



中山大学软件工程学院

SCHOOL OF SOFTWARE ENGINEERING

第5讲 软件体系结构设计-Part I

集成通信建模+子系统划分&设计

授课教师：张能 助理教授

zhangn279@mail.sysu.edu.cn

综合实验楼A323-3

2023年05月24、26日

目录



- 实验内容
- 任务讲解
- Draw.io绘图演示

■ 1. 集成通信建模

- 合并基于用例的通信图, 绘制**集成通信图**

■ 2. 子系统划分&设计

- 确定**子系统**, 以及各个子系统的**并发任务**和**任务通信**类型
- 绘制**子系统间的并发通信图**及各个**子系统的并发通信图**



集成通信图的绘制方法

子系统组织准则

并发任务组织准则及任务通信

并发通信图的绘制方法

集成通信建模



- 为了实现从分析到设计的过渡并确定子系统, 需要在分析的基础上, 通过**集成基于用例的交互图**, 合成一个**初始设计**
- **集成通信图**: 所有用例的通信图的合成
- 对象之间的动态交互可用通信图和顺序图描述, 但**集成使用通信图的原因**: 通信图能够清晰地描述对象之间的相互连接和传递的消息

- **通信图集成方式**: 从**第一个用例**的通信图开始, **依次**将后续用例的通信图进行叠加; 每次叠加的通信图会向集成通信图中添加新的对象和消息交互 → 在多个通信图中出现的对象和消息交互只会显示一次
- **Q: 怎样确定用例通信图的集成顺序?**
 - 若不同用例之间存在执行的优先顺序, 则**通信图合成的顺序应与用例执行的顺序一致**, 如PIN验证→取款

■ 集成通信图采用泛化的通信图

- 1) 只显示**对象**和**消息**, **无需显示消息序列编号**
- 2) 消息会被描述成简单消息, **无需确定消息的类型**(如同步或异步)
- 3) 对象名不加下划线 // 本教材里的所有交互图中的对象名均未加下划线
- 4) 无需**确定对象的类型**(如主动或被动)

- 大型系统的集成通信图会非常复杂，可采取两种方法**减少信息量**
 - **1) 对消息进行聚合**: 如果一个对象向另一个对象发送了多条消息，可使用一个**聚合消息**而不在图上显示所有消息
 - **2) 按子系统划分**: 为每个子系统开发**集成通信图**，并开发一个高层的**子系统通信图**以显示子系统之间的交互 → **前提: 根据系统描述及特点可以确定高层子系统**

子系统划分



- **关注点分离(separation of concerns)**: 在子系统分解的设计过程中, 让不同的子系统具有不同的关注点, 使得子系统之间的独立性更好



如何实现关注点分离?



设计考虑

组合&聚合子系统

地理位置

客户端和服务

用户交互

外部对象的接口

控制范围

子系统组织准则

子系统组织准则



1. 这些准则是**Should NOT Must**
2. 一个子系统可满足不止一个组织准则

组合&聚合子系统

1. 同一组合对象的组成对象**应**在同一子系统中;
2. 一个聚合子系统包括一组根据功能相似性聚集在一起的对象, 这些对象可以跨越地理位置边界

地理位置

若两个对象可能分布在不同的地理位置, 则它们**应**位于不同的子系统

客户端和服务

客户端和服务**应**位于不同的子系统;
可看作地理位置准则的特例, 因为客户端和服务通常位于不同的地理位置

用户交互

让用户交互对象位于独立的子系统;
用户交互对象通常是客户端, 此准则可视为客户端/服务准则的特例

外部对象的接口

一个外部现实世界对象**应**只与一个子系统之间存在接口

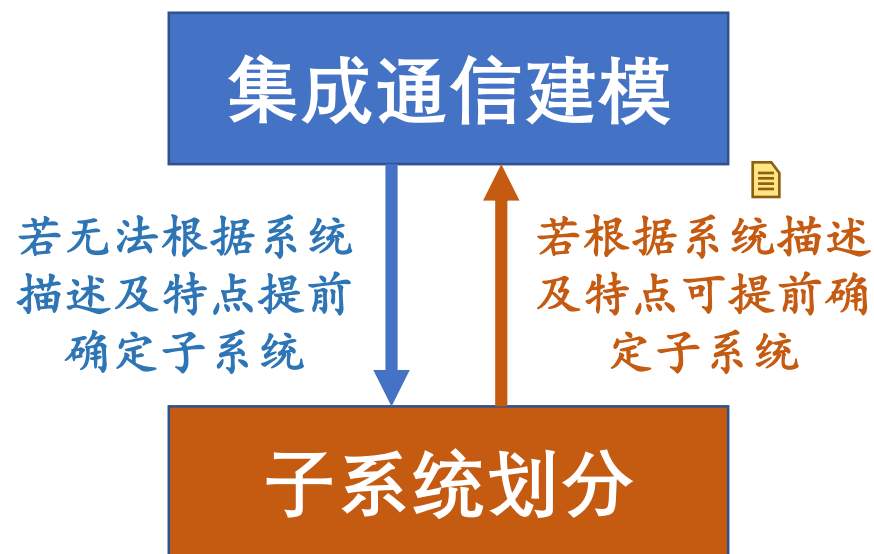
控制范围

一个控制对象及其直接控制的实体对象和输入/输出(I/O)对象**应**位于同一子系统, 而不是分散在多个子系统中

集成通信建模与子系统划分的关系



1. 两个任务的用途?
2. 两个任务的执行先后?



此任务是从分析到设计的过渡, 所绘制的集成通信图

1. 可用于确定子系统(包括细分子系统)
2. 可用于类/任务/服务接口设计(结合用例通信图)

此任务将整个软件系统分解为不同的子系统

1. 可提供系统的高层设计
2. 可简化系统的设计(每个子系统可独立设计)
3. 可提升系统的开发效率(每个子系统可并行开发)

SSE212课程示例: 银行系统



- **系统描述**: 一家银行拥有一些**ATM机**, 分布在不同的地理位置并通过广域网连接到一个**中央服务器**上。每个ATM机由一个读卡器、一个吐钞机、一个键盘/显示器和一个凭条打印机组成。通过使用ATM机, **客户**能够从**支票账户**或**储蓄账户**提取现金、查询账户余额, 或者在账户间转账。客户将一个ATM卡插入读卡器后会启动一个交易。...
- 根据系统描述, 按照**地理位置准则**或**客户端和服务准则**, 该银行系统可分为**ATM客户端子系统**和**银行服务子系统**

