiate alarm; turn on lights around sites of positive sensors; call police with location of suspected break-in.
Voltage drop of between 10% and 20%	Switch to battery backup; run power supply test.

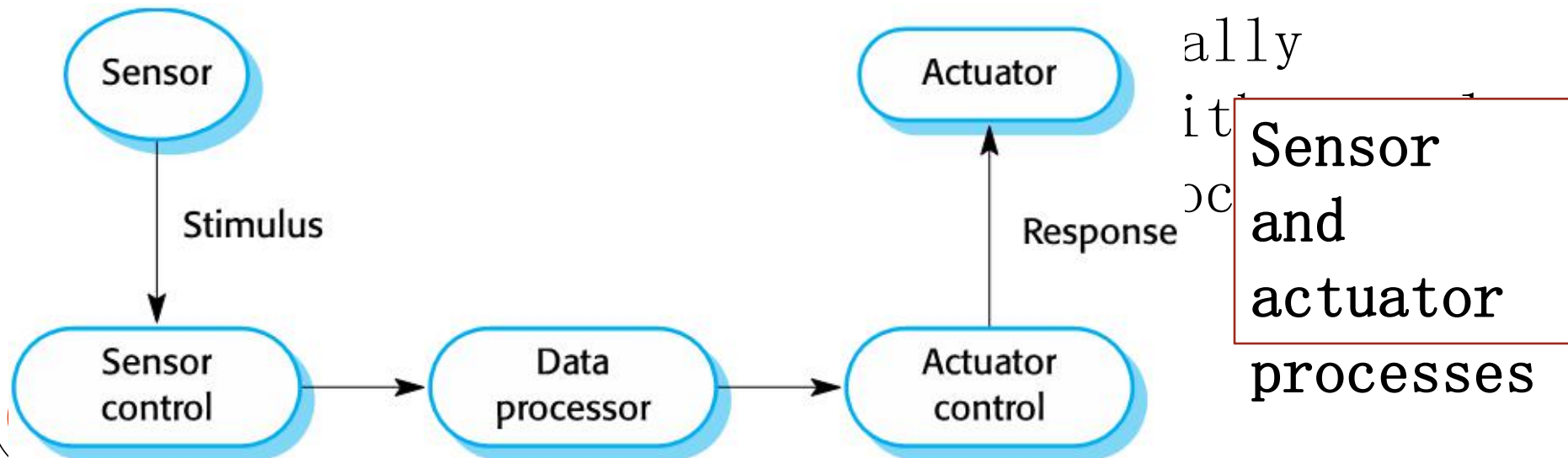
Embedded system design...

A general model of an embedded real-time system



Embedded system design...

- Because of the need to respond to timing demands made by different stimuli/responses, the **system architecture** must allow for fast switching between stimulus handlers
- Timing demands of different stimuli are different so a simple sequential loop is not usually adequate.



Embedded system design...

✧ System elements

✧ 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

Embedded system design...

- ✧ There is no standard embedded system design process. Rather, different processes are used that depend on
 - ✧ The type of system, available hardware, and the organization that is developing the system.
- ✧ The activities that may be included in a real-time software design process:
 - ✧ *Platform selection* – hardware and the real-time operating system are selected. The choice depends on timing constraints on the system, limitations on power available, the experience of the development team, and the price target for the delivered system.
 - ✧ *Stimuli/response identification* – identify

Embedded system design...

- ❧ *Process design* – aggregate the stimulus and response processing into a number of concurrent processes. Architectural patterns are good starting points
- ❧ *Algorithm design* – For each stimulus and response, design algorithms to carry out the required computations.
- ❧ *Data design* – specify the information that is exchanged by processes and the events that coordinate information exchange, and design data structures to manage this information exchange. Several concurrent processes may share these data structures.
- ❧ *Process scheduling* – design a scheduling system that will ensure that processes are started in

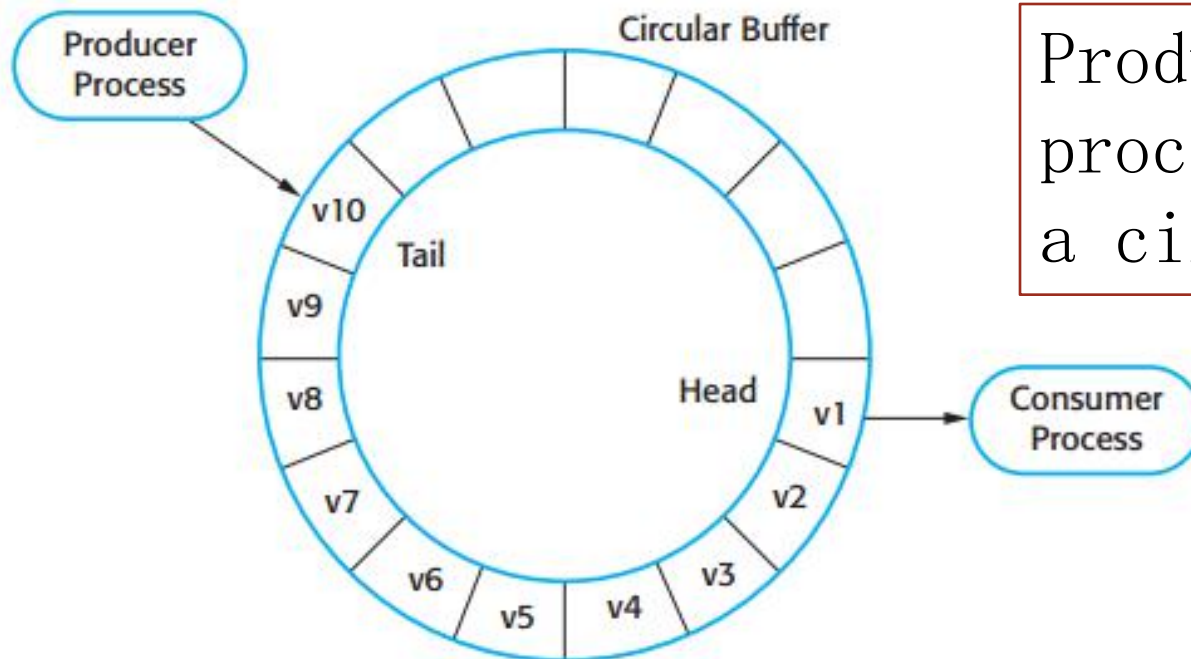
Embedded system design...

