CS 320
Fundamentals of Software Engineering

Lecture 12: Design and Implementation
Software design and implementation is the stage at which an executable software system is developed.

Software design and implementation activities are invariably interleaved.
Process Stages

- Common activities
  - Define the context
  - Design the system architecture
  - Identify the principle system objects
  - Develop design models
  - Specify object interfaces
System Context and Interactions

• Understand the relationships between the software and its external environment

• Establish the boundaries
System Context for a Weather Station System

- Control system
- Weather information system
- Satellite
- Weather station

1..n
1
1
1
1
1
1
1
1
Weather Station Use Cases

- Weather information system
  - Report weather
  - Report status
- Control system
  - Restart
  - Shutdown
  - Reconfigure
  - Powersave
  - Remote control
# Use Case Description - Report Weather

<table>
<thead>
<tr>
<th>System</th>
<th>Weather station</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case</td>
<td>Report weather</td>
</tr>
<tr>
<td>Actors</td>
<td>Weather information system, Weather station</td>
</tr>
<tr>
<td>Description</td>
<td>The weather station sends a summary of the weather data that has been collected from the instruments in the collection period to the weather information system. The data sent are the maximum, minimum, and average ground and air temperatures; the maximum, minimum, and average air pressures; the maximum, minimum, and average wind speeds; the total rainfall; and the wind direction as sampled at five-minute intervals.</td>
</tr>
<tr>
<td>Stimulus</td>
<td>The weather information system establishes a satellite communication link with the weather station and requests transmission of the data.</td>
</tr>
<tr>
<td>Response</td>
<td>The summarized data is sent to the weather information system.</td>
</tr>
<tr>
<td>Comments</td>
<td>Weather stations are usually asked to report once per hour but this frequency may differ from one station to another and may be modified in the future.</td>
</tr>
</tbody>
</table>
Architectural Design

- Once interaction between the system and its environment have been understood, you use this information for designing the system architecture.
- You identify the major components that make up the system and their interactions, and then may organize components using an architectural pattern, such as a layered or client-server model.
- The weather station is composed of independent subsystems that communicate by broadcasting messages on a common infrastructure.
High-level Architecture of the Weather Station

- Data collection
- Communications
- Configuration manager
- Fault manager
- Power manager
- Instruments
- Communication link
Architecture of Data Collection System

Data collection

Transmitter

Receiver

WeatherData
Object Class Identification

- Identify object classes is often a difficult part of object oriented design
- There is no ‘magic formula’ for object identification. It relies on the skill, experience and domain knowledge of system designers
- Object identification is an iterative process. You are unlikely to get it right first time
Approaches to Identification

- Use a grammatical approach based on a natural language description of the system
- Base the identification on tangible things in the application domain
- Use a behavioral approach and identify objects based on what participates in what behavior
- Use a scenario-based analysis. The objects, attributes and methods in each scenario are identified.
Weather Station Description

* A weather station is a package of software controlled instruments which collects data, performs some data processing and transmits this data for further processing. The instruments include air and ground thermometers, an anemometer, and a barometer. Data is collected periodically.

* When a command is issued to transmit the weather data, the weather station processes and summarizes the collected data. The summarized data is transmitted to the mapping computer when a request is received.
Weather Station Object Classes

- Object class identification in weather station system may be based on the tangible hardware and data in the system:
  - Ground thermometer, anemometer, barometer
    - Application domain objects that are ‘hardware’ objects related to the instruments in the system.
  - Weather station
    - The basic interface of the weather station to its environment. It therefore reflects the interactions identified in the use-case model.
  - Weather Data
    - Encapsulates the summarized data from the instruments.
Weather Station Object Classes

<table>
<thead>
<tr>
<th>WeatherStation</th>
<th>WeatherData</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>airTemperatures</td>
</tr>
<tr>
<td>reportWeather ()</td>
<td>groundTemperatures</td>
</tr>
<tr>
<td>reportStatus ()</td>
<td>windSpeeds</td>
</tr>
<tr>
<td>powerSave (instruments)</td>
<td>windDirections</td>
</tr>
<tr>
<td>remoteControl (commands)</td>
<td>pressures</td>
</tr>
<tr>
<td>reconfigure (commands)</td>
<td>rainfall</td>
</tr>
<tr>
<td>restart (instruments)</td>
<td>collect ()</td>
</tr>
<tr>
<td>shutdown (instruments)</td>
<td>summarize ()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ground thermometer</th>
<th>Anemometer</th>
<th>Barometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>gt_Ident</td>
<td>an_Ident</td>
<td>bar_Ident</td>
</tr>
<tr>
<td>temperature</td>
<td>windSpeed</td>
<td>pressure</td>
</tr>
<tr>
<td>get ()</td>
<td>get ()</td>
<td>get ()</td>
</tr>
<tr>
<td>test ()</td>
<td>test ()</td>
<td>test ()</td>
</tr>
</tbody>
</table>

---

gt_Ident
an_Ident
bar_Ident
Design Models

- Design models show the objects and object classes and relationships between entities
- Static models describe the static structure of the system in terms of object classes and relationships
- Dynamic models describe the dynamic interactions between objects
Examples of Design Models

- Subsystem models that show logical grouping of objects into coherent subsystems
- Sequence models that show the sequence of object interactions
- State machine models that show how individual objects change their state in response to events
Subsystem Models

- Shows how the design is organized into logically related group of objects
- In the UML, these are shown using packages
Sequence Models

- Sequence models show the sequence of object interactions that take place
  - Objects are arranged horizontally across the top
  - Time is represented vertically so models are read top to bottom
  - Interactions are represented by labelled arrows, different styles of arrow represented different types of interaction
  - A thin rectangle in an object lifeline represents the time when the object is the controlling object in the system
Sequence Diagram Describing Data Collection
State Diagram

- State diagram are used to show how objects respond to difference service requests and the state transitions triggered by these requests.
- State diagrams are useful high-level models of a system or an object’s run-time behavior.
- You don’t usually need a state diagram for all of the objects in the system. Many of the objects in a system are relatively simple and a state model adds unnecessary detail to the design.
Weather Station State Diagram

- **Shutdown**
  - shutdown()
  - restart()
  - reconfigure()
  - powerSave()

- **Configuring**
  - configuration done
  - clock

- **Running**
  - reportStatus()
  - collection done

- **Collecting**
  - reportWeather()

- **Transmitting**
  - transmission done
  - test complete
  - weather summary complete

- **Testing**
  - test complete

- **Summarizing**
  - test complete

- **Operation**
  - remoteControl()

- **Controlled**
Object interfaces have to be specified so that the objects and other components can be designed in parallel.

Designers should avoid designing the interface representation but should hide this in the object itself.

Objects may have several interfaces which are viewpoints on the methods provided.
Weather Station Interfaces

**Reporting**


**Remote Control**

- `startInstrument(instrument): iStatus`
- `stopInstrument (instrument): iStatus`
- `collectData (instrument): iStatus`
- `provideData (instrument): string`
Key Points

- Software design and implementation are inter-leaved activities

- The process of object-oriented design includes activities to design the system architecture, identify objects in the system, describe the design using different object models and document the component interfaces

- A range of difference models may be produced during an object-oriented design process. These include static models (class), and dynamic models (sequence, state)

- Component interfaces must be defined precisely so that other objects can use them