Structural and Behavioural Modelling
Text 5.3 – 5.5

CMPT 276
© Dr. B. Fraser

Topics

1) How can we model the structure of a system?
2) How can we model the behaviour of a system?
3) Can we use modules to generate a system?

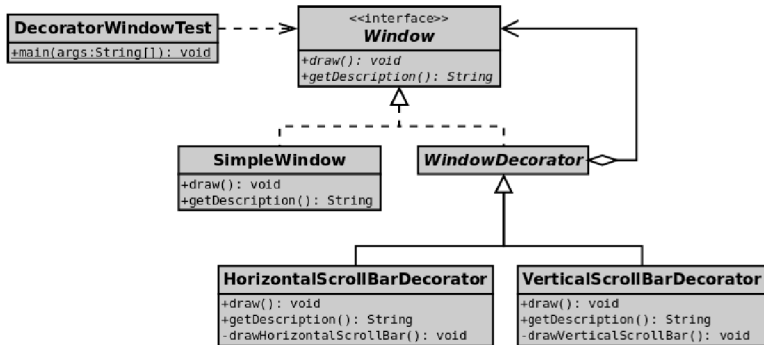# Structural models

Structural models

- Structural models of software:

- Structural models may be:
  - static models:
    show the structure of the...
    Ex: Classes
  - dynamic models:
    show organization of system...
    - Ex: Interacting threads.
- Use structural models of a system when discussing and designing the system architecture.

UML Class Diagram

- UML Class Diagram
  - A diagram showing..

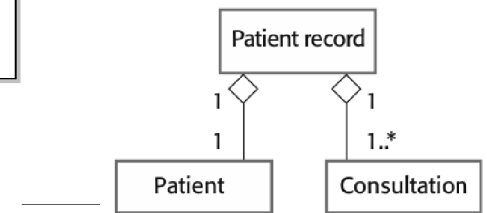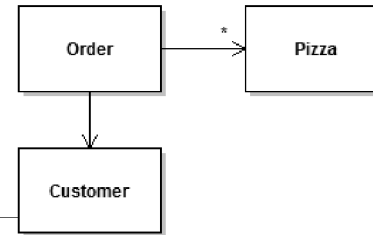# Relationships: Aggregation

- Aggregation: ...
  - Shows an object composed of other objects.

    Ex: A cell-phone has-a screen, or has many buttons.

- Shown as either:
  - line with a..   (on side of whole)
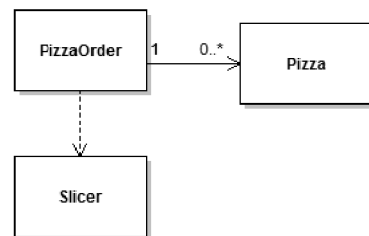 - solid arrow (open)

- Show number: 1, 0..1, *

Relationship: Dependency

- Dependency:
  Class X depends on class Y if..

  - Usually said: "X uses Y"
  - If X knows of Y's existence, then..
  -

- Example:

```
class PizzaOrder {
    ArrayList<Pizza> pizzas;
    // ...
    public void slicePizzas() {
        Slicer slicer = new Slicer();
        slicer.slicePizzas(pizzas);
    }
}
```
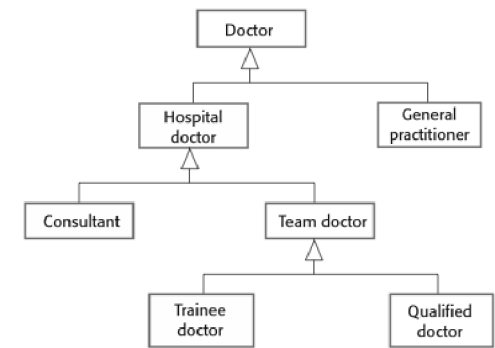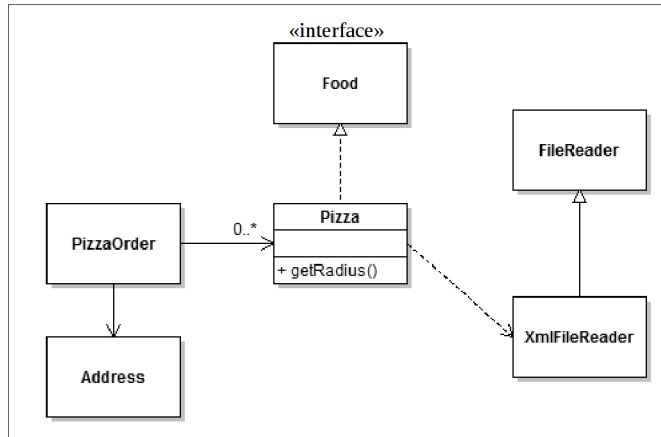

Relationships: Inheritance

- Inheritance:

  - A cell-phone is a type of phone: cell-phone inherits from phone.
  -

    pointing from the subclass to the superclass (more general class).

Exercise: Label the relationships

«interface» Food, FileReader, Pizza + getRadius(), PizzaOrder 0.., Address, XmlFileReader*

# Behavioural models

---

Behavioural models

- Behavioural models:

  - Shows what happens when a system responds to a stimulus from its environment.
- Two types of stimuli:
  -

    data arrives that has to be processed by the system.
  -

    an event happens that triggers system processing.