✂ Process coordination and Mutual exclusion

- ✂ Processes in a real-time system have to be coordinated and share information.
- ✂ Process coordination mechanisms ensure mutual exclusion to shared resources.
- ✂ When one process is modifying a shared resource, other processes should not be able to change that resource.
- ✂ When designing the information exchange between processes, you have to take into account the fact that these processes may be running at different speeds
- ✂ Producer processes collect data and add it to the buffer. Consumer processes take data from

Embedded system design...

- ⌘ Producer and consumer processes must be mutually excluded from accessing the same element
- ⌘ **Mechanisms:** Semaphores, monitors, Critical Regions
- ⌘ The buffer must stop producer processes adding information to a full buffer and consumer processes trying to take information from an empty buffer

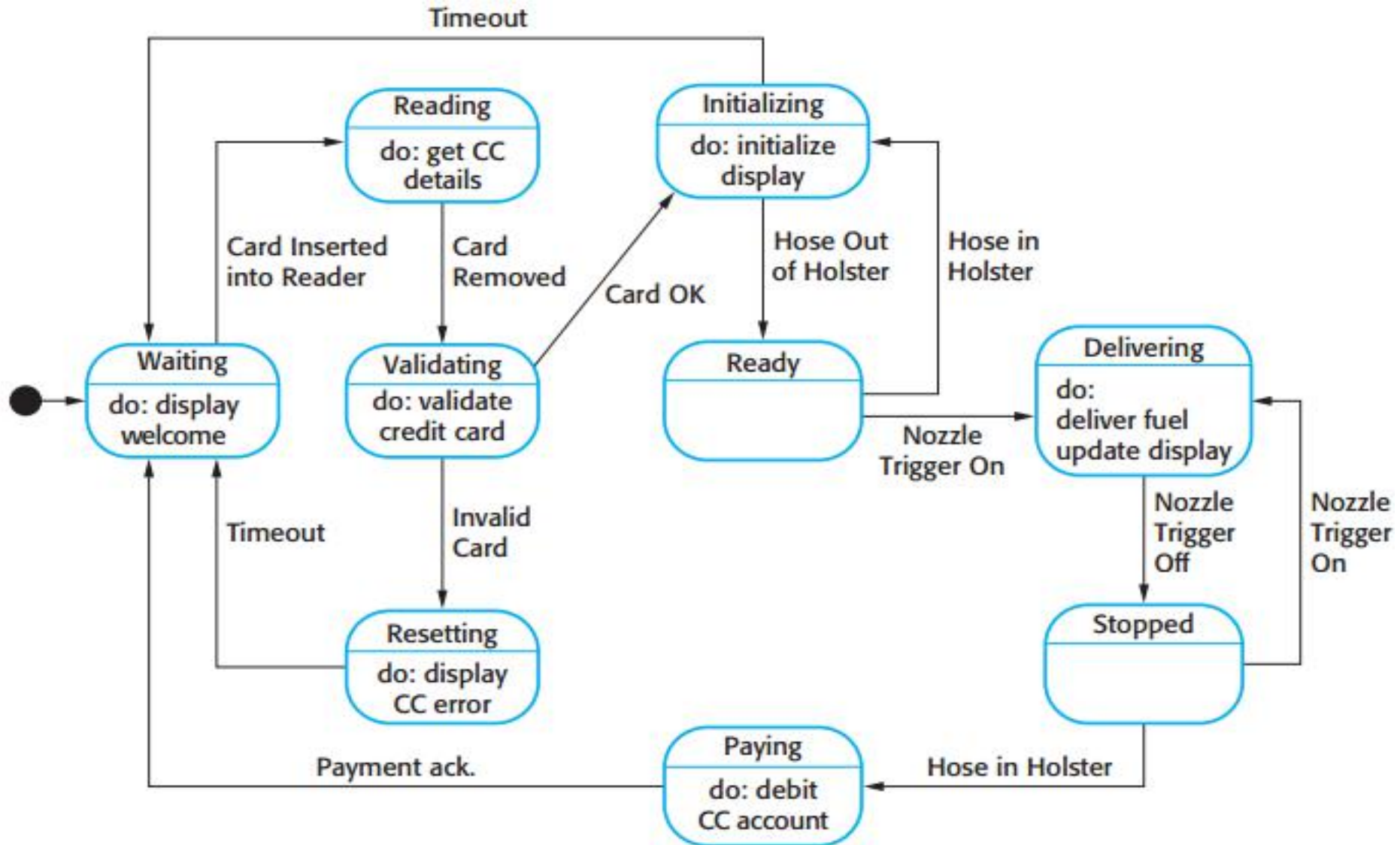


Producer/consumer processes sharing a circular buffer

Real-time system modelling

- ✧ The effect of a stimulus in a real-time system may trigger a transition from one state to another.
- ✧ For example, a system controlling a valve may move from a state 'Valve open' to a state 'Valve closed' when an operator command (the stimulus) is received.
- ✧ State models are therefore often used to describe embedded real-time systems.
- ✧ UML state diagrams may be used to show the states and state transitions in a real-time system.

