

Overview of Software Architecture

Lecture 7

Reference: H. Gomma, Chapters 12, 13 - *Software Modeling and Design*, Cambridge University Press, February 2011

1

1

Sequential & Concurrent Problems

Sequential problems

Activities happen in strict sequence

E.g., compiler, payroll

Concurrent problems

Many activities happen at the same time

E.g., multi-user interactive system, air traffic control system

2

Concurrent Systems

- Concurrent System
 - Many tasks execute at the same time
 - Tasks need to interact with each other
- Real-Time system
 - Concurrent system with timing deadlines
- Distributed application
 - Concurrent system executing on geographically distributed nodes

3

Active and Passive Objects

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

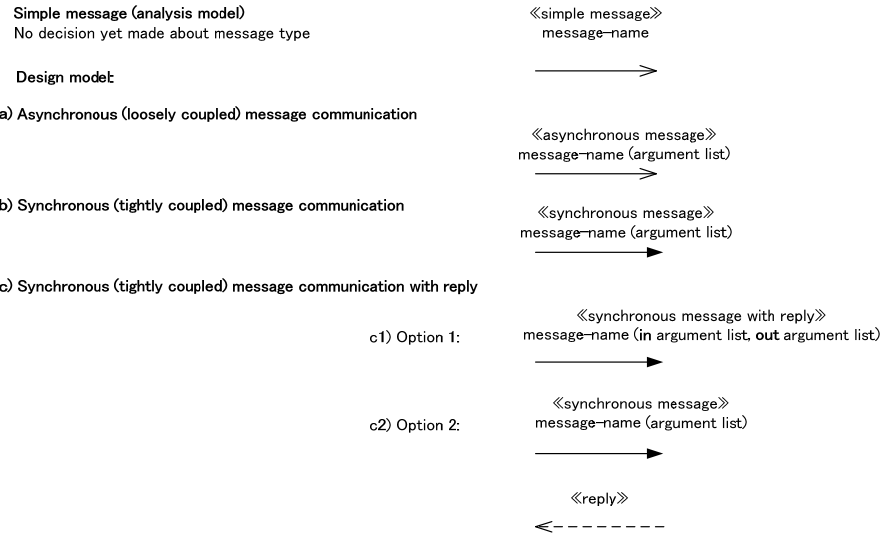
«Task»

«Object»

4

4

UML notation for messages



5

5

Design of Software Architecture

- Software Architecture
 - Structure of software system
 - Software elements (e.g., subsystems or components)
 - Externally visible properties of software elements (e.g., component interfaces)
 - Relationships among software elements (e.g., connectors)
- Develop initial software architecture
 - Synthesize from communication diagrams
 - Structure system into subsystems
- Subsystems determined using subsystem structuring criteria
 - Use stereotypes for subsystem structuring criteria
 - E.g., <<client>>, <<service>>
 - Depict subsystems on subsystem communication diagrams

6

6

Transition from Analysis to Design: Develop initial software architecture (High-level)

- Start with dynamic interaction model
 - Use case-based interaction diagrams
 - Sequence diagrams
 - Communication diagrams
- Integrate use case-based interaction diagrams
 - Initial version of software architecture
- Structure system into subsystems
 - Subsystem contains multiple objects
- Depict subsystems on subsystem communication diagram
 - High-level communication diagram
 - Shows subsystems and their interactions
 - Use stereotypes for **subsystem structuring criteria**

7

7

Transition from Analysis to Design: Integration of Communication Diagrams

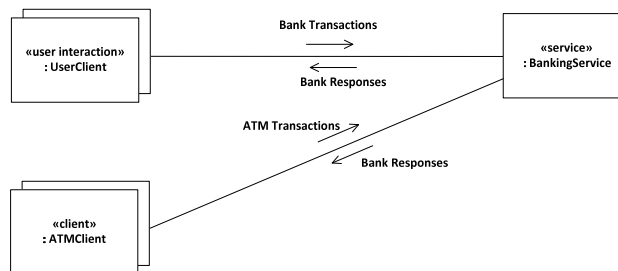
- Integration of communication diagrams
 - To determine overall structure of system
- Merging of communication diagrams
 - Start with first communication diagram
 - Superimpose other communication diagrams
 - Add new objects and new message interactions from each subsequent diagram
 - Objects and interactions that appear on multiple diagrams are only shown once
 - Consider alternative scenarios for each use case
- Integrated communication diagram
 - Shows all objects and their interactions

8

8

Design of Client/Server Systems

- Client Subsystem sends request to Service Subsystem
 - Waits for response
- Service Subsystem
 - Receives Client requests
 - Processes each Client request in the order received
 - First-In First-Out (FIFO)
 - Sends response to Client



