

Kỹ thuật phần mềm

Thiết kế hướng đối tượng

Mục tiêu

- Thiết kế phần mềm kiểu hướng đối tượng được trình bày như là một tập các đối tượng tương tác với nhau
 - Các đối tượng tự quản lý trạng thái và các thao tác của mình
- Mô tả các hoạt động trong quy trình thiết kế hướng đối tượng
- Giới thiệu một số mô hình có thể dùng để mô tả một thiết kế hướng đối tượng
- Minh họa cách trình bày các mô hình này bằng ngôn ngữ UML

Các chủ đề

- Đối tượng và các lớp đối tượng
- Một quy trình thiết kế hướng đối tượng
- Tiến hóa thiết kế

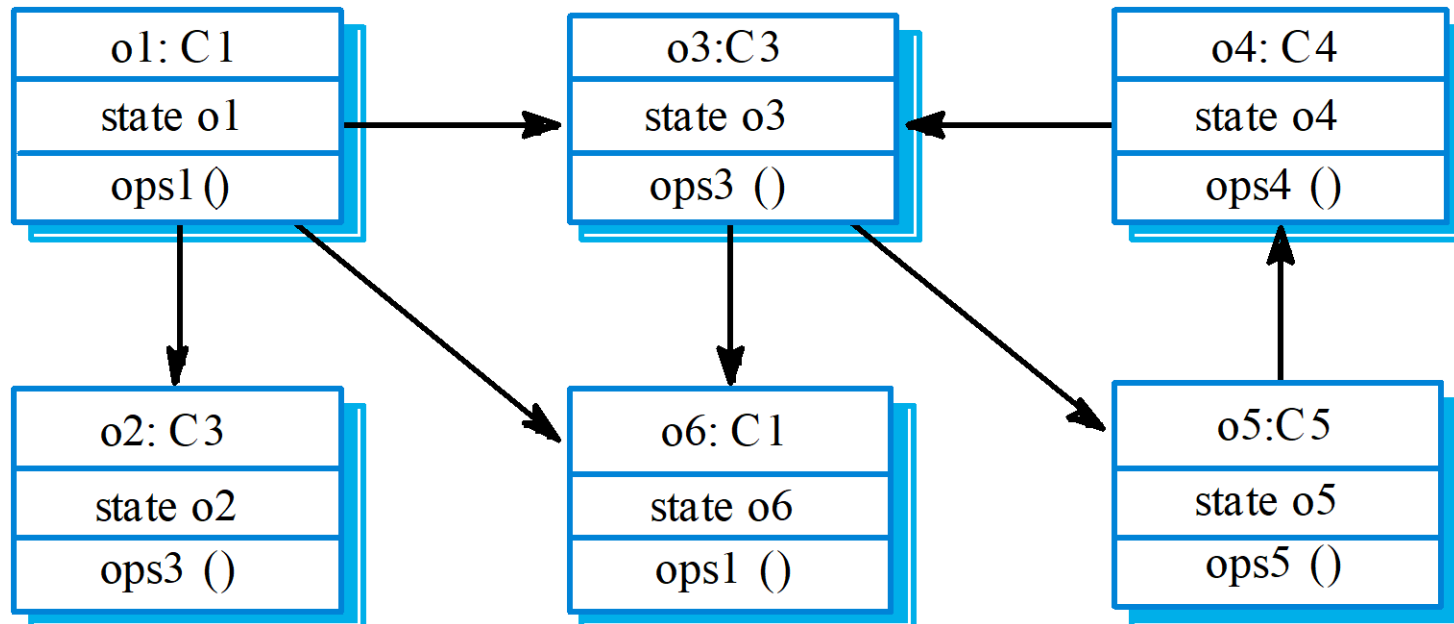
Phát triển hướng đối tượng

- Phân tích (OOA), thiết kế (OOD) và lập trình hướng đối tượng (OOP) có quan hệ với nhau nhưng không trùng nhau
 - Phân tích hướng đối tượng nhằm phát triển một mô hình đối tượng của miền ứng dụng
 - Thiết kế hướng đối tượng nhằm phát triển một mô hình hệ thống hướng đối tượng để thực thi bộ các yêu cầu hệ thống
 - Lập trình hướng đối tượng nhằm hiện thực hóa một thiết kế hướng đối tượng bằng cách sử dụng một ngôn ngữ lập trình hướng đối tượng như Java hoặc C++

Đặc điểm của OOD

- Các đối tượng là các trừu tượng hóa của các thực thể trong **thế giới thực** hoặc các thực thể hệ thống, chúng tự quản lý bản thân
- Các đối tượng có tính độc lập, chúng đóng gói trạng thái và các thao tác liên quan
- Các chức năng hệ thống được diễn đạt theo các **dịch vụ đối tượng**
 - Loại bỏ các vùng dữ liệu dùng chung
 - Các đối tượng liên lạc với nhau bằng cách truyền thông điệp
- Các đối tượng có thể phân tán tại các máy tính khác nhau và có thể thực thi tuần tự hoặc song song

Các đối tượng tương tác



Ưu điểm của OOD

- Dễ bảo trì hơn.
 - Có thể xem các đối tượng như là các thực thể độc lập (stand-alone).
- Các đối tượng là các thành phần tái sử dụng được về tiềm năng. Vì chúng đóng gói độc lập trạng thái và thao tác
- Dễ mô hình hóa hơn
 - Có thể có một ánh xạ dễ thấy từ các thực thể trong thế giới thực tới các thực thể hệ thống.

Đối tượng và lớp đối tượng

- Đối tượng là thực thể trong một hệ thống phần mềm, nó đại diện cho các thể hiện của các thực thể trong thế giới thực và thực thể hệ thống.
- Lớp đối tượng là các khuôn mẫu (template) cho các đối tượng.
 - Dùng để tạo các đối tượng.
- Các lớp đối tượng có thể thừa kế các thuộc tính và dịch vụ từ các lớp đối tượng khác.