SSE212课程示例: 银行系统

- 例: 银行系统的子系统
间的(高层)通信图

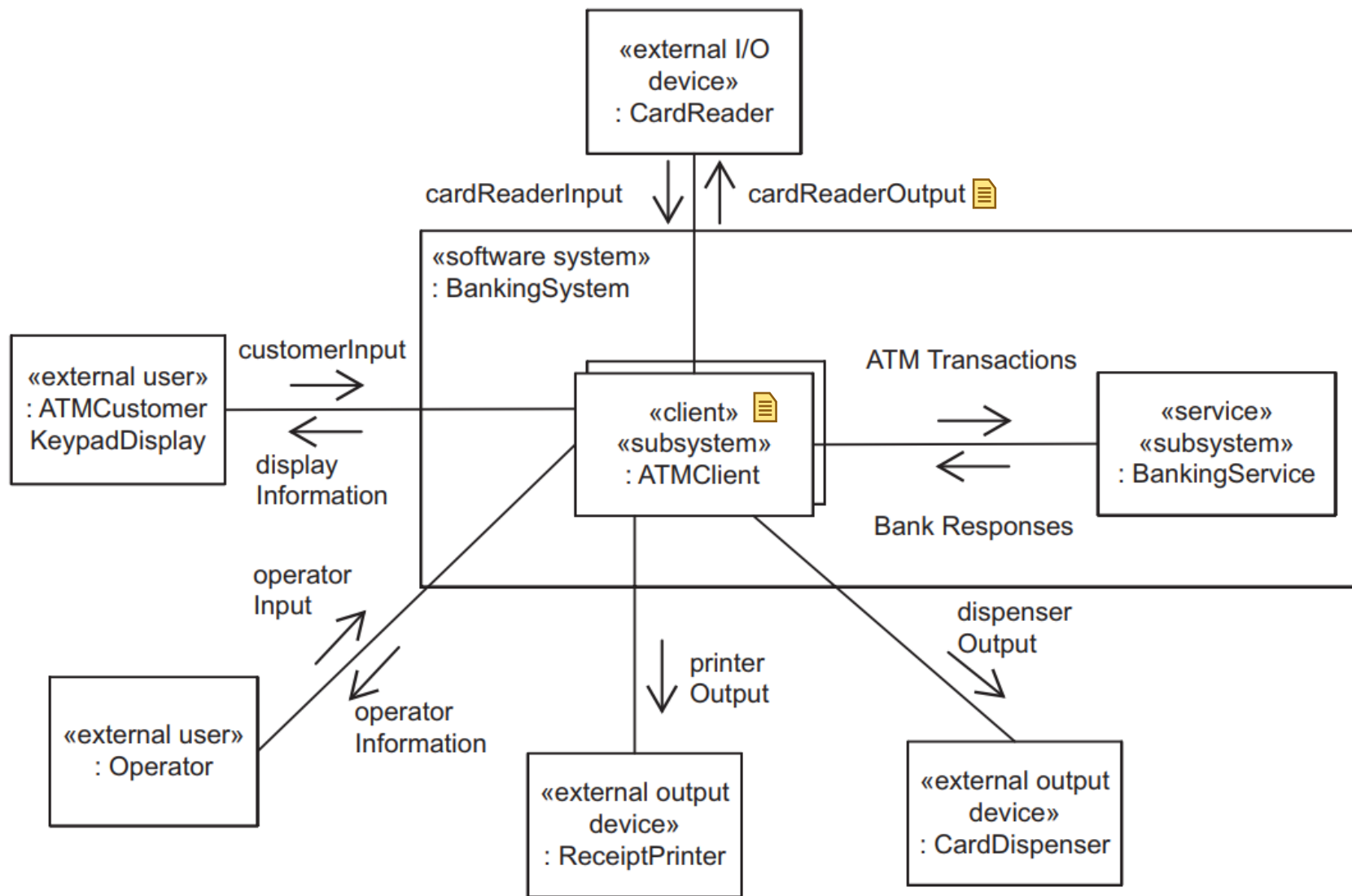


Figure 21.27. Subsystem design: high-level communication diagram for Banking System

SSE212课程示例· 银行系统

➤ 例：银行系统的ATM 客户端子系统的集成 通信图

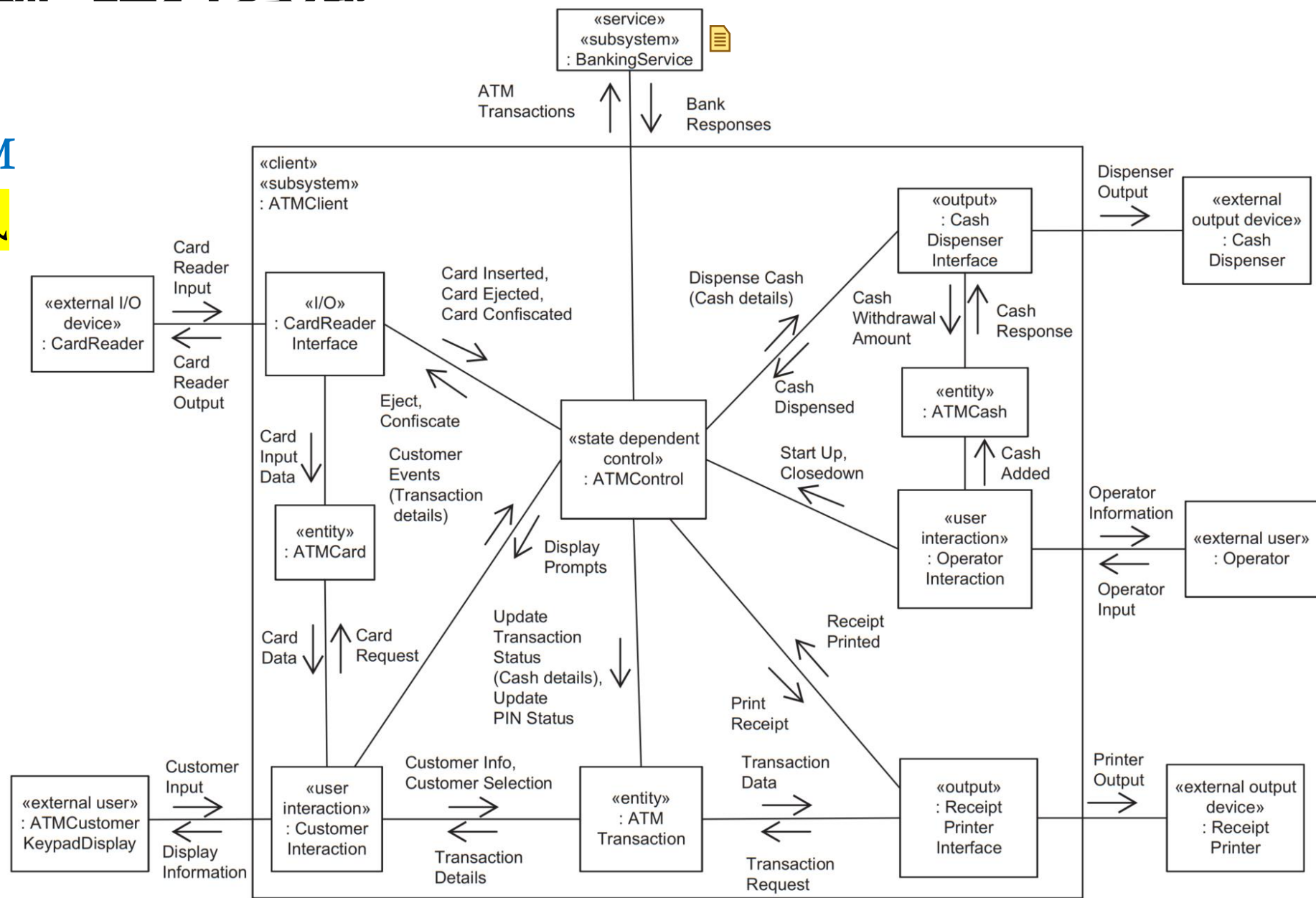


Figure 13.2. Integrated communication diagram for ATM Client subsystem

SSE212课程示例:

