



澳門理工學院
Instituto Politécnico de Macau
Macao Polytechnic Institute



Online Learning Materials

COMP223: Software Engineering Design and Implementation

Dr. Kim, Song-Kyoo (Amang)
Associate Professor,

Computer Science Program
MACAO POLYTECHNIC INSTITUTE
Macau, SAR



Session (Chapter 7) Objectives



- Object-oriented design using the UML
- Design patterns
- Implementation issues
- Open source development



澳門理工學院
Instituto Politécnico de Macau
Macao Polytechnic Institute

Design & Implementation (Part 1)



Design and implementation (1/2)



- Software design and implementation is the stage in the software engineering process at which an executable software system is developed.
- Software design and implementation activities are invariably inter-leaved.
 - Software design is a creative activity in which you identify software components and their relationships, based on requirements of customers.
 - Implementation is the process of realizing the design as a program.

Design and implementation (2/2)



● Build or buy

- In a wide range of domains, it is now possible to buy off-the-shelf systems (COTS) that can be adapted and tailored to the users' requirements.
- For example, if you want to implement a medical records system, you can buy a package that is already used in hospitals. It can be cheaper and faster to use this approach rather than developing a system in a conventional programming language.
- When you develop an application in this way, the design process becomes concerned with how to use the configuration features of that system to deliver the system requirements.

Object-oriented design using the UML (1/20)



- An object-oriented design process
 - Structured object-oriented design processes involve developing a number of different system models.
 - They require a lot of effort for development and maintenance of these models and, for small systems, this may not be cost-effective.
 - However, for large systems developed by different groups design models are an important communication mechanism.

Object-oriented design using the UML (2/20)



- **Process stages**: there are a variety of different object-oriented design processes that depend on the organization using the process.
- Common activities in these processes include:
 - Define the context and modes of use of the system;
 - Design the system architecture;
 - Identify the principal system objects;
 - Develop design models;
 - Specify object interfaces.
- Process illustrated here using a design for a wilderness weather station.

Object-oriented design using the UML (3/20)



- System context and interactions

- Understanding the relationships between the software that is being designed and its external environment is essential for deciding how to provide the required system functionality and how to structure the system to communicate with its environment.
- Understanding of the context also lets you establish the boundaries of the system. Setting the system boundaries helps you decide what features are implemented in the system being designed and what features are in other associated systems.

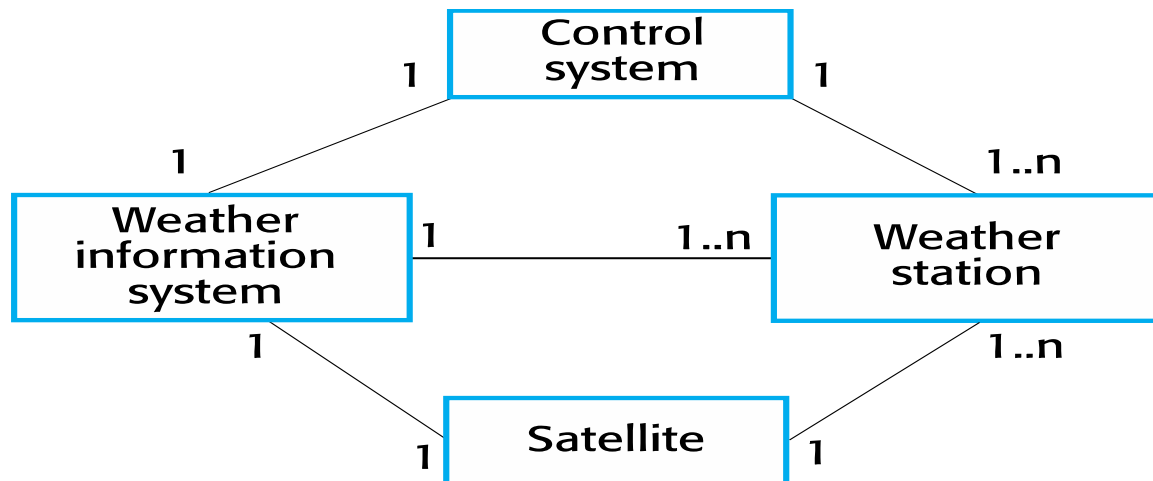
Object-oriented design using the UML (4/20)



● Context and interaction models

- A system context model is a structural model that demonstrates the other systems in the environment of the system being developed.
- An interaction model is a dynamic model that shows how the system interacts with its environment as it is used.