Real-time system modelling...



Real-time system modelling...

- Sequence of actions in real-time pump control system
 - The buyer inserts a credit card into a card reader built into the pump.
 - Removal of the card triggers a transition to a Validating state where the card is validated.
 - If the card is valid, the system initializes the pump and, when the fuel hose is removed from its holster, transitions to the Delivering state.
 - After the fuel delivery is complete and the hose replaced in its holster, the system moves to a Paying state.
 - After payment, the pump software returns to the Waiting state

Real-time programming

- ❧ Programming languages for real-time systems development have to include facilities to access system hardware, and it should be possible to predict the timing of particular operations in these languages.
- ❧ Systems-level languages, such as C, which allow efficient code to be generated are widely used in preference to languages such as Java.
- ❧ There is a performance overhead in object-oriented systems because extra code is required to mediate access to attributes and handle calls to operations.

Architectural patterns for embedded systems

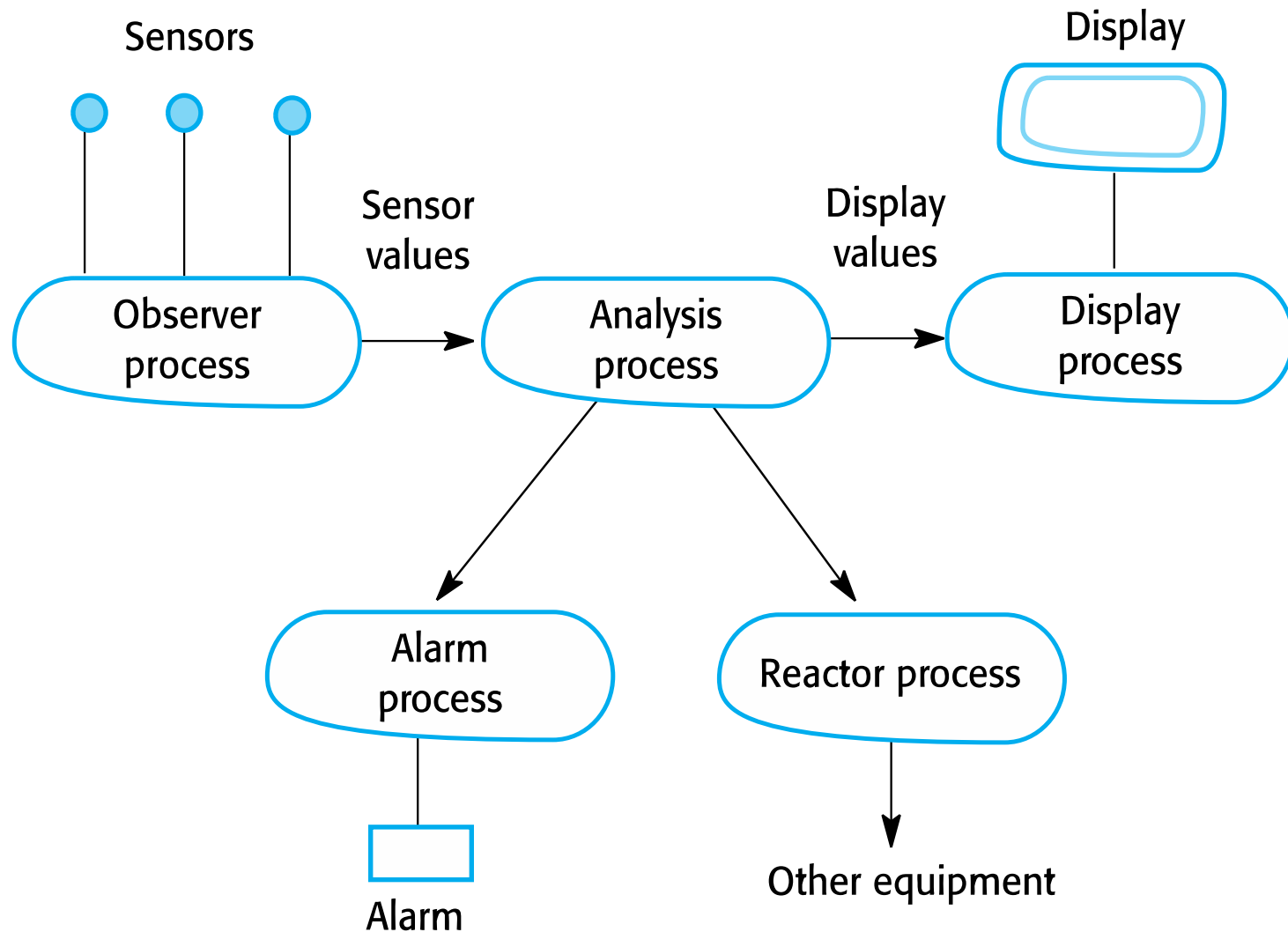
- ✧ Architectural patterns are abstract, stylized descriptions of good design practice.
- ✧ architectural pattern are not generic design to be instantiated.
- ✧ Rather, it can be used to understand an architecture and as starting point for creating your own specific architectural design.
- ✧ Three real-time architectural patterns are seen here
 - ✧ Observe and React
 - ✧ Environmental Control
 - ✧ Process Pipeline

The Observe and React pattern

This pattern is used when a set of sensors are routinely monitored and displayed.

Name	Observe and React
Description	The input values of a set of sensors of the same types are collected and analyzed. These values are displayed in some way. If the sensor values indicate that some exceptional condition has arisen, then actions are initiated to draw the operator's attention to that value and, in certain cases, to take actions in response to the exceptional value.
Stimuli	Values from sensors attached to the system.
Responses	Outputs to display, alarm triggers signals to reacting systems.