9

9

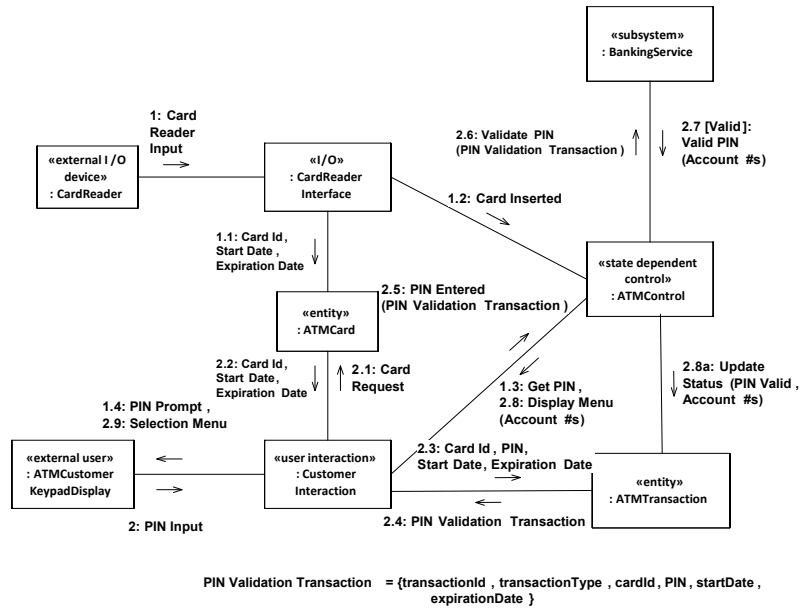
Example of Integration of Communication Diagrams - Banking System

- Client/Server architecture
 - Multiple ATM Clients
 - Banking Service
 - Processes ATM client requests
- Develop subsystem communication diagram
 - Banking System
- Develop integrated communication diagram for each subsystem
 - ATM Client
 - Banking Service
- Start from use case-based communication diagrams
 - Validate PIN
 - Withdraw Funds
 - Etc.

