

Arquitetura de Software – exemplo usando UML

- O exemplo, a seguir, foi tirado do capítulo 12 do livro do Ian Sommerville
- A idéia é design orientado a objetos, não necessariamente design arquitetural
- Ele descreve o design da arquitetura de um sistema de mapeamento meteorológico e outros modelos de design
- São apresentados vários modelos usando UML

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Sistema de mapeamento meteorológico

A weather data collection 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 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.

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

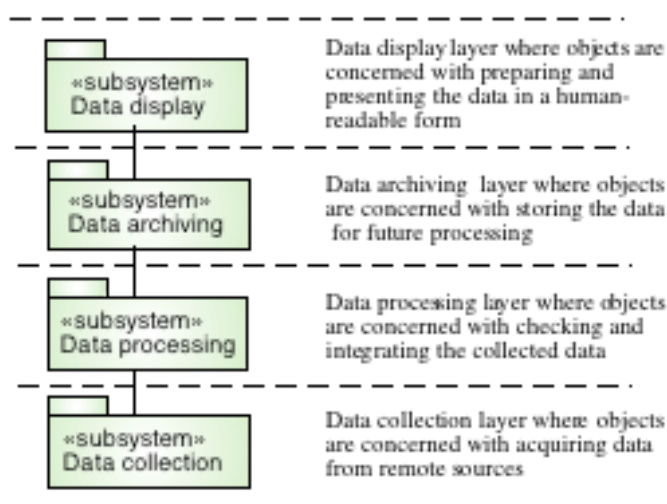
Descrição da estação meteorológica

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, a wind vane, a barometer and a rain gauge. Data is collected every five minutes.

When a command is issued to transmit the weather data, the weather station processes and summarises the collected data. The summarised data is transmitted to the mapping computer when a request is received.

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

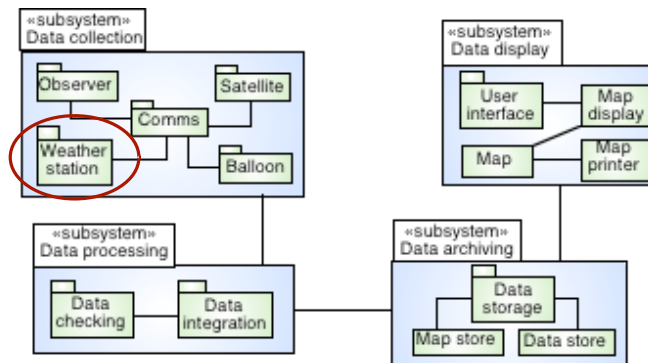
Arquitetura em camadas



Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Contexto do Sistema

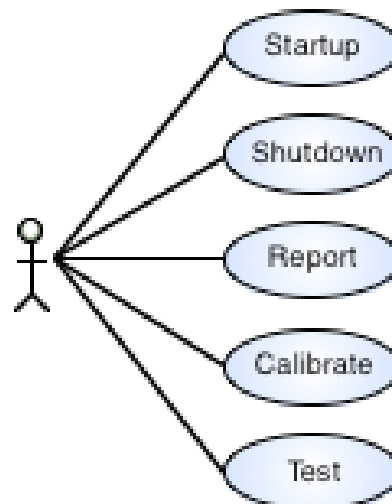
- Desenvolver uma compreensão dos relacionamentos entre o software e o seu ambiente externo
- Descrito usando o estereótipo <<subsistema>> da UML.



Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Modelo do Uso do Sistema – Casos de Uso

- Descreve como o sistema interage com o seu ambiente.
- Os casos de uso descrevem esta interação
- Cada caso de uso precisa ser descrito na forma de cenários



Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

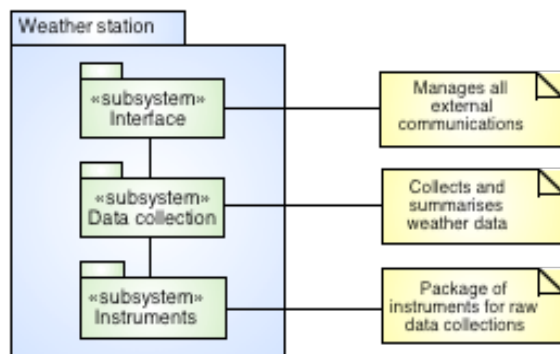
Exemplo de cenário

- Sistema: estação meteorológica
- Caso de Uso: Relatar (report)
- Atores: Sistema de coleta de dados, Estação meteorológica
- Cenário:
 - A estação meteorológica envia para o SCDC um resumo dos dados sobre o clima, que foram coletados a partir de instrumentos, no período de coleta. Os dados enviados referem-se às temperaturas máximas, mínimas e médias do solo e do ar; à pressão máxima, mínima e média do ar; à precipitação total das chuvas e à direção do vento, conforme a amostragem em intervalos de cinco minutos.
- Pré-condição (estímulo):
 - link para transmissão dos dados
- Pós-condição (resposta):
 - os dados resumidos são enviados para o sistema de coleta

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Arquitetura da estação meteorológica

- As interações entre o sistema e o seu ambiente podem ser a base para a arquitetura do sistema.
- Uma arquitetura em camadas é apropriada para o sistema



- Uma camada Interface para tratar a comunicação

- A camada coleta de dados para gerenciar os instrumentos

- A camada Instrumentos para a coleta dos dados

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Identificação dos objetos

- Não existe mágica, a identificação depende da experiência e do conhecimento do domínio.
- É um processo iterativo, que pode ser modificado a cada rodada.
- Utiliza-se a descrição dos requisitos (cenários), onde são identificados os objetos e atributos.
 - São substantivos nas sentenças dos cenários.

