



拥抱变化的软件架构 (Agility Architecture) --- Rest Oriented Architecture (ROA) 设计

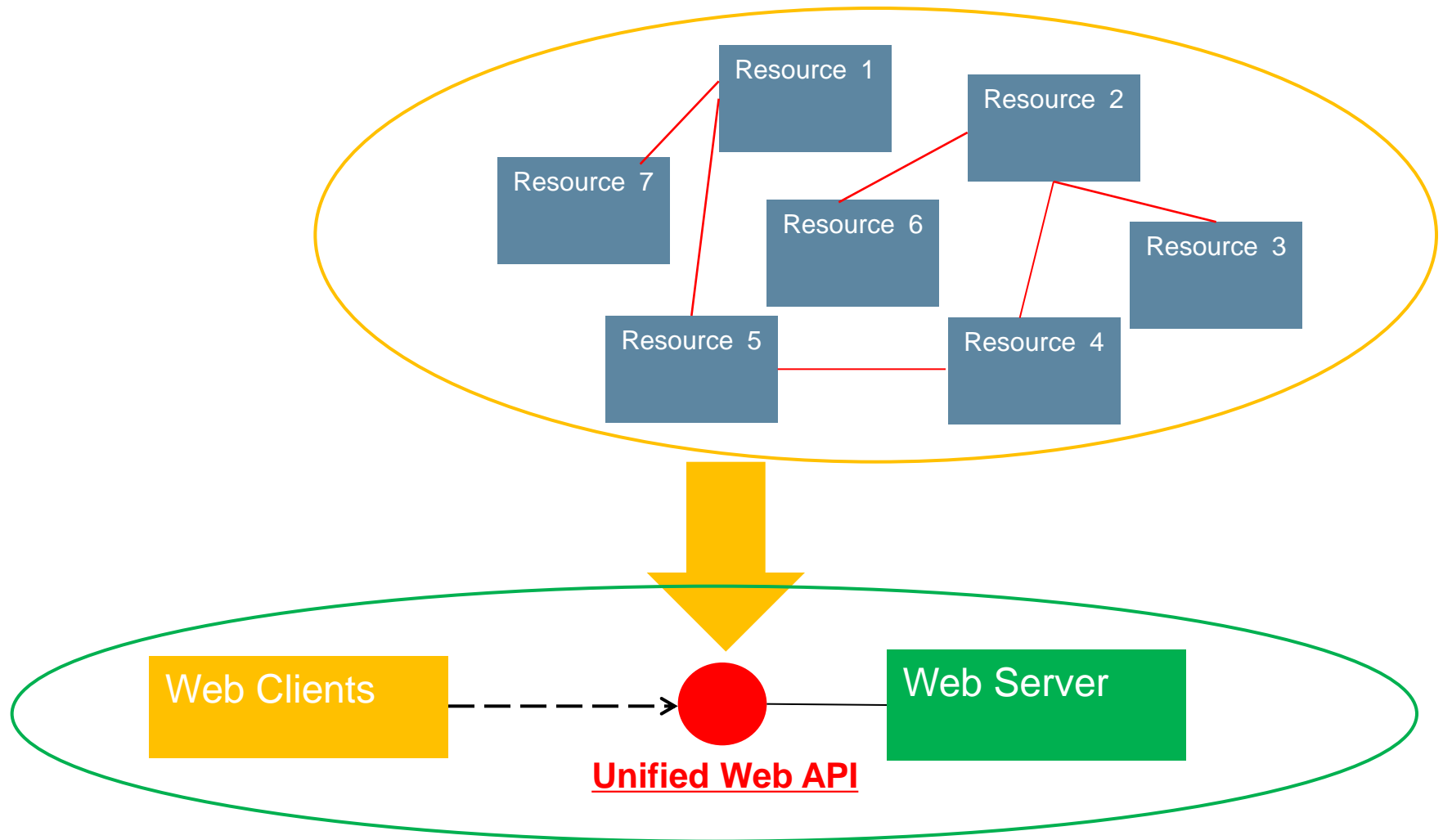
方俊贤 (Ken Fang)