➤ 例：银行系统的服务子系统的集成通信图

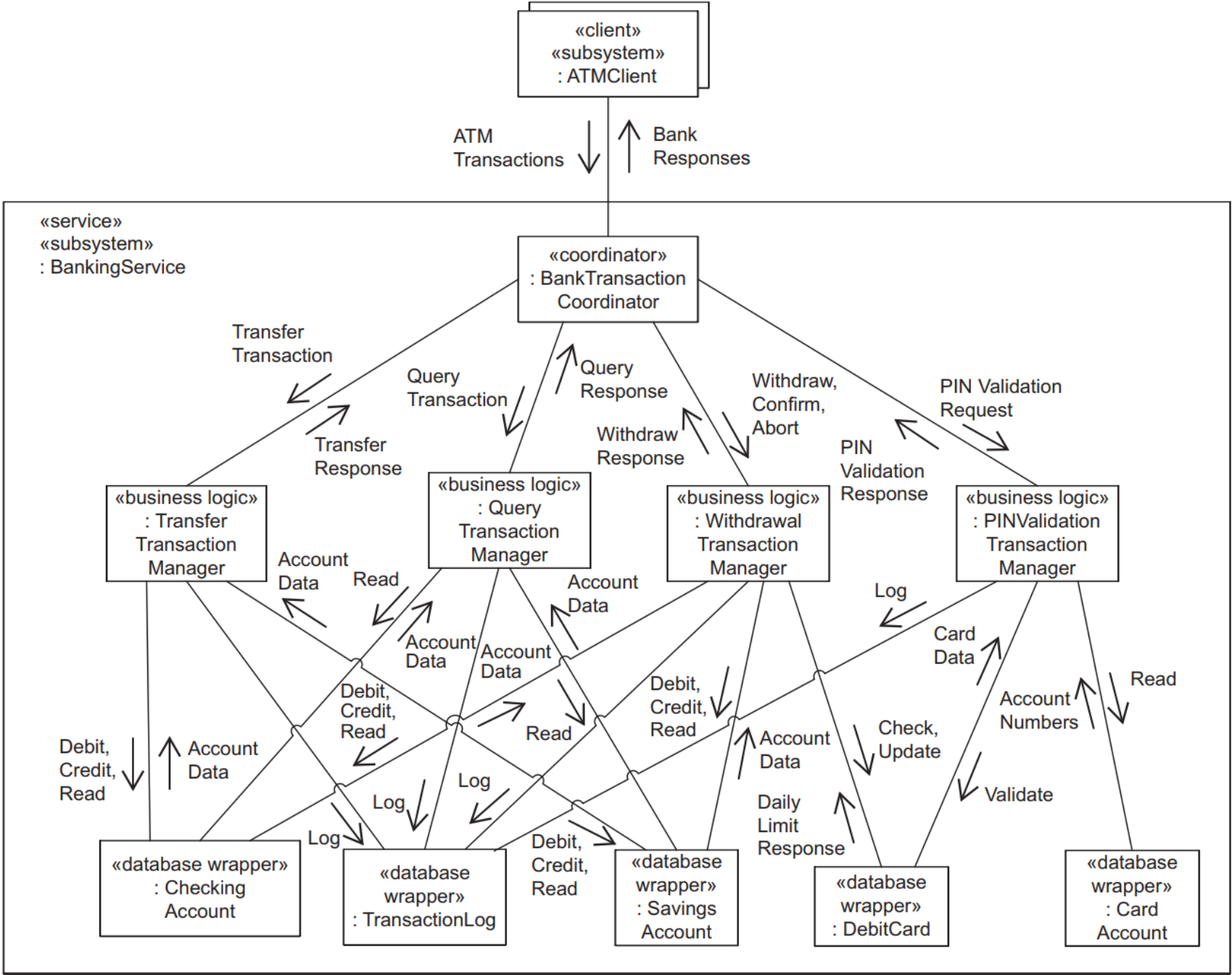


Figure 21.26. Integrated communication diagram for Banking Service subsystem

子系统设计: 并发任务组织



- 在**分析模型**中, 系统表示为相互协作的对象和它们之间的消息通信
- 在**并发任务组织阶段**, **系统的并发特性**被表示为一系列并发任务和它们之间的通信接口 → 根据系统的动态特性, 需要**确定哪些对象应并发执行、哪些对象应顺序执行**

并发任务组织



■ 并发任务可使用两种构造型

- **对象角色(object role)**: 在**对象组织阶段**完成, 如《I/O》
- **并发类型(type of concurrency)**: 在**并发任务组织阶段**, 若确定一个对象是主动的, 则需要进一步确定它的并发类型, 以显示其特性

➤ **例**: 若一个《I/O》对象是主动/并发的, 则需要确定它属于哪种并发类型: **事件驱动任务**《event driven》、**周期性任务**《periodic》、**按需触发任务**《demand》

并发任务组织



■ 按以下顺序将任务组织准则应用于分析模型(即用例通信图) 📄

- **I/O任务**: 分析每个与外部世界交互的**设备I/O对象**, 将其构造为事件驱动I/O任务、周期性I/O任务或按需驱动I/O任务
- **控制任务**: 分析每个**状态相关的控制对象**和**协调者对象**, 将其构造为状态相关的控制任务或协调者任务
- **周期性任务**: 分析**内部周期性活动**, 将其构造为周期性任务
- **按需驱动任务**: 分析每个由**内部事件**触发的**内部任务**, 将其构造为按需驱动任务

并发任务组织: 部分



事件驱动I/O任务

当系统中存在由事件驱动(也称中断驱动)的I/O设备并需要与之交互时,就需要事件驱动I/O任务 📖

周期性I/O任务

定期轮询一个被动I/O设备; 📖
周期性I/O任务会被外部计时器发送的计时器事件触发, 然后执行I/O动作

按需驱动I/O任务

处理不需定期轮询的被动I/O设备; 📖
使用场合: 当计算与I/O需要同时进行, 如按需驱动输出任务

周期性任务

周期性内部活动被构造为周期性任务;
周期性任务会被计时器事件触发并执行

按需驱动任务

一个按需触发(当接收到来自不同任务的内部消息或事件时)的对象被构造为一个按需驱动任务

用户交互任务

用户交互对象会被设计为用户交互任务, 且通常是事件驱动的

状态相关的控制任务

一个严格顺序执行的控制活动表示为一个状态相关的控制任务;
控制任务通常是按需驱动任务

并发任务组织



Table 18.1. Mapping from analysis model objects to design model tasks

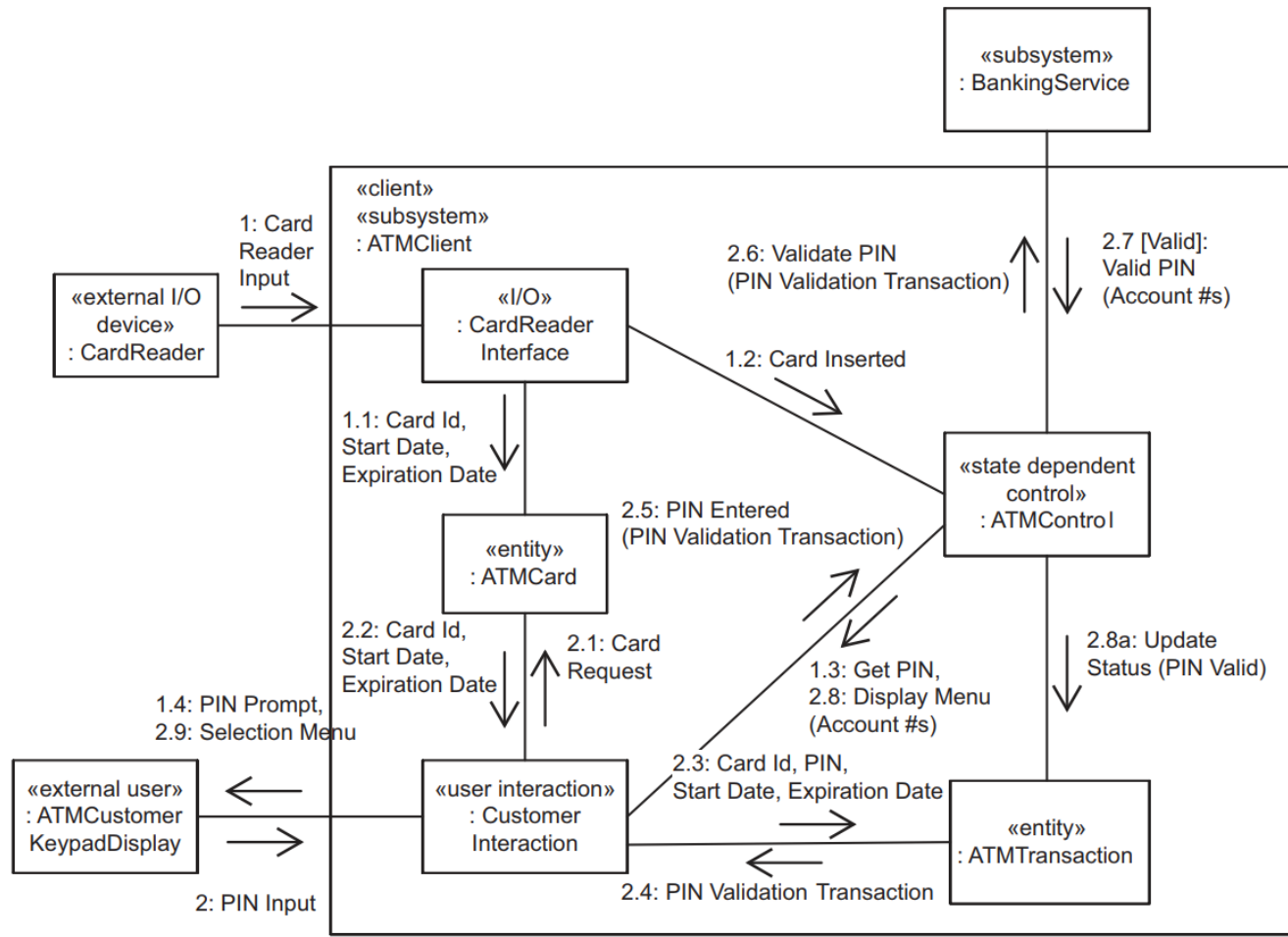
Analysis model (Object)	Design model (Task)
User interaction	Event driven user interaction
Input/Output (input, output, I/O)	Event driven I/O (input, output, I/O) Periodic I/O (input, output, I/O) Demand driven I/O (usually output)
Proxy	Event driven proxy
Timer	Periodic timer
State-dependent control	Demand driven state-dependent control
Coordinator	Demand driven coordinator
Algorithm	Demand driven algorithm Periodic algorithm