10

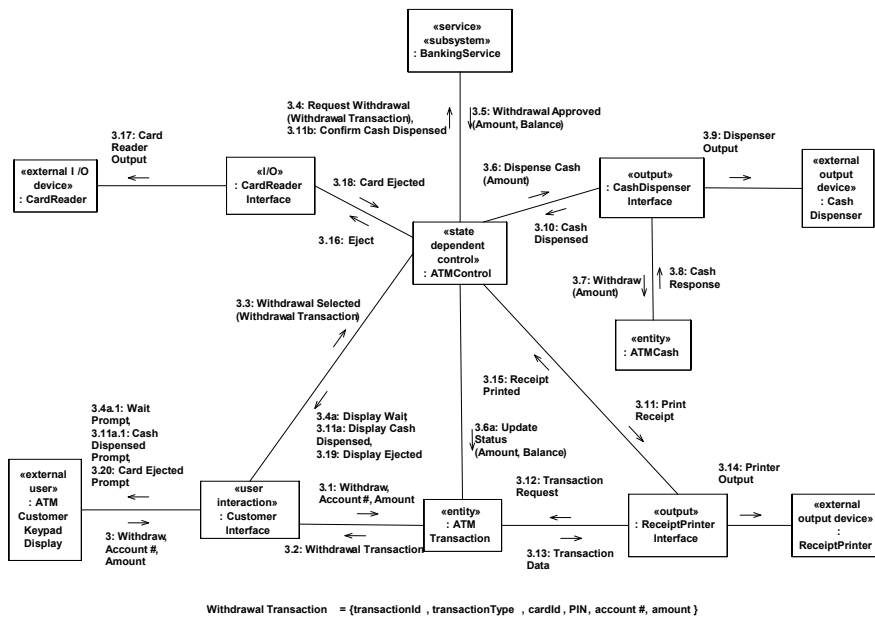
10

**Figure 11.1 Communication diagram: ATM Client –
Validate PIN use case – Valid Pin**



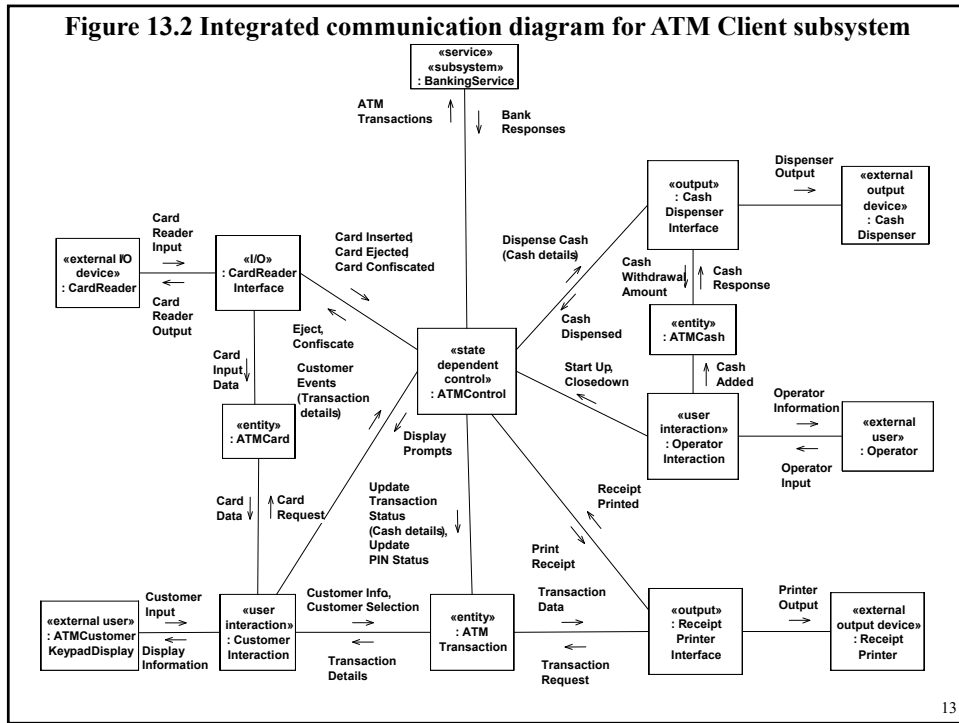
11

Figure 21.16 Communication diagram: ATM Client – Withdraw Funds use case



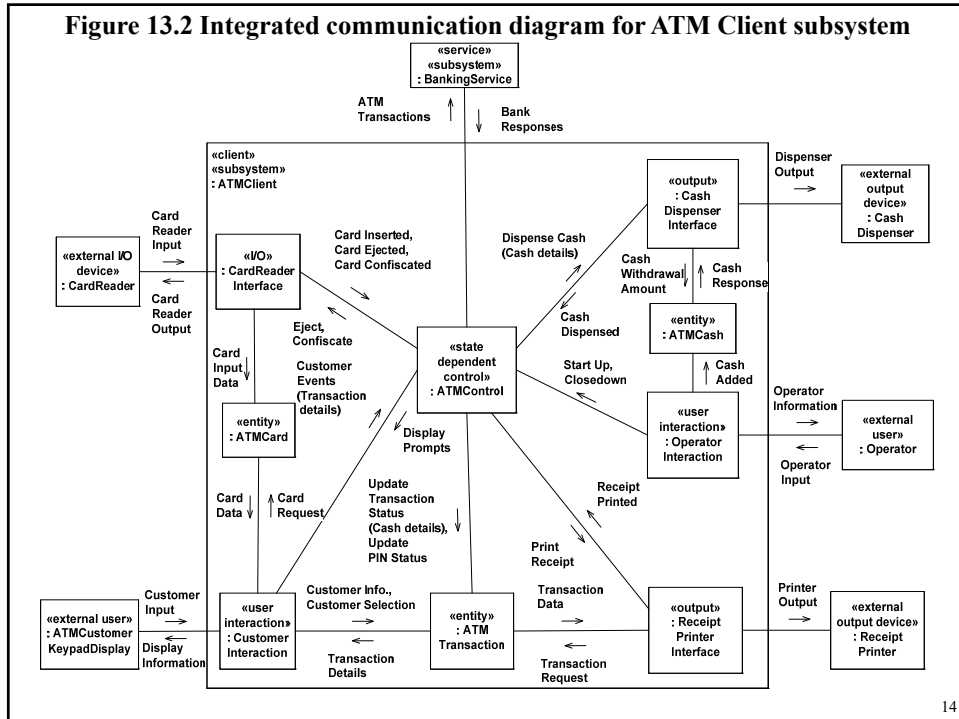
12

Figure 13.2 Integrated communication diagram for ATM Client subsystem



13

Figure 13.2 Integrated communication diagram for ATM Client subsystem



14

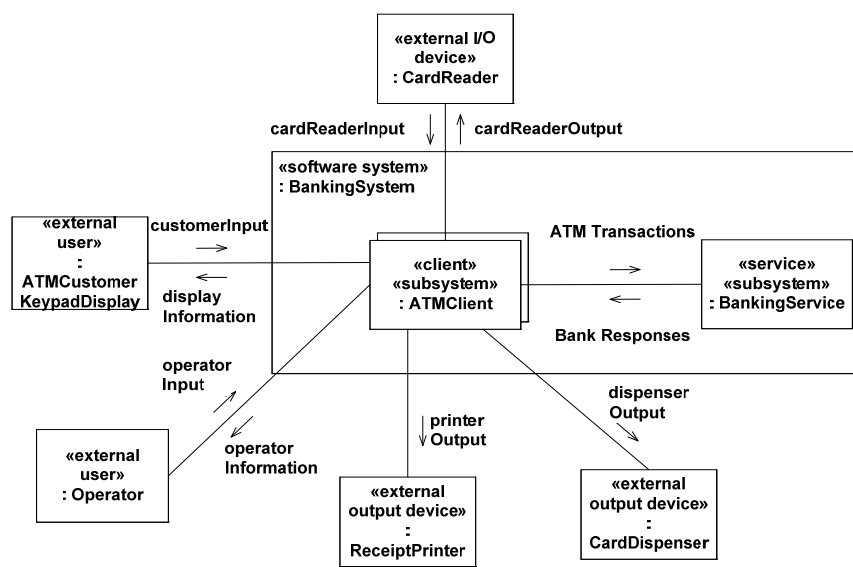
Characteristics of Integrated Communication Diagram – ATM Client

- Shows all objects and interactions
 - From main and alternative sequences of each use case
 - Validate PIN, Withdraw Funds, Transfer Funds, Query
- For clarity
 - Sequence numbers are removed
 - Individual messages between objects can be aggregated
 - E.g., between ATM Control and Customer Interaction
 - Aggregate message = Display Prompts
 - Consists of simple messages
 - Get PIN, Invalid PIN Prompt, Display Menu, etc.