首席咨询顾问, 雅各布森软件 (北京) 有限公司

议题

- Agility Architecture: Rest Oriented Architecture (ROA)
- ROA 设计的原则
- ROA 设计

Agility Architecture: Rest Oriented Architecture (ROA)



ROA 设计的原则

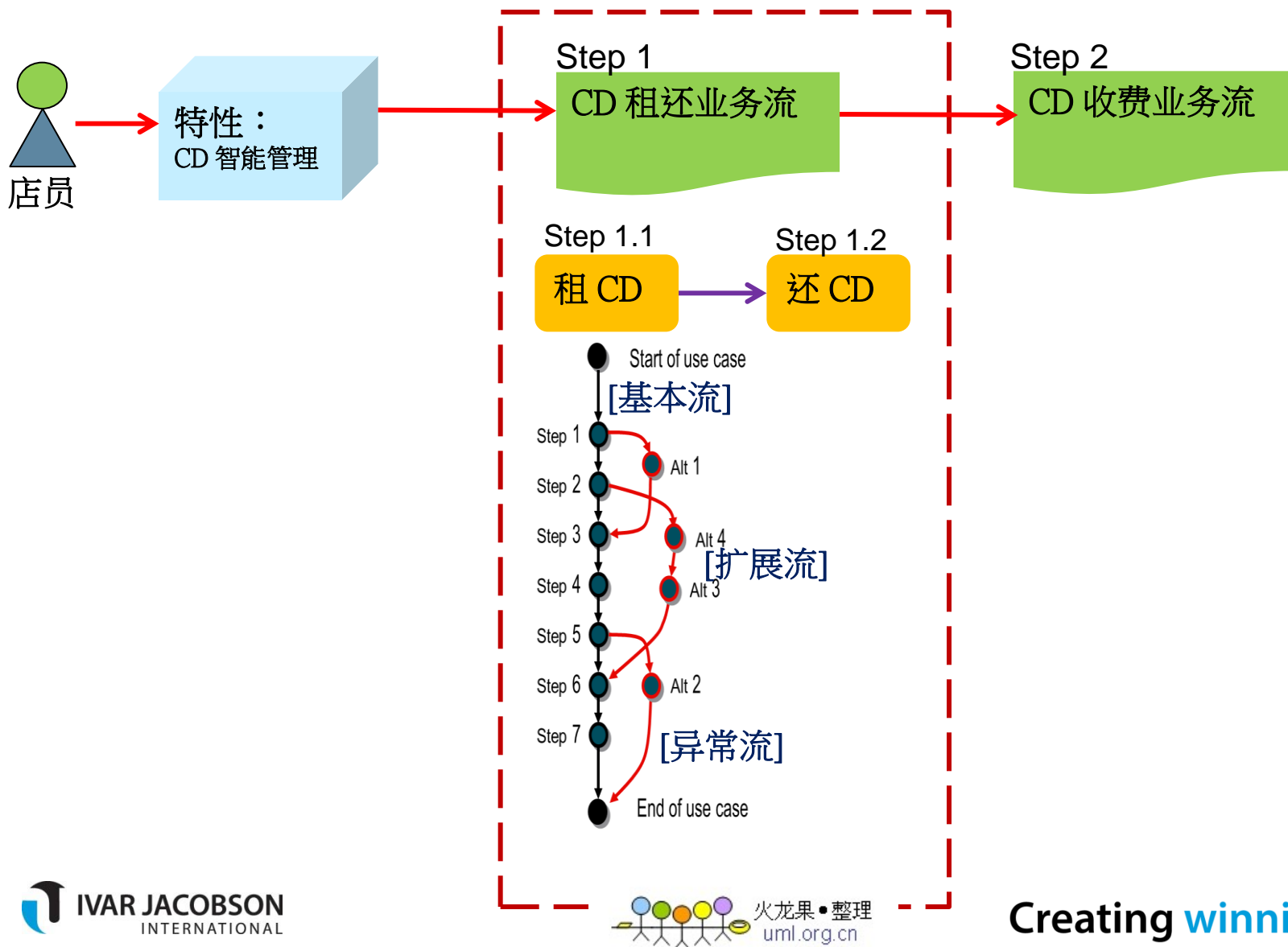
- 特性驱动
- 视觉化团队协作
- 轻量级文档

ROA 设计

- ✓ Step 1: 以特性为纬度分析业务流程
- ✓ Step 2: 识别 Entity Resources, Utility Resources
- ✓ Step 3: 识别特性的启动业务 Resources
- ✓ Step 4: 识别 HTTP 方法
- ✓ Step 5: 识别 Resource 组合
- ✓ Step 6: 定义 URI
- ✓ Step 7: 定义 representation class

