CptS 487
Software Design and Architecture

Lesson 15

Intro to Software Architecture

WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

Instructor: Bolong Zeng

Overview

- Introduction to the concept of Software Architecture
- Understand Subsystem, Component, Service and Interface

Software Architecture

- *Software architecture* is process of designing the global organization of a software system, including:
  - Dividing software into subsystems.
  - Deciding how these will interact.
  - Defining global control flow
  - Determining their interfaces.
    - The architecture is the core of the design.
    - The architecture will often constrain the overall efficiency, reusability and maintainability of the system.

The Importance of Software Architecture

- Why you need to develop an architectural model:
  - To enable everyone to better understand the system (deal with complexity)
  - To allow people to work on individual pieces of the system in isolation
  - To prepare for extension of the system
  - To facilitate reuse and reusability

Contents of a Good Architectural Model

- A system's architecture will often be expressed in terms of several different *views*
  - The logical breakdown into subsystems
  - The interfaces among the subsystems
  - The dynamics of the interaction among components at run time
  - The data that will be shared among the subsystems
  - The components that will exist at run time, and the machines or devices on which they will be located

Design a Stable Architecture

- To ensure the maintainability and reliability of a system, an architectural model must be designed to be *stable*.
  - Being stable means that the new features can be easily added with only small changes to the architecture

Developing an Architectural Model

- Start by sketching an outline of the architecture
  - Based on the principal requirements and use cases
  - Determine the main components that will be needed
  - Choose among the various architectural patterns
    - Discussed next
  - *Suggestion*: have several different teams independently develop a first draft of the architecture and merge together the best ideas

Developing an Architectural Model

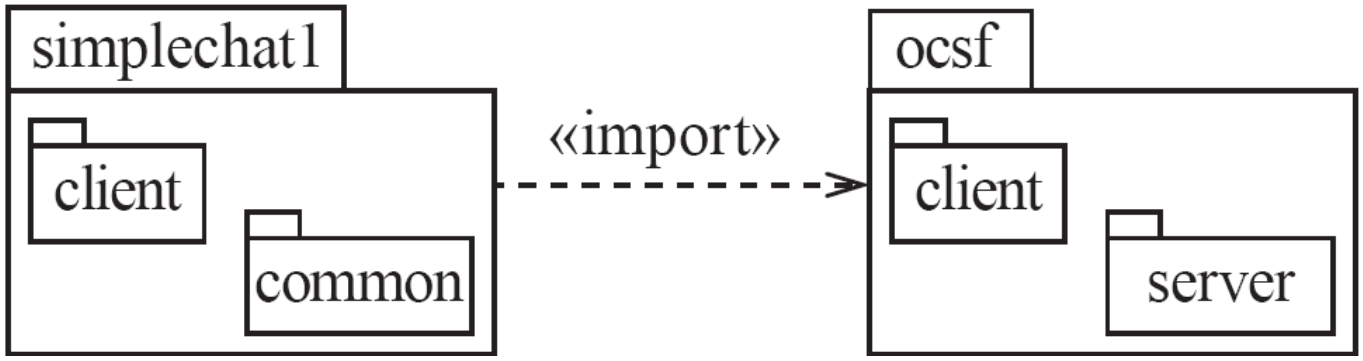
- Refine the architecture
  - Identify the main ways in which the components will interact and the interfaces between them
  - Decide how each piece of data and functionality will be distributed among the various components
  - Determine if you can re-use an existing framework, if you can build a framework
- Consider each use case and adjust the architecture to make it realizable
- Mature the architecture

Describing an Architecture Using UML

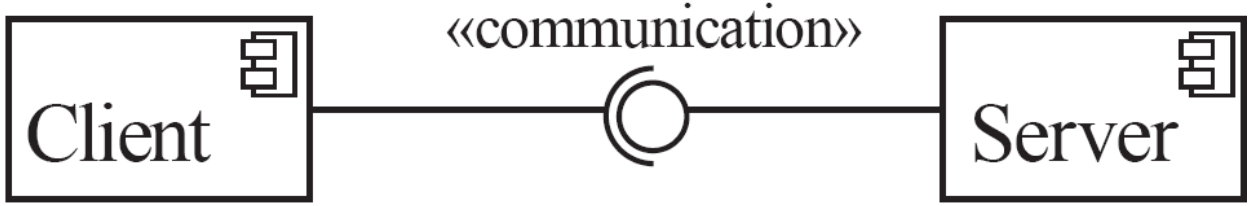
- All UML diagrams can be useful to describe aspects of the architectural model
- Three UML diagrams are particularly suitable for architecture modelling:
  - Package diagrams
  - Component diagrams
  - Deployment diagrams