并发任务组织: 示例



■ 验证PIN码用例的通信图

1. **读卡器接口**: 一个与读卡器交互的I/O对象, 而读卡器是事件驱动的外部I/O设备 → 读卡器接口被构造为一个**事件驱动的I/O任务**
2. **ATM卡**: 一个**被动的实体对象**
3. **ATM控制**: 一个状态相关的控制对象, 需要独立的控制线程处理不同来源的信息, 被构造为一个**状态相关的(按需驱动)控制任务**
4. **客户交互**: 一个用户交互对象, 在显示器上输出消息并从键盘上接收输入, 需要独立的控制线程, 被构造为一个**事件驱动的用户交互任务**
5. **ATM交易**: 一个**被动的实体对象**



PIN Validation Transaction = {transactionId, transactionType, cardId, PIN, starDate, expirationDate}

Figure 21.11. Communication diagram: ATM client Validate PIN use case

并发任务组织: 示例

ATM客户端子系统 的初始并发通信图

确定了并发任务,
但未确定消息类型

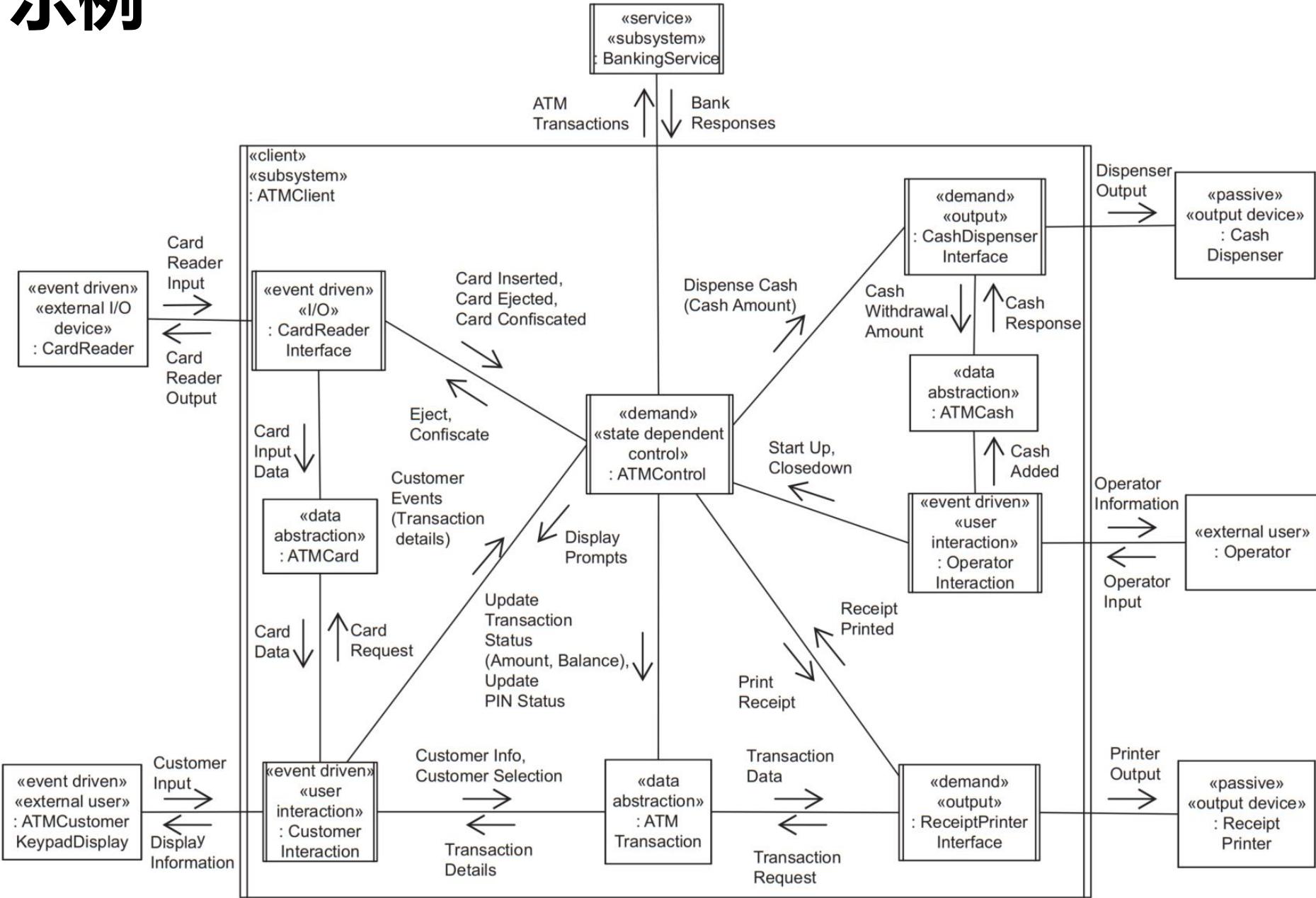


Figure 21.29. Task architecture: initial concurrent communication diagram for ATM client subsystem

