

DDD & ENODE

作者：汤雪华

博客：<http://www.cnblogs.com/netfocus>

内容提要

- DDD理论
 - DDD案例分析
 - ENode框架简介
 - ENode – Conference案例简介
-

DDD – 什么是DDD

- 领域驱动设计（Domain-driven Design）
- 一种模型驱动的软件设计方式



DDD相关领导人物

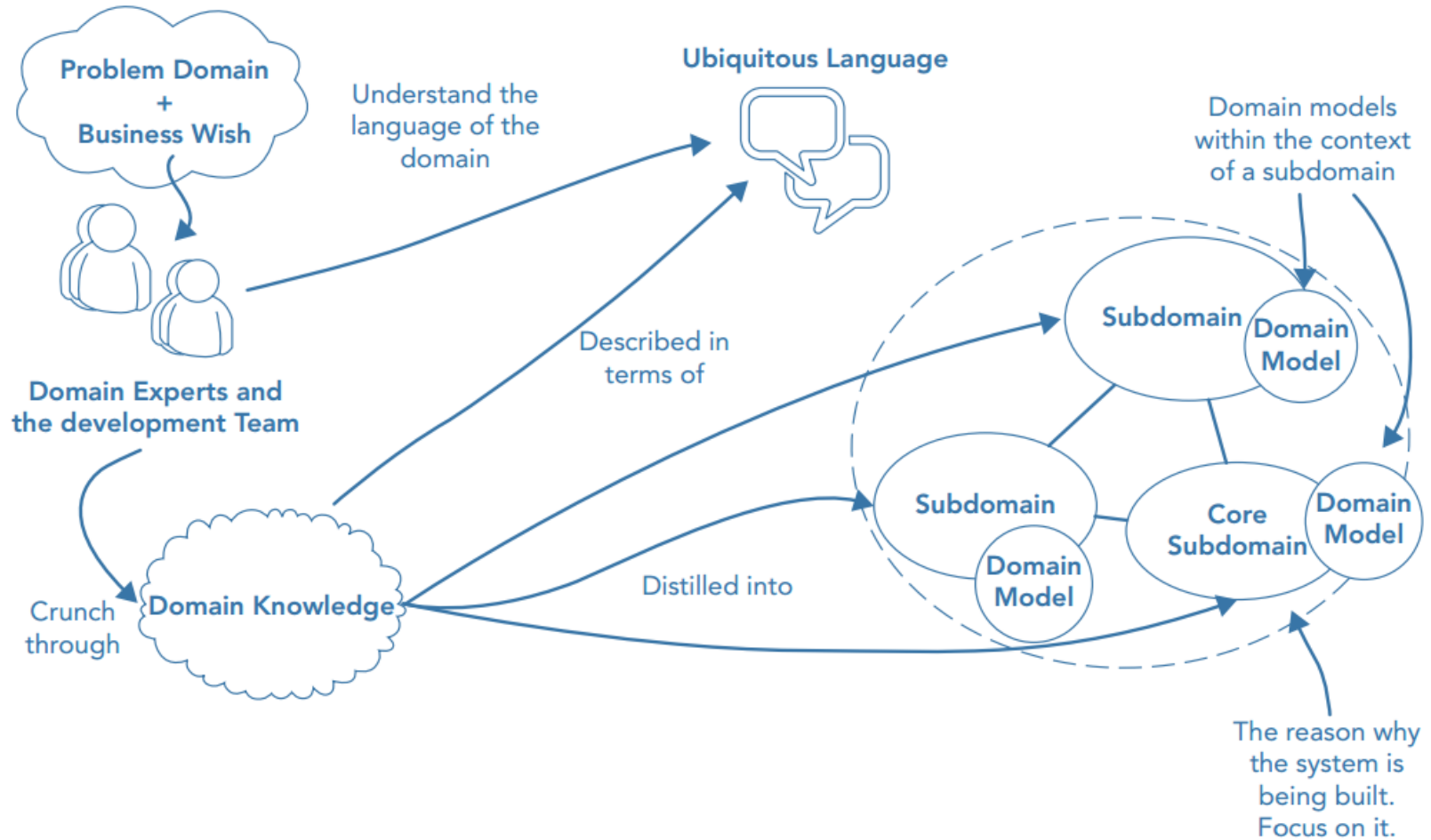
Attendees (Left to right): Paul Rayner, Randy Stafford, Eric Evans, Udi Dahan, Alberto Brandolini, Greg Young, Jimmy Nilsson, Niclas Hedhman, Vaughn Vernon. Down in front (and our photographer) Martin Fowler.