Note: 此处的设计主要是针对功能性需求。

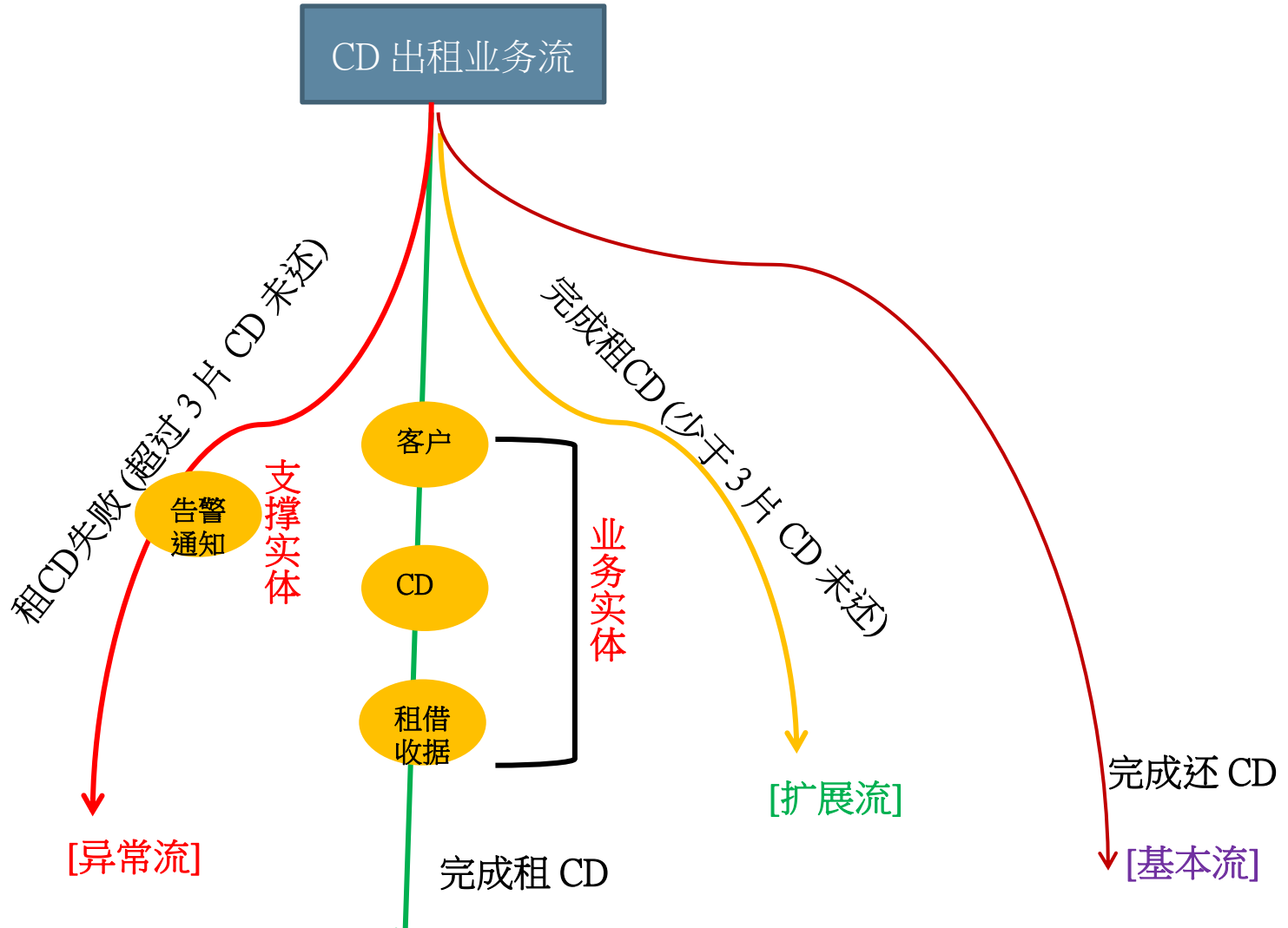
Step 1: 以特性为纬度分析业务流程



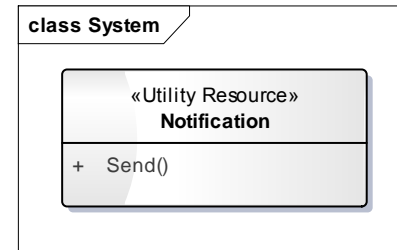
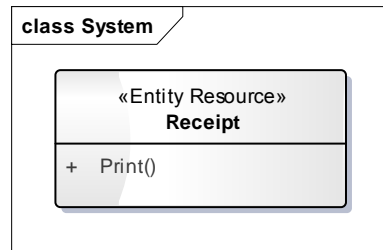
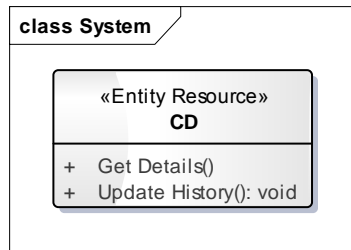
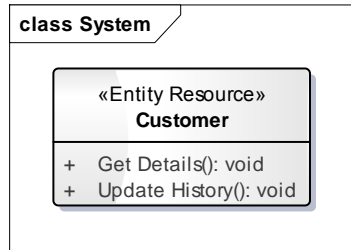
以特性为纬度分析业务流程工具: 特性业务流程看板

| 业务流 | CD 租还 业务流 | | CD 收费 业务流 |
|-----|-------------------------|------|--------------|
| 活动 | 租 CD | 还 CD | |
| 基本流 | 完成租CD | | |
| 扩展流 | 完成租CD (少于 3 片 CD 未还) | | |
| 异常流 | 租CD失败 (超过 3 片 CD 未还) | | |