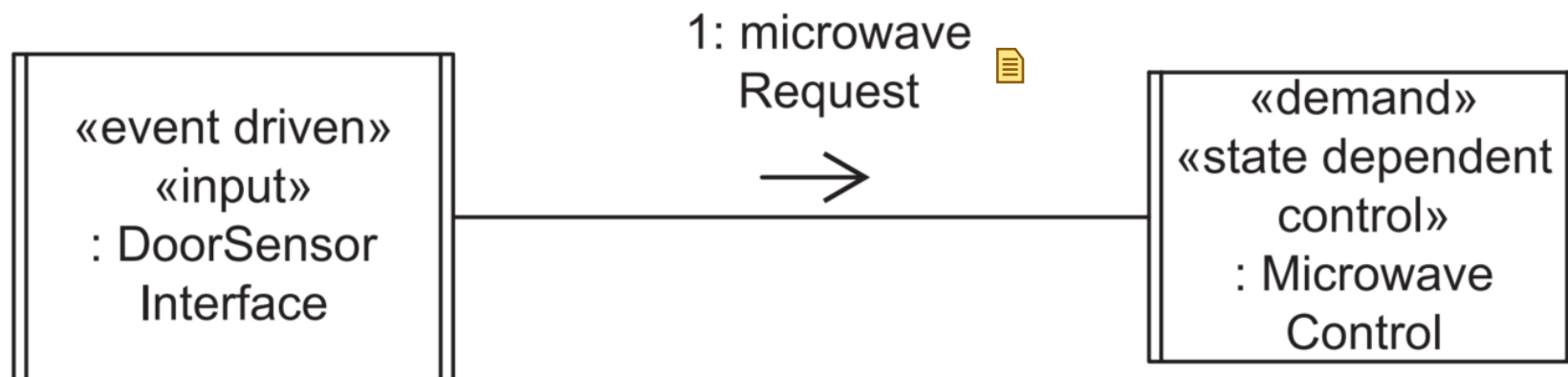
- **任务之间的消息映射**: 消息通信(message communication)、事件同步(event synchronization)、对信息隐藏对象的访问(access to information hiding objects)
- 任务之间的消息可以是**异步的(松耦合)**或**同步的(紧耦合)**

任务通信: 示例1



- **异步消息通信**: 消息生产者给消息消费者发送一个消息后, **不用等待回复**便可继续执行

- **例**: 生产者任务(门传感器接口)与消费者任务(微波炉控制)之间设计为异步消息通信



NOTE: 此例与P27页示例的区别与联系 → 将 `microwaveRequest` 作为 内部事件 更合适!

Figure 18.14. Example of Asynchronous Message Communication

任务通信: 示例2



■ **带回复的同步消息通信**: 常用于**客户端/服务器系统**中, 也可用于**单个消息生产者向单个消息消费者**发送消息并等待回复的场景

➤ **例**: 生产者任务(车辆控制)向消费者任务(发动机接口)发送开始和停止消息, 等待回复

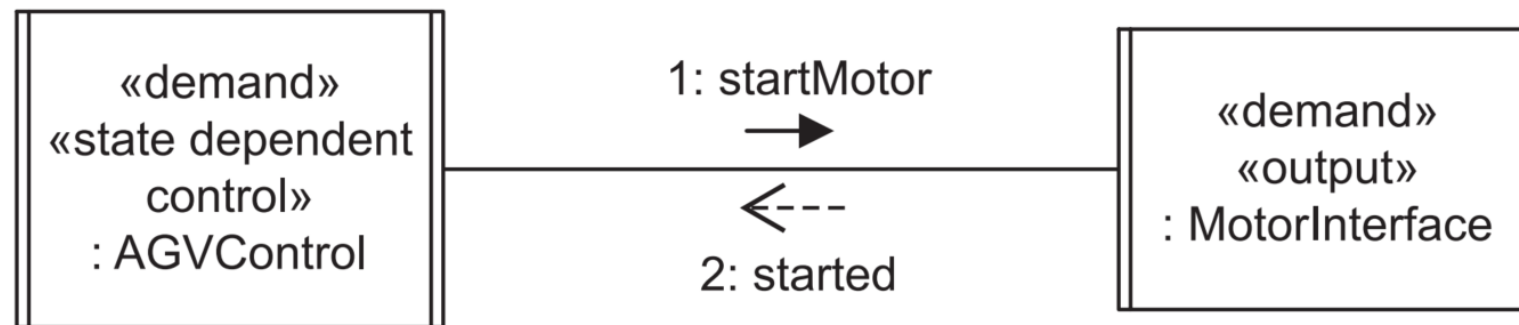


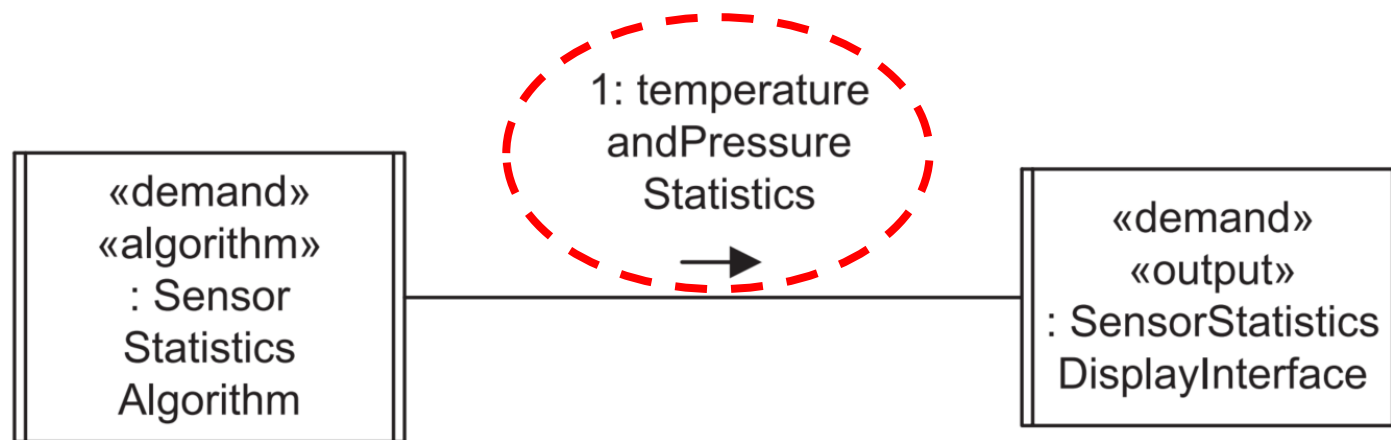
Figure 18.15. Example of Synchronous Message Communication with Reply

任务通信: 示例3



■ **不带回复的同步消息通信**: 消息生产者向消息消费者发送消息, **等待消息被接收**; 当消息到达时, 消费者接收消息并释放生产者

➤ **例**: 传感器数据显示接口是一个**按需输出任务**; 生产者任务(传感器数据算法)向消费者任务(传感器数据显示接口)发送温度和压力信息



按需驱动
输出任务

Figure 18.16. Example of Synchronous Message Communication without Reply

■ 三种事件同步类型

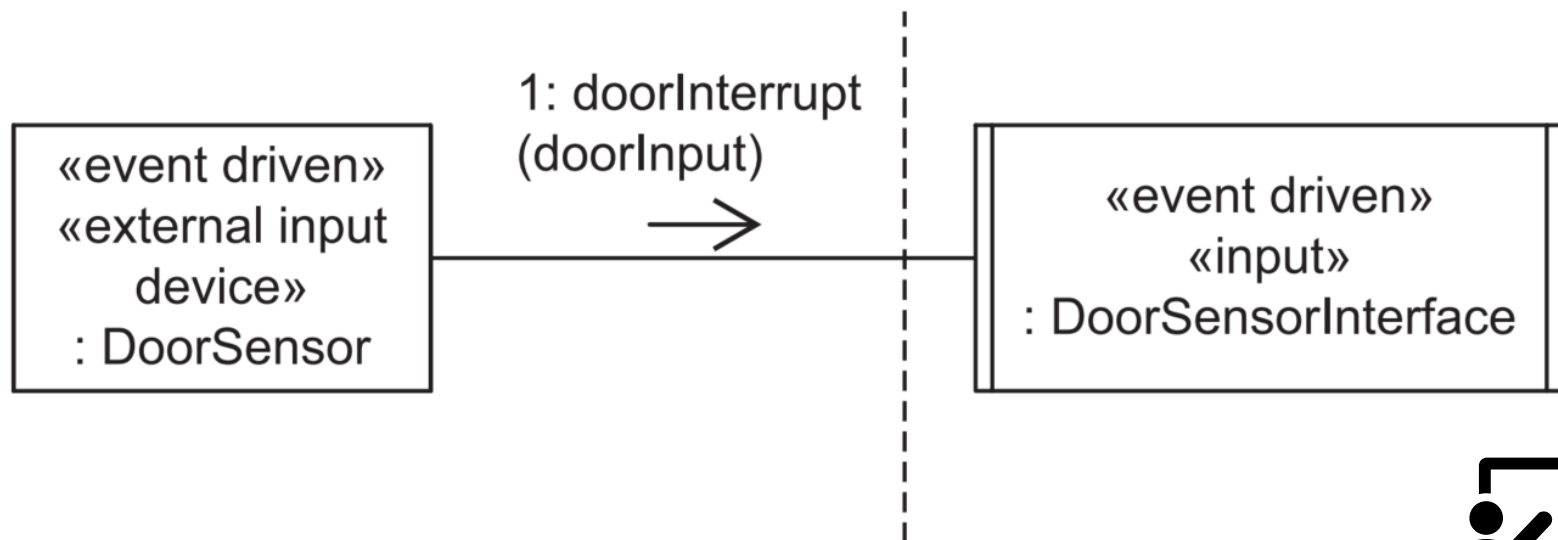
- **外部事件**: 来自外部对象的事件，通常是外部I/O设备的**中断**
- **内部事件**: 源任务与目标任务之间的内部事件
- **计时器事件**: 一个任务的周期性触发