15

15

Figure 21.27 Subsystem design – high level communication diagram for Banking System

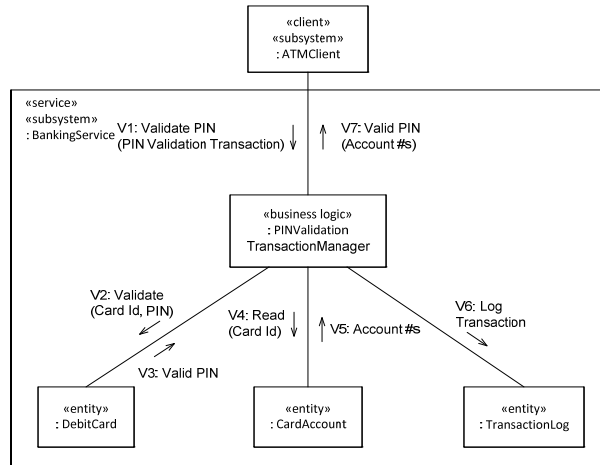


16

16

Banking Service Subsystem Design

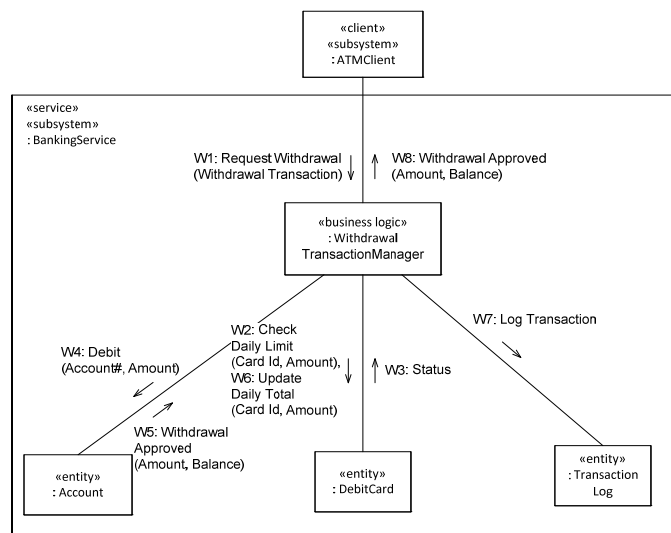
- Start with use case-based communication diagrams
 - Figure 21.14 Communication diagram: Banking Service Validate PIN



17

Banking Service Subsystem Design

- Start with use case-based communication diagrams
 - Figure 21.19 Communication diagram: Banking Service Withdraw Funds



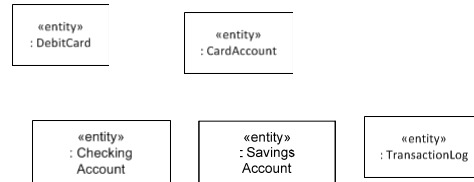
18

Integrated Communication Diagram – Banking Service

- **Banking Service Objects**

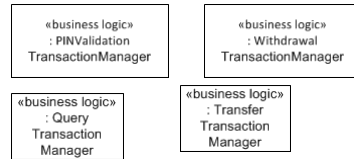
- **Entity Objects**

- Debit Card
 - Card Account
 - Checking Account
 - Savings Account
 - Transaction Log



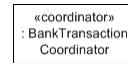
- **Business Logic Objects include**

- PIN Validation Transaction Manager
 - Withdrawal Transaction Manager
 - Query Transaction Manager
 - Transfer Transaction Manager