Đối tượng và lớp đối tượng

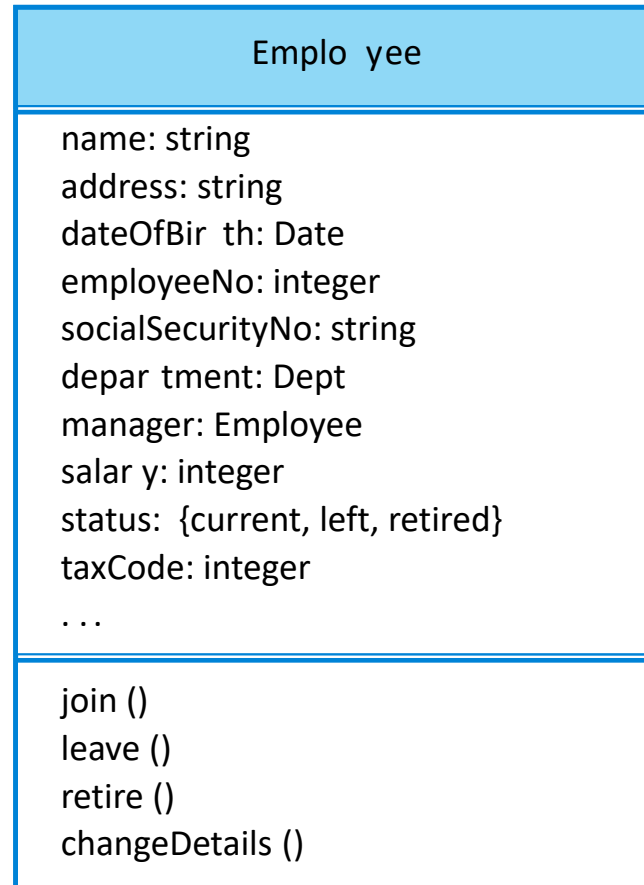
Một **đối tượng** là một thực thể có một trạng thái (state) và một tập xác định trước gồm các thao tác (operation) mà chúng sẽ hoạt động trên trạng thái đó. Trạng thái được đại diện bởi một tập các thuộc tính đối tượng. Các thao tác gắn với một đối tượng cung cấp dịch vụ cho các đối tượng khác (client). Khi cần tính toán, các đối tượng client sẽ gọi các dịch vụ đó.

Các đối tượng được tạo theo một định nghĩa **lớp đối tượng**. Một định nghĩa lớp đối tượng là template cho các đối tượng. Nó bao gồm khai báo cho tất cả các thuộc tính và dịch vụ gắn với mỗi đối tượng thuộc lớp đó.

The Unified Modeling Language

- Một số **kí pháp khác nhau** cho việc mô tả các thiết kế hướng đối tượng đã được đề xuất trong các thập kỉ 1980 và 1990.
- Unified Modeling Language (UML) là một tích hợp của các kí pháp này.
 - Nó mô tả các kí pháp cho một loạt các mô hình khác nhau có thể được tạo ra trong quá trình phân tích thiết kế hướng đối tượng.
 - Nó hiện là một chuẩn không chính thức cho mô hình hóa hướng đối tượng.

Employee object class (UML)



Liên lạc giữa các đối tượng

- Về mặt khái niệm, các đối tượng liên lạc với nhau bằng cách truyền thông điệp.
- Các thông điệp (message)
 - Tên của dịch vụ mà đối tượng gọi yêu cầu chạy;
 - Bản sao của thông tin cần thiết để chạy dịch vụ và tên của nơi chứa kết quả của dịch vụ.
- Trong thực tiễn, các thông điệp thường được cài đặt bằng các lời gọi hàm
 - tên = tên hàm;
 - Thông tin = danh sách tham số.

Ví dụ về thông điệp

```
// Call a method associated with a buffer  
// object that returns the next value  
// in the buffer
```

```
    v = circularBuffer.Get () ;
```

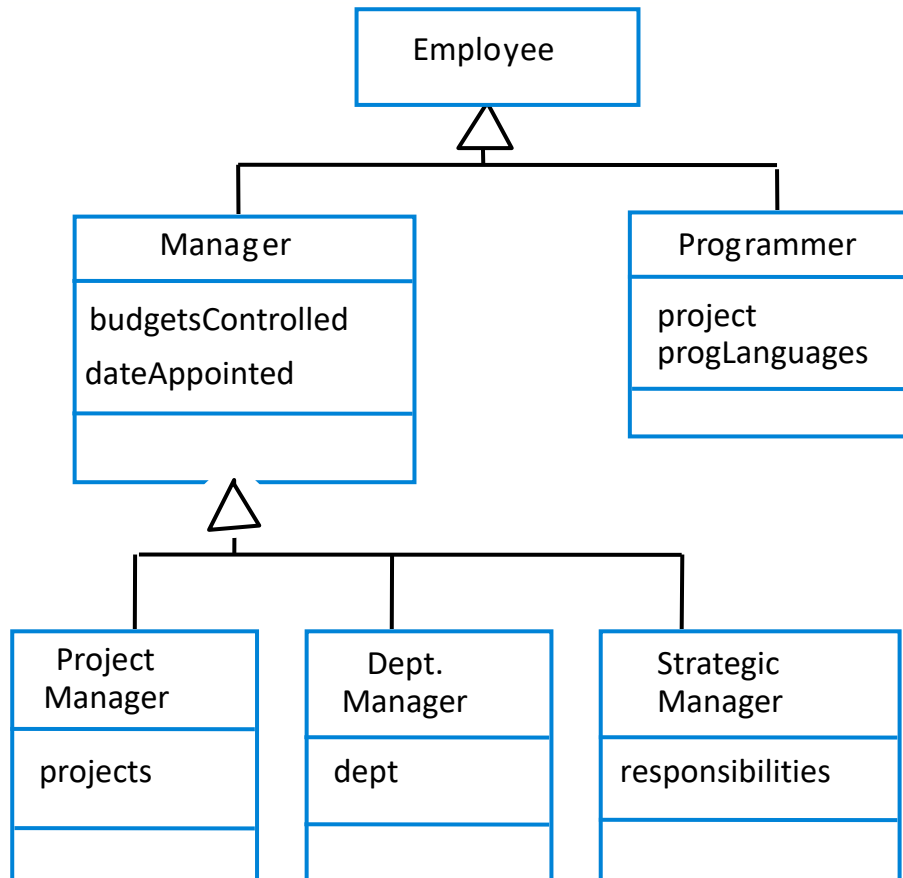
```
// Call the method associated with a  
// thermostat object that sets the  
// temperature to be maintained
```

```
    thermostat.setTemp (20) ;
```

Tổng quát hóa và thừa kế

- Các lớp đối tượng có thể được sắp xếp trong một cấu trúc phân cấp
 - Trong đó một lớp (một lớp cha) là một tổng quát hóa (generalisation) của một vài lớp khác (lớp con).
- Một lớp con thừa kế các thuộc tính và thao tác của lớp cha của nó và có thể bổ sung các phương thức mới hoặc thuộc tính mới của riêng nó.
- Quan hệ tổng quát hóa trong UML được cài bằng quan hệ thừa kế trong các ngôn ngữ OOP.