■ UML中使用异步消息表示一个事件信号

事件同步: 示例1



- **例:** 门传感器接收到输入时会产生一个中断，触发门传感器接口任务



在事件驱动的I/O任务中，
中断应表示为异步消息

Hardware / software boundary

(Note: the dashed line for the hardware/software boundary is for illustrative purposes only and does not conform to the UML notation.)

Figure 18.17. Example of external event

事件同步: 示例2



■ 当任务之间需要同步但不进行数据通信时使用内部事件同步

- 源任务发出事件信号，目标任务在信号达到之前保持等待并挂起

■ UML中，事件信号表示为不包含任何数据的异步消息

- 例：“拾取-放置机器人”发出事件信号 partReady 唤醒“钻孔机器人”执行钻孔操作，完成后发出“拾取-放置机器人”正在等待的事件信号 partCompleted

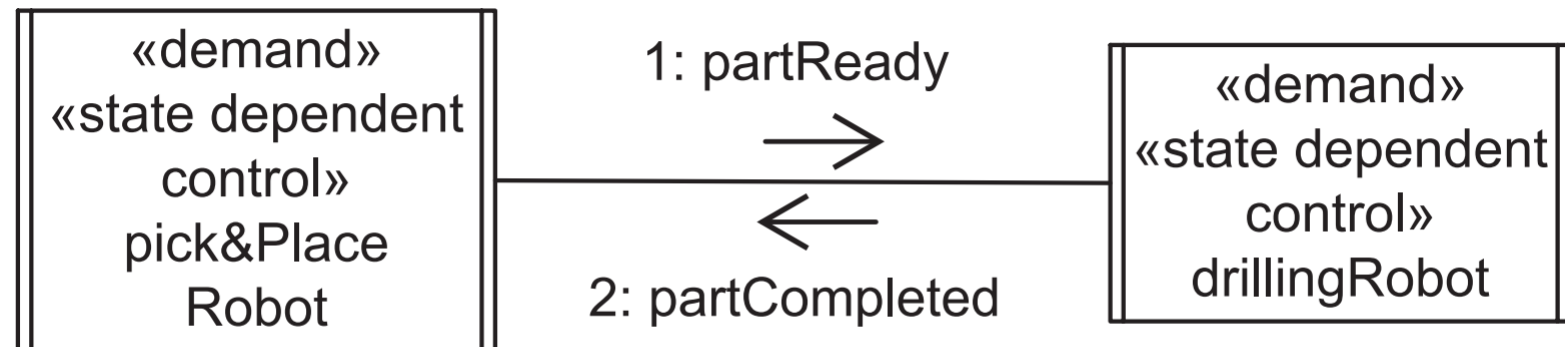
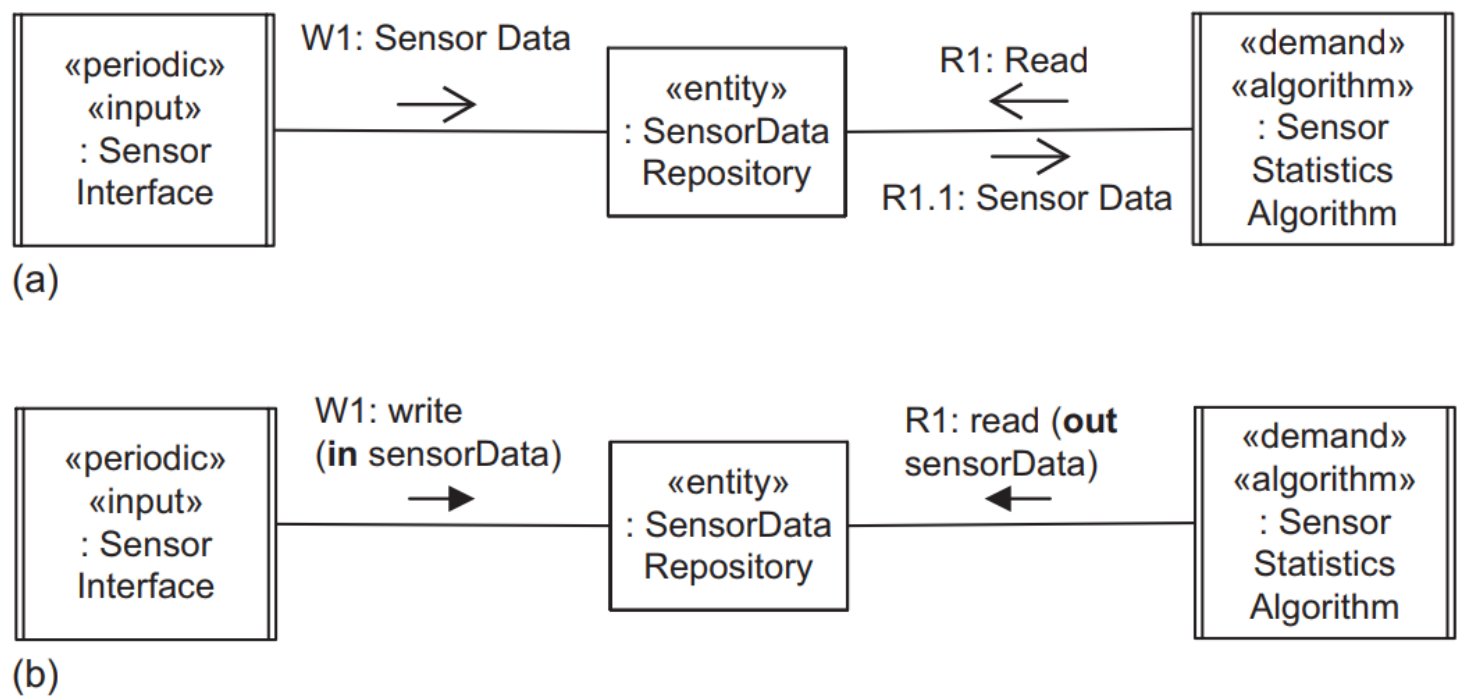


Figure 18.19. Example of internal events



■ 任务之间可通过被动信息隐藏对象交换信息



“传感器数据算法”任务从“传感器数据存储”实体对象读取数据，而“传感器接口”任务则更新此实体对象

从被动信息隐藏对象读取数据对应于一个**操作调用**，用**同步消息**表示

Figure 18.20. Example of tasks invoking operations of passive object:
(a) Initial concurrent communication diagram with simple messages.
(b) Revised concurrent communication diagram with tasks invoking operations of passive object

