Lecture 9 – Concurrent Task Structuring

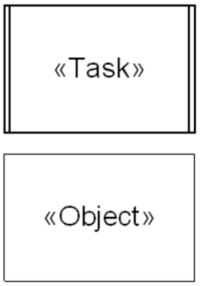
Reference: H. Gomaa, Chapter 18 - *Software Modeling and Design*, Cambridge University Press, February 2011

Structure Subsystem into Concurrent Tasks

- Concurrent task structuring criteria
  - Structure subsystem into concurrent tasks
    - Task is an active object
    - Task has thread of control
  - Consider concurrent nature of system activities
  - Determine concurrent tasks
- Define task interfaces

Active and Passive Objects

- Objects may be **active** or **passive**
- **Active object**
  - **Concurrent Task**
  - Has thread of control
- **Passive object**
  - a.k.a. **Information Hiding Object**
  - Has no thread of control
  - Operations of passive object are executed by task
- Software Design terminology
  - **Task** refers to active object
  - **Object** refers to passive object



3

Task Structuring Criteria

- Each task structured using two orthogonal criteria
  - Represented using stereotypes
    - Object role criterion (from analysis model)
    - Concurrency criterion (from task structuring)
- Concurrency criteria
  - Define how task is activated
    - Event driven task
      - Activated by external event (e.g., interrupt)
    - Periodic task
      - Activated by timer
    - Demand driven task
      - Activated by arrival of internal message

4

I/O Task Structuring Criteria

- Event driven I/O task
  - Task for each event (interrupt) driven I/O device
  - Event driven device generates interrupt
- Periodic I/O task
  - Task for each passive I/O device
  - I/O device (usually input) sampled at regular intervals (polling)
- Demand driven I/O task
  - Task for each passive I/O device (usually output)
  - Computation overlapped with output

5

Event Driven I/O Task

- One task for each event driven I/O device
  - Activated by device I/O interrupt
  - Reads input
  - Converts to internal format
- Disposes of input
  - Sends message containing data to another task
  - Signals event (message with no data)
  - Writes to data store
- Event driven proxy task
  - Interfaces to an external system by using messages

6

Figure 18.5 Example of event driven I/O task

Figure 18.5a Analysis model – communication diagram

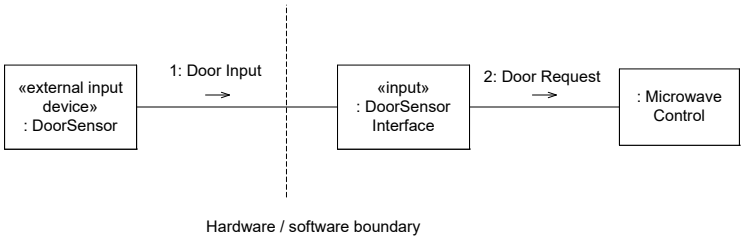
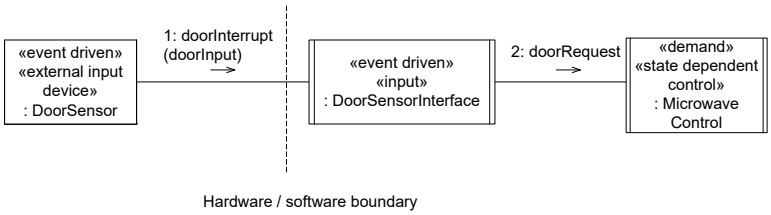


Figure 18.5b Design model – concurrent communication diagram



7

7

Periodic I/O Task

- Periodic I/O task for passive I/O device
  - Passive I/O device does not generate an interrupt
  - Activated periodically by timer event
  - Performs I/O operation (Samples I/O device)
  - Waits for next timer event
- Periodic I/O Task
  - Used for passive sensor devices
    - For sensor-based industrial system
      - Has many digital and analog sensors
    - Engine sensor