The Observe and React pattern...



Observe and React process

The Observe and React pattern...

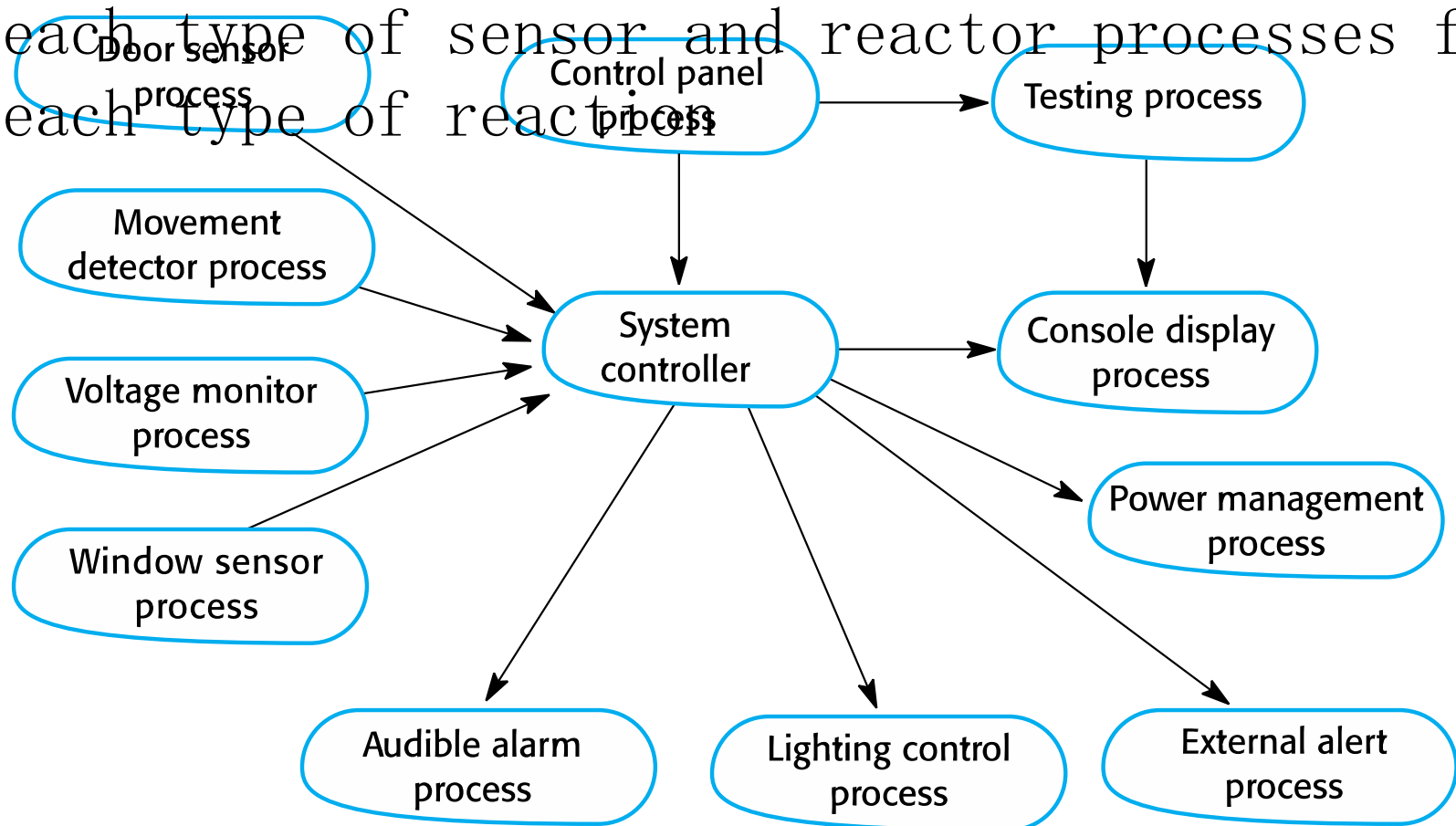
Alarm system description

A software system is to be implemented as part of a burglar alarm system for commercial buildings. This uses several different types of sensor. These include movement detectors in individual rooms, door sensors that detect corridor doors opening, and window sensors on ground-floor windows that detect when a window has been opened.

When a sensor detects the presence of an intruder, the system automatically calls the local police and, using a voice synthesizer, reports the location of the alarm. It switches on lights in the rooms around the active sensor and sets off an audible alarm. The sensor system is normally powered by mains power but is equipped with

The Observe and React pattern...

- ❧ A possible process architecture for the burglar alarm system
- ❧ There are observer processes associated with each type of sensor and reactor processes for each type of reaction