SSE212课程示例. 银行系统

- 例：银行系统的
子系统间的(高层)
并发通信图

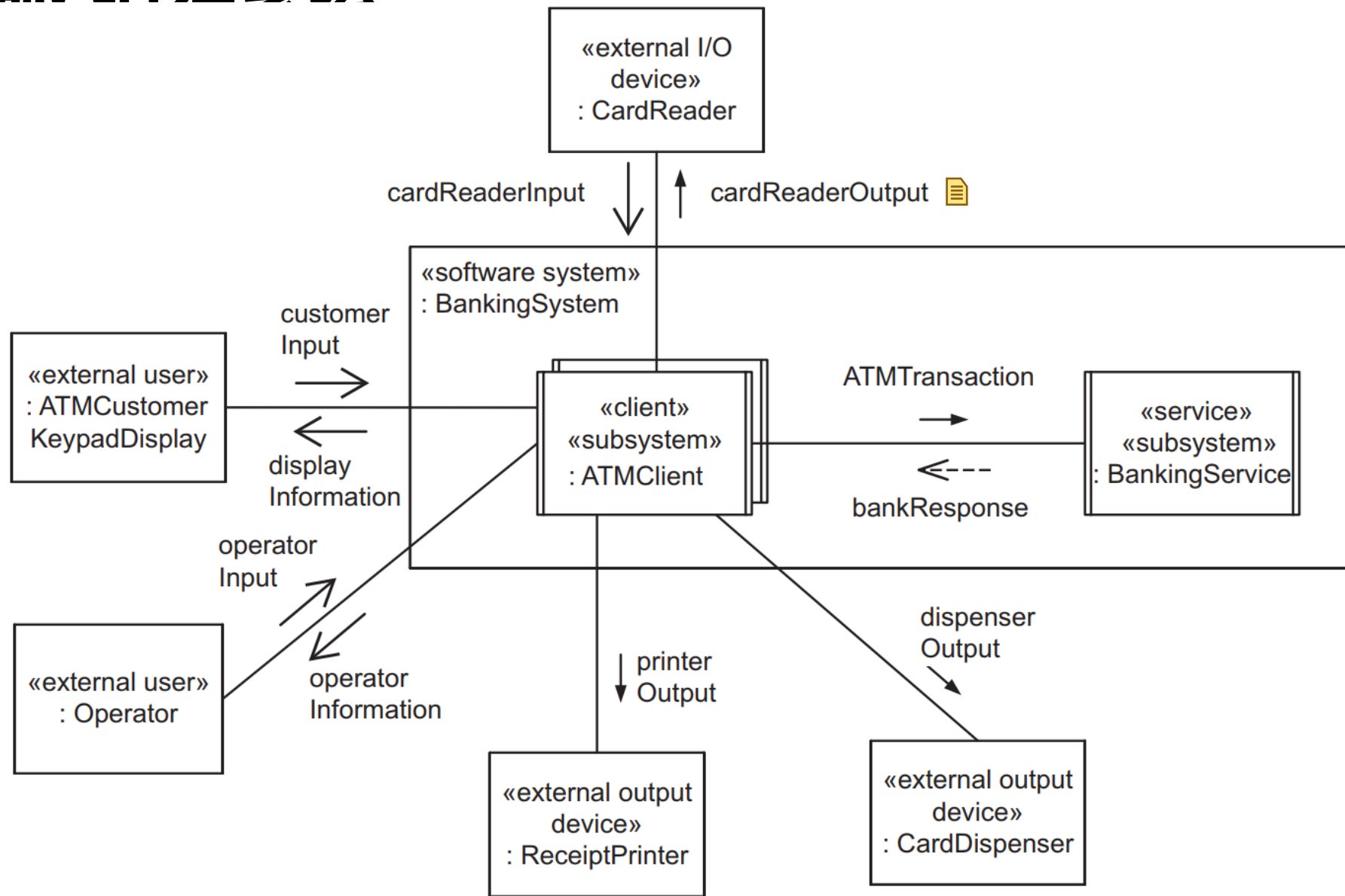


Figure 21.28. Subsystem interfaces: high-level concurrent communication diagram for Banking System

SSE212课程

➤ 例：银行系统的 ATM客户端子系 统的并发通信图

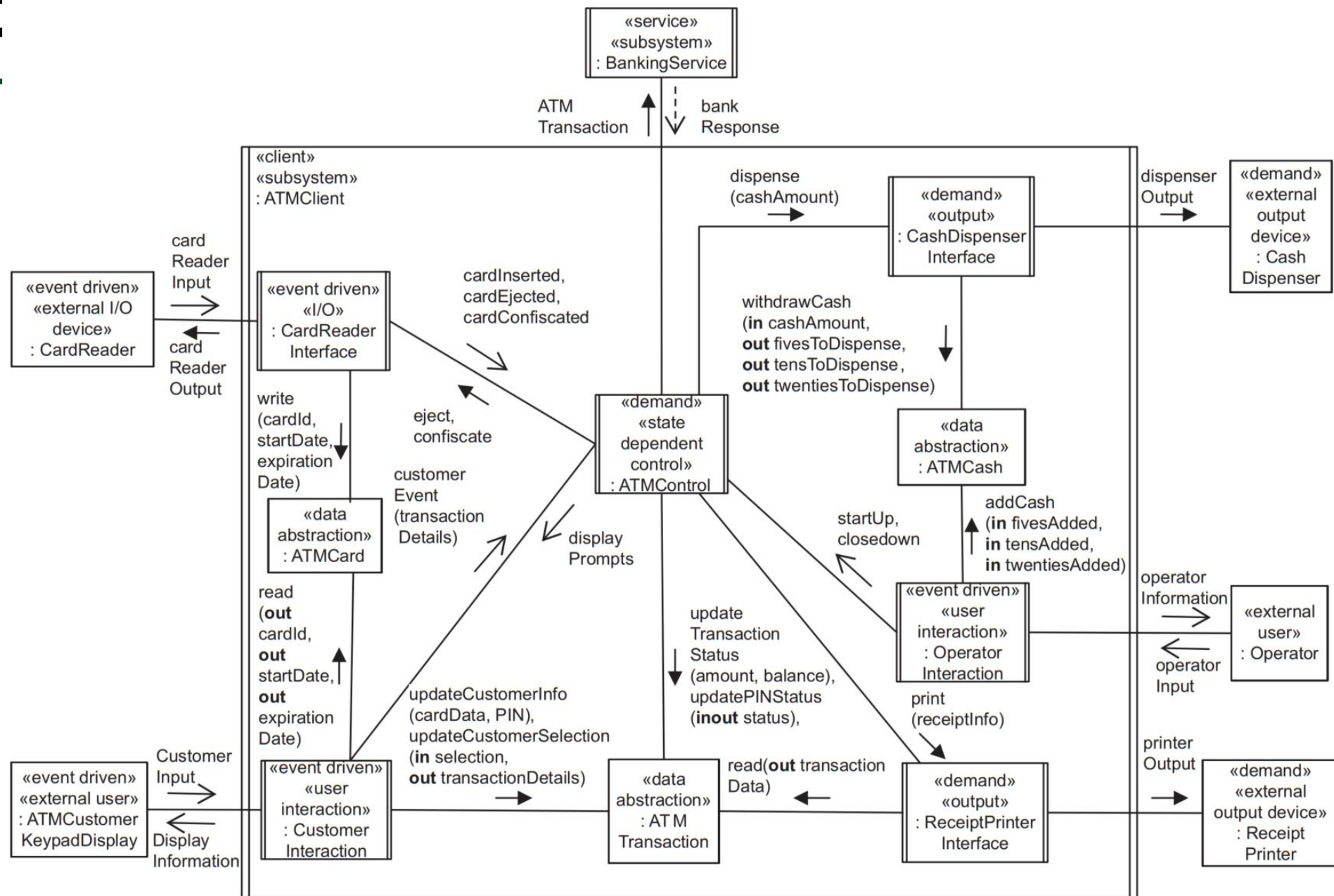


Figure 21.30. Task architecture: revised concurrent communication diagram for ATM client subsystem