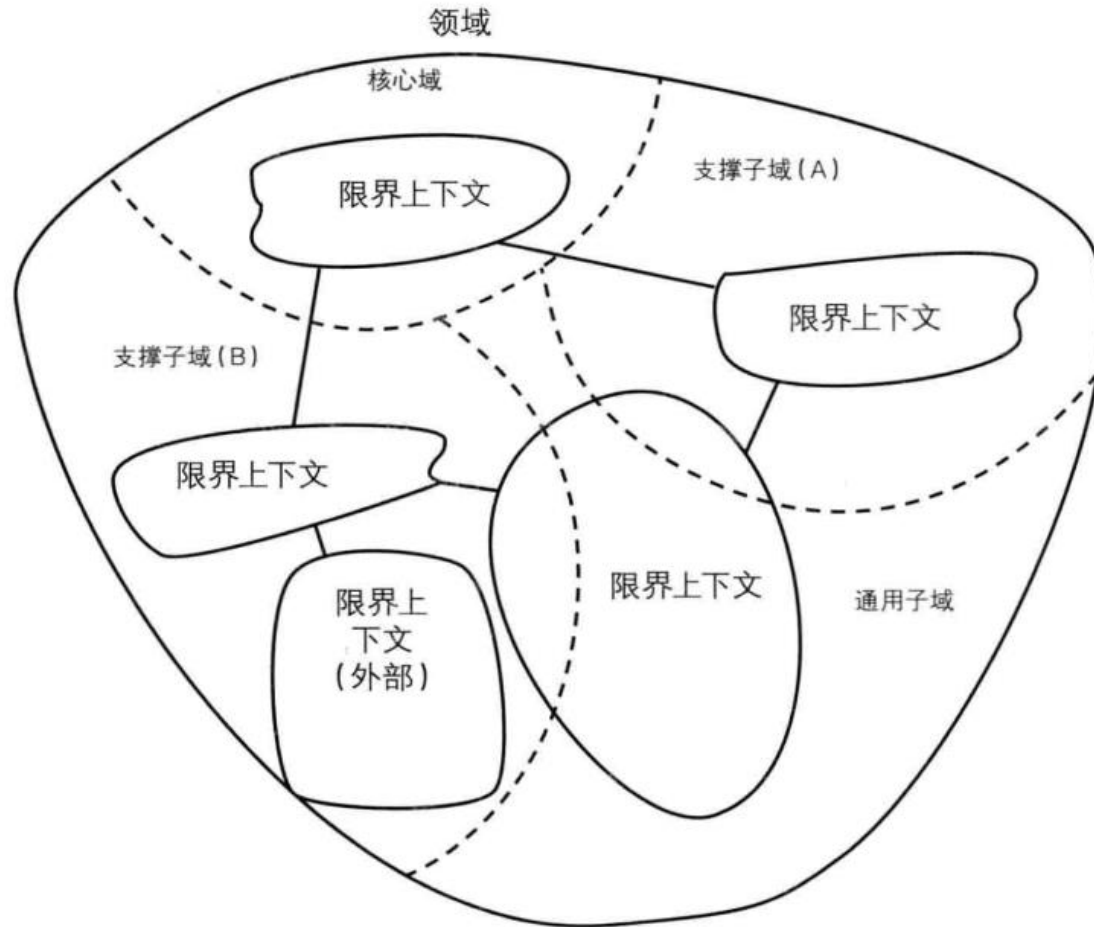
DDD – 核心概念

- 领域划分
- 统一语言
- 领域模型
- 界定上下文（Bounded Context）

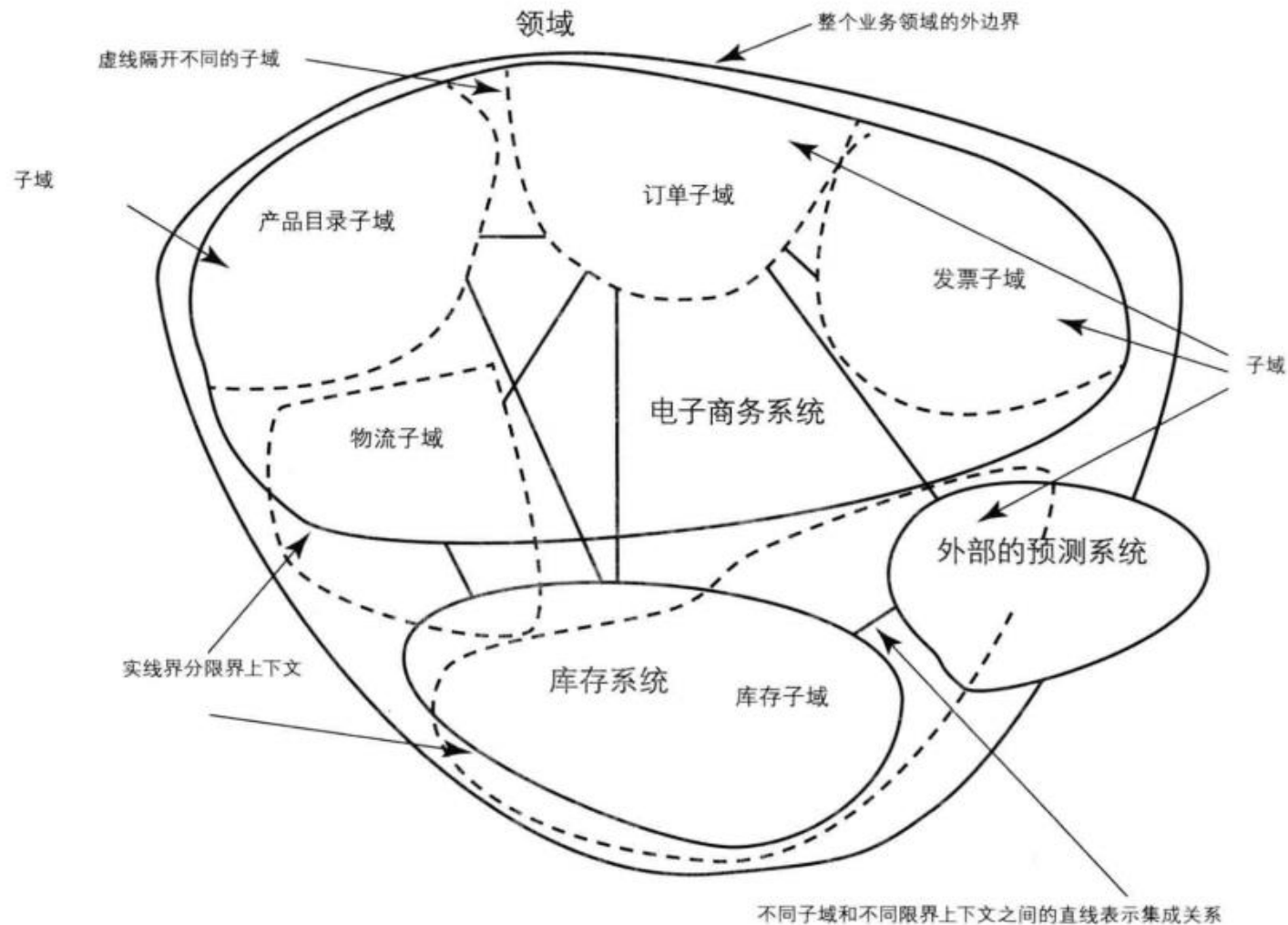
DDD – 领域划分、统一语言



DDD – Bounded Context



DDD – Bounded Context