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Classes de objetos identificados

- Ground thermometer, Anemometer, Barometer
 - Baseados nos instrumentos do sistema
- Weather station
 - Interface da estação com o ambiente
- Weather data
 - Encapsula um resumo dos dados dos instrumentos

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

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

Ground thermometer
temperature
test ()
calibrate ()

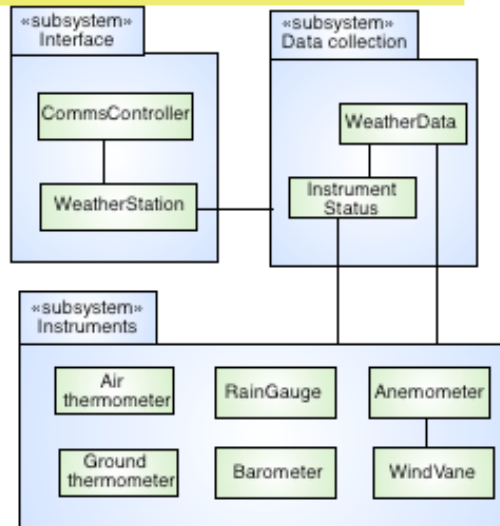
Anemometer
windSpeed
windDirection
test ()

Barometer
pressure
height
test ()
calibrate ()

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Objetos e sub-sistemas

- Mostra como os objetos estão agrupados em sub-sistemas e módulos
- Descreve a visão lógica dos objetos, segundo Sommerville
- Descreve a visão de módulos, segundo Hofmeister e outros.



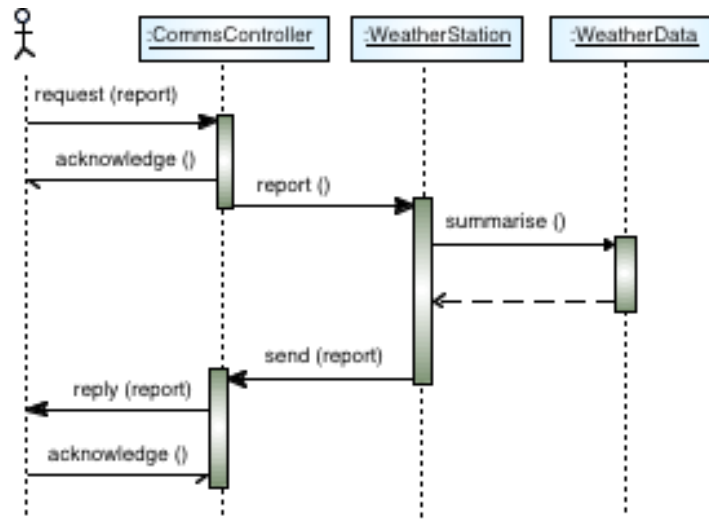
Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Outros modelos – visão lógica

- Modelos de sub-sistemas mostram o agrupamentos de lógico de objetos. Podem incluir módulo. (E)
- Modelos de seqüência mostram a interação entre objetos ao longo do tempo. (C)
- Máquinas de estados modelam como um objeto muda de estado em resposta a eventos.(C)
- Outros modelos podem mostrar outras visões.
- E = estrutura; C = comportamento

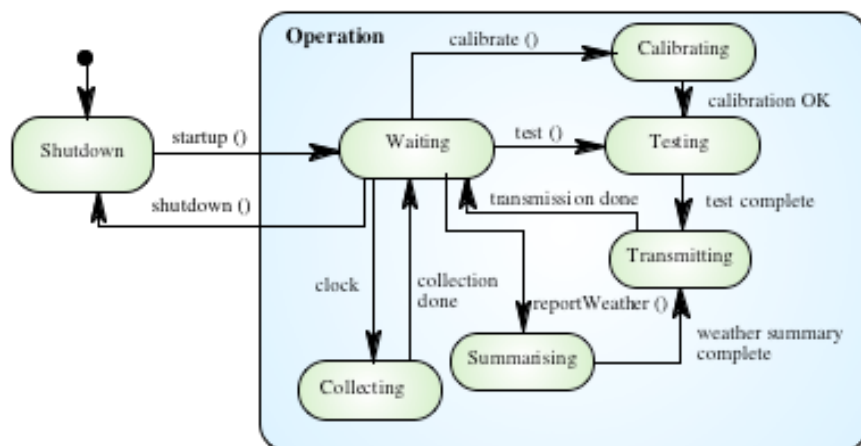
Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Diagrama de seqüência



Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Diagrama de estados da estação



Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite

Interface da estação

```
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
```

Baseando em Engenharia de Software, cap. 12, Ian Sommerville, 2000, adaptado por Jair C Leite