SSE212课程示例: 银行系统

➤ 例：银行系统的服务子系统的并发通信图

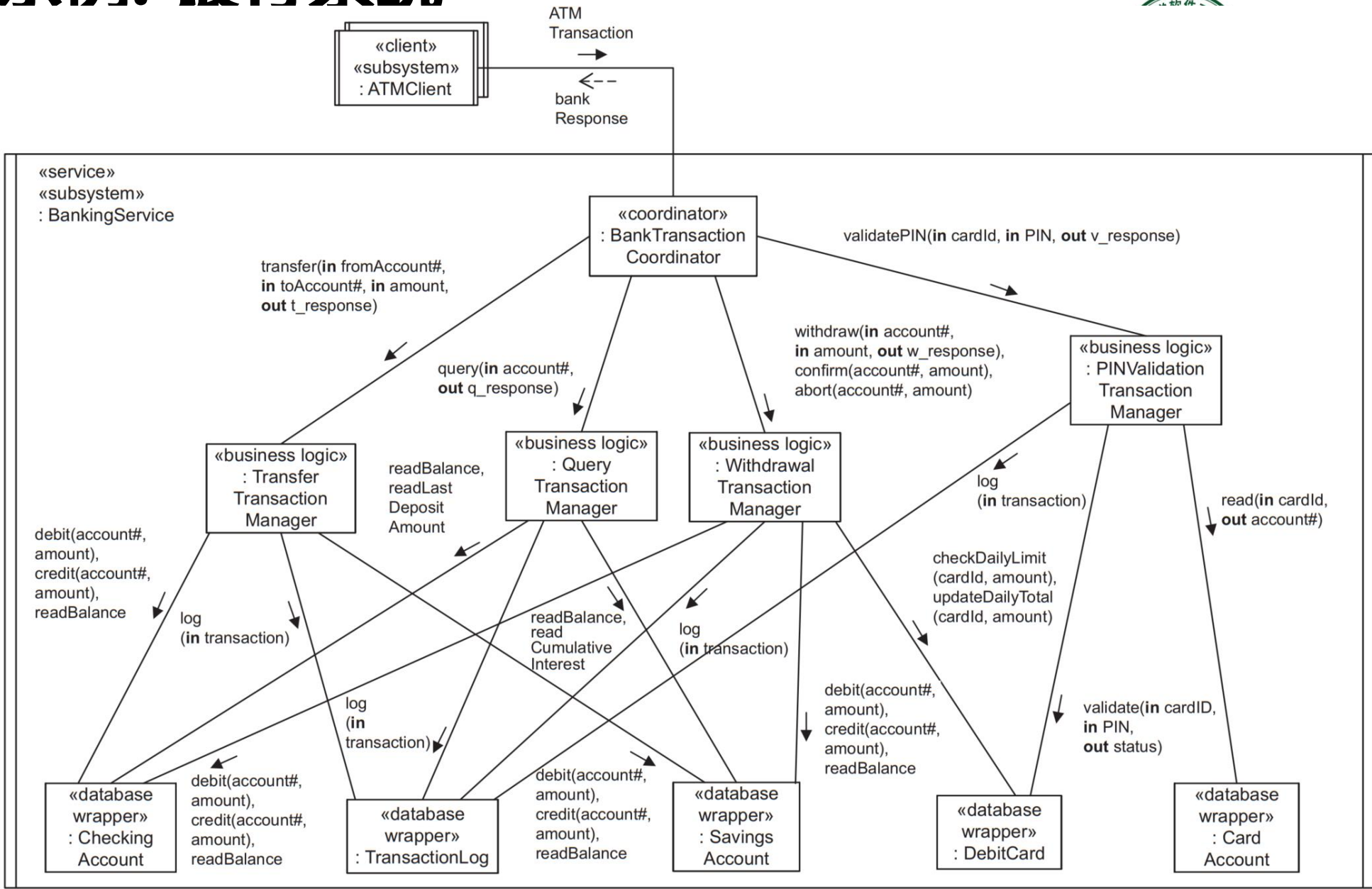


Figure 21.35. Revised concurrent communication diagram for Banking Service subsystem



SSE212课程实践系统-- 图书自助借还系统

SSE212课程实践: 集成通信建模+子系统划分&设计



■ 实验产出

- ✓ 子系统划分结果, 即存在哪些子系统: 给出划分依据
- 子系统之间的并发通信图: [建议](#)给出并发任务及通信类型的依据
- 各个子系统的并发通信图: [建议](#)给出并发任务及通信类型的依据

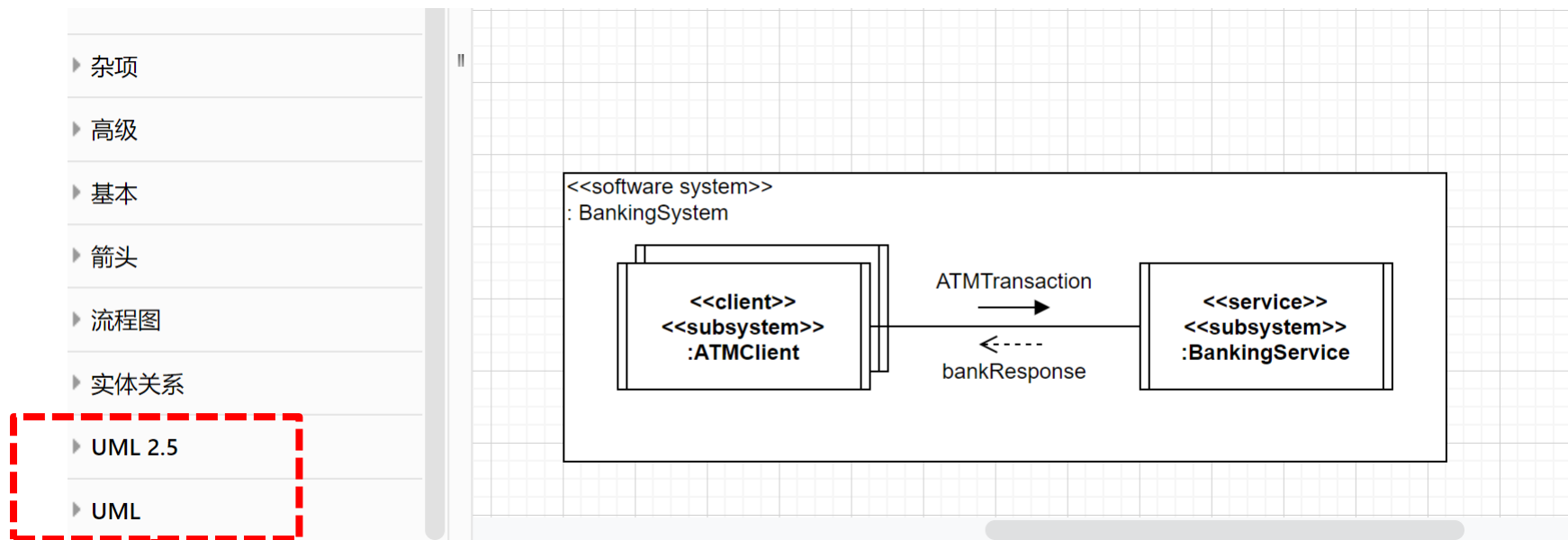


StarUML不支持并发对象和多重构造型的表示
并发通信图的绘制采用**draw.io**

Draw.io演示



- 两种使用draw.io的方法
- 1. 网页版: <https://app.diagrams.net/> 注意保存.svg文件
- 2. 桌面版: 从<https://www.drawio.com/> 下载安装之后使用



涵盖建模所有UML图的元素