以特性为纬度分析业务流程工具: 路径树



Step 2: 识别 Entity Resources, Utility Resources



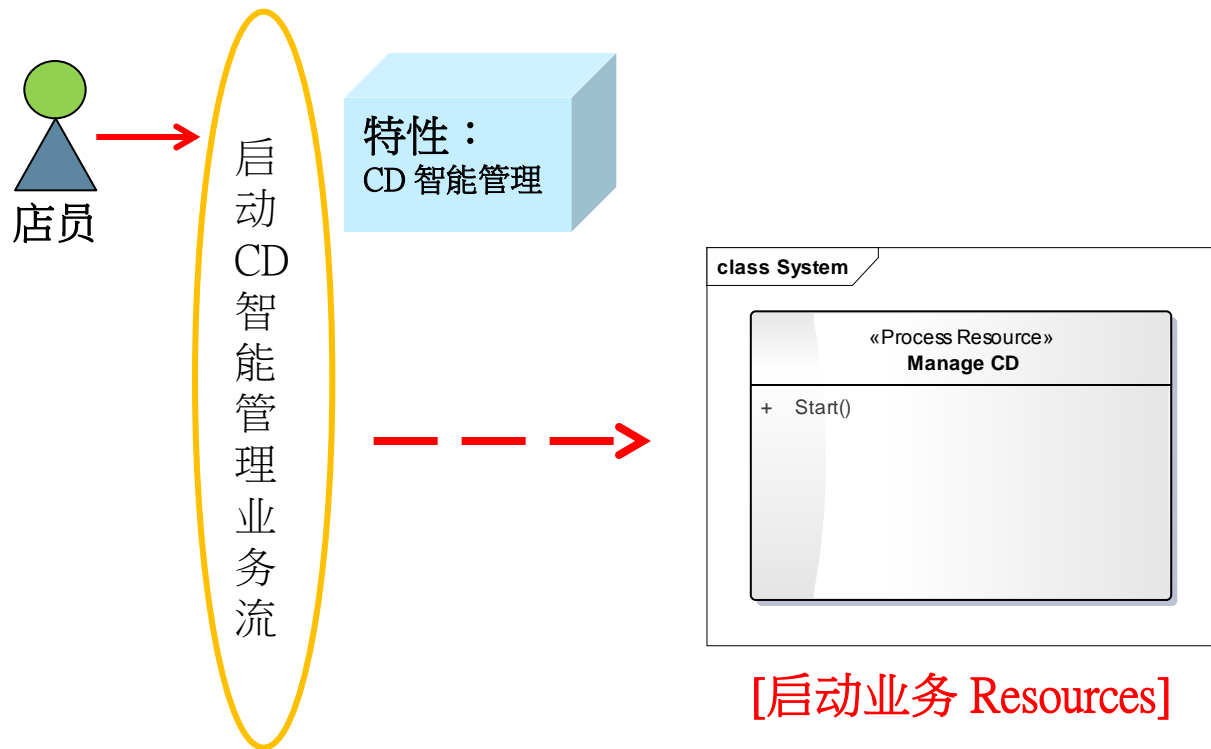
由路径树中的业务实体, 识别出 Entity Resources

| 业务实体 | Entity Resources |
|------|------------------|
| 客户 | Customer |
| CD | CD |
| 租借收据 | Receipt |

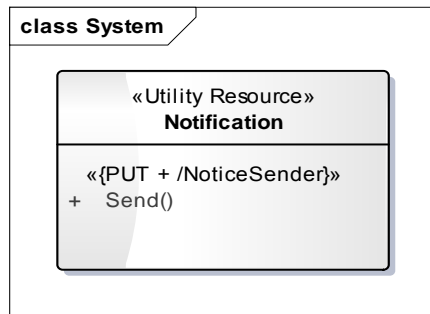