- **Design decision**

- **Need to add Coordinator object**
 - Bank Transaction Coordinator

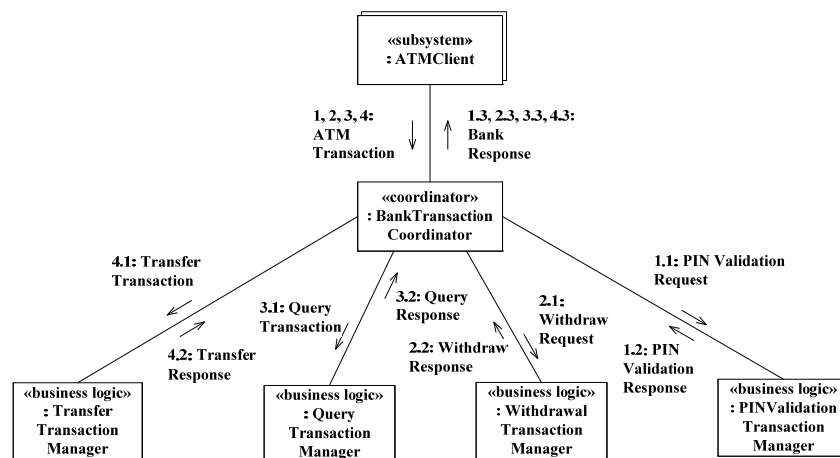


dm-19

19

Coordinator object

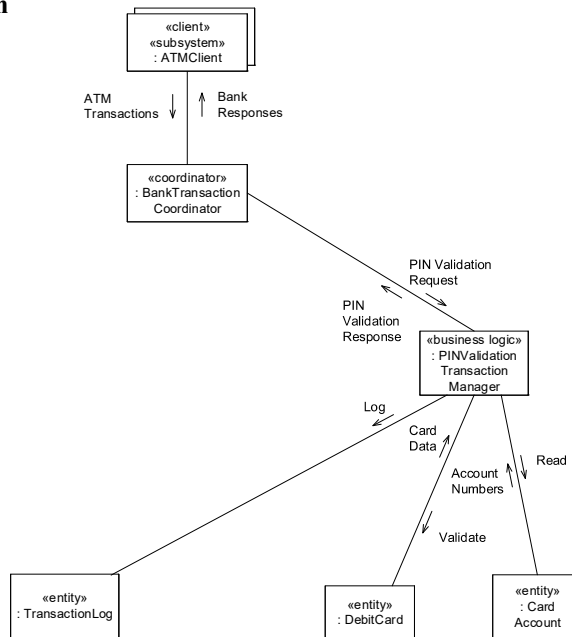
- Decision making object
 - Decision based on input received
- Bank Transaction Coordinator
 - Forward ATM transaction to a business logic object



20

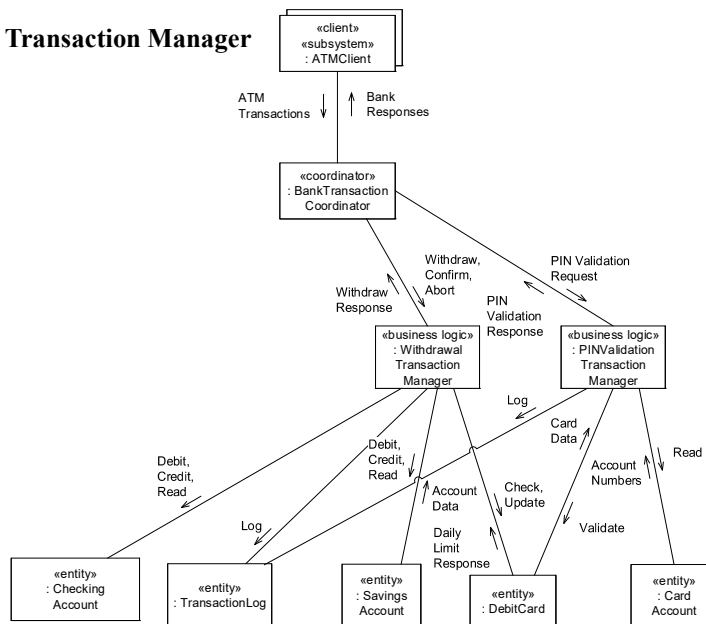
20

Figure 21.26 Integrated communication diagram for Banking Service subsystem
Start with PIN Validation
Transaction Manager



21

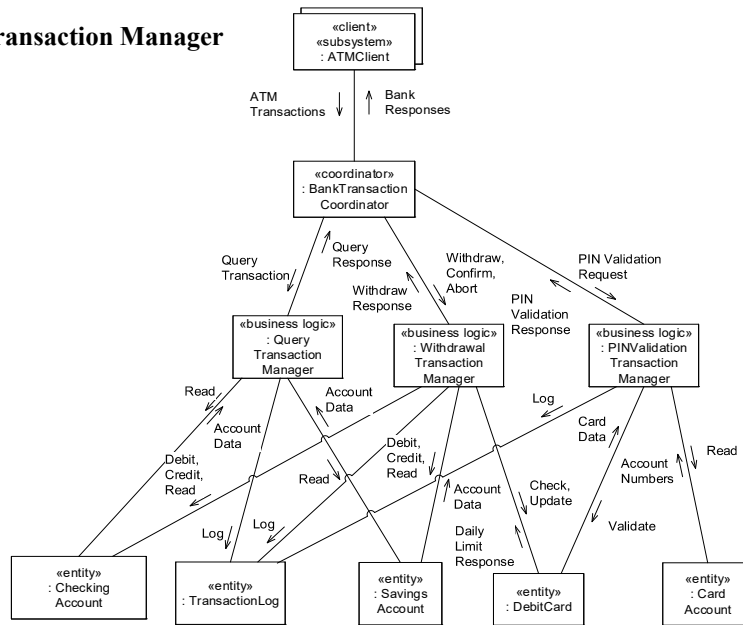
Figure 21.26 Integrated communication diagram for Banking Service subsystem
Add
Withdrawal Transaction Manager