phân cấp của một tổng quát hóa



Ưu điểm của thừa kế

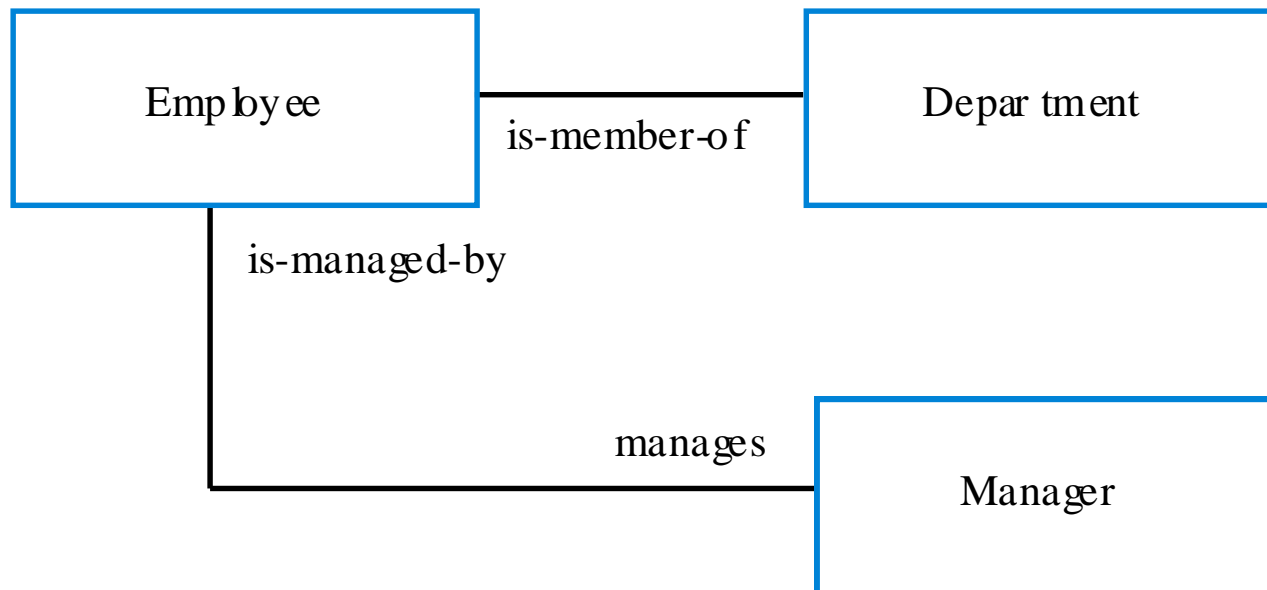
- Là một cơ chế trừu tượng hóa có thể dùng để phân lớp các thực thể.
- Là một cơ chế **tái sử dụng** ở cả hai mức thiết kế và lập trình.
- Đồ thị thừa kế là một nguồn cho kiến thức tổ chức về miền ứng dụng và hệ thống.

Nhược điểm của thừa kế

- Các lớp đối tượng không độc lập tự chủ
 - Không thể hiểu chúng mà không cần biết đến các lớp cha.
- Các nhà thiết kế có xu hướng tái sử dụng biểu đồ thừa kế đã được lập trong pha phân tích.
 - Có thể dẫn đến sự không hiệu quả đáng kể.
- Các biểu đồ thừa kế sản phẩm của phân tích, thiết kế và cài đặt có các chức năng khác nhau và nên được duy trì ở trạng thái tách biệt.

UML association

- Đối tượng và các lớp đối tượng có quan hệ với các đối tượng và lớp khác.



Các đối tượng song song

- Về bản chất, các đối tượng có tính độc lập (self-contained). Do đó, chúng phù hợp cho cài đặt song song.
- Mô hình truyền thông điệp của các đối tượng có thể được cài trực tiếp nếu các đối tượng chạy trên các bộ vi xử lý khác nhau trong một hệ phân tán.

Server và các đối tượng chủ động

- Server
 - Đối tượng được cài như là một tiến trình song song (server) với các điểm vào (entry point) tương ứng với các thao tác của object.
 - Nếu không có lời gọi nào được thực hiện đối với object, nó tự vào trạng thái nghỉ (suspend) và đợi các yêu cầu dịch vụ.
- Các đối tượng chủ động
 - Các đối tượng được cài như là các tiến trình song song và trạng thái trong của đối tượng có thể được thay đổi bởi chính đối tượng chứ không chỉ bởi các lời gọi thao tác từ bên ngoài.

Active transponder object

- thuộc tính của các đối tượng chủ động có thể bị sửa đổi bởi các thao tác kích hoạt từ ngoài nhưng cũng có thể được tự động cập nhật bởi các thao tác bên trong
- Một đối tượng **transponder** gửi broadcast vị trí của một máy bay.
 - Vị trí của máy bay có thể được cập nhật bằng cách sử dụng hệ thống định vị vệ tinh.
 - Đối tượng định kì cập nhật vị trí bằng thông tin nhận được từ các vệ tinh (phương pháp tam giác).

An active transponder object

```
class Transponder extends Thread {  
    Position currentPosition ;  
    Coords c1, c2 ;  
    Satellite sat1, sat2 ;  
    Navigator theNavigator ;
```

Thread (luồng) trong Java là một cấu trúc đơn giản để cài đặt các đối tượng song song

```
    public Position givePosition () {  
        return currentPosition ;  
    }
```

Các thread phải chứa một phương thức run() và nó được khởi động bởi Java run-time system.

```
    public void run () {  
        while (true) {  
            c1 = sat1.position () ;  
            c2 = sat2.position () ;  
            currentPosition = theNavigator.compute (c1, c2) ;  
        }  
    }  
} //Transponder
```

Các đối tượng chủ động thường có một vòng lặp vô hạn để chúng liên tục thực hiện tính toán.

Một quy trình thiết kế hướng đối tượng

- Các quy trình thiết kế có cấu trúc bao gồm việc phát triển một loạt các mô hình hệ thống.
- Việc này đòi hỏi nhiều công sức phát triển và bảo trì các mô hình này, và có thể không hiệu quả về chi phí đối với các hệ thống nhỏ.
- Tuy nhiên, với các hệ thống lớn do các nhóm khác nhau cùng phát triển, các mô hình hệ thống lại là một cơ chế giao tiếp quan trọng.

Các giai đoạn của quy trình

- Đánh dấu các hoạt động quan trọng mà không bị phụ thuộc vào bất cứ một quy trình có bản quyền nào (chẳng hạn RUP).
 - Xác định ngữ cảnh và các hình thức sử dụng hệ thống;
 - Thiết kế kiến trúc hệ thống;
 - Xác định các đối tượng quan trọng của hệ thống;
 - Phát triển các mô hình thiết kế;
 - Đặc tả giao diện đối tượng.

Miêu tả hệ thống thời tiết

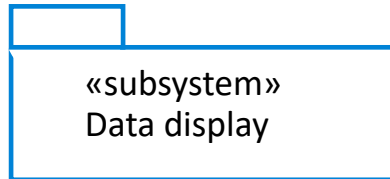
A **weather mapping system** is required to generate weather maps on a regular basis using data collected from remote, unattended weather stations and other data sources such as weather observers, balloons and satellites. Weather stations transmit their data to the area computer in response to a request from that machine.