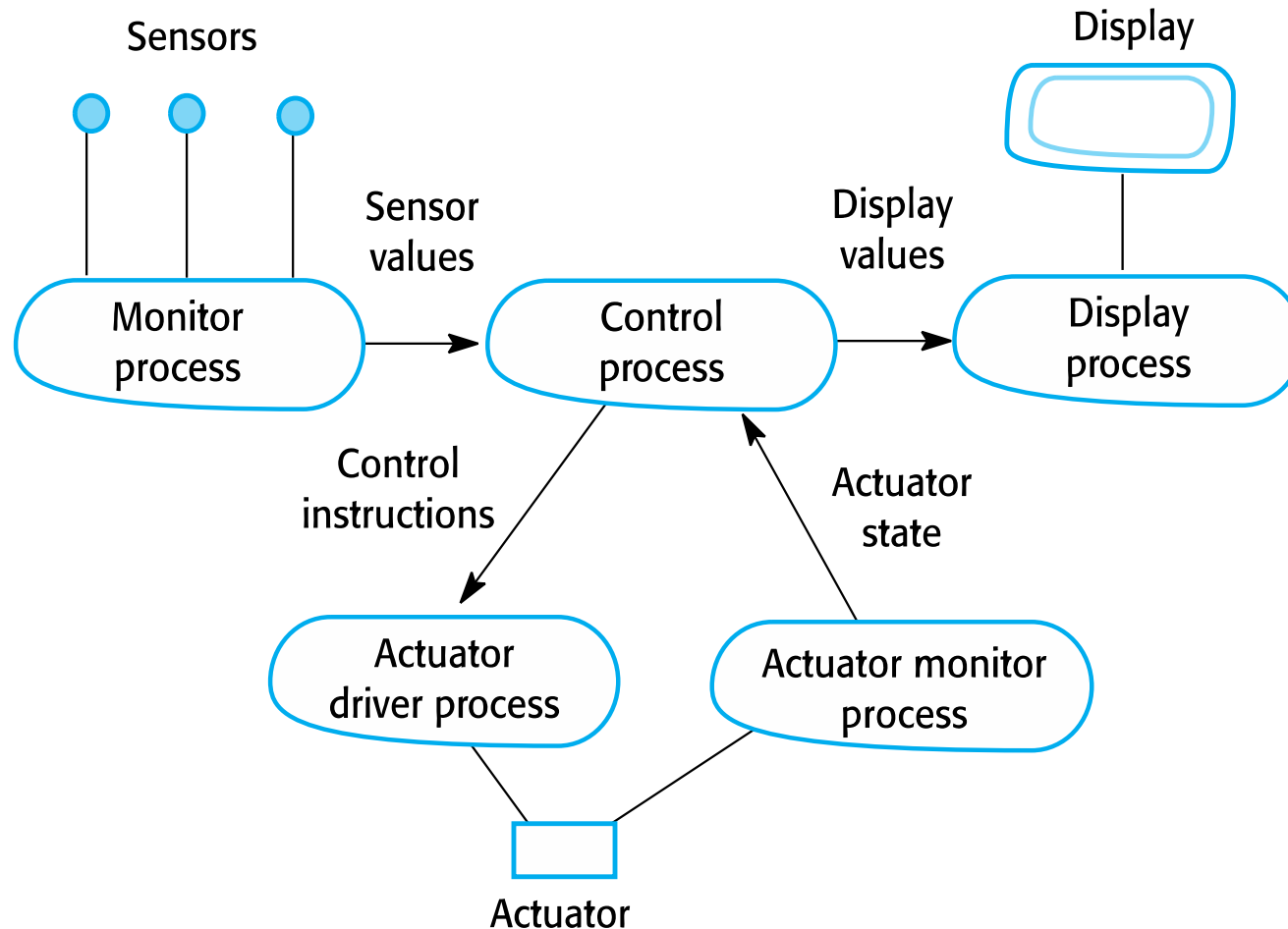


The Environmental Control pattern

is used when a system includes sensors, which provide information about the environment and actuators that can change the environment

Name	Environmental Control
Description	The system analyzes information from a set of sensors that collect data from the system's environment. Further information may also be collected on the state of the actuators that are connected to the system. Based on the data from the sensors and actuators, control signals are sent to the actuators that then cause changes to the system's environment. Information about the sensor values and the state of the actuators may be displayed.
Stimuli	Values from sensors attached to the system and the state of the system actuators.
Responses	Control signals to actuators, display

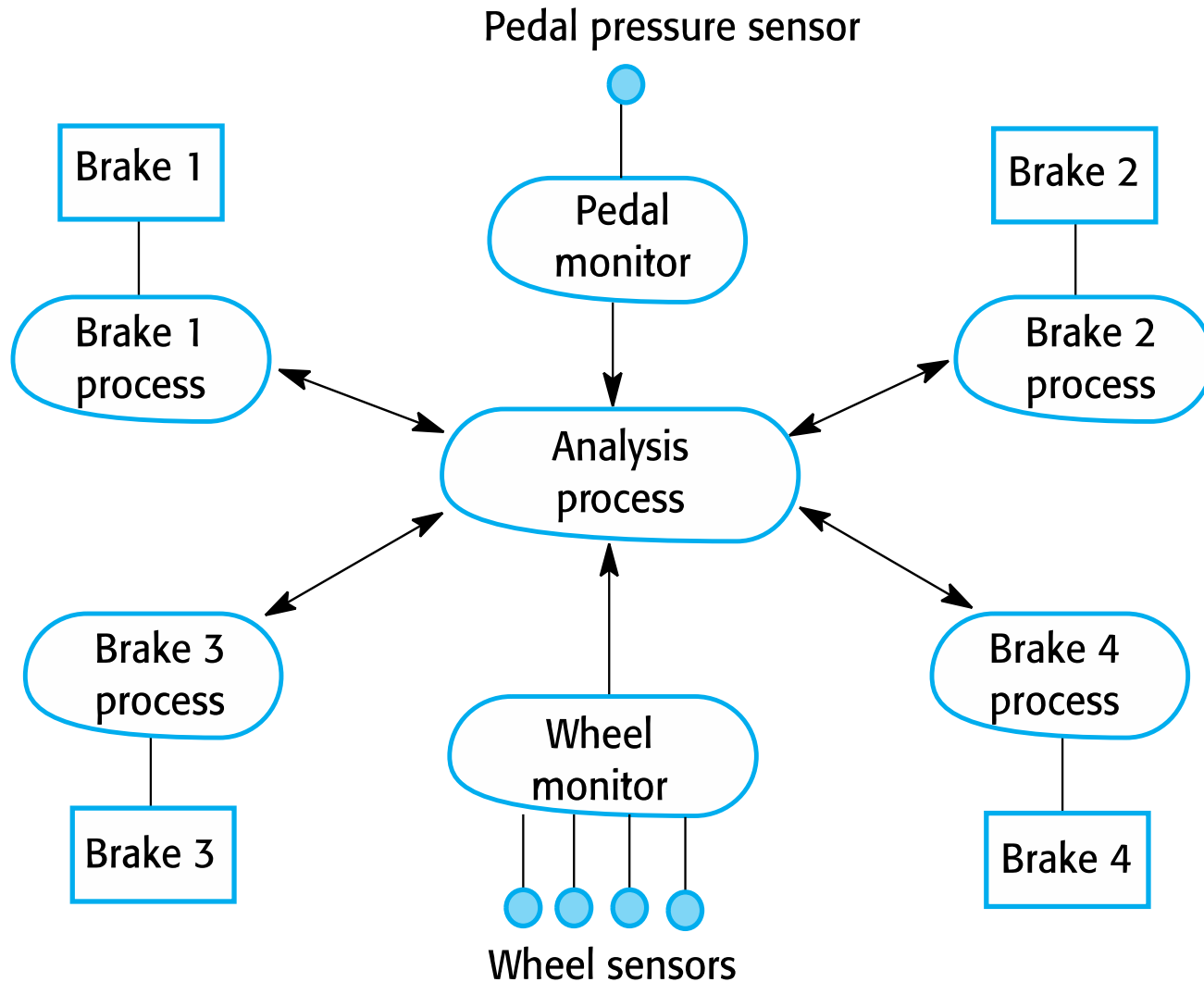
The Environmental Control pattern...