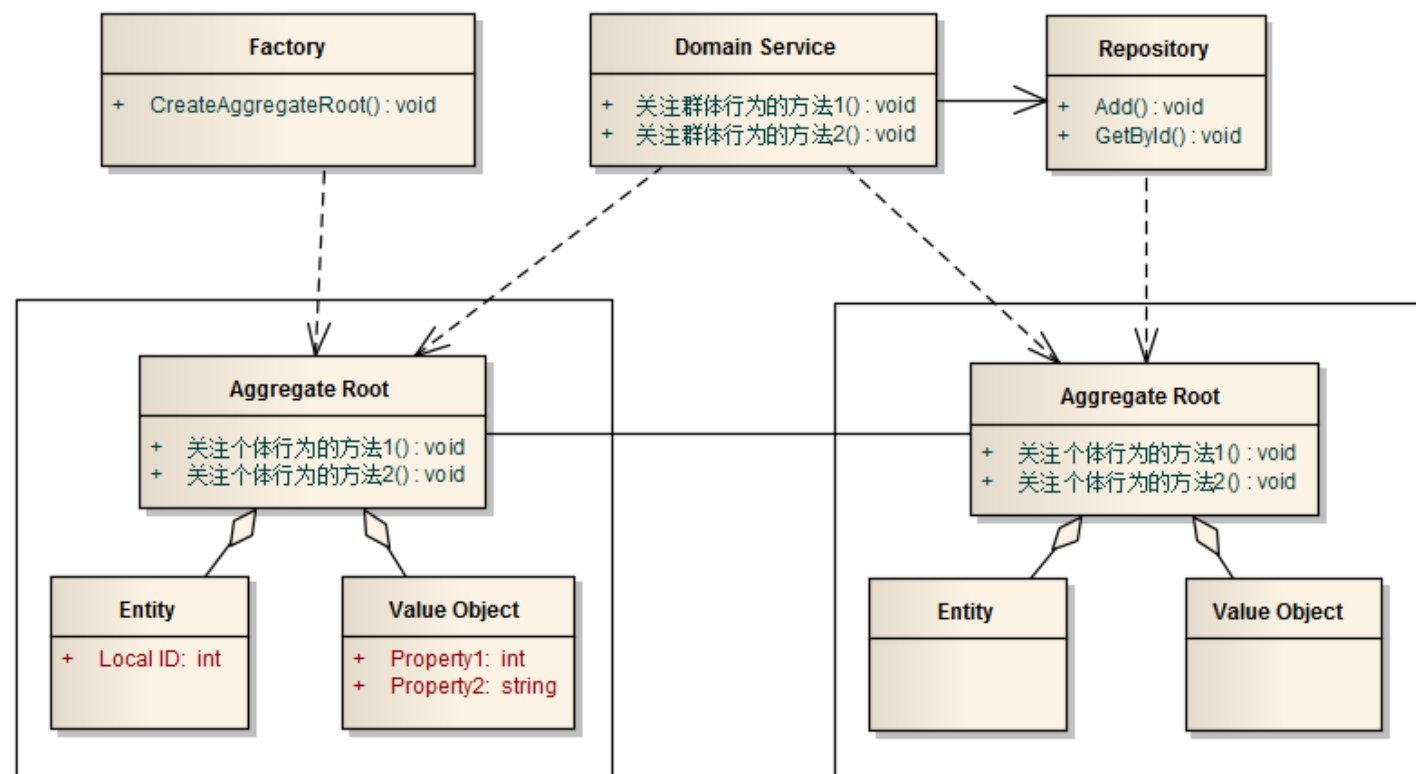


DDD – 领域模型基本构造块

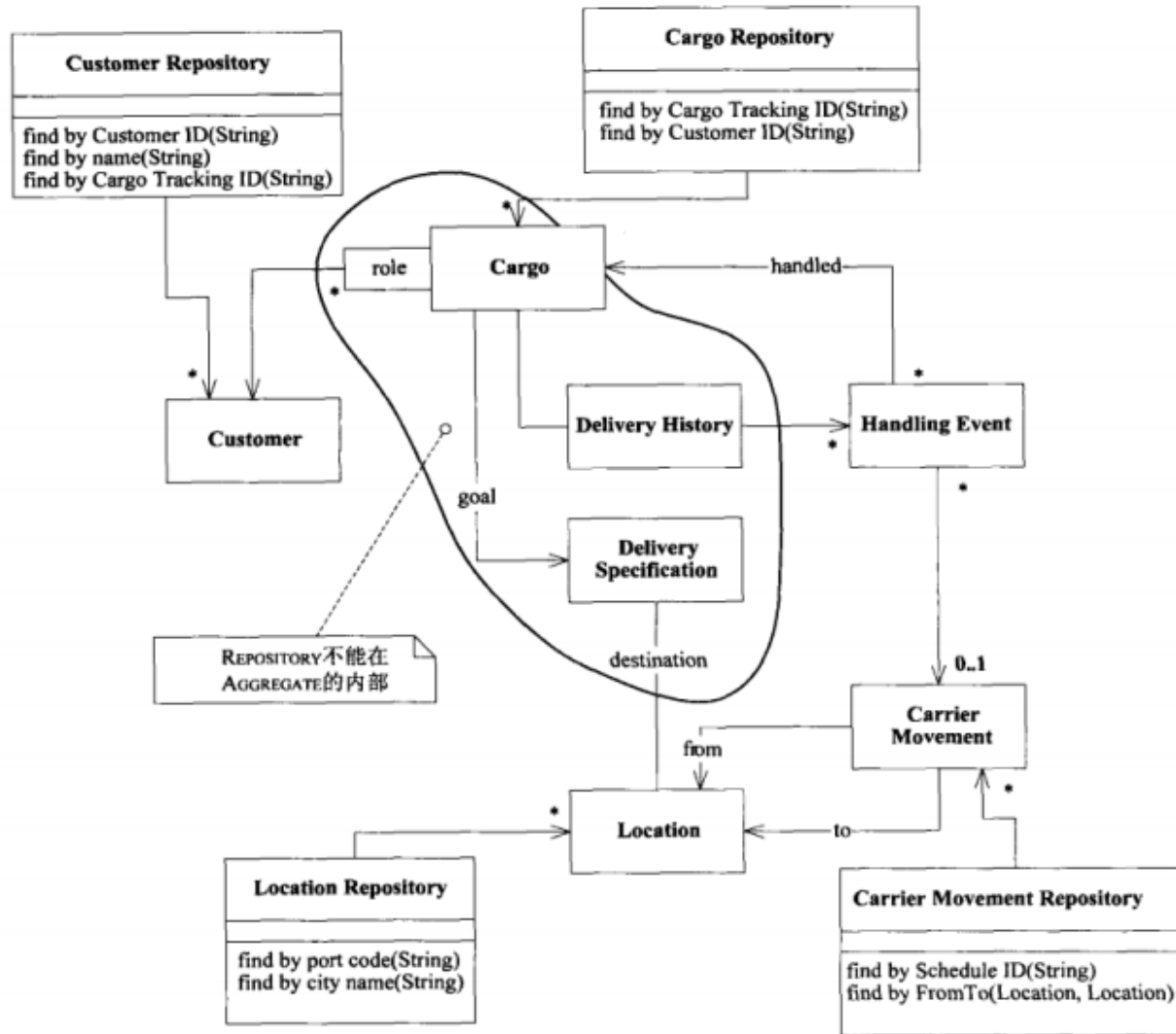
- 实体（Entity）
- 值对象（Value Object）
- 服务（Service）
- 聚合（Aggregate）
- 聚合根（Aggregate Root）
- 仓储（Repository）
- 工厂（Factory）
- 领域事件（Domain Event）