8

8

Figure 18.6 Example of a periodic I/O task

Figure 18.6a Analysis model – communication diagram

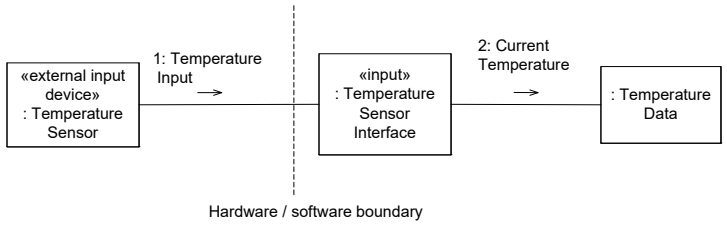
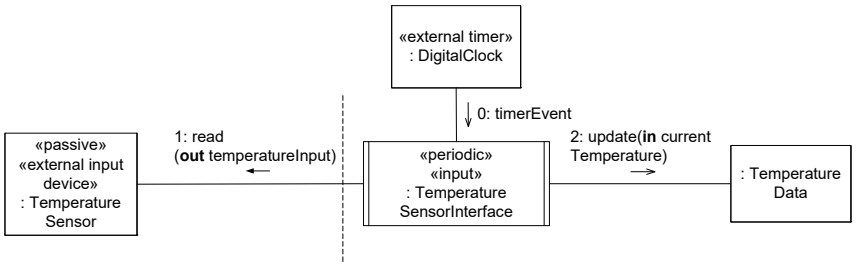


Figure 18.6b Design model – concurrent communication diagram



9

9

Demand Driven I/O Task

- Task for each passive I/O device
  - Used for passive I/O device that does not need to be polled
  - Computation overlapped with input/output
    - In output case:
      - Overlap the output to the device with computational task that produces the data
- Usually for passive output device
  - Demand driven I/O task

10

10

Figure 18.7 Example of a Demand Driven Output Task

Figure 18.7a Analysis model – communication diagram

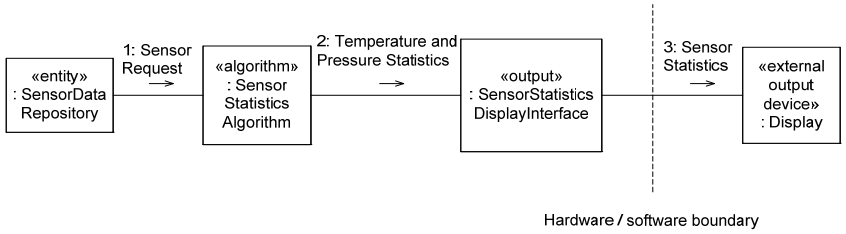
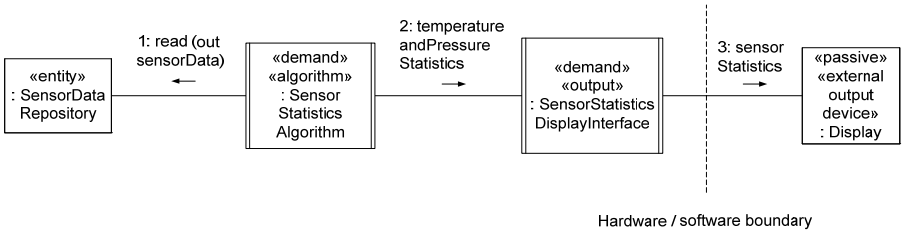


Figure 18.7b Design model – concurrent communication diagram



11

11

Internal Task Structuring Criteria

- Periodic task
  - Task for each periodic activity
- Demand task
  - Task for each demand-driven internal activity
- Control task (Demand Driven)
  - Task executes state machine
- User interaction task (Event Driven)
  - Task for each sequential user activity

12

12

Periodic Task

- Task for each periodic activity
- Task activated periodically
  - Activated by timer event
  - Performs activity
  - Waits for next timer event

13

Figure 18.8 Example of periodic task

Figure 18.8a Analysis model – communication diagram

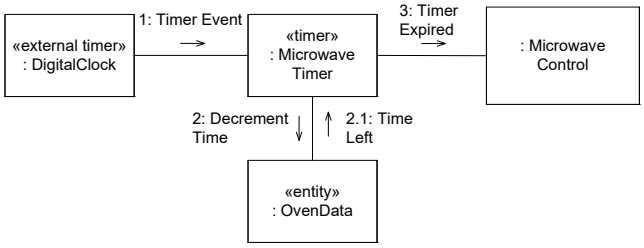
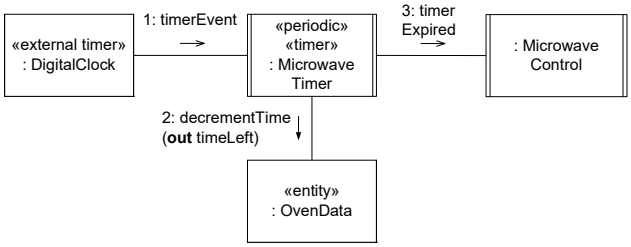