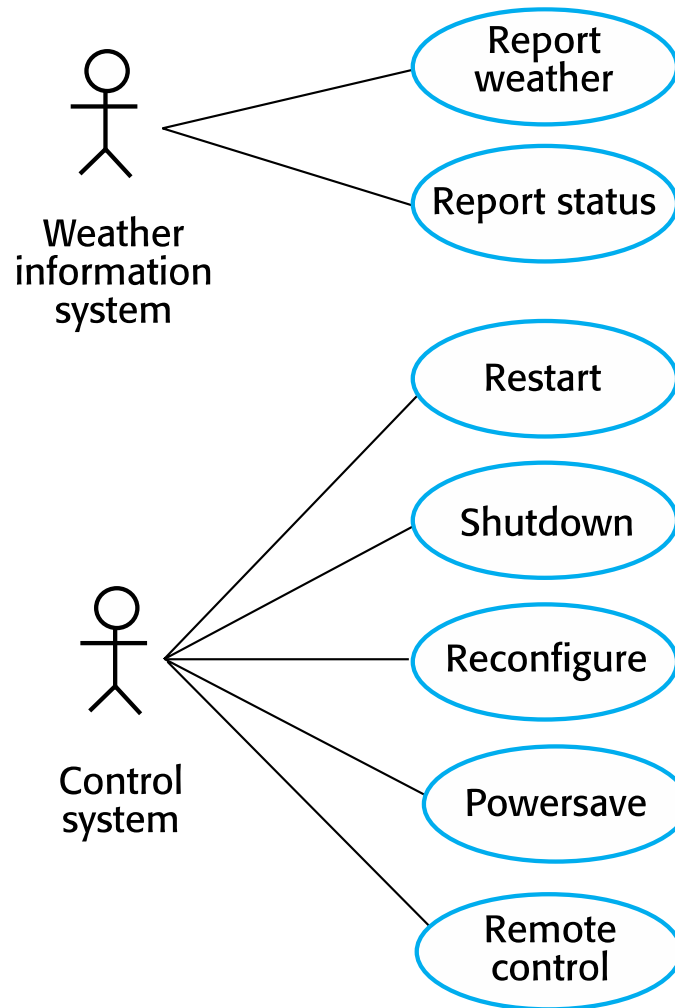
● System context for the weather station



Object-oriented design using the UML (5/20)



● Weather station use cases



Object-oriented design using the UML (6/20)



● Use case description—Report weather

System	Weather station
Use case	Report weather
Actors	Weather information system, Weather station
Description	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.
Stimulus	The weather information system establishes a satellite communication link with the weather station and requests transmission of the data.
Response	The summarized data is sent to the weather information system.
Comments	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.

Object-oriented design using the UML (7/20)