DDD – 构造块关系图

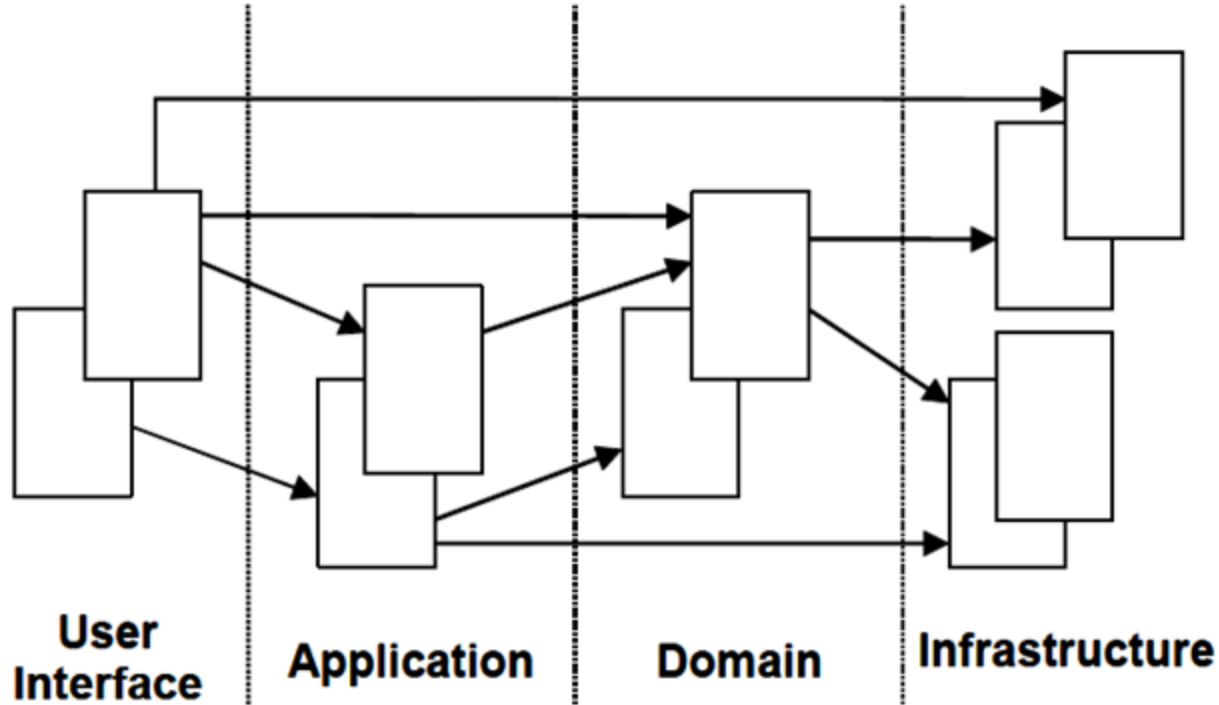
DDD领域模型各构造块关系图



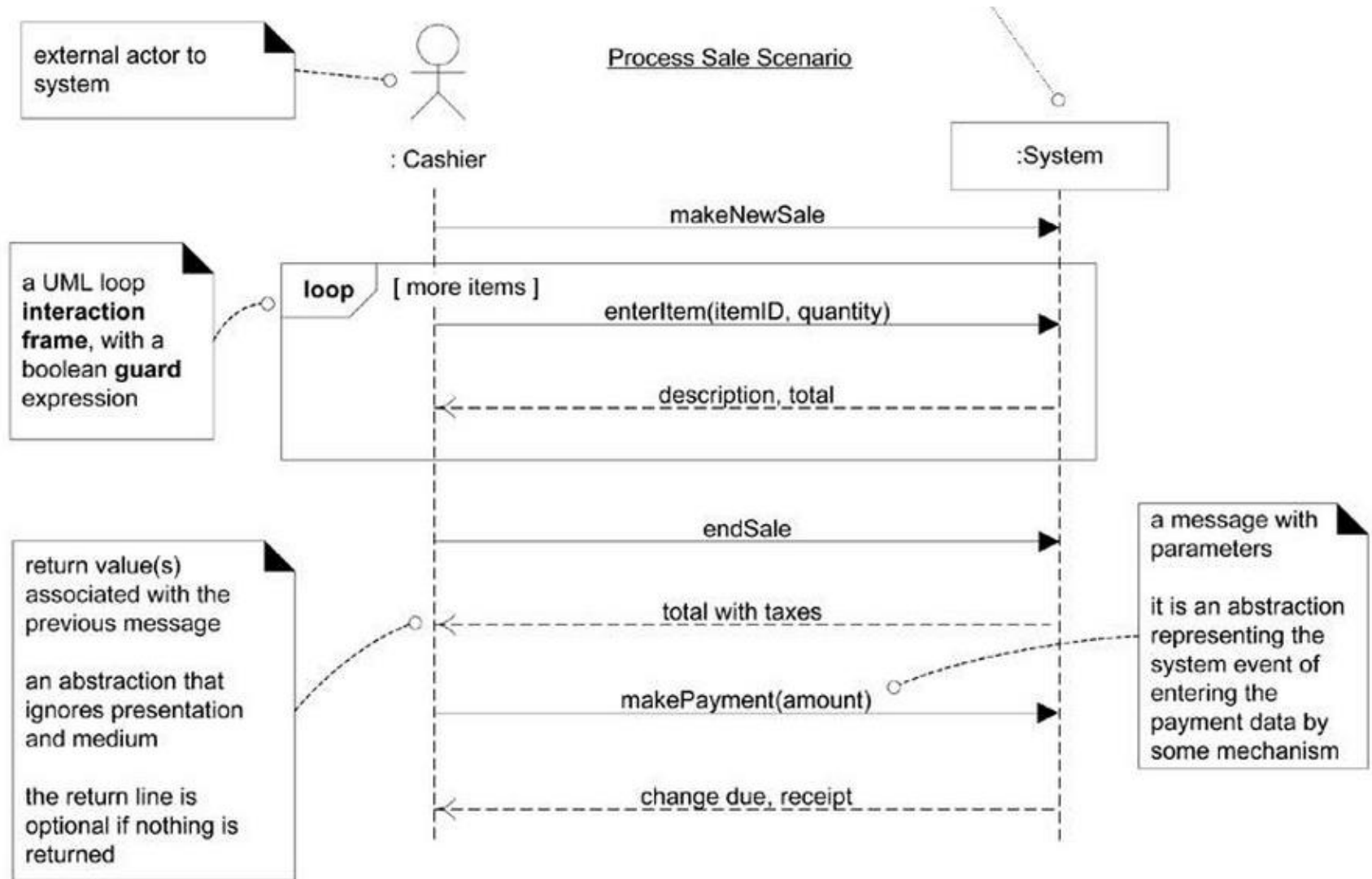
DDD – 领域模型



DDD – 隔离领域



DDD – 用户与系统的关系



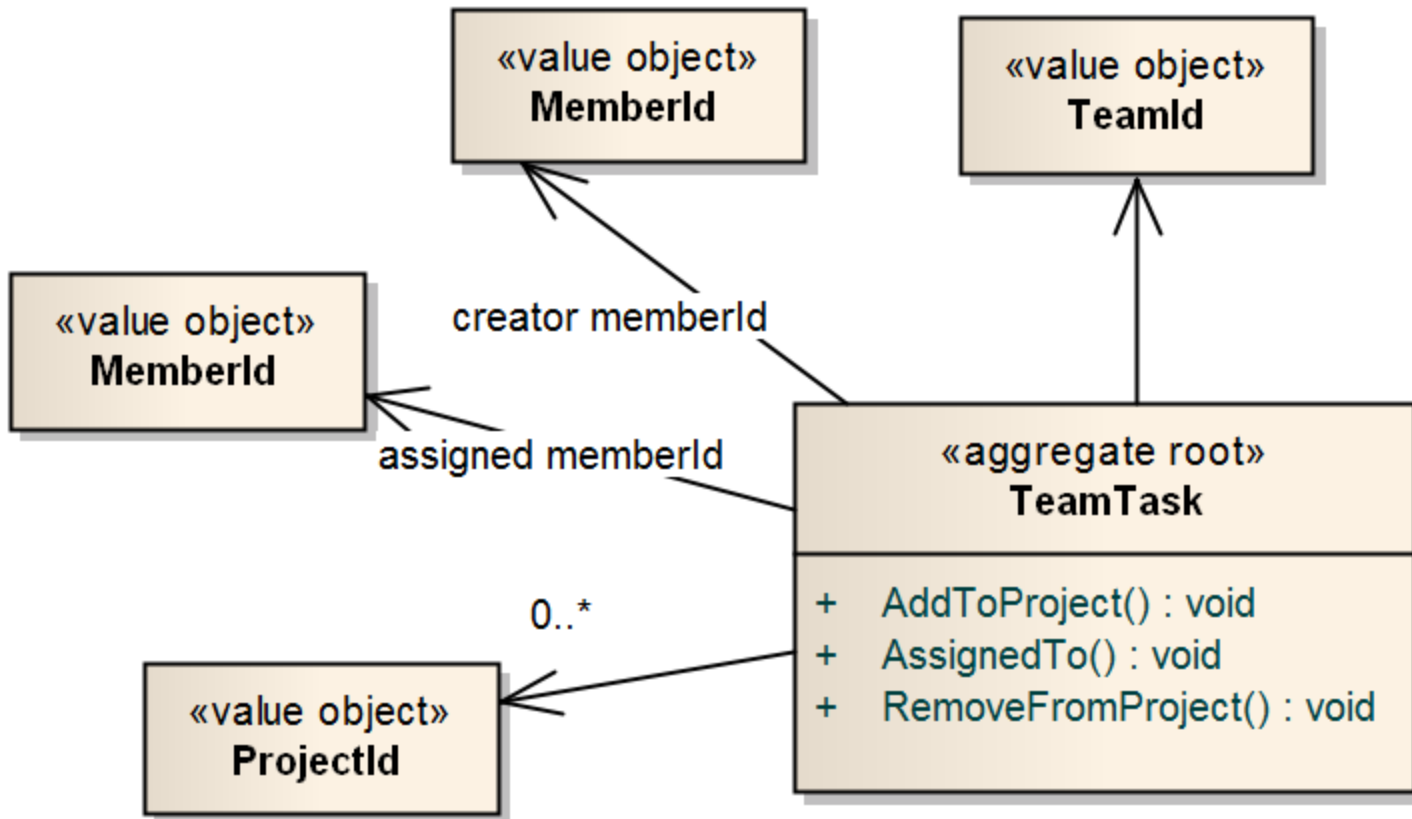
DDD - 领域建模时分析问题的角度

- 以用户为中心 != 以用户的需求为中心
- 领域模型是主观的，是人们对领域知识、需求分析后的产物，目的是容纳需求，解决领域内的问题；比喻：只有空杯子才能装水
- 理解：用户、账号、角色的差别

DDD – 聚合设计原则

- 聚合 != 大对象，相反聚合应尽量设计的小
- 聚合是用来封装真正的不变性，而不是简单的将对象组合在一起
- 聚合之间的关联通过**ID**，而不是对象引用
- 聚合内强一致性，聚合之间最终一致性
- 信息专家模式：将职责分配给拥有执行该职责所需信息的对象