Environ
mental
Control
process
structu
re

In this type of pattern software controls the operation of equipment, based on stimuli from the equipment's environment.

The Environmental Control pattern...



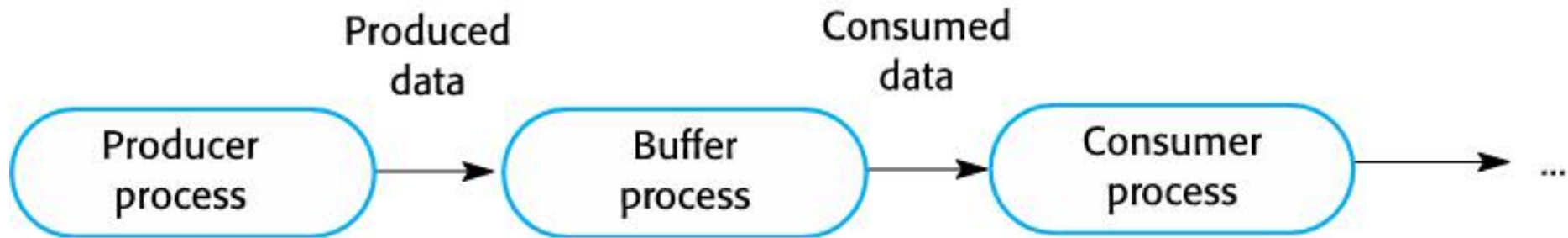
The Process Pipeline pattern

This pattern is used when data has to be transformed from one representation to another before it can be processed.

Name	Process Pipeline
Description	A pipeline of processes is set up with data moving in sequence from one end of the pipeline to another. The processes are often linked by synchronized buffers to allow the producer and consumer processes to run at different speeds. The culmination of a pipeline may be display or data storage or the pipeline may terminate in an actuator.
Stimuli	Input values from the environment or some other process

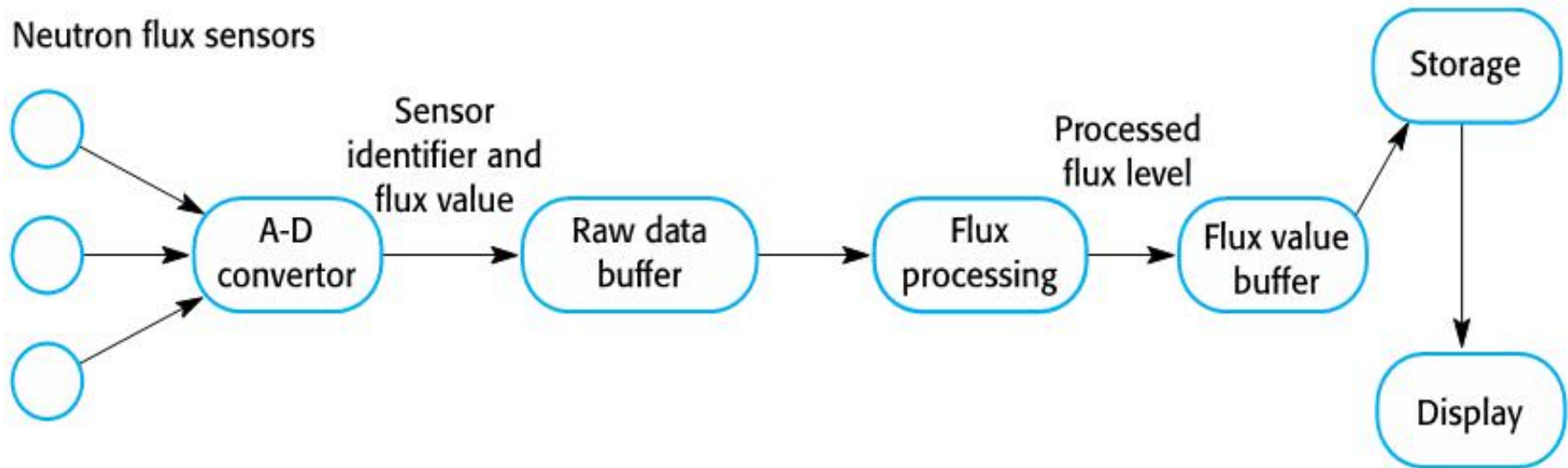
The Process Pipeline pattern...