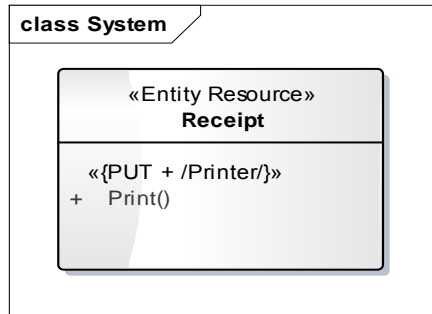
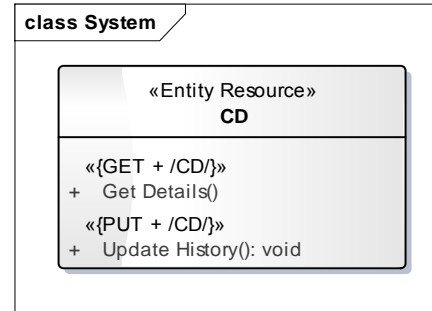
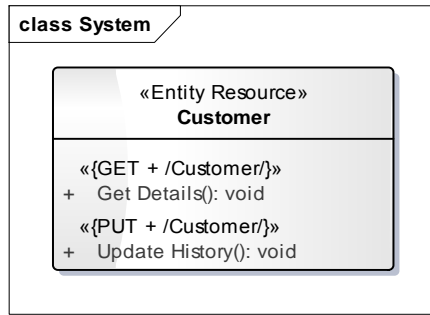
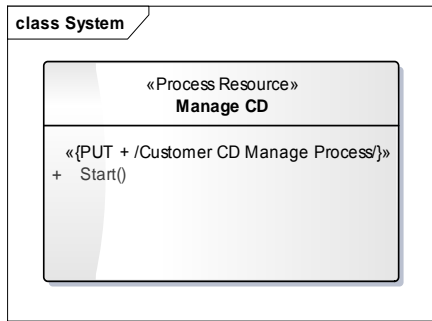
由路径树的支撑实体, 识别出 Utility Resources

| 支撑实体 | Utility Resources |
|------|-------------------|
| 告警通知 | Notification |

