A model of a data collector component

Requires interface
Provides interface

sensorManagement
sensorData

Data collector

addSensor
removeSensor
startSensor
stopSensor
testSensor
initialize
report
listAll
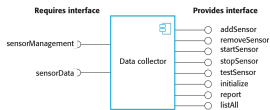
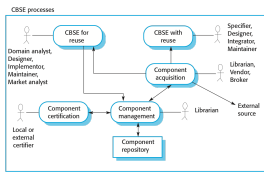## An adaptor linking a data collector and a sensor

start
stop
getdata

sensor

Adapter

sensorManagement
sensorData

Data collector

addSensor
removeSensor
startSensor
stopSensor
testSensor
initialize
report
listAll

Basic elements of a component model

Customisation
Naming convention
Composition
Documentation

Interface definition
Specific interfaces
Meta-data access
Packaging
Evolution support

Interfaces
Usage information
Deployment and use

Component model

CBSE processes

CBSE processes

CBSE for reuse
CBSE with reuse

Domain analyst, Designer, Implementor, Maintainer, Market analyst

Specifier, Designer, Integrator, Maintainer

Component acquisition

Librarian, Vendor, Broker

Component certification
Component management
Librarian
External source

Local or external certifier
Component repository

CBSE with reuse

Outline system requirements
Identify candidate components
Modify requirements according to discovered components

Architectural design
Identify candidate components
Compose components to create system

Component interfaces - Note UML notation. Ball and sockets can fit together.

Requires interface
Provides interface

Defines the services that are needed and should be provided by other components

Component

Defines the services that are provided by the component to other components

Components with incompatible interfaces

phoneDatabase (string command)

addressFinder

string location(string pn)
string owner (string pn)
string propertyType (string pn)

mapDB (string command)

mapper

displayMap (string postCode, scale)
printMap (string postCode, scale)

components_with_incompatible

phoneDatabase (string command)

addressFinder

string location(string pn)
string owner (string pn)
string propertyType (string pn)

mapDB (string command)

mapper

displayMap (string postCode, scale)
printMap (string postCode, scale)

Data collection and report generation components

(a)

Data collection
Data management
Report generator
Report

(b)

Data collection
Data base
Report
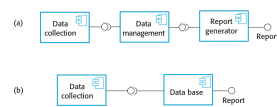
Middleware services defined in a component model

Support services

Component management
Transaction management
Resource management

Concurrency
Persistence
Security

Platform services

Addressing
Interface definition
Exception management
Component communications
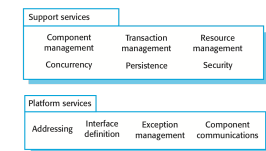
Photo library composition

Photo Library

addItem
retrieve
catEntry

adaptor

getImage

Image Manager

getCatalogEntry

User Interface
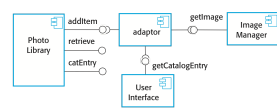
The component identification process
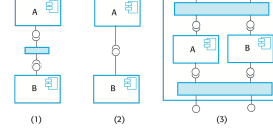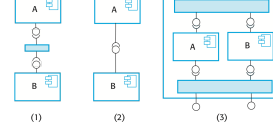
Component search
Component selection
Component validation

Types of component composition

A
A
A
B

B
B

(1)
(2)
(3)

types_of_component_composit

A
A
A
B

B
B

(1)
(2)
(3)

A decentralized p2p architecture

n4
n6
n8
n13

n2
n3
n7
n12
n9

n4
n10
n11
n13

n1
n5

A distributed component architecture

Comp 1
Common services

Comp 2
Common services

Comp 4
Common services

Comp 4
Common services

Communication middleware

Client
Client
Client
Client
Client
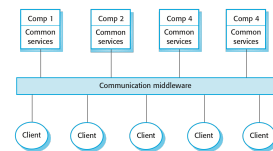
A distributed component architecture for a data mining system

Database 1
Integrator 1
Report gen.