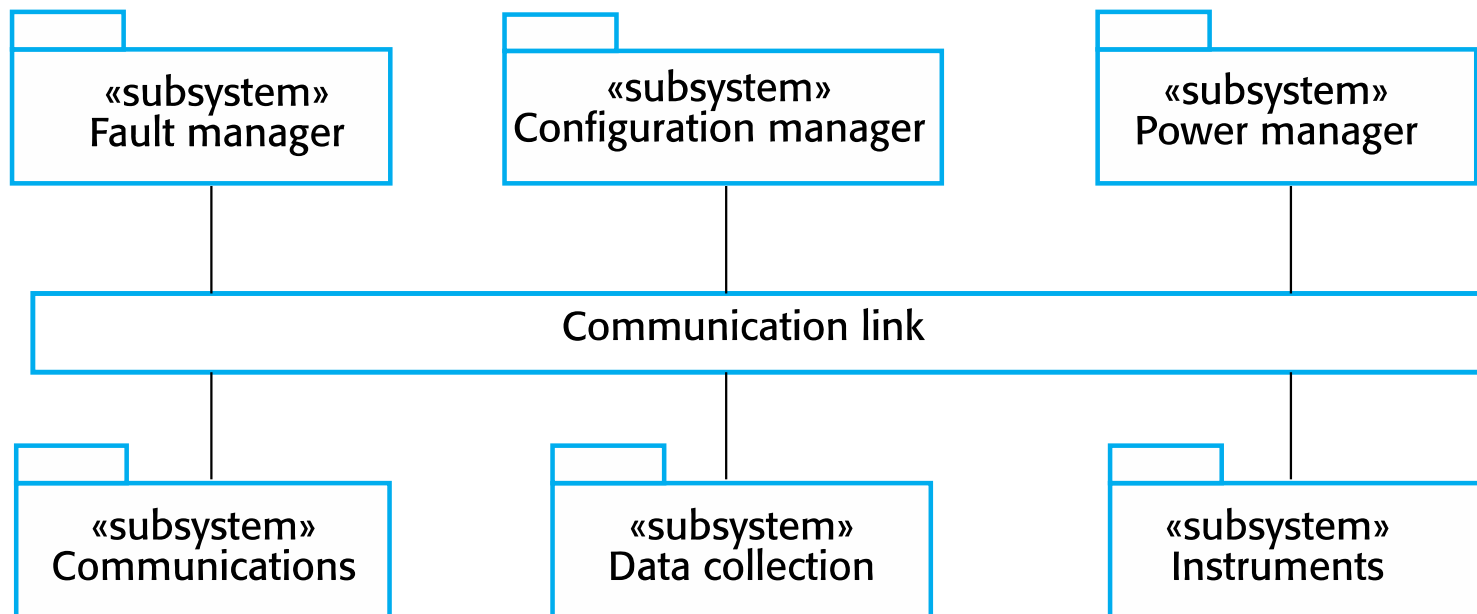
● Architectural design

- Once interactions 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 the 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.

Object-oriented design using the UML (8/20)



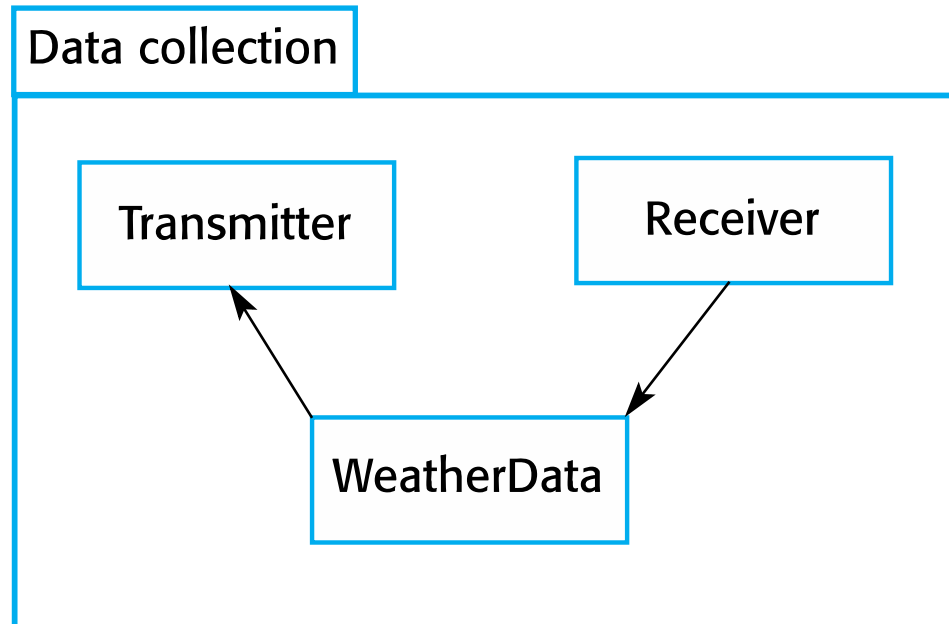
- High-level architecture of the weather station



Object-oriented design using the UML (9/20)



- Architecture of data collection system



OOD using the UML (10/20)



● Object class identification

- Identifying 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.