Figure 18.8b Design model – concurrent communication diagram



14

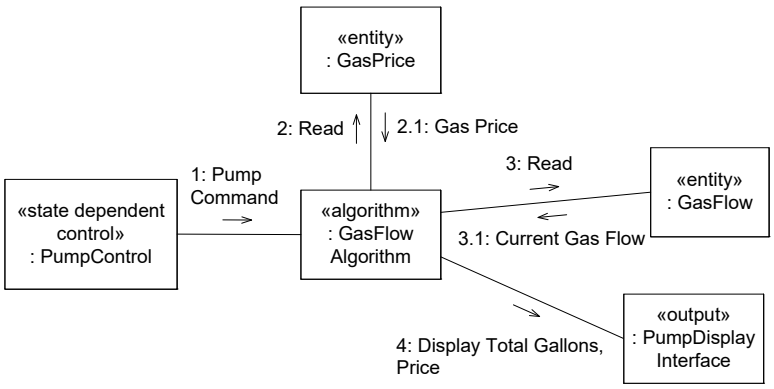
Demand Task

- Demand task
  - Activated on demand by event or message sent by different task
  - Performs demanded action
  - Waits for next event or message

15

Figure 18.9 Example of demand task

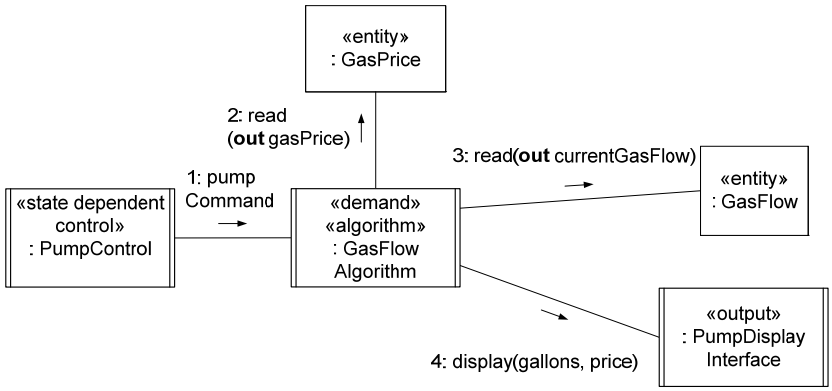
Figure 18.9a Analysis model – communication diagram



16

Figure 18.9 Example of demand task

Figure 18.9b Design model – concurrent communication diagram



17

17

Control Task

- Task executes statechart
  - State dependent control object executes statechart
  - Execution of statechart is sequential
- One task for each control object
- Can have multiple tasks of same type
  - E.g., multiple Elevator control tasks
- For coordinator object - designed as coordinator tasks
  - The job of the task is to control other tasks

18

18

Figure 18.10 Example of control task

Figure 18.10a Analysis model – communication diagram

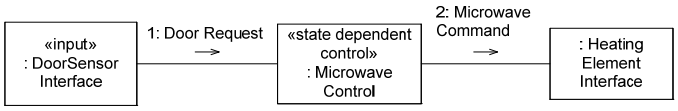
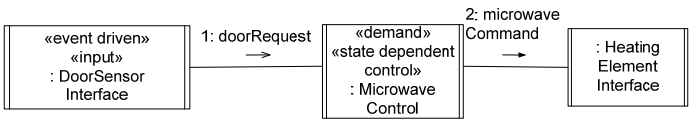


Figure 18.10b Design model – concurrent communication diagram



19

User Interaction Task

- User's interaction with system is sequential activity
  - User interaction objects are handled by user interaction tasks
- Usually event driven
  - User interaction task awakened by inputs from the external users
- No necessary to develop special-purpose I/O tasks to handle keyboard, mouse, display
  - because OS handles them

20

User Interaction Task

- One task for each sequential user activity
- Multi-user system
  - One task per user
- Windowing system
  - User engaged in multiple activities
  - Each window executes sequential activity
  - One task for each window