22

Figure 21.26 Integrated communication diagram for Banking Service subsystem
Add

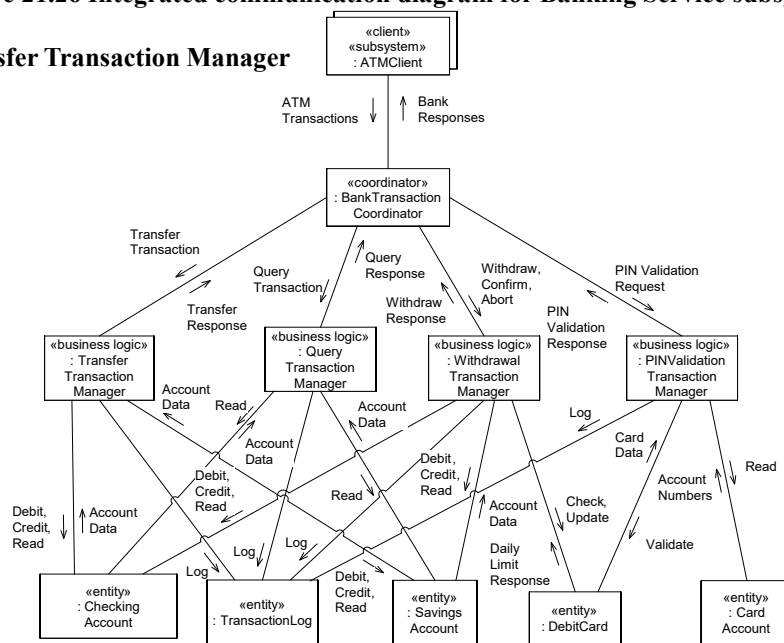
Query Transaction Manager



23

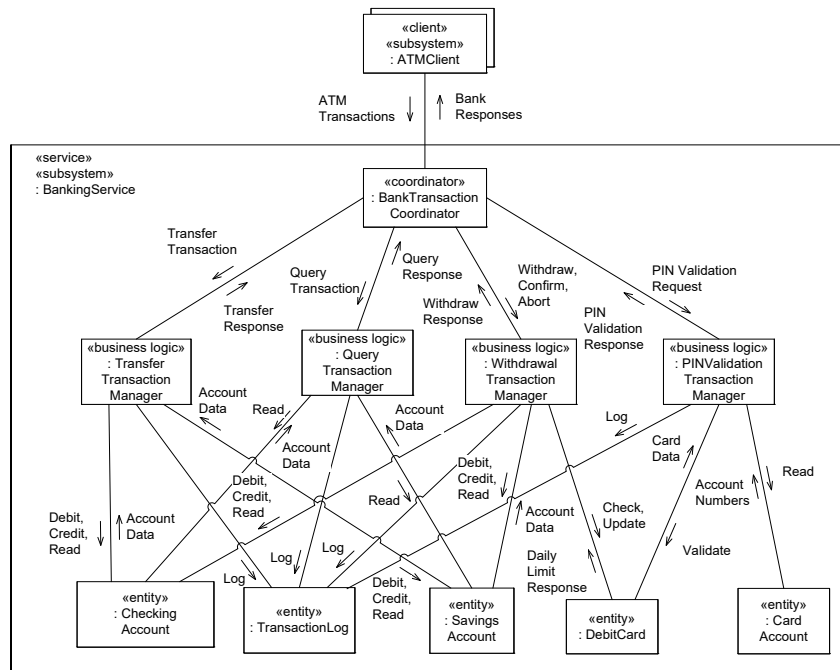
Figure 21.26 Integrated communication diagram for Banking Service subsystem
Add

Transfer Transaction Manager



24

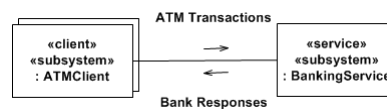
Figure 21.26 Integrated communication diagram for Banking Service subsystem



25

Characteristics of Integrated Communication Diagram – Banking Service

- Shows all objects and interactions
 - From main and alternative sequences of each use case
- For clarity
 - Sequence numbers are removed
 - Individual messages between objects can be aggregated
 - E.g., between ATM Client and Banking Service
 - Aggregate message = ATM Transactions
 - Consists of simple messages
 - PIN Validation Transaction,
 - Withdrawal Transaction, Query Transaction, Transfer Transaction



26

Design of Software Architecture

- Software Architecture
 - Define overall structure of system
 - Component interfaces and interconnections
 - Separately from component internals
- Each subsystem performs major service
 - Contains highly coupled objects
 - Relatively independent of other subsystems
 - May be decomposed further into smaller subsystems
 - Subsystem is aggregate or composite object