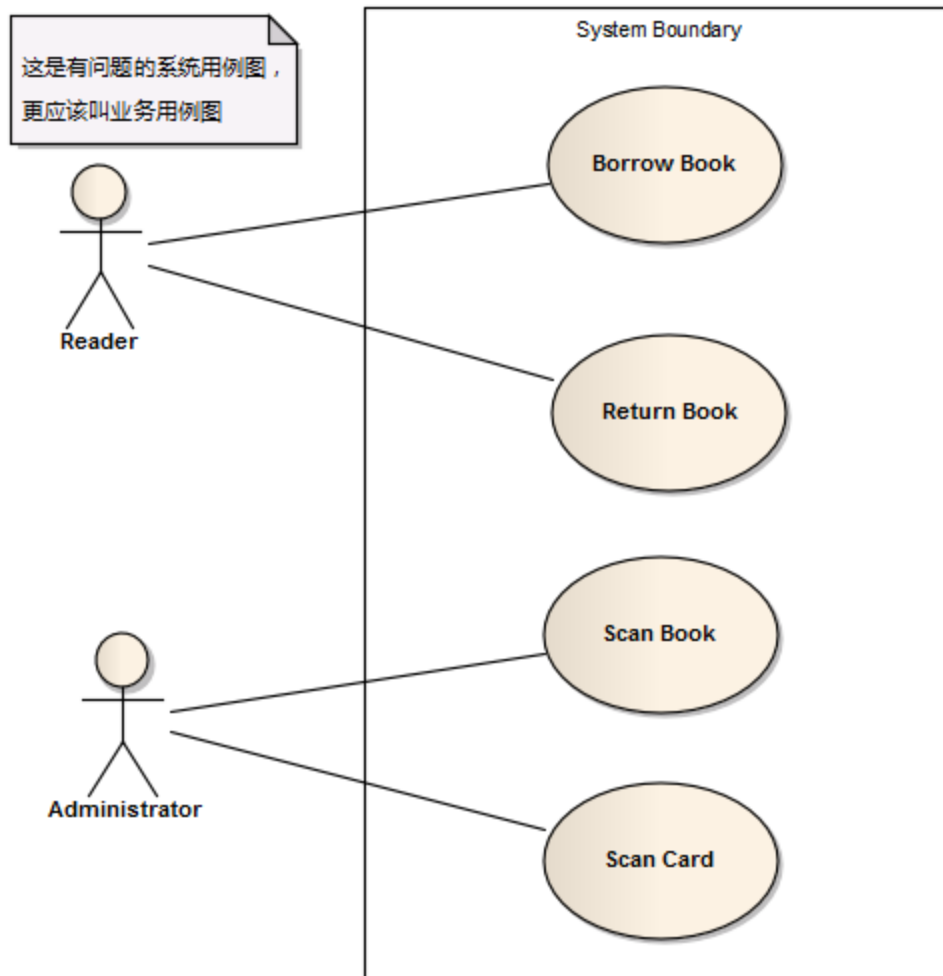
DDD – 聚合之间ID引用例子



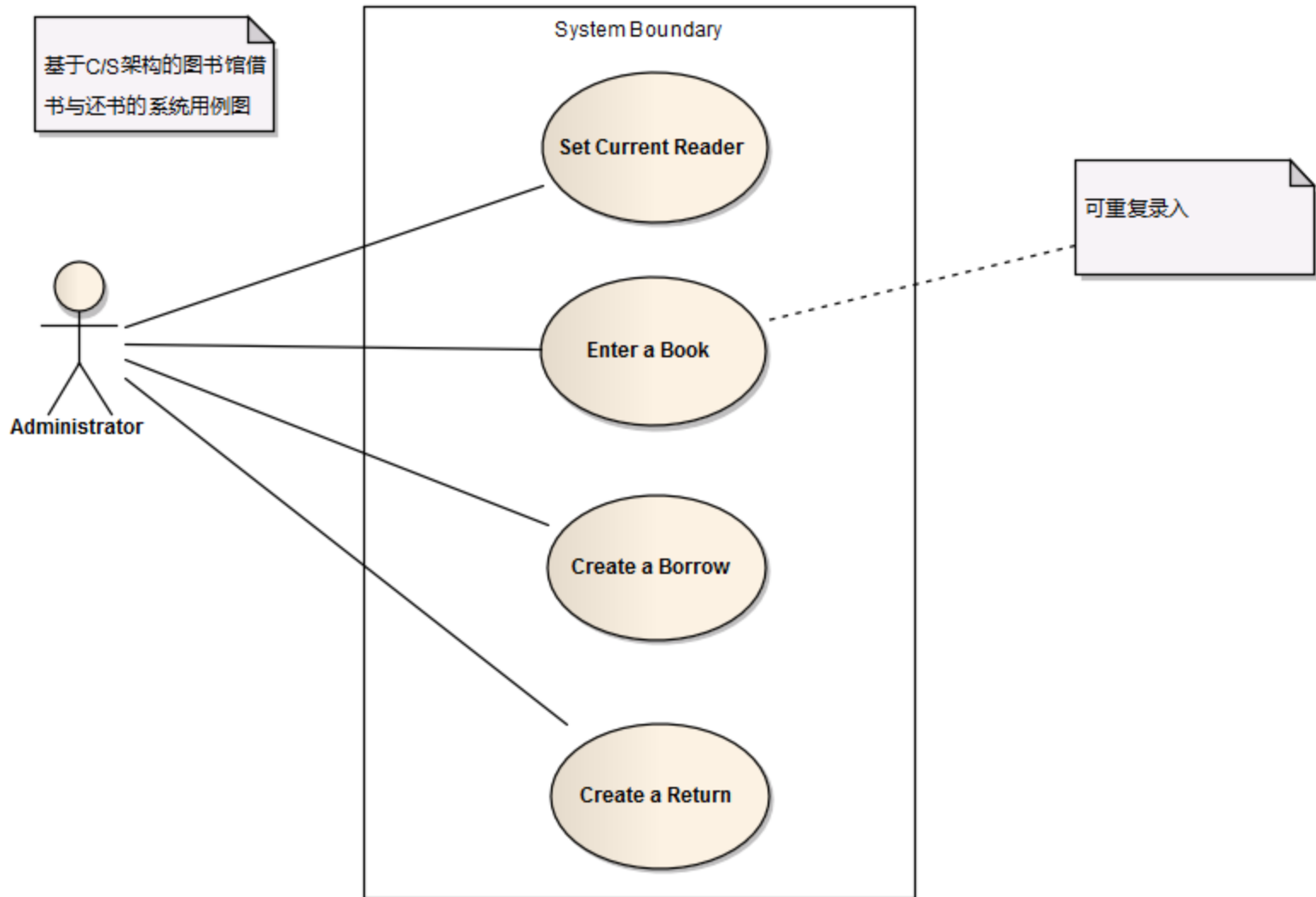
DDD - 领域建模步骤参考

- 先从需求中考虑一些**业务场景**，和领域专家交谈场景的过程，从中识别出一些明显的**领域概念**，以及它们的**关联**，关联可以暂时没有方向但需要有（1: 1, 1: N, M: N）这些关系；可以用文字精确的没有歧义的描述出每个领域概念的涵义以及包含的主要信息；
- 根据上面分析得到的领域概念建立一个**初步的领域模型**
- 分析主要的软件应用程序功能，识别出主要的应用层的类；这样有助于及早发现哪些是应用层的职责，哪些是领域层的职责；
- 进一步分析领域模型，识别出哪些是**实体**，哪些是**值对象**，哪些是**领域服务**；
- 分析关联，通过对业务的更深入分析以及各种**软件设计原则及性能方面的权衡**，明确关联的方向或者**去掉一些不需要的关联**；
- **找出聚合边界及聚合根**，这是一件很有难度的事情；因为你在分析的过程中往往会碰到很多模棱两可的难以清晰判断的选择问题，所以，需要我们平时一些分析经验的积累才能找出正确的聚合根；
- 为聚合根**分配仓储**，为每个聚合分配一个仓储，此时只要设计好仓储的接口即可；