Figure 18.12 Example of user interaction task

Figure 18.12a Analysis model – communication diagram

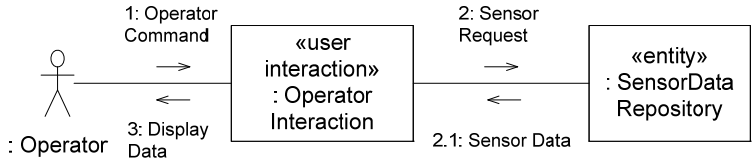


Figure 18.12b Design model – concurrent communication diagram

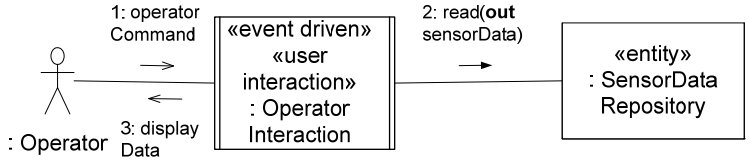
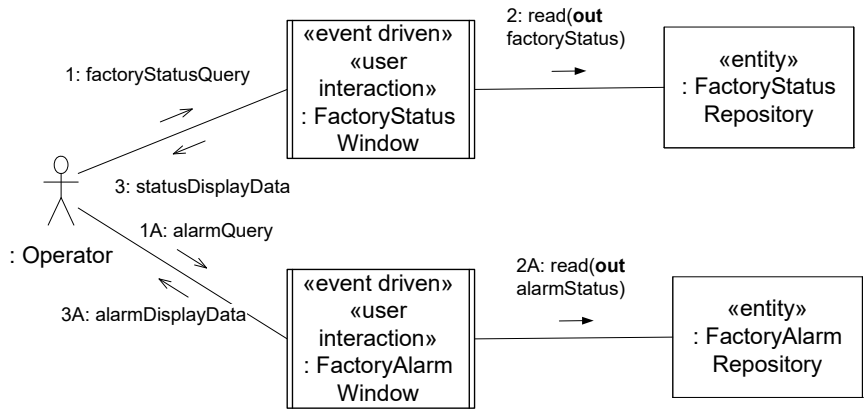


Figure 18.12 Example of user interaction task

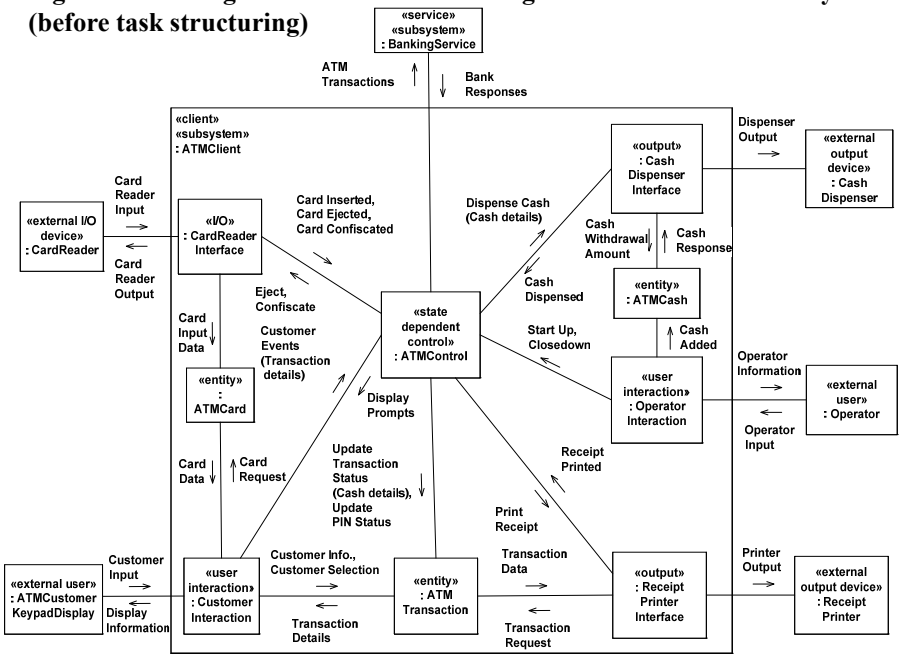
Figure 18.12c Design model – concurrent communication diagram