27

27

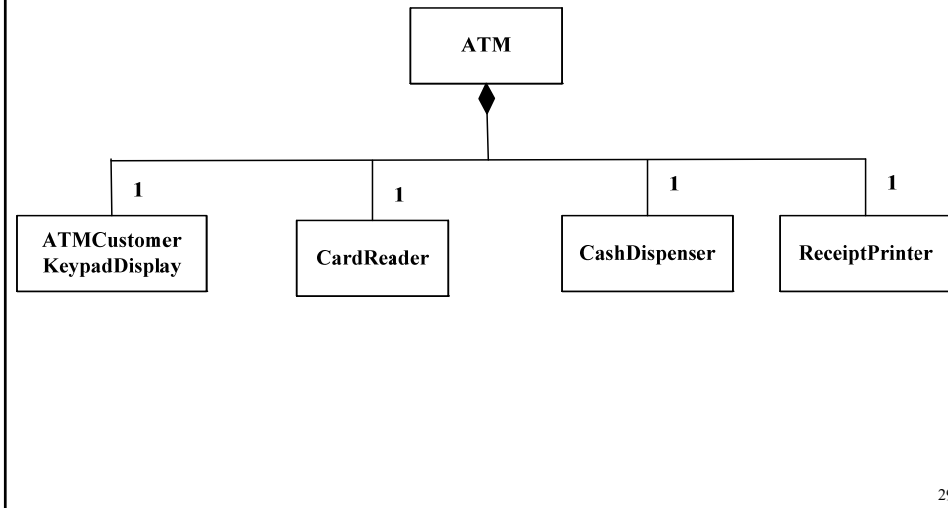
Separation of Subsystem Concerns

- **Aggregate/composite object**
 - Objects that are part of aggregate/composite object
 - Structure in same subsystem (e.g., Fig. 13.3)
- **Interface to external objects**
 - External real-world object should interface to 1 subsystem (e.g., Fig. 13.7)
- **Scope of Control**
 - Control object & objects it controls are in same subsystem (e.g., Fig. 13.2)
- **Geographical location**
 - Objects at different locations are in separate subsystems (e.g., Fig. 17.12)
- **Clients and Services**
 - Place in separate subsystems (e.g., Fig. 17.12, 13.7)
- **User Interaction**
 - Separate client subsystem (e.g., Fig. 17.12, Fig. 13.6)