Process Pipeline process structure



The process pipeline pattern...

Neutron flux high-speed data acquisition



Data acquisition systems collect data from sensors for subsequent processing and analysis. These systems are used in situations where the sensors are collecting a lot of data from the system's environment and it isn't possible or necessary to process that data in real time. Rather, it is collected and stored for later analysis often used

Timing analysis

- ✧ The correctness of a real-time system depends not just on the correctness of its outputs but also on the time at which these outputs were produced.
 - ✧ In a timing analysis, you calculate how often each process in the system must be executed to ensure that all inputs are processed and all system responses produced in a timely way.
 - ✧ The results of the timing analysis are used to decide how frequently each process should execute and how these processes should be scheduled by the real-time operating system.
- ✧ Timing analysis for real-time systems is

Timing analysis...

⌘ Factors in timing analysis

⌘ *Deadlines*

⌘ The times by which stimuli must be processed and some response produced by the system.

⌘ Important, especially for hard real-time systems

⌘ *Frequency*

⌘ The number of times per second that a process must execute so that you are confident that it can always meet its deadlines.

⌘ *Execution time*

⌘ The time required to process a stimulus and produce a response.

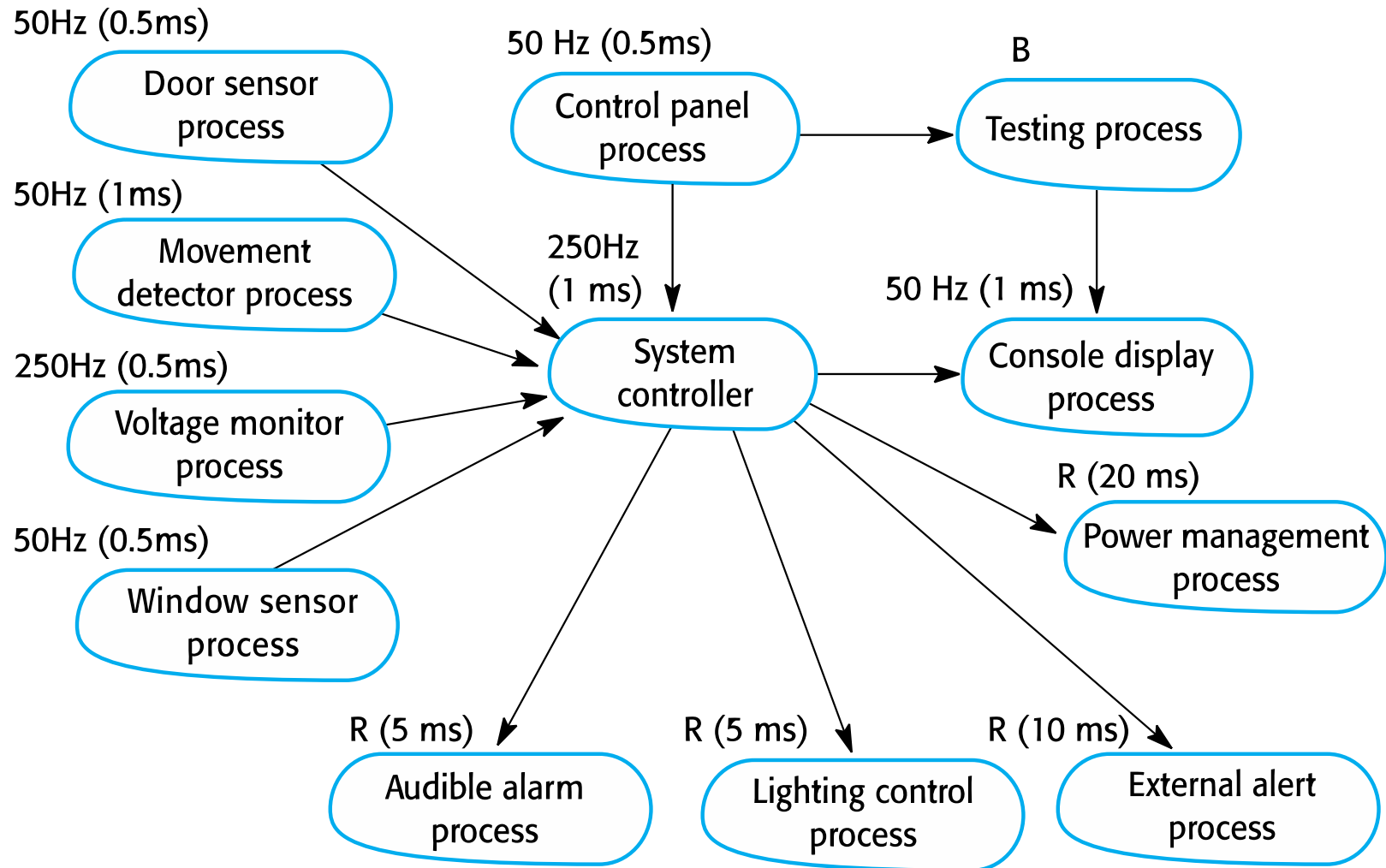
⌘ Often, you have to take two execution times

Timing analysis...