23

23

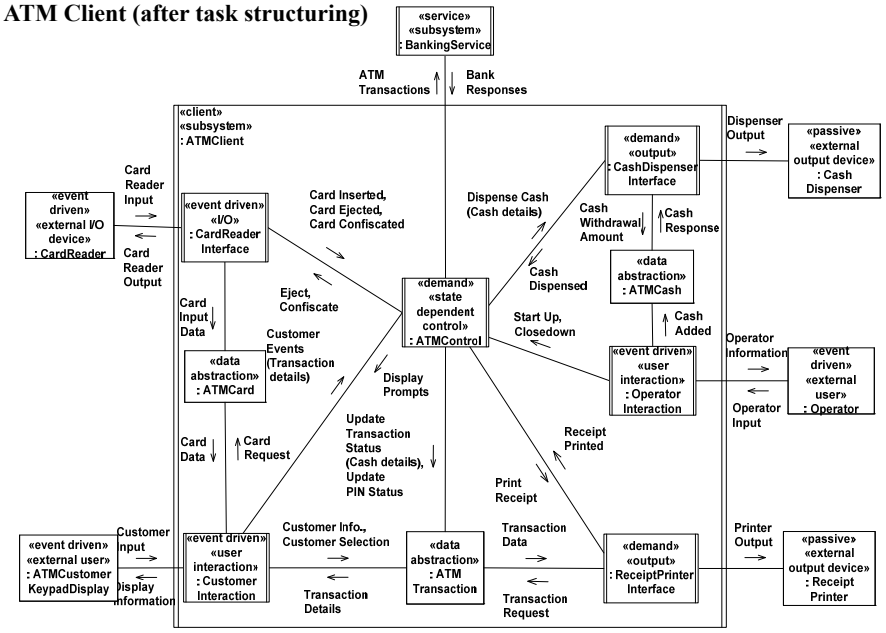
Figure 21.25 Integrated communication diagram for ATM Client subsystem (before task structuring)



24

24

Figure 18.13 Task architecture – initial concurrent communication diagram for ATM Client (after task structuring)



25

25

Banking System Case Study - Task Structuring Criteria

- Event driven I/O task
  - Card Reader Interface
- Demand driven output task
  - Cash Dispenser Interface
  - Receipt Printer Interface
- Event driven user interaction Task
  - Customer Interaction
  - Operator Interaction
- Demand driven state dependent control task
  - ATM Control

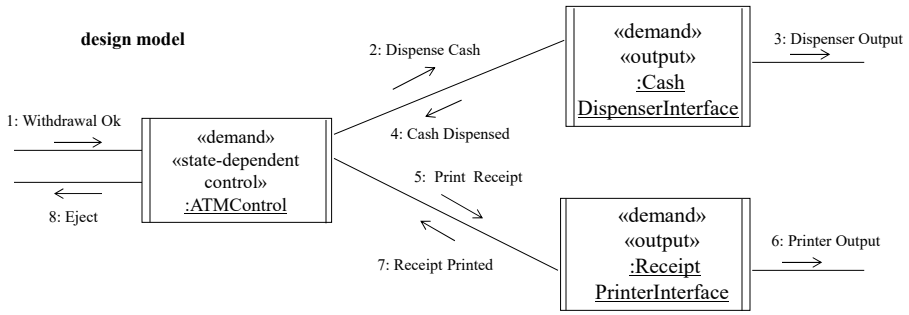
26

26

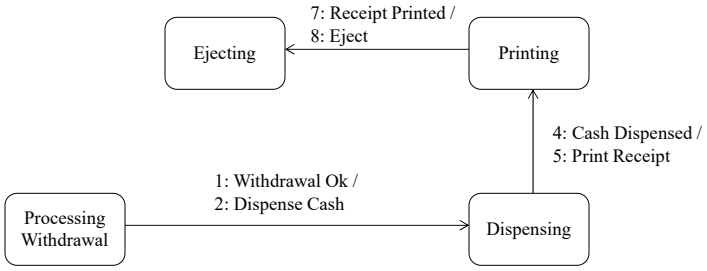
Effect of Multithread program

- Using Multithreads
  - Reasonable number of threads
    - Can create very efficient program
  - Too many threads
    - More CPU time spent changing contexts than executing programs
    - Need to reduce the number of threads