Step 3: 识别特性的启动业务 Resources

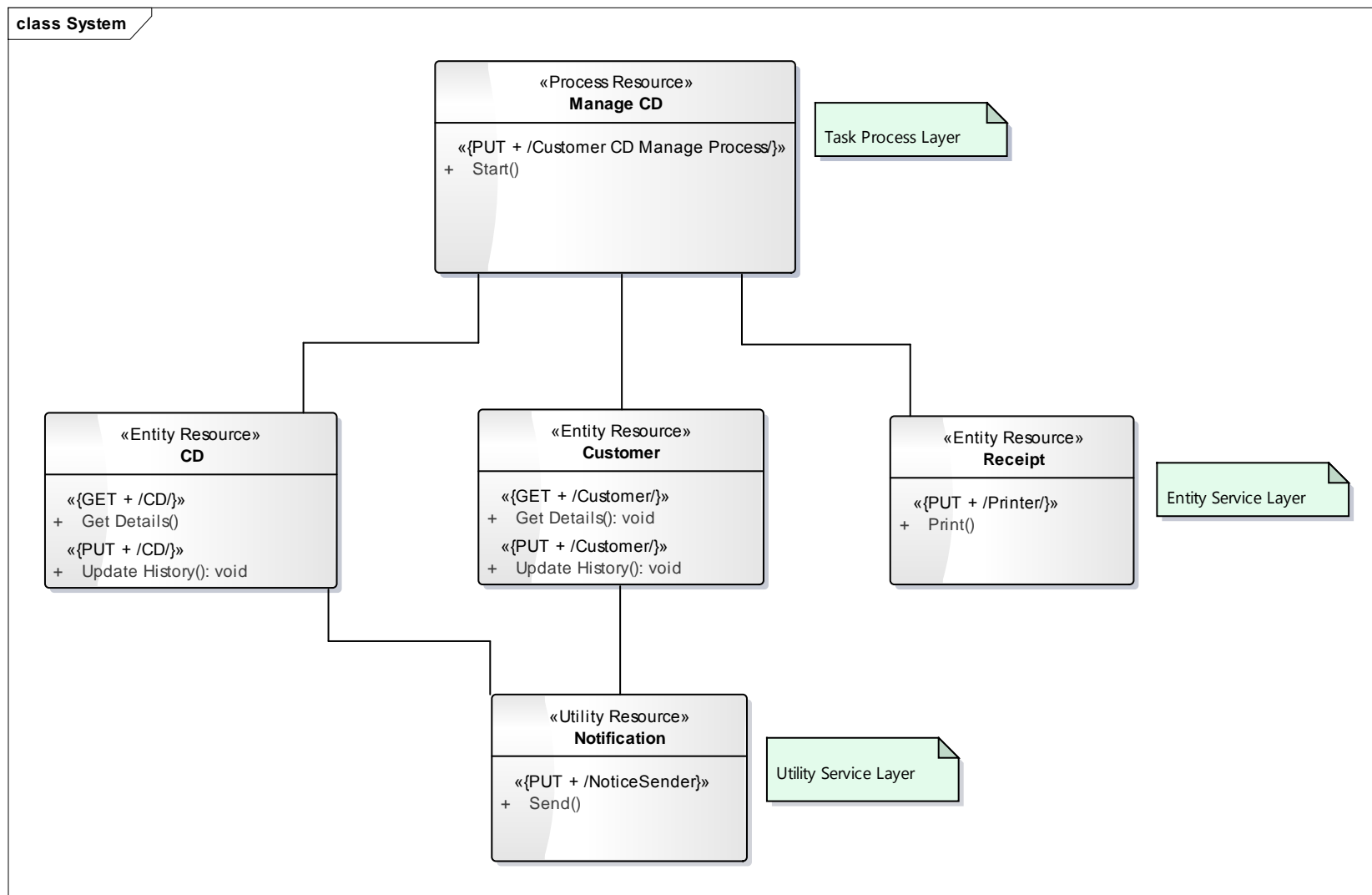


Step 4: 识别 HTTP 方法

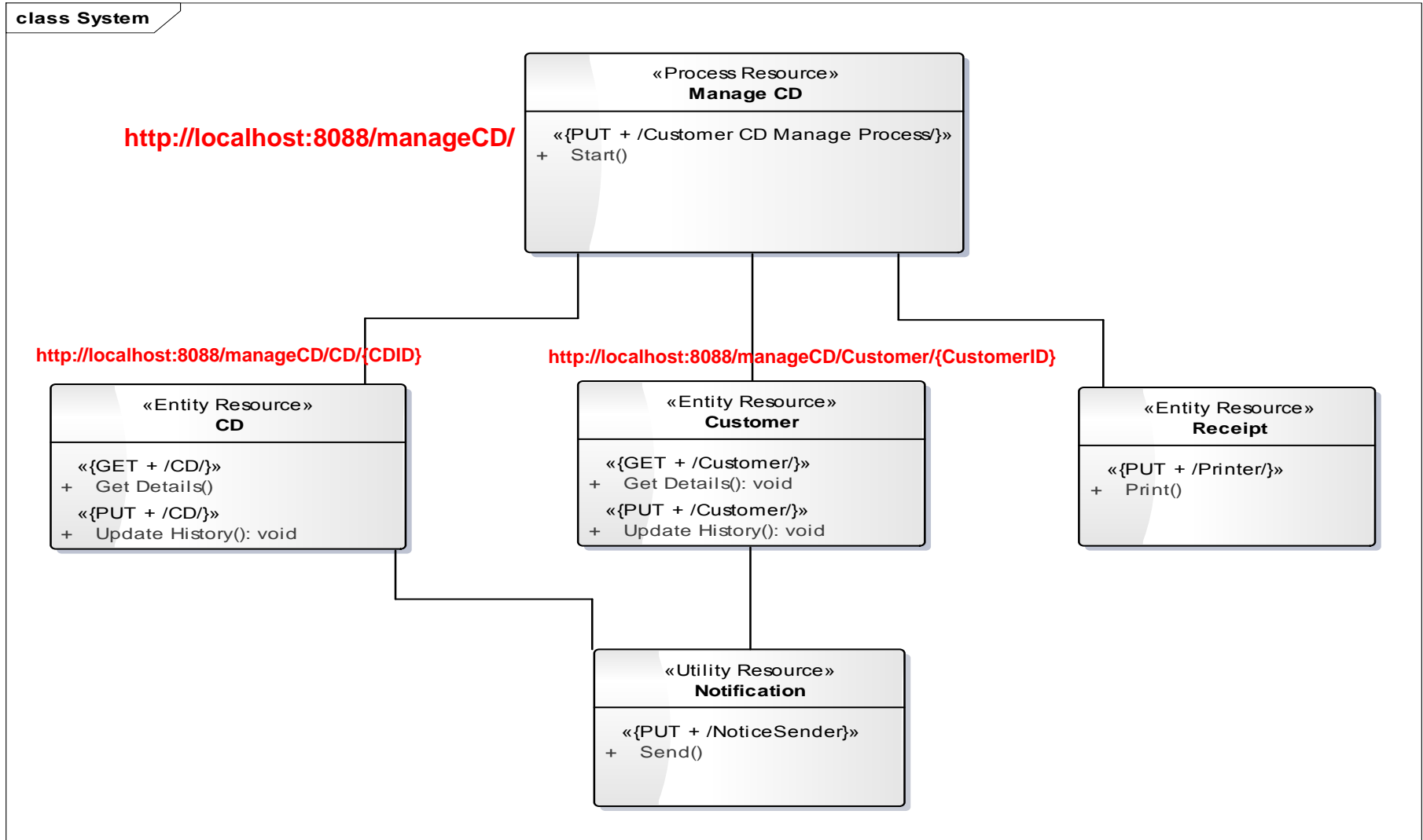


| HTTP 主要的方法 | Client 要求做的事 | 类型 |
|------------|---|---------------------------|
| GET | Give me back a representation of the target resource. | Safe and idempotent |
| PUT | Create or update the target resource with the representation I am sending you in the body of this request | Nonsafe and idempotent |
| POST | Make the target resource process the data I am sending to you in the body of this request | Nonsafe and nonidempotent |
| DELETE | Delete the target resource and its associated state | Nonsafe and idempotent |