The area computer system validates the collected data and integrates it with the data from different sources. The integrated data is archived and, using data from this archive and a digitised map database a set of local weather maps is created. Maps may be printed for distribution on a special-purpose map printer or may be displayed in a number of different formats.

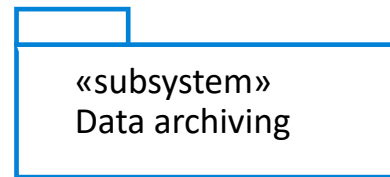
Ngữ cảnh hệ thống và mô hình sử dụng

- Các quan hệ giữa phần mềm đang được thiết kế và môi trường bên ngoài
- Ngữ cảnh hệ thống
 - Một mô hình tĩnh mô tả các hệ thống khác trong môi trường.
 - Dùng một mô hình hệ thống con (subsystem model) để mô tả các hệ thống khác.
- Mô hình sử dụng hệ thống
 - Một mô hình động mô tả tương tác của hệ thống với môi trường.
 - Dùng các use-case để mô tả các tương tác

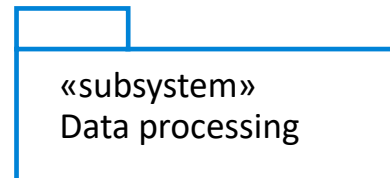
Kiến trúc phân tầng



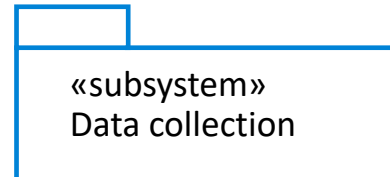
Tầng hiển thị dữ liệu, nơi các đối tượng có nhiệm vụ chuẩn bị và trình bày dữ liệu ở dạng con người đọc được



Tầng lưu trữ dữ liệu, nơi các đối tượng có nhiệm vụ lưu trữ dữ liệu để phục vụ các xử lý sau này

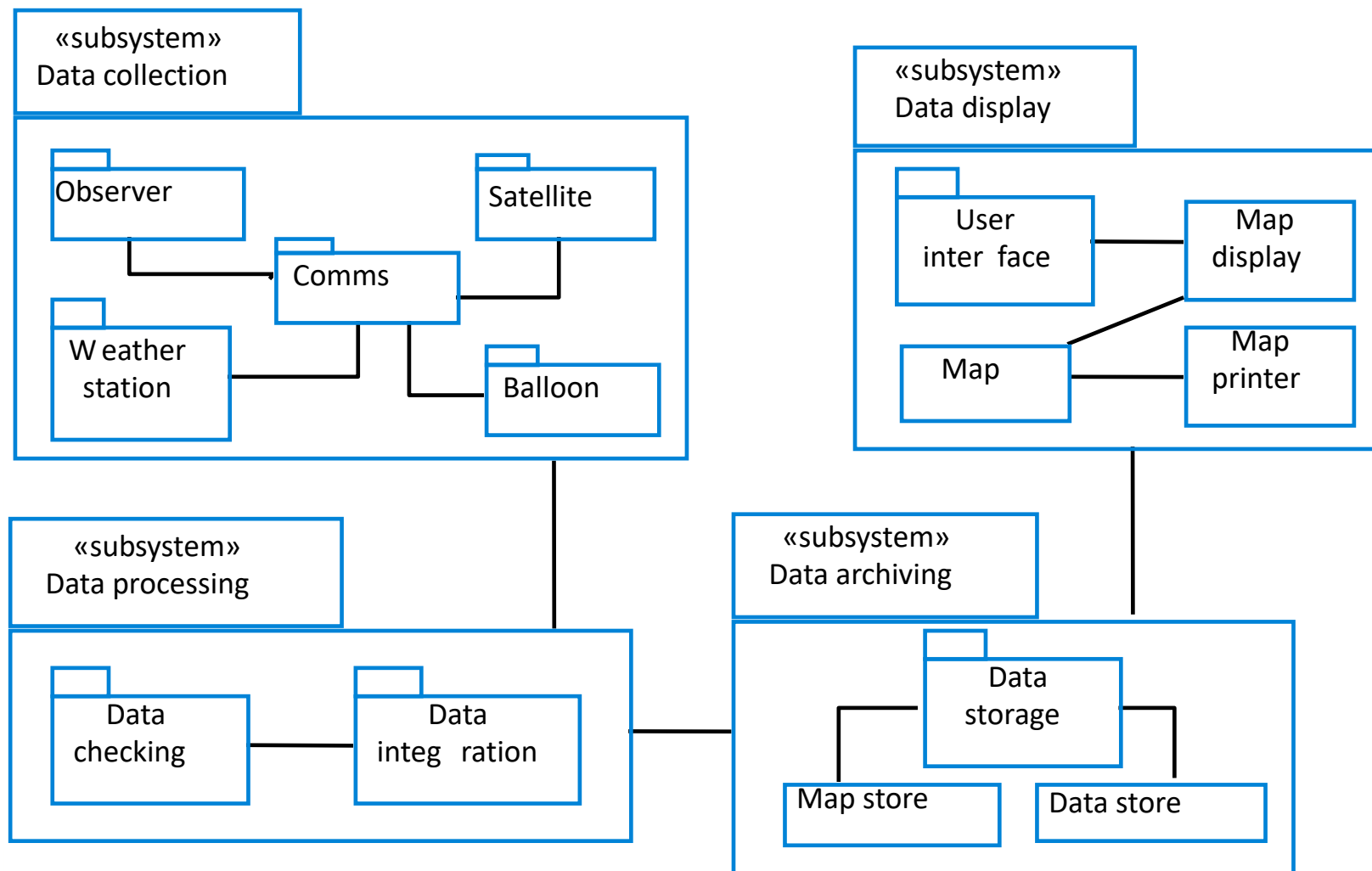


Tầng xử lý dữ liệu, nơi các đối tượng có nhiệm vụ kiểm tra và tích hợp dữ liệu đã thu thập được