Example of control clustering

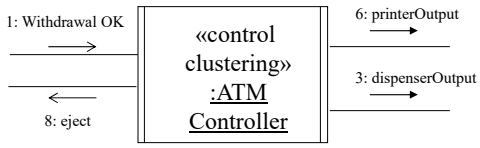


Statechart for ATM Control

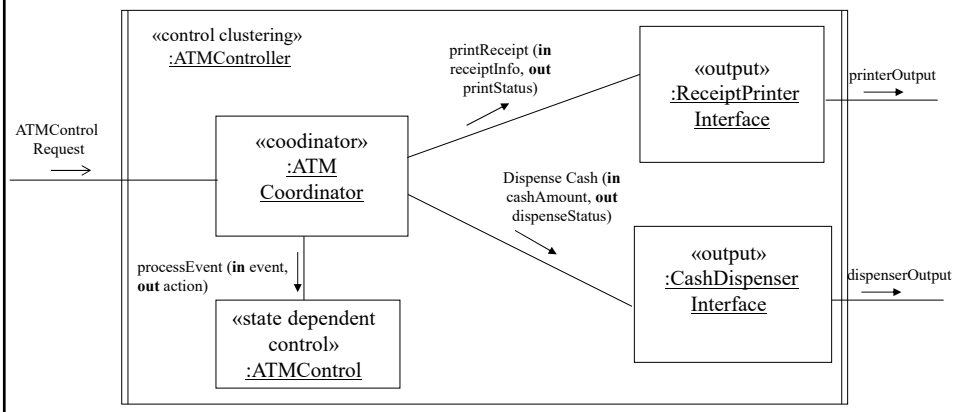


Example of control clustering

Design model -
concurrent collaboration diagram with 1 task

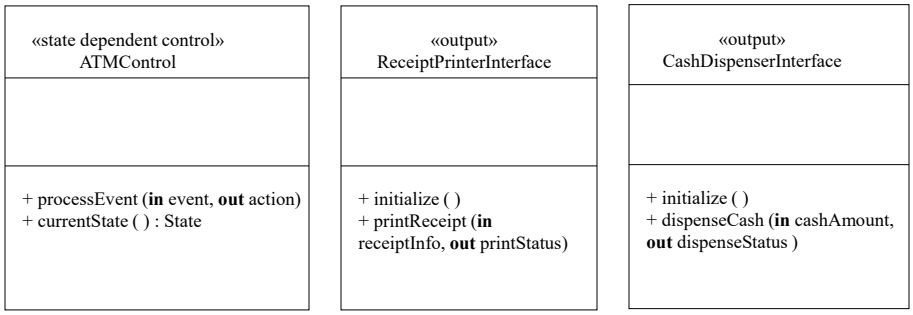


Example of control clustering task with passive objects



Example of control clustering task with passive objects

Information hiding classes



31

Design Task Interfaces

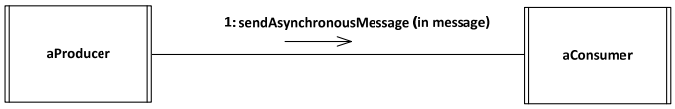
- Based on Analysis Model simple message interfaces
- Need to determine type of message communication
  - Also referred to as message communication patterns
- Asynchronous message communication
- Synchronous message communication
  - With reply
  - Without reply
- Event synchronization
  - External event (interrupt)
  - Timer event
  - Internal event
- Passive objects
  - Task interfaces to information hiding object

32

Asynchronous Message Communication

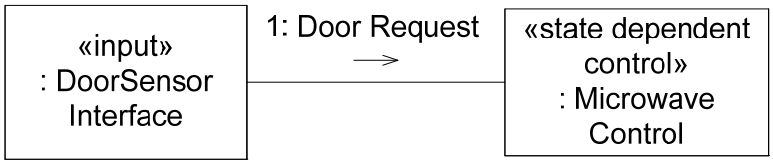
- Modeled as Design Pattern (reusable template)
  - Producer sends message and continues
  - Consumer receives message
    - Suspended if no message is present
    - Activated when message arrives
  - Message queue may build up at Consumer

Figure A-9 Asynchronous Message Communication Pattern



Example of Asynchronous Message Communication

Analysis model – communication diagram

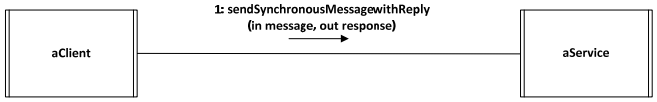


Design model – communication diagram

Synchronous Message Communication With Reply

- Modeled as Design Pattern
  - Producer (Client) task sends message and waits for reply
  - Consumer (Service) receives message
    - Suspended if no message is present
    - Activated when message arrives
    - Generates and sends reply
  - Producer (Client) and Consumer (Service) continue