- **走查场景**，确定我们设计的领域模型能够有效地解决业务需求；
- 考虑如何创建领域实体或值对象，是通过工厂还是直接通过构造函数；
- 停下来**重构模型**。寻找模型中觉得有些疑问或者是蹩脚的地方，比如思考一些对象应该通过关联导航得到还是应该从仓储获取？聚合设计的是否正确？考虑模型的性能怎样，等等；

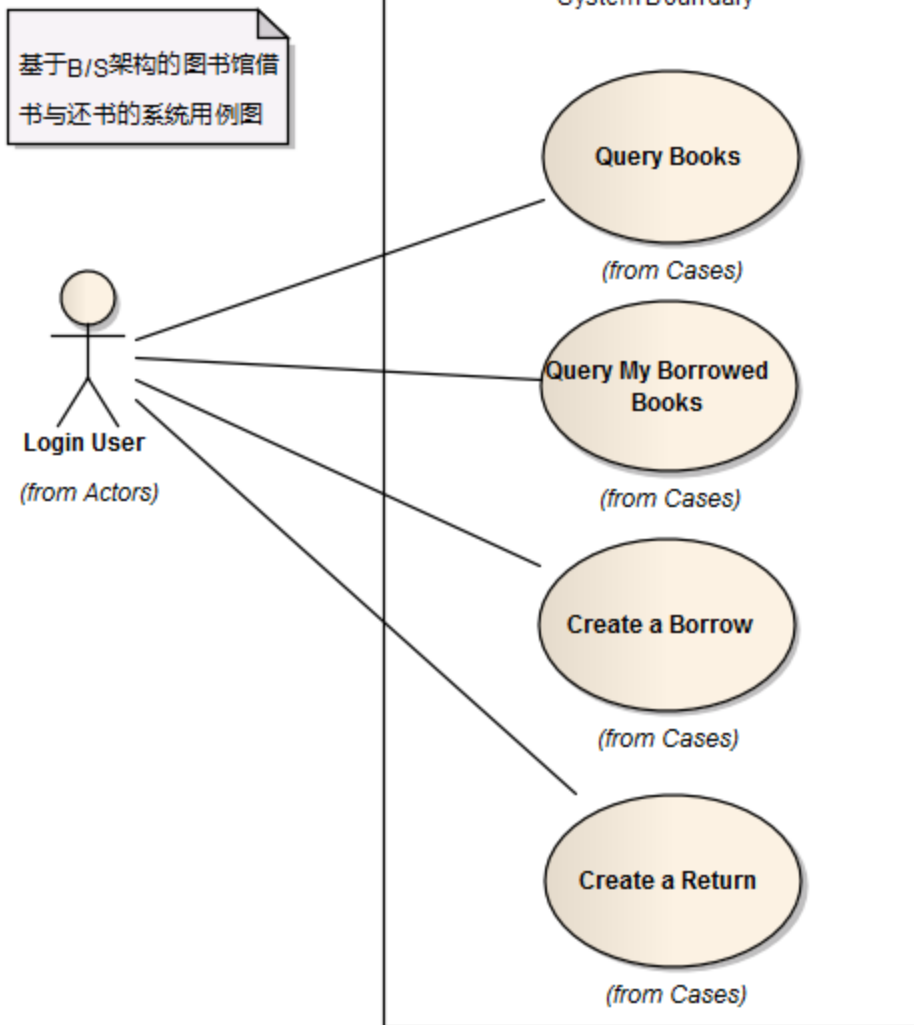
DDD案例 – 图书借还场景



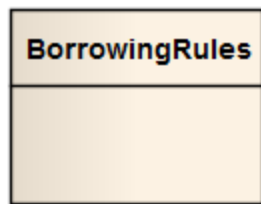
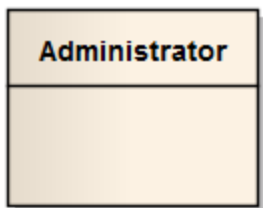
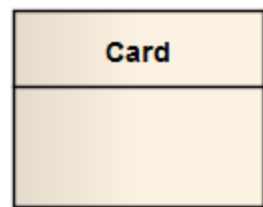
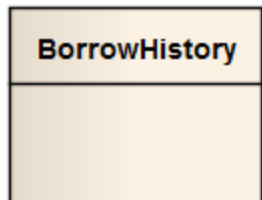
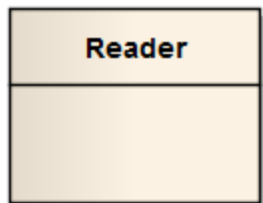
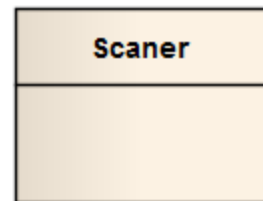
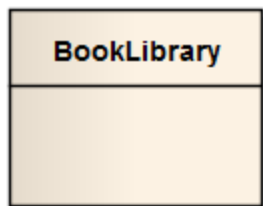
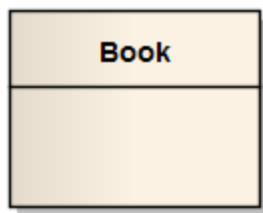
DDD案例 – 图书借还场景