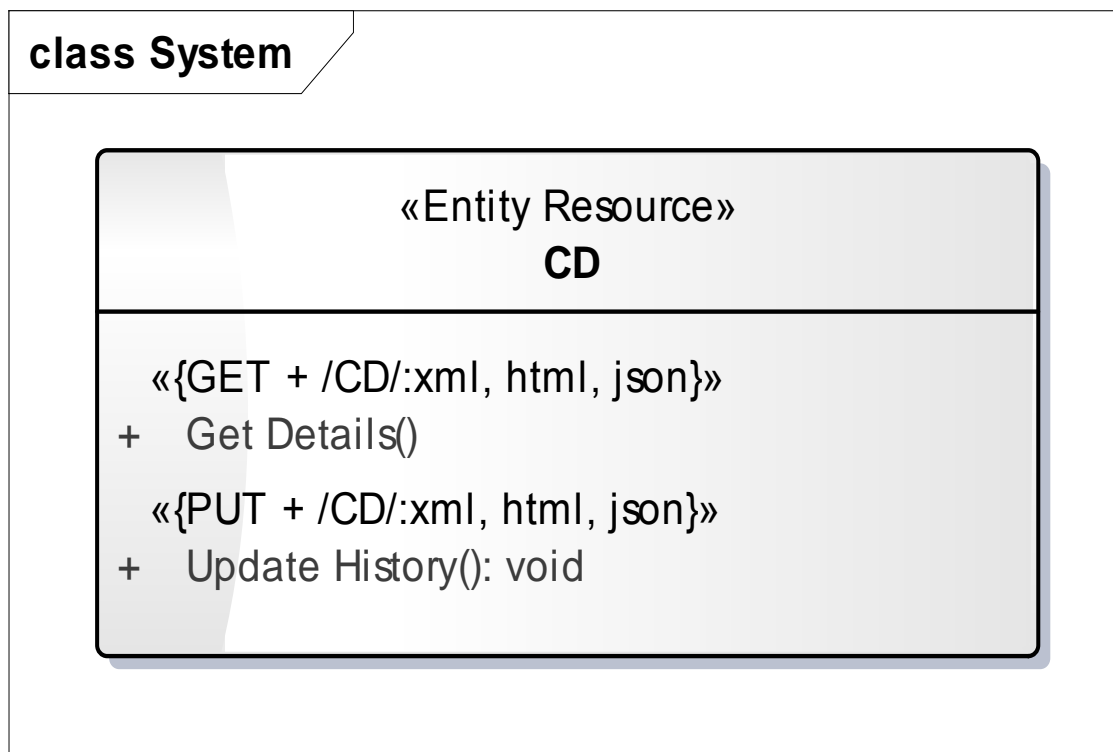
Step 5: 识别 Resource 组合



Step 6: 定义 URI



Step 7: 定义 representation class



Rest Oriented Architecture (ROA) 设计看板

| Resource | 识别 HTTP 方法 | 识别 Resource 组合 | 定义 URI | 定义 representation class |
|-----------|------------|----------------|--------|-------------------------|
| [Process] | | | | |
| [Entity] | | | | |
| [Utility] | | | | |

结论

- ✓ 经由 Rest Oriented Architecture (ROA) 设计工具
 - 特性业务流程看板
 - 路径树
 - 设计看板
- ✓ 使得 Rest Oriented Architecture (ROA) 设计……
 - 特性驱动
 - 视觉化团队协作
 - 轻量级文档