OOD using the UML (11/20)

- **Weather station object classes:** object class identification in the 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.



OOD using the UML (12/20)

- Weather station object classes

WeatherStation
identifier
reportWeather () reportStatus () powerSave (instruments) remoteControl (commands) reconfigure (commands) restart (instruments) shutdown (instruments)

WeatherData
airTemperatures groundTemperatures windSpeeds windDirections pressures rainfall
collect () summarize ()

Ground thermometer
gt_Ident temperature
get () test ()

Anemometer
an_Ident windSpeed windDirection
get () test ()

Barometer
bar_Ident pressure height
get () test ()

OOD using the UML (13/20)



- **Design models** show the objects and object classes and relationships between these entities.
- There are two kinds of design model:
 - **Structural models** describe the static structure of the system in terms of object classes and relationships.
 - **Dynamic models** describe the dynamic interactions between objects.

OOD using the UML (14/20)



- Examples of design models
 - Subsystem models that show logical groupings 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.
 - Other models include use-case models, aggregation models, generalization models, etc.

OOD using the UML (15/20)



- **Subsystem models** shows how the design is organized into logically related groups of objects.
 - In the UML, these are shown using packages - an encapsulation construct.
 - This is a logical model.
 - The actual organization of objects in the system may be different.

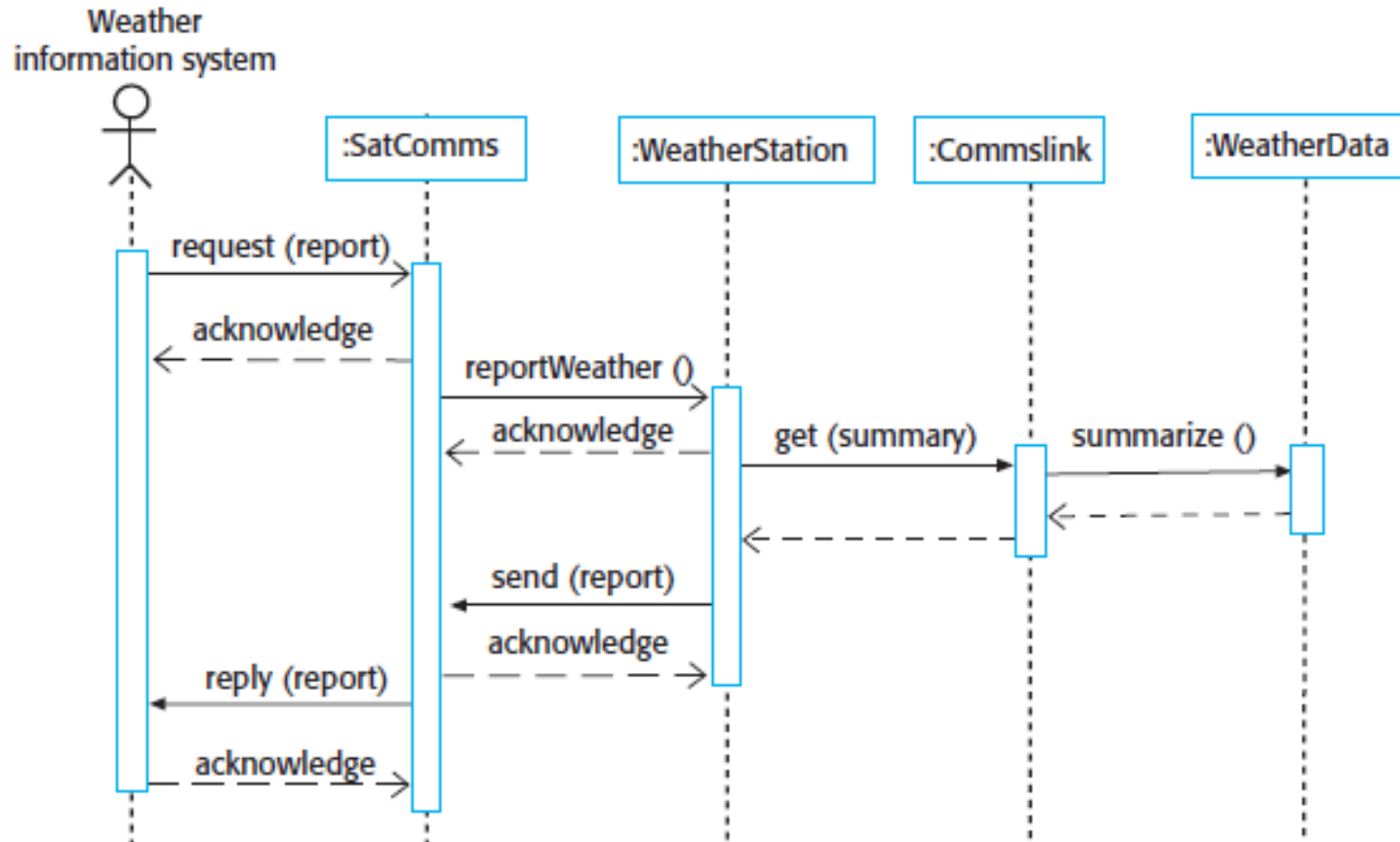
OOD using the UML (16/20)