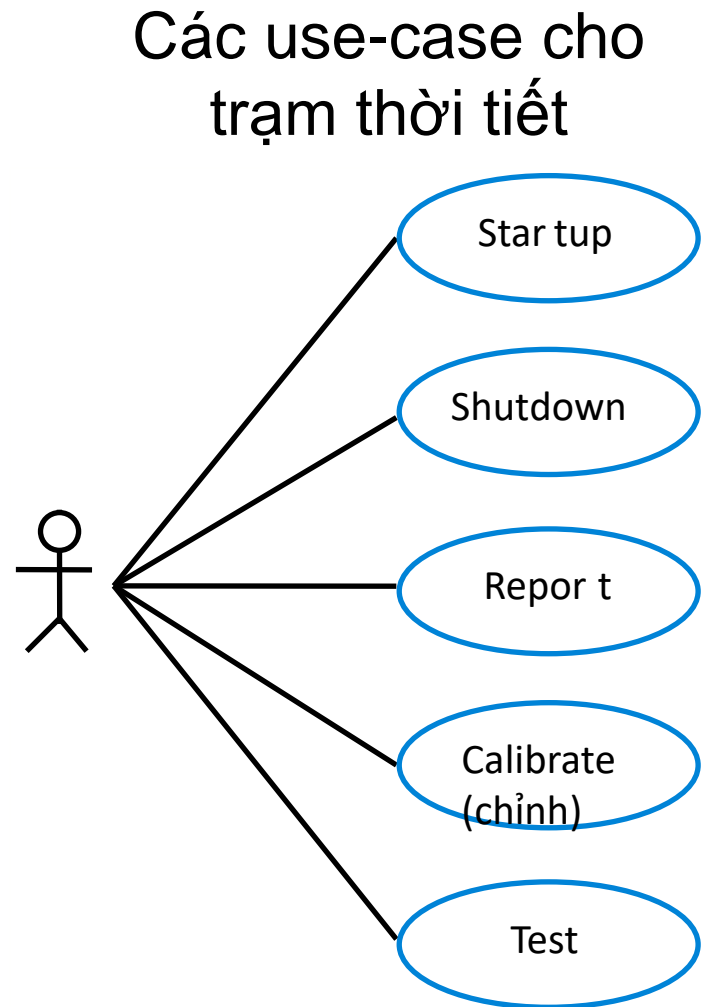
Tầng thu thập dữ liệu, nơi các đối tượng có nhiệm vụ thu thập dữ liệu từ các nguồn ở xa

Các hệ thống con trong hệ thống bản đồ thời tiết



Các mô hình use-case

- Các mô hình use-case dùng để biểu diễn từng tương tác với hệ thống.
- Các hình e-líp biểu thị các chức năng của hệ thống, hình người biểu thị thực thể tương tác.



Mô tả use-case

System Weather station

Use-case Report

Actors Weather data collection system, Weather station

Data The weather station sends a summary of the weather data that has been collected from the instruments in the collection period to the weather data collection 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 5 minute intervals.

Stimulus The weather data collection system establishes a modem link with the weather station and requests transmission of the data.

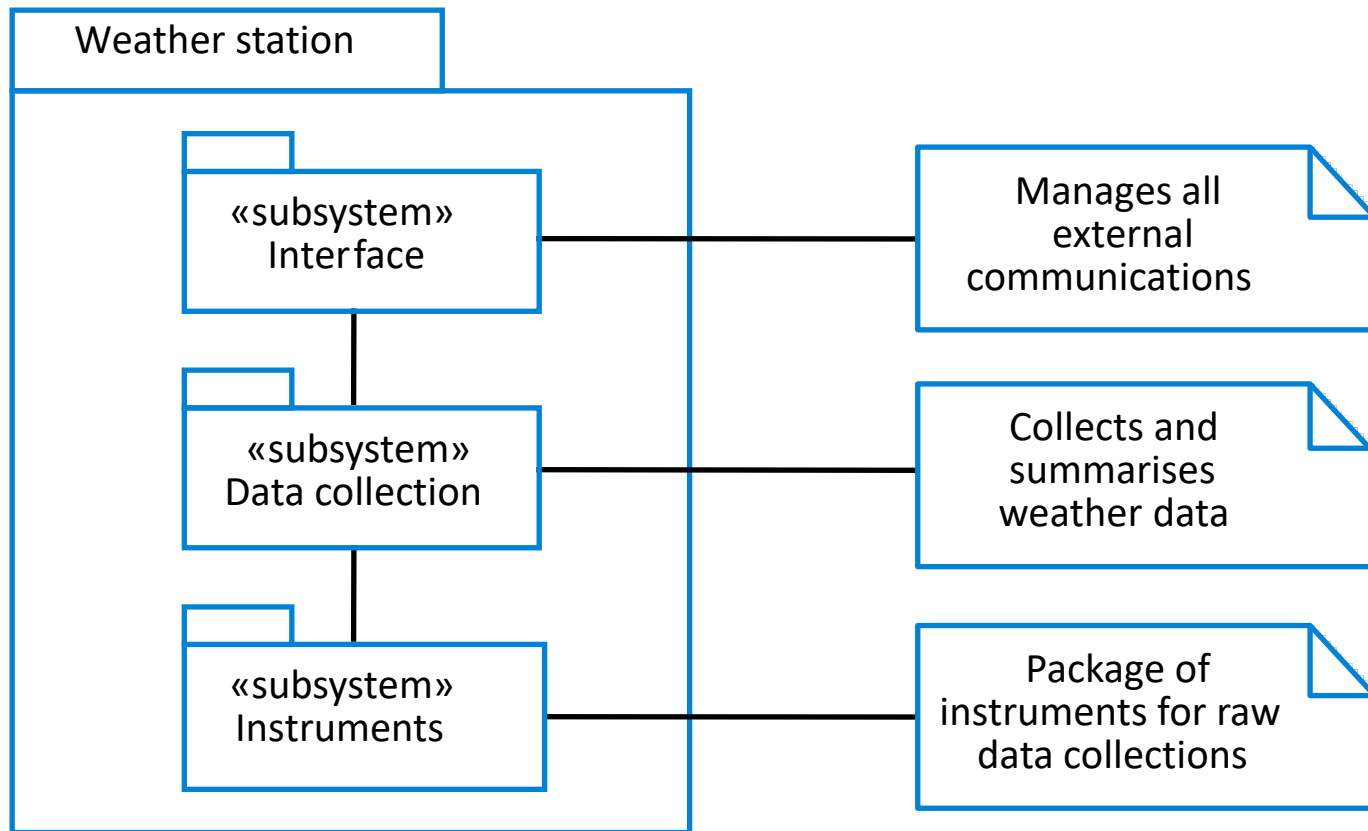
Response The summarised data is sent to the weather data collection system

Comments Weather stations are usually asked to report once per hour but this frequency may differ from one station to the other and may be modified in future.

Thiết kế kiến trúc

- Kiến trúc phân tầng (chương 11) thích hợp cho trạm thời tiết
 - Tầng interface dành cho việc xử lý các liên lạc;
 - Tầng thu thập dữ liệu quản lý các thiết bị;
 - Tầng thiết bị thu thập dữ liệu.
- Thông thường không nên có nhiều hơn 7 thực thể trong một mô hình kiến trúc.