Figure A.20 Synchronous Message Communication with Reply Pattern

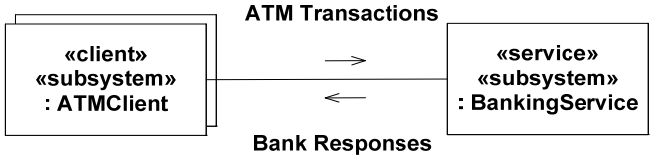


35

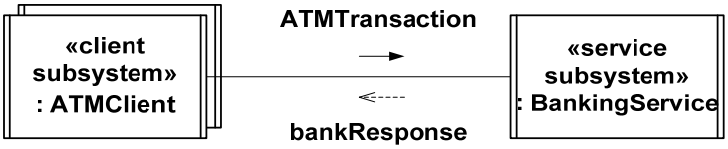
35

Example of Synchronous Message Communication With Reply

Communication diagram – Before task interface design



Communication diagram – After task interface design



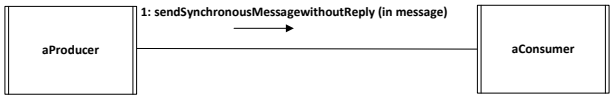
dm-36

36

Synchronous Message Communication Without Reply

- Modeled as Design Pattern
  - Producer task sends message and waits for acceptance
  - Consumer receives message
    - Suspended if no message is present
    - Activated when message arrives
    - Accepts message, Releases producer
  - Producer and Consumer continue

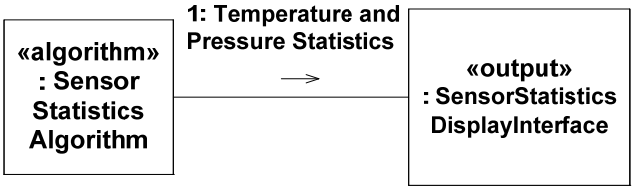
Figure A.21 Synchronous Message Communication Without Reply Pattern



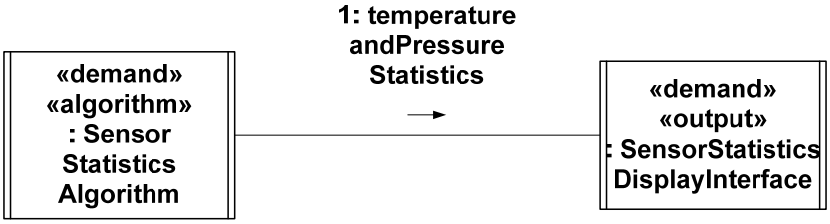
37

Example of Synchronous Message Communication Without Reply

Communication diagram – Before task interface design



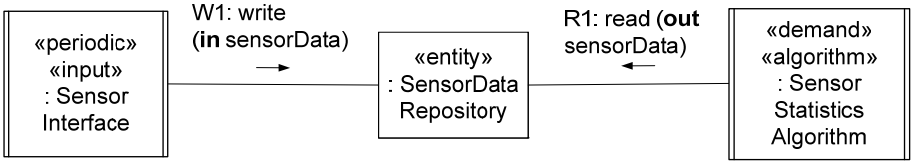
Communication diagram – After task interface design