Database 2
Visualizer

Integrator 2

Database 3
Display
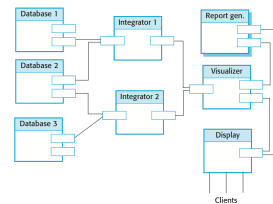
Clients

A fat-client architecture for an ATM system

ATM
ATM
ATM
ATM

Account server

Tele-processing monitor
Customer account database
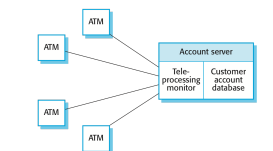
A multitenant database

Tenant	Key	Name	Address
234	C100	XYZ Corp	43, Anystreet, Sometown
234	C110	BigCorp	2, Main St, Motown
435	X234	J. Bowie	56, Mill St, Starville
592	PP37	R. Burns	Alloway, Ayrshire

A semicentralized p2p architecture

Discovery server (Super peer)

n1
n4
n6
n3
n5
n2
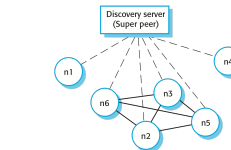
A traffic management system with a master-slave architecture

Control room processor

Sensor processor
Coordination and display process
Traffic light control processor

Sensor control process
Coordination and display process
Light control process

Master
Slave
Slave

Traffic flow sensors and cameras
Operator consoles
Traffic lights
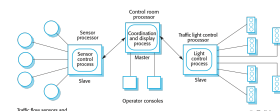
Client–server interaction

c2
c3
c10
c12

c1
s1
s4
c11

c5
c10

s2
c9

c6
c7
c8

Server process
Client process

Configuration of a software system offered as a service

User 1
User 2
User 3
User 4
User 5

Profile C1
Profile C2
Profile C3

Application service
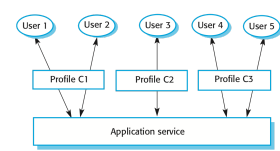
Layered architectural model for client–server applications

Presentation

Data handling

Application processing

Database
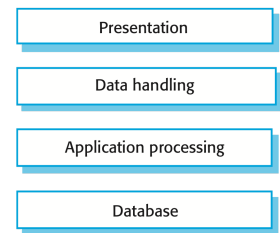
Mapping of clients and servers to networked computers

c1
s1, s2
c2
c3, c4

CC1
SC2
CC2
CC3

Network

c5, c6, c7
c8, c9
c10, c11, c12

CC4
SC1
CC5
CC6

s3, s4

Server computer
Client computer
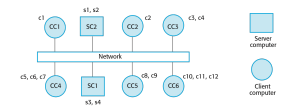
Message-based interaction between a waiter and the kitchen

```
<starter>
    <dish name = "soup" type = "tomato" />
    <dish name = "soup" type = "fish" />
    <dish name = "pigeon salad" />
</starter>
<main course>
    <dish name = "steak" type = "sirloin" cooking = "medium" />
    <dish name = "steak" type = "fillet" cooking = "rare" />
    <dish name = "sea bass" />
</main>
<accompaniment>
    <dish name = "french fries" portions = "2" />
    <dish name = "salad" portions = "1" />
</accompaniment>
```
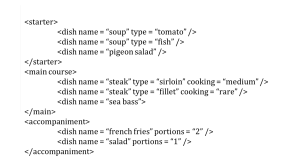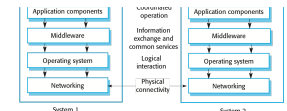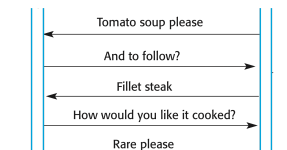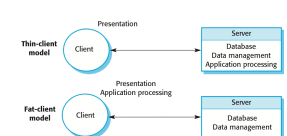
Middleware in a distributed system

Application components
Middleware
Operating system
Networking

System 1

Coordinated operation
Information exchange and common services
Logical interaction
Physical connectivity

Application components
Middleware
Operating system
Networking

System 2

Procedural interaction between a diner and a waiter

Tomato soup please
And to follow?
Fillet steak
How would you like it cooked?
Rare please

Thin- and fat-client architectural models

Presentation

Thin-client model
Client
Server
Database
Data management
Application processing

Presentation
Application processing

Fat-client model
Client
Server
Database
Data management

Three-tier architecture for an Internet banking system

Tier 1. Presentation

Client
Client
Client
Client

HTTPS interaction

Web server
Account service provision

SQL query

Database server
Customer account database

SQL

Tier 2. Application processing and data management

Tier 3. Database processing