Kiến trúc trạm thời tiết



Xác định các đối tượng

- Việc xác định các đối tượng (hoặc lớp đối tượng) là phần khó khăn nhất của thiết kế hướng đối tượng.
- Không có 'công thức kì diệu' cho việc xác định đối tượng.
 - Nó phụ thuộc vào kĩ năng, kinh nghiệm, và kiến thức miền ứng dụng của người thiết kế hệ thống.
- Việc xác định đối tượng là một quy trình lặp đi lặp lại.
 - Bạn khó có thể làm đúng ngay từ đầu.

Các cách xác định đối tượng

- Dùng phân tích ngữ pháp dựa trên miêu tả hệ thống bằng ngôn ngữ tự nhiên (dùng trong phương pháp thiết kế hướng đối tượng Hood).
- Dựa trên những thứ hữu hình trong miền ứng dụng.
- Dùng cách tiếp cận hành vi và xác định các đối tượng dựa trên những gì tham gia vào hành vi nào.
- Dùng phân tích dựa theo kịch bản.
 - Xác định các đối tượng, thuộc tính và phương thức trong từng kịch bản.

Miêu tả trạm thời tiết

Một *trạm thời tiết* là một package gồm các thiết bị do phần mềm điều khiển, nó thu thập dữ liệu, xử lý một chút và truyền dữ liệu này đi để được xử lý tiếp. Các thiết bị bao gồm: máy đo nhiệt độ không khí, máy đo nhiệt độ đất, máy đo gió, chong chóng gió, khí áp kế, máy đo lượng mưa. Dữ liệu được thu thập định kì.

Khi nhận lệnh gửi dữ liệu thời tiết, trạm thời tiết xử lý và tổng hợp dữ liệu đã thu thập được. Dữ liệu kết quả được gửi tới máy tính tạo bản đồ khi nhận được yêu cầu.

Các lớp đối tượng trạm thời tiết

- Ground thermometer, Anemometer, Barometer
 - Các đối tượng thuộc miền ứng dụng là các đối tượng 'phần cứng' liên quan đến các thiết bị trong hệ thống.
- Weather station
 - Giao diện cơ bản của trạm thời tiết với môi trường của nó. Do đó nó phản ánh các tương tác được xác định trong mô hình use-case.
- Weather data
 - Đóng gói dữ liệu đã được tổng hợp từ các thiết bị.

Các lớp đối tượng trạm thời tiết

WeatherStation
identifier
reportWeather () calibrate (instruments) test () startUp (instruments) shutdown (instruments)

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

Ground thermometer
temperature
test () calibrate ()

Anemometer
windSpeed windDirection
test ()

Barometer
pressure height
test () calibrate ()

Các đối tượng khác và làm mịn đối tượng

- Dùng kiến thức về miền ứng dụng để xác định các đối tượng và thao tác khác
 - Các trạm thời tiết nên có định danh riêng;
 - Các trạm thời tiết thường đặt ở những nơi xa xôi nên các sự cố thiết bị phải được báo cáo tự động. Do đó cần đến các thuộc tính và thao tác cho việc tự kiểm tra.
- Các đối tượng chủ động hoặc thụ động
 - Trong trường hợp này, các đối tượng thụ động và thu thập dữ liệu theo yêu cầu thay vì tự chủ. Nếu muốn có sự linh động thì sẽ cần đến thời gian xử lý của bộ điều khiển.

Các mô hình thiết kế

- Các mô hình thiết kế mô tả các đối tượng, lớp đối tượng và mối quan hệ giữa các thực thể đó.
 - Các mô hình tĩnh mô tả cấu trúc tĩnh của hệ thống bằng các lớp đối tượng và quan hệ giữa chúng.
 - Các mô hình động mô tả các tương tác động giữa các đối tượng.

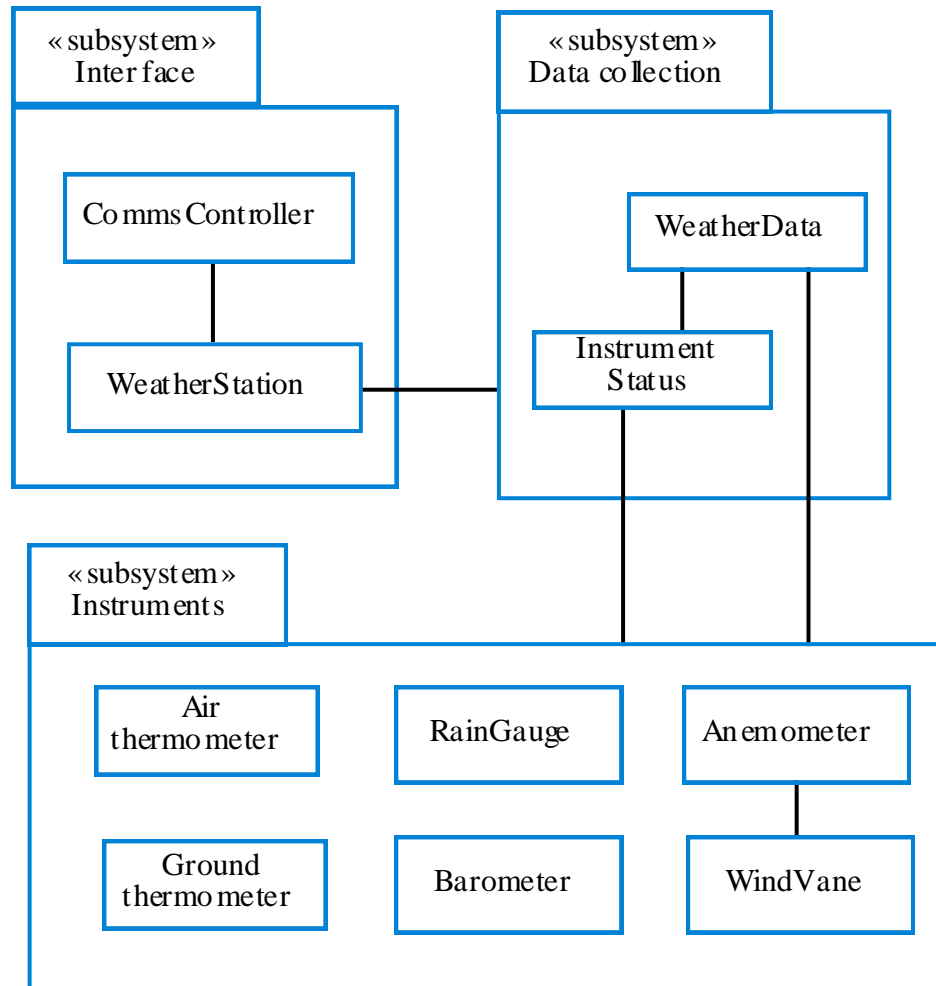
Ví dụ về các mô hình thiết kế

- **Mô hình hệ thống con** mô tả sự phân nhóm lô-gic các đối tượng thành các hệ thống con có tính gắn kết cao.
- **Mô hình tuần tự** mô tả trình tự các tương tác đối tượng.
- **Sơ đồ trạng thái** mô tả các đối tượng thay đổi trạng thái như thế nào để đáp ứng các sự kiện.
- Các mô hình khác **use-case models, aggregation models, generalisation models, etc.**

Các mô hình hệ thống con

- Miêu tả thiết kế được tổ chức thành các nhóm đối tượng có liên quan đến nhau về lô-gic.
- Trong UML, các hệ thống con được thể hiện bằng các package – một cấu trúc đóng gói. Đây chỉ là mô hình lô-gic, tổ chức thực tế của các đối tượng trong hệ thống có thể không như vậy.