Data-driven modelling

- Many business systems are data-processing systems:
  - Ex: phone billing system processes raw data into invoices.
  - no real-time events; just data to process.
- Data-driven modelling great for requirements analysis to show...
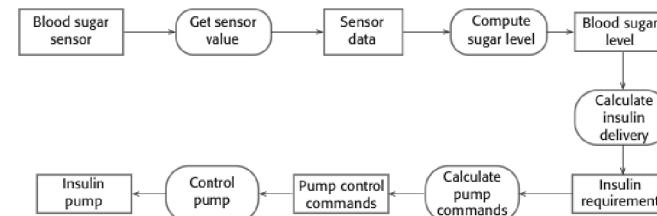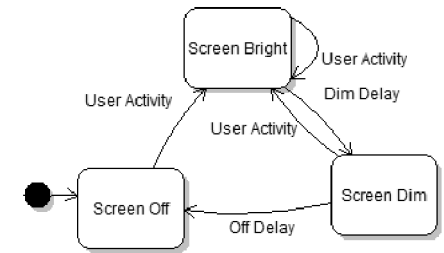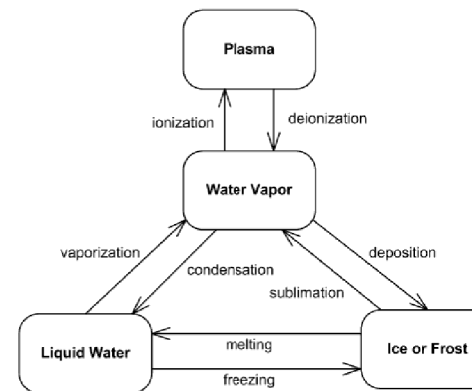

An activity model of the insulin pump's operation

Event-driven modelling

- Real-time systems are often event-driven, with minimal data processing.
  - Ex: microwave oven, alarm clock, etc.
- Event-driven modelling shows how a system

  - System has states, and events (stimuli) cause...
  - Called state diagram, or FSM: Finite state machine.

State Machines
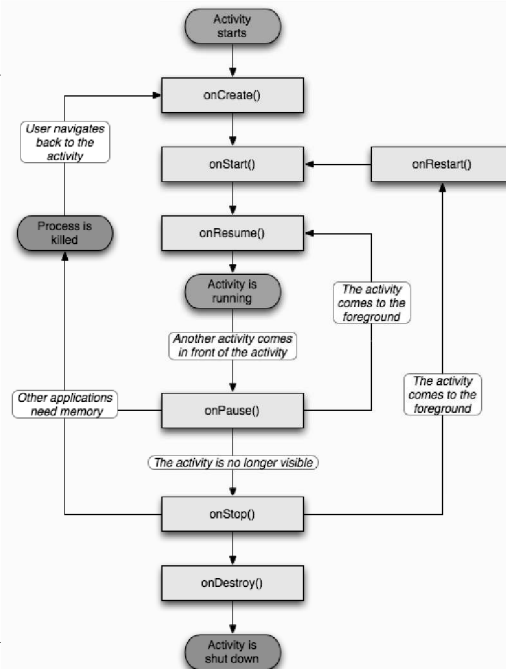
- What are each of the following state machines for?

http://www.uml-diagrams.org/examples/state-machine-example-water.png   http://cphacker0901.wordpress.com/1900/01/01/android-power-management/

Android

- Many events can occur in the lifetime of an Android activity.
- Demo Examples:
  - Creation
  - While running, switch to home screen.
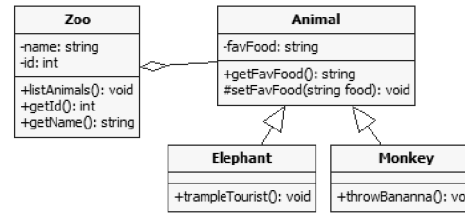 - While in background, killed by OS.

Model-driven engineering

Model-driven engineering

- Model-driven engineering
  - An approach to software development where models rather than programs are the principal outputs of the development process.
  -

- Pros
  - Work at...
  - Cheaper port to new platforms: code is generated!

- Cons
  - Models for abstraction not always suited to implementation.
  -

Model-driven engineering example

- StarUML Generates C++ code from class diagram
  - Generates all .h files and function stubs in .cpp files.

```
//  Generated by StarUML(tm) C++ Add-In
//
//  @ Project : Untitled
//  @ File Name : Zoo.h
//  @ Date : 20/02/2014
//  @ Author :

#if !defined(_ZOO_H)
#define _ZOO_H

class Zoo {
public:
        void listAnimals();
        int getId();
        string getName();
private:
        string name;
        int id;
};

#endif  //_ZOO_H
```

Summary

- Structural models show the organization and architecture of a system.
  - Class diagrams define the static structure of classes in a system and their associations.

- Behavioural models describe the dynamic behaviour of an executing system.
  - Data processing perspective – activity diagram: show flow of data through steps.
  - Event driven perspective – state diagrams: show states and internal or external events.

- Model-driven engineering: build the model, and then tools automatically transformed to executable code.