Timing requirements for the burglar alarm

Stimulus/Response	Timing requirements
Audible alarm	The audible alarm should be switched on within half a second of an alarm being raised by a sensor.
Communications	The call to the police should be started within 2 seconds of an alarm being raised by a sensor.
Door alarm	Each door alarm should be polled twice per second.
Lights switch	The lights should be switched on within half a second of an alarm being raised by a sensor.
Movement detector	Each movement detector should be polled twice per second.
Power failure	The switch to backup power must be completed within a deadline of 50 ms.
Police synthesizer	A synthesized message should be available within 2 seconds of an alarm being raised by a sensor.

Timing analysis...



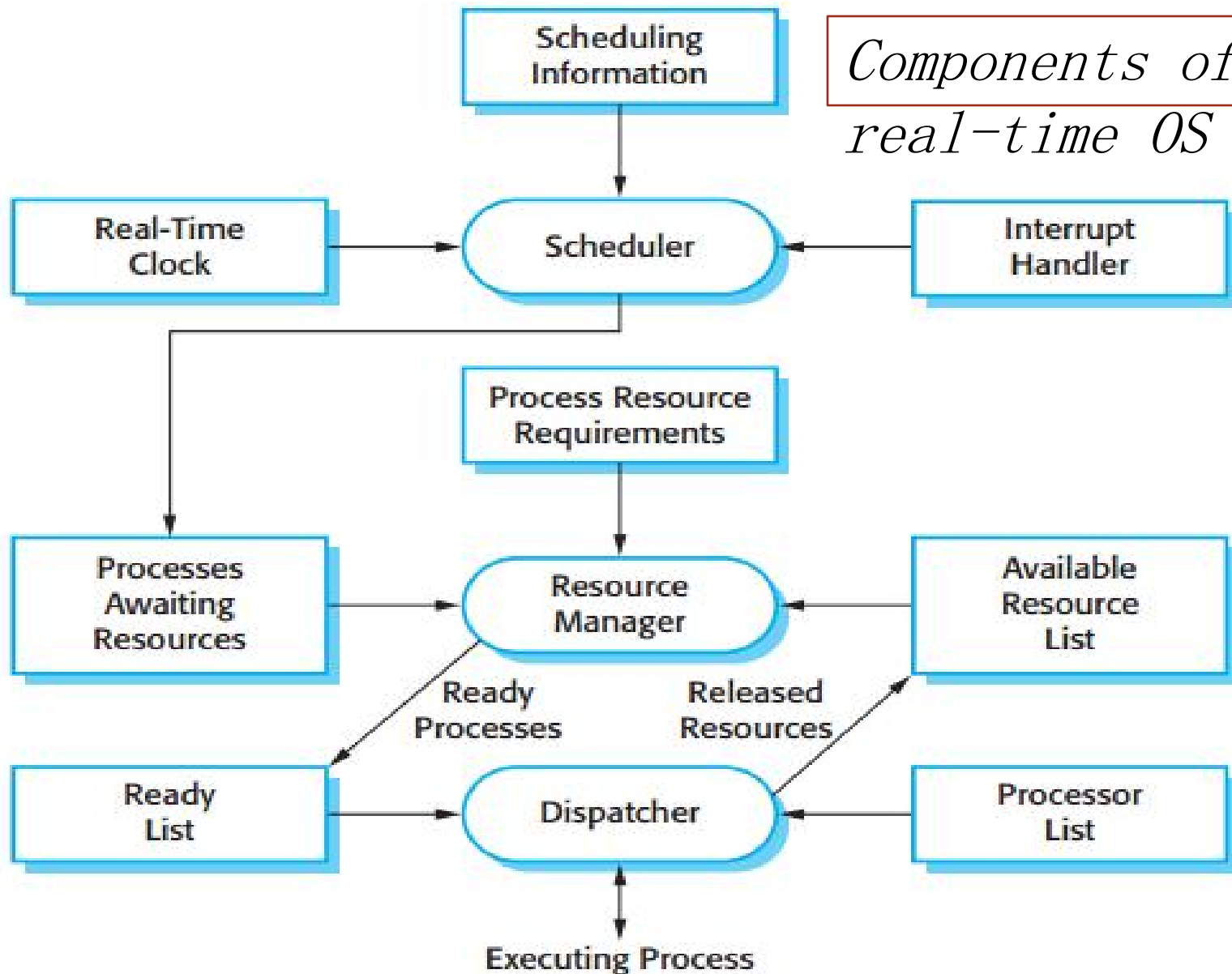
Alarm process timing

Real-time operating systems

- ❧ Real-time operating systems are specialised operating systems which manage the processes in the RTS.
- ❧ Responsible for process management and resource (processor and memory) allocation.
- ❧ May be based on a standard kernel which is used unchanged or modified for a particular application.
- ❧ Do not normally include facilities such as file management.
- ❧ Standard operating systems, such as Linux and Windows, are not normally used as the execution platform

Real-time operating systems

*Components of a
real-time OS*



Summary

- ✧ An embedded software system is part of a hardware/software system that reacts to events in its environment. The software is ‘embedded’ in the hardware. Embedded systems are normally real-time systems.
- ✧ A real-time system is a software system that must respond to events in real time. System correctness does not just depend on the results it produces, but also on the time when these results are produced.
- ✧ Real-time systems are usually implemented as a set of communicating processes that react to stimuli to produce responses.
- ✧ State models are an important design

Summary...

- ✧ There are several standard patterns that can be observed in different types of embedded system. These include a pattern for monitoring the system's environment for adverse events, a pattern for actuator control and a data-processing pattern.
- ✧ Designers of real-time systems have to do a timing analysis, which is driven by the deadlines for processing and responding to stimuli. They have to decide how often each process in the system should run and the expected and worst-case execution time for processes.
- 39 ✧ A real-time operating system is responsible for