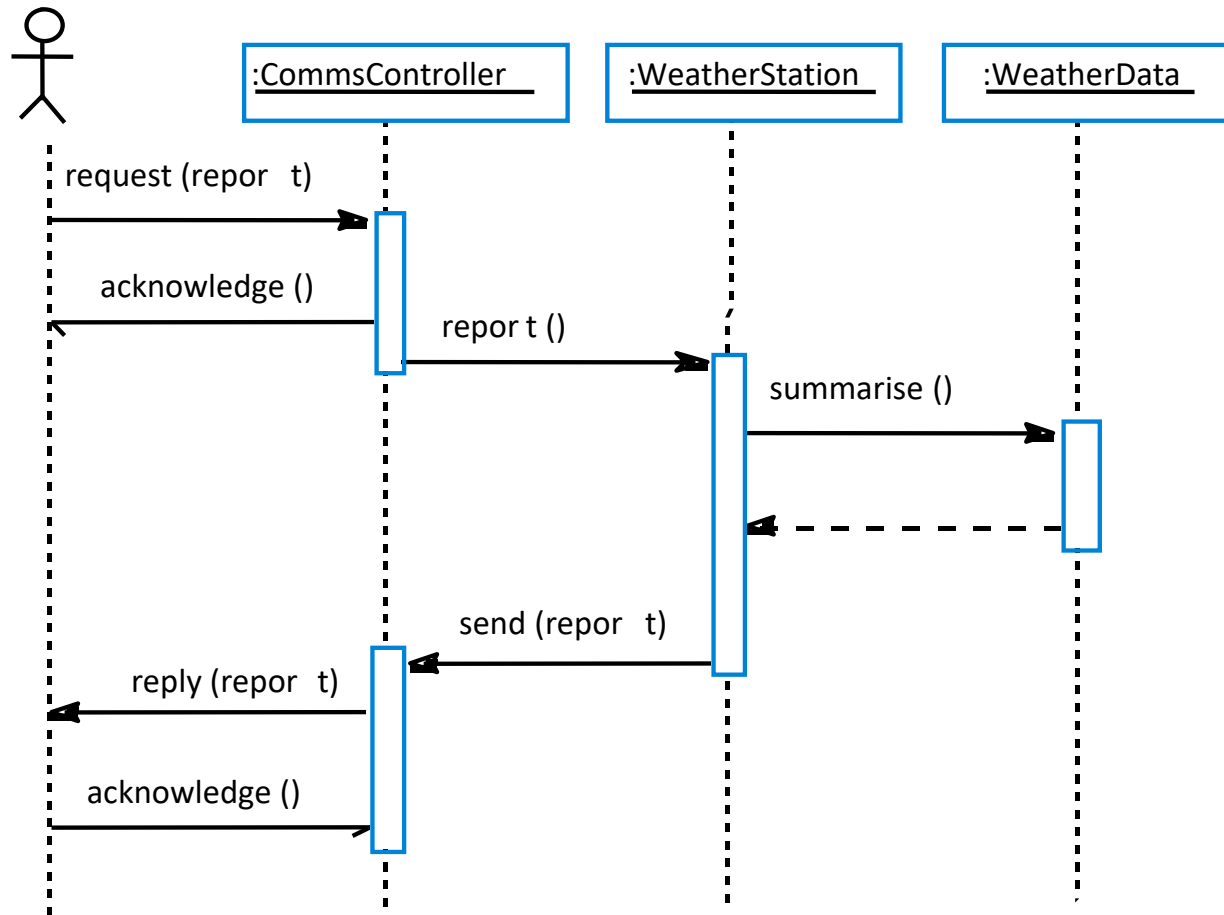
Weather station subsystems



Các mô hình tuần tự

- Các mô hình tuần tự mô tả trình tự của các tương tác xảy ra giữa các đối tượng
 - Các đối tượng xếp thành 1 hàng ngang trên đỉnh;
 - Thời gian được biểu diễn theo chiều dọc, mô hình được đọc từ trên xuống dưới;
 - Các tương tác được biểu thị bằng các mũi tên có nhãn. Các dạng mũi tên khác nhau đại diện cho các loại tương tác khác nhau;
 - Hình chữ nhật mỏng trong khoảng thời gian sống của một đối tượng đại diện cho khoảng thời gian mà đối tượng đó là đối tượng giữ vai trò kiểm soát hệ thống.