dm-38

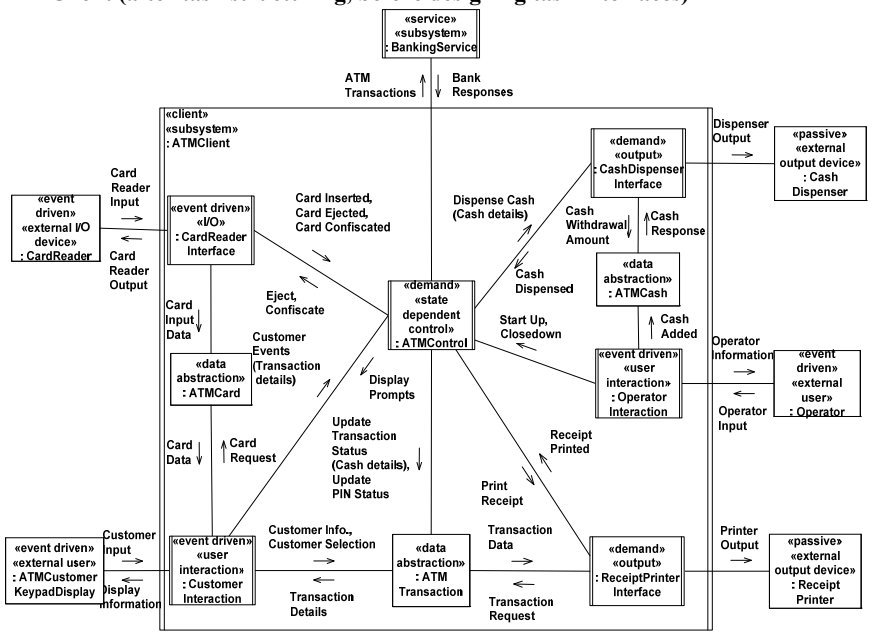
Information Hiding Object

- Passive object
  - Encapsulates data
  - Hides contents of data structure
  - Data accessed indirectly via operations
- Passive object accessed by two or more tasks
  - Operations must synchronize access to data
  - Design of class operations described in Class Design



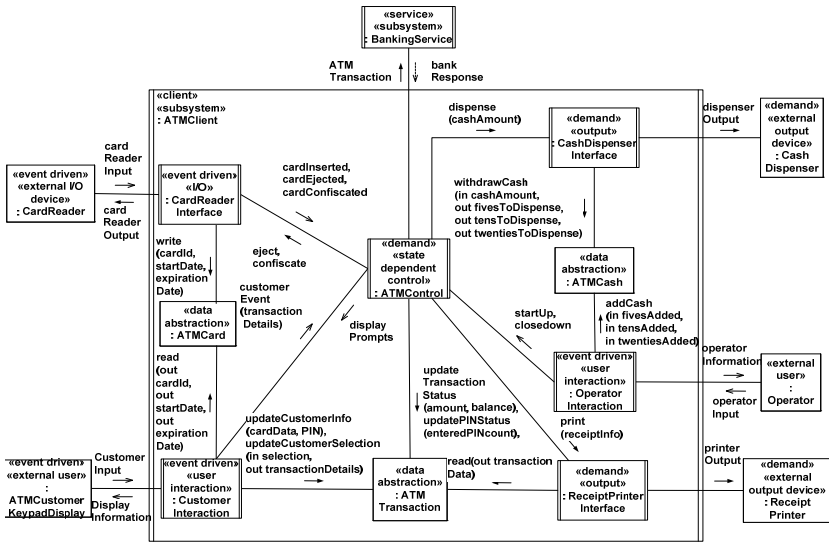
39

Figure 18.13 Task architecture – initial concurrent communication diagram for ATM Client (after task structuring, before designing task interfaces)



40

Figure 21.30 Task architecture – revised concurrent communication diagram for ATM Client subsystem (after defining task interfaces)



dm-41

41

Task Interface Specifications (TIS)

- Developed during Task Structuring
- Describes concurrent task's external view
  - Information hidden
  - Structuring criteria
    - Role (e.g., input)
    - Concurrency (e.g., event driven)
  - Anticipated changes
  - **Task interface**
  - Errors detected by task

42

42

Task Interface Specification

- Task interface
  - Event inputs and outputs
  - External inputs or outputs
  - Message inputs and outputs
    - Type of message interface (e.g., asynchronous)
    - Each message: name, parameters

dm-43

Example of Task Interface Specification

Name: Card Reader Interface

Information hidden: Details of processing input from and output to card reader.

Structuring criteria: role criterion: input/output; concurrency criterion: event driven

Assumptions: only one ATM card input and output is handled at one time.

Anticipated Changes: Possible additional information will need to be read from ATM card.

Task interface:

Task inputs:

Event input: Card reader external interrupt to indicate that a card has been input.

External input: `cardReaderInput`.

Synchronous message communication without reply:

- `eject`
- `confiscate`

Task outputs:

External output: `cardReaderOutput`

Asynchronous message communication:

- cardInserted.
- cardEjected
- cardConfiscated.

Passive objects accessed: ATMCard

Errors detected: Unrecognized card, Card reader malfunction.

44