28

28

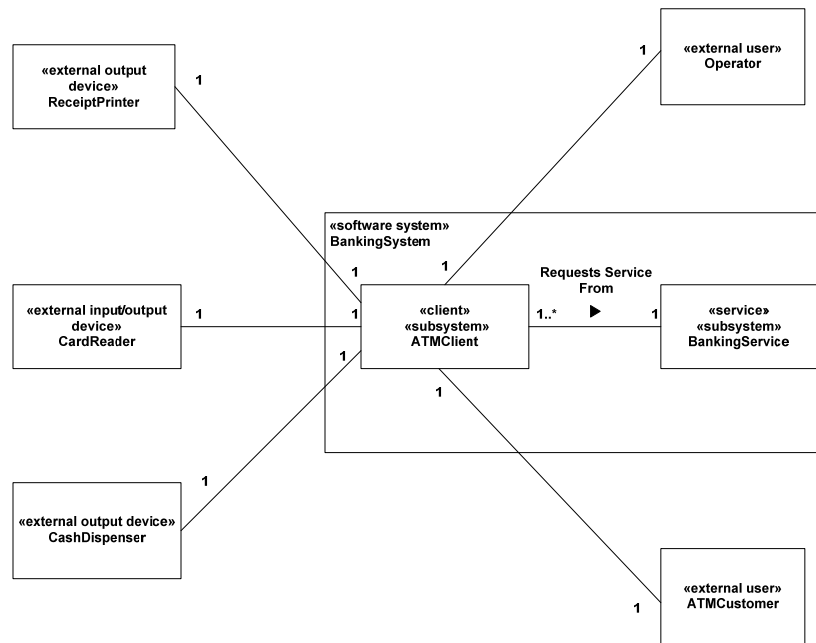
Figure 13.3 Example of composite class



29

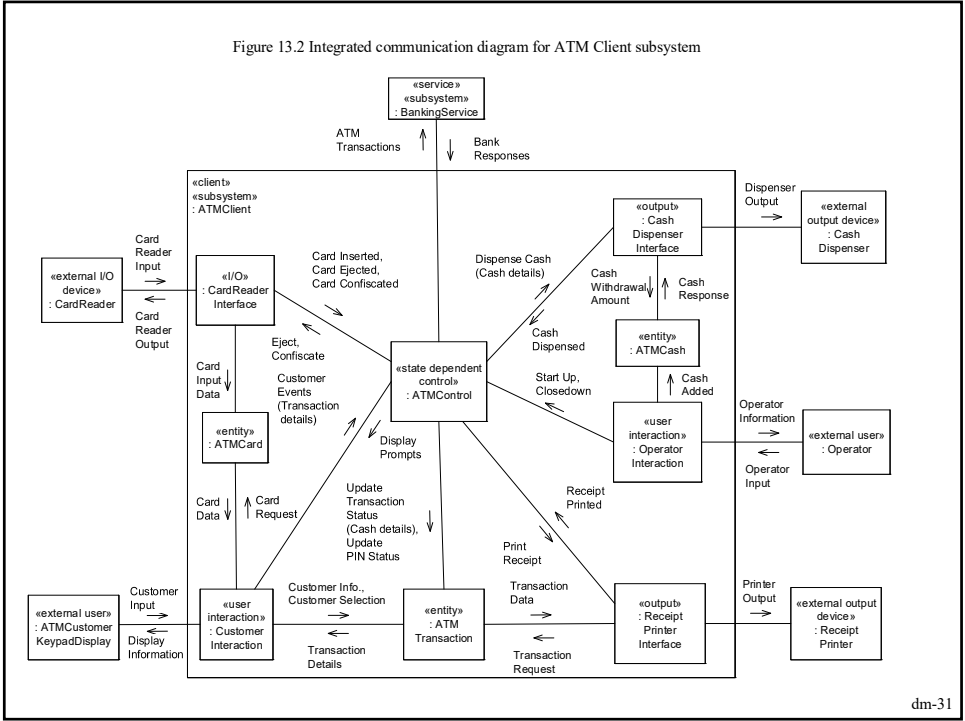
29

Figure 13.7 Interface to external classes – Banking System

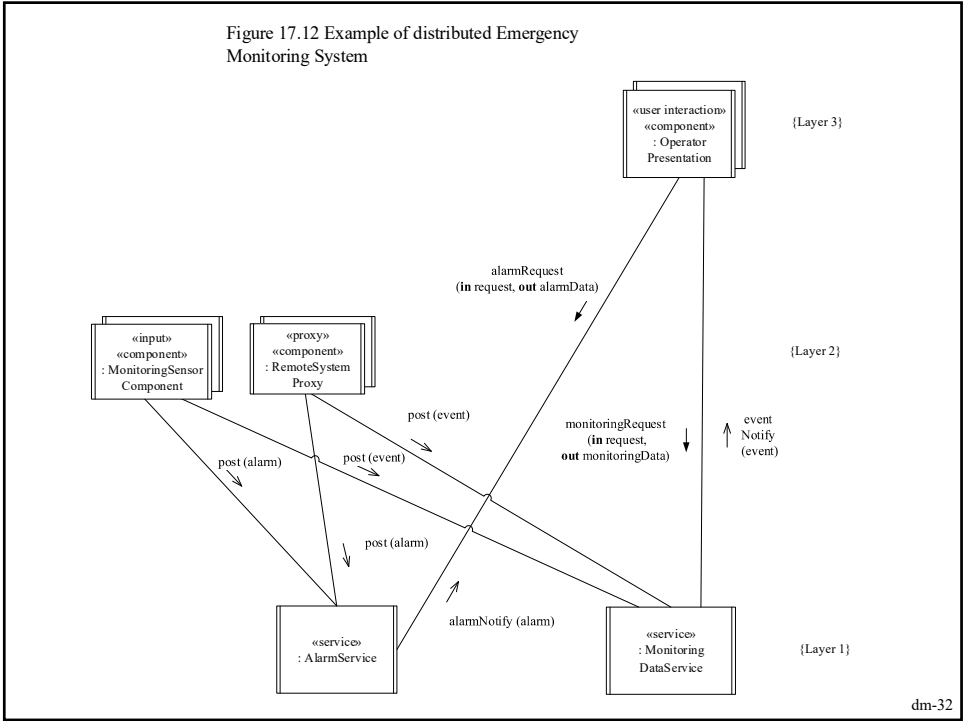


30

30

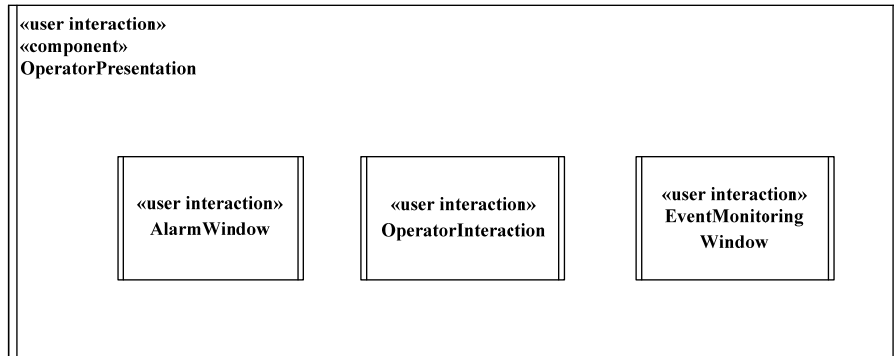


31



32

Figure 13.6 Example of user interaction subsystem: Operator Presentation component



33

33

Subsystem Structuring Criteria

- Client
 - Requester of one or more services (e.g., Fig. 13.7)
- User Interaction
 - Collection of objects supporting needs of user (e.g., Fig. 13.6, 13.10)
- Service
 - Provides service for client subsystems (e.g., Fig. 17.12, 13.7)
- Control
 - Subsystem controls given part of system (e.g., Fig. 13.10)
- Coordinator
 - Coordinates several control subsystems (e.g., Fig. 13.10)
- Input / Output
 - Performs I/O operations for other subsystems (e.g., Fig. 17.12)

34

34

Figure 13.10 Example of coordinator and control subsystems - Factory Automation System