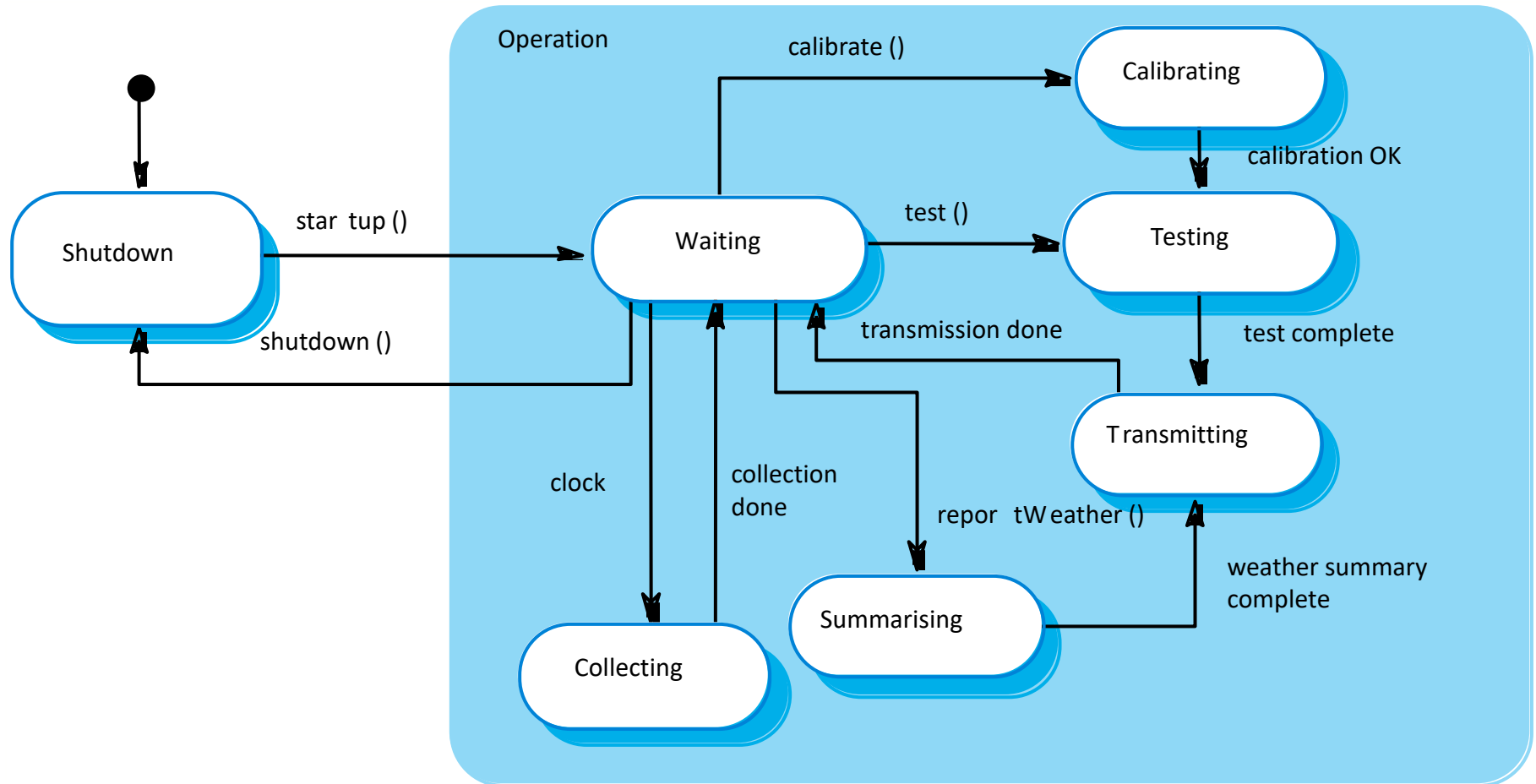
Biểu đồ tuần tự thu thập dữ liệu



Sơ đồ trạng thái

- Minh họa cách các đối tượng phản ứng với các yêu cầu dịch vụ khác nhau và việc chuyển trạng thái được kích hoạt bởi các yêu cầu đó
 - Nếu trạng thái của đối tượng là Shutdown thì nó đáp ứng một thông điệp Startup();
 - Trong trạng thái waiting, đối tượng đang đợi các thông điệp tiếp theo;
 - Nếu reportWeather () thì hệ thống chuyển sang trạng thái tổng hợp (summarising state);
 - Nếu calibrate () thì hệ thống chuyển sang trạng thái điều chỉnh (calibrating);
 - Hệ thống vào trạng thái collecting khi nhận được một tín hiệu đồng hồ.

Sơ đồ trạng thái trạm thời tiết



Đặc tả giao diện đối tượng

- Các giao diện đối tượng phải được đặc tả để các đối tượng và các thành phần khác có thể được thiết kế song song.
- Người thiết kế cần tránh thiết kế biểu diễn giao diện (interface representation) mà nên giấu nó trong chính đối tượng.
- Các đối tượng có thể có vài giao diện, đó là các viewpoint dựa trên các phương thức được cung cấp.
- UML dùng biểu đồ lớp để đặc tả giao diện hoặc có thể dùng cả Java.

Giao diện trạm thời tiết

```
interface WeatherStation {  
    public void WeatherStation();  
    public void startup();  
    public void startup(Instrument i);  
    public void shutdown();  
    public void shutdown(Instrument i);  
    public void reportWeather();  
    public void test();  
    public void test(Instrument i);  
    public void calibrate(Instrument i);  
    public int getID();  
} //WeatherStation
```


Tiến hóa thiết kế

- Che dấu thông tin ở bên trong các đối tượng có nghĩa là các thay đổi đối với đối tượng không ảnh hưởng tới các đối tượng khác một cách không đoán trước được.
- Giả sử các thiết bị đo ô nhiễm được bổ sung cho các trạm thời tiết. Các thiết bị này lấy mẫu không khí và tính lượng các chất ô nhiễm trong không khí.
- Số liệu về tình trạng ô nhiễm được gửi cùng dữ liệu thời tiết.

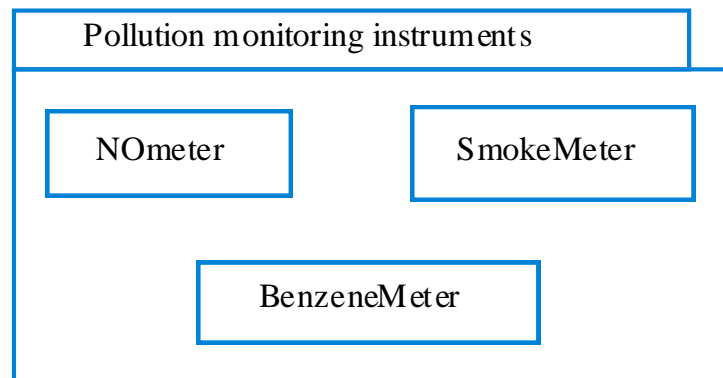
Các thay đổi cần thiết

- Thêm lớp đối tượng AirQuality vào thành một phần của WeatherStation.
- Thêm thao tác reportAirQuality vào WeatherStation. Sửa phần mềm điều khiển để thu thập số liệu ô nhiễm.
- Thêm các đối tượng đại diện cho các thiết bị đo độ ô nhiễm.

Pollution monitoring

WeatherStation
identifier
reportWeather () reportAirQuality () calibrate (instruments) test () startUp (instruments) shutdown (instruments)

Air quality
NOData smokeData benzeneData
collect () summarise ()



Tổng kết

- OOD là cách tiếp cận đối với thiết kế sao cho các thành phần thiết kế có trạng thái riêng và các thao tác riêng.
- Các đối tượng nên có constructor và các thao tác kiểm tra.
 - Cung cấp dịch vụ cho các đối tượng khác.
- Các đối tượng có thể được cài đặt lần lượt hoặc song song.
- Ngôn ngữ UML cung cấp các kí pháp khác nhau để định nghĩa các mô hình đối tượng khác nhau.

Tổng kết

- Quy trình OOD dùng đến một loạt các mô hình khác nhau.
 - Trong đó có các mô hình hệ thống tĩnh và động.
- Các giao diện đối tượng nên được định nghĩa chính xác, chẳng hạn bằng cách sử dụng một ngôn ngữ lập trình như Java/C++/Python.
- Thiết kế hướng đối tượng có tiềm năng đơn giản hóa việc tiến hóa hệ thống.