- **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 represent different types of interaction;
 - A thin rectangle in an object lifeline represents the time when the object is the controlling object in the system.

OOD using the UML (17/20)

- Sequence diagram describing data collection



OOD using the UML (18/20)

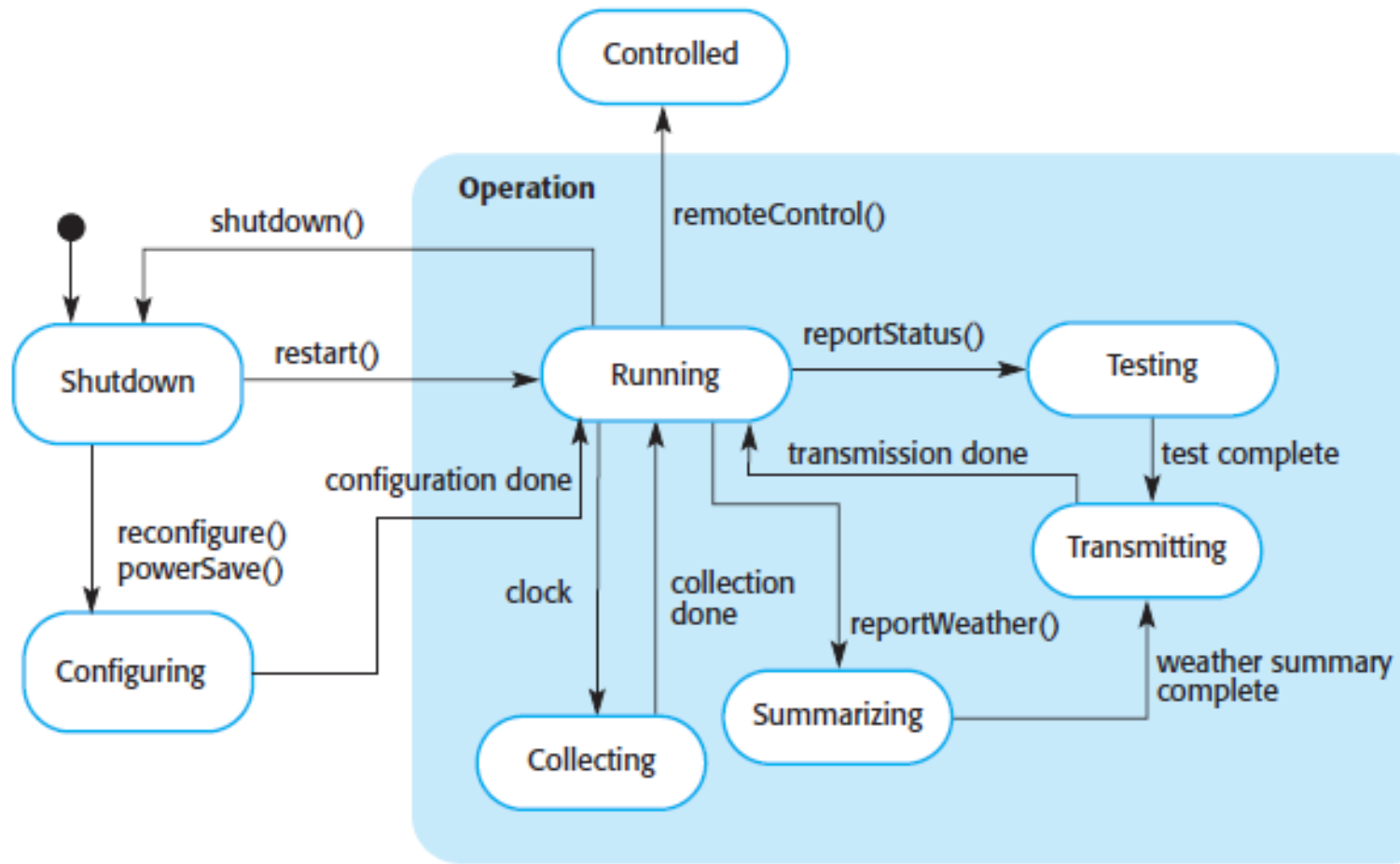


- **State diagrams** are used to show how objects respond to different service requests and the state transitions triggered by these requests.
- State diagrams are useful high-level models of a system or an object's run-time behavior.
- You do not usually need a state diagram for all of the objects in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

OOD using the UML (19/20)



- Weather station state diagram



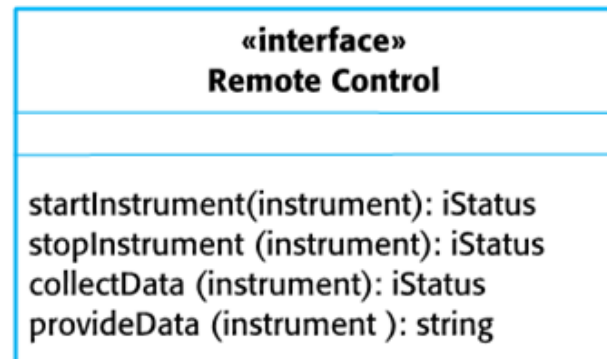
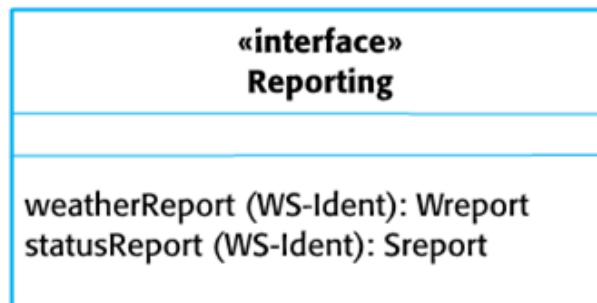


OOD design using the UML (20/20)

● Interface specification:

- 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



Session Summary (1/4)



- Software design and implementation are inter-leaved activities.
- The level of detail in the design depends on the type of system and whether you are using a plan-driven or agile approach.
- 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.



Session Summary (2/4)

- A range of different models may be produced during an object-oriented design process.
- These include static models (class models, generalization models, association models) and dynamic models (sequence models, state machine models).
- Component interfaces must be defined precisely so that other objects can use them.
- A UML interface stereotype may be used to define interfaces.



SCAN ME



SCAN ME