  - We'll be focusing on Component diagram

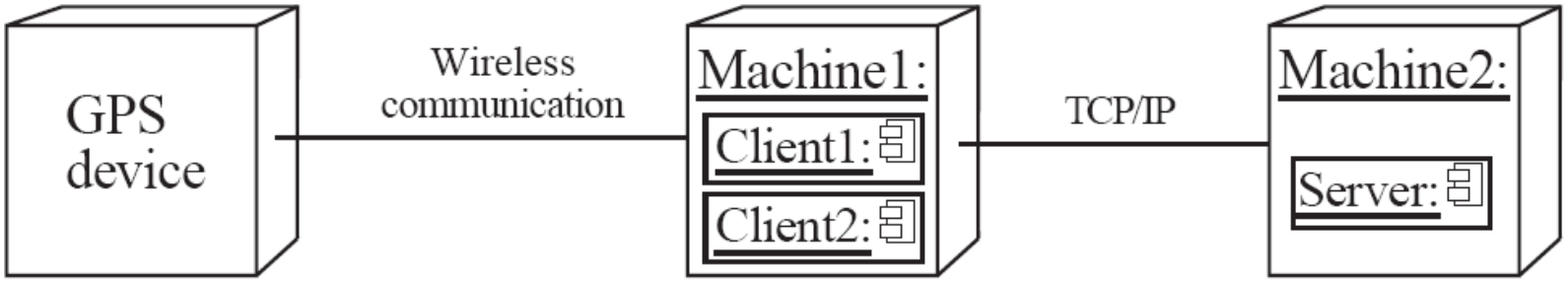
Package Diagrams



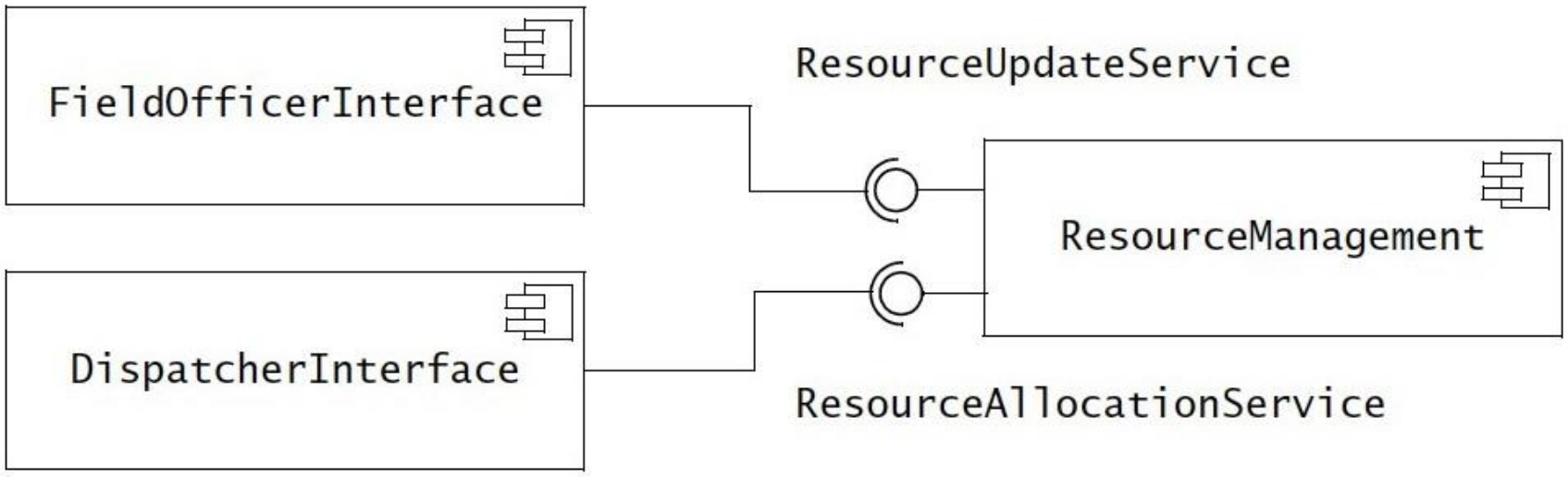
Component Diagrams

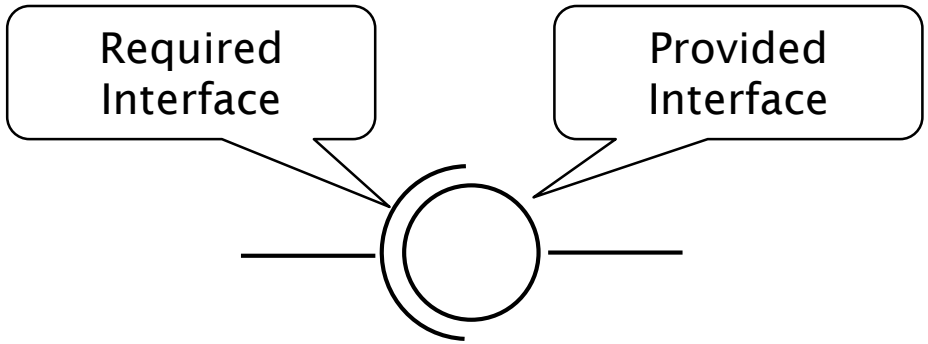


Deployment Diagrams



Component Diagram Example





- Ball-socket notation showing provided and required interfaces

Subsystem, Component

- Concrete example in daily life:
  - A home entertainment system.

Subsystem, Component

- Subsystem: a collection of one or more components that serve similar purposes

- Component: a collection of one or more classes that work together to achieve certain goals/purposes

Services, Interfaces

- How do we connect the subsystems/components together?

- Through interfaces!
  - In this example, what are the “interfaces”?

Services, Interfaces

- The whole system is connected together via interfaces, with different components providing and using each others' services.

Architectural Patterns

- The notion of patterns can be applied to software architecture.
  - These are called *architectural patterns* or *architectural styles*.
  - Each allows you to design flexible systems using components
    - The components are as independent of each other as possible.