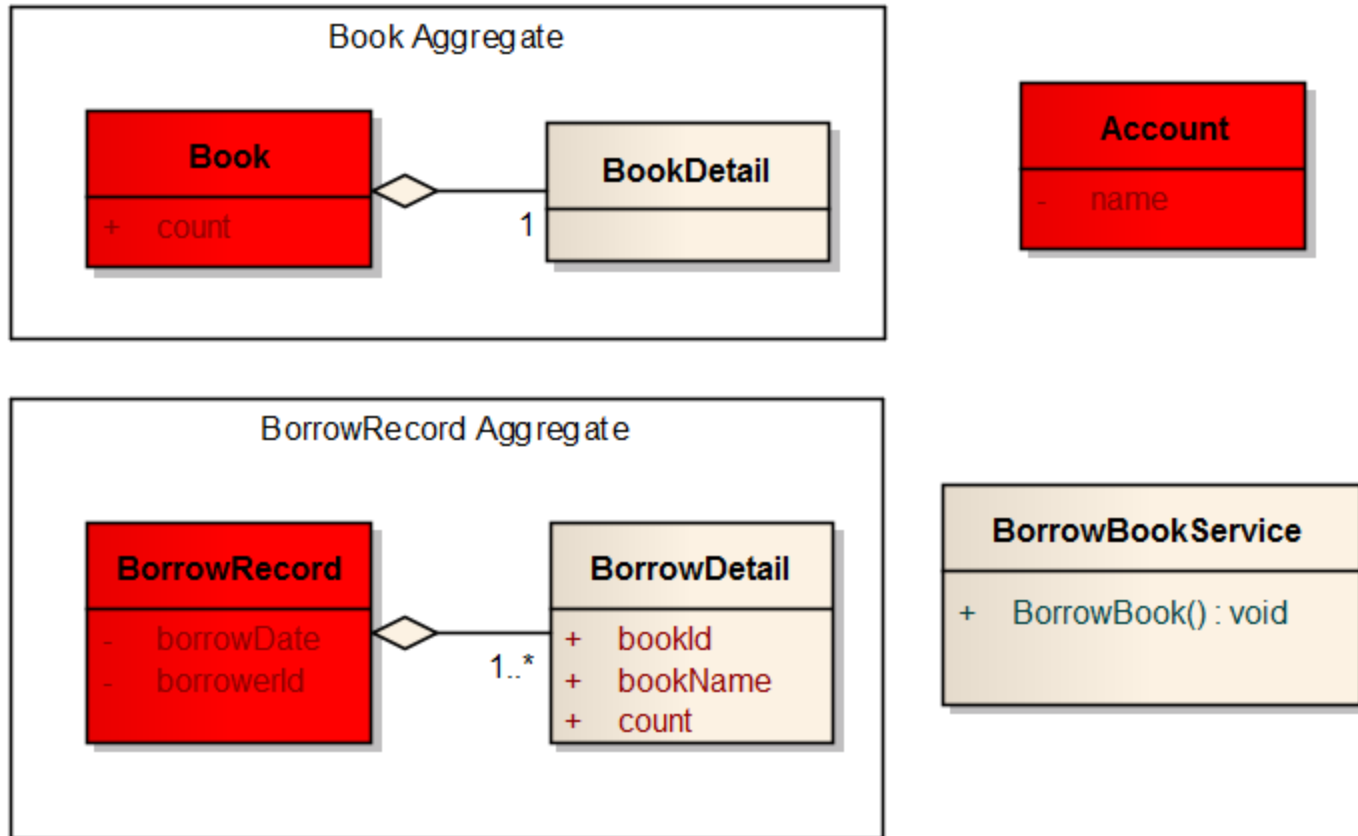
DDD案例 – 图书借还场景



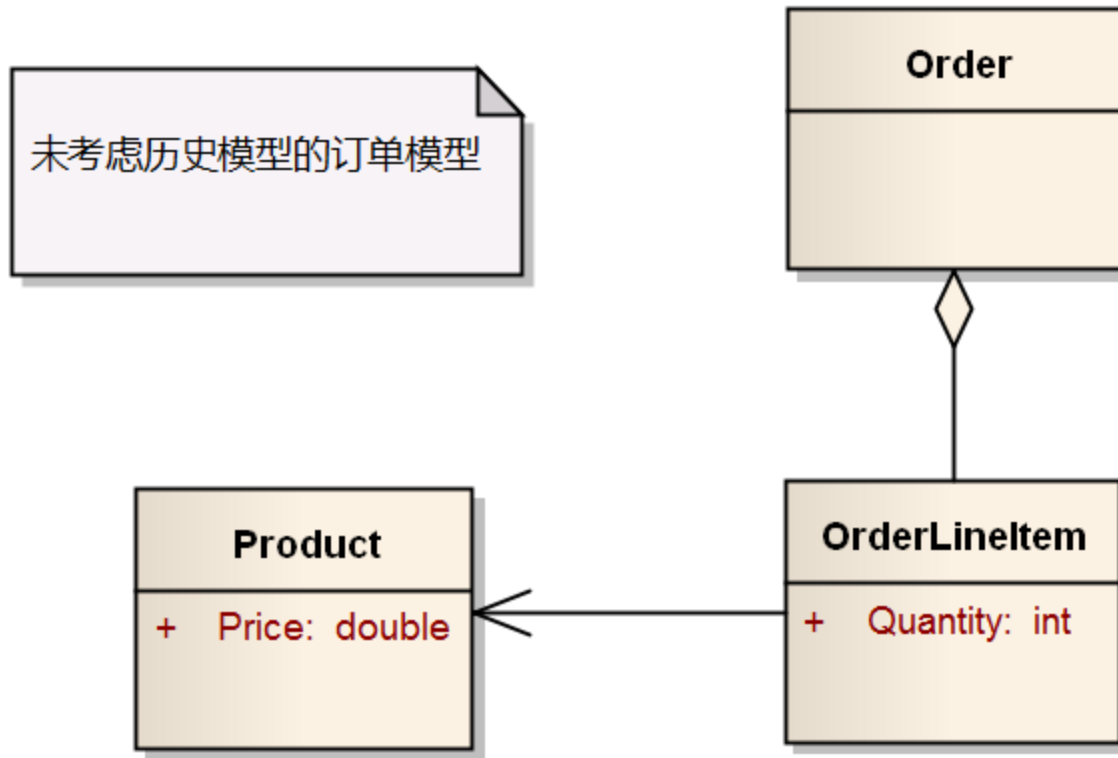
DDD案例 – 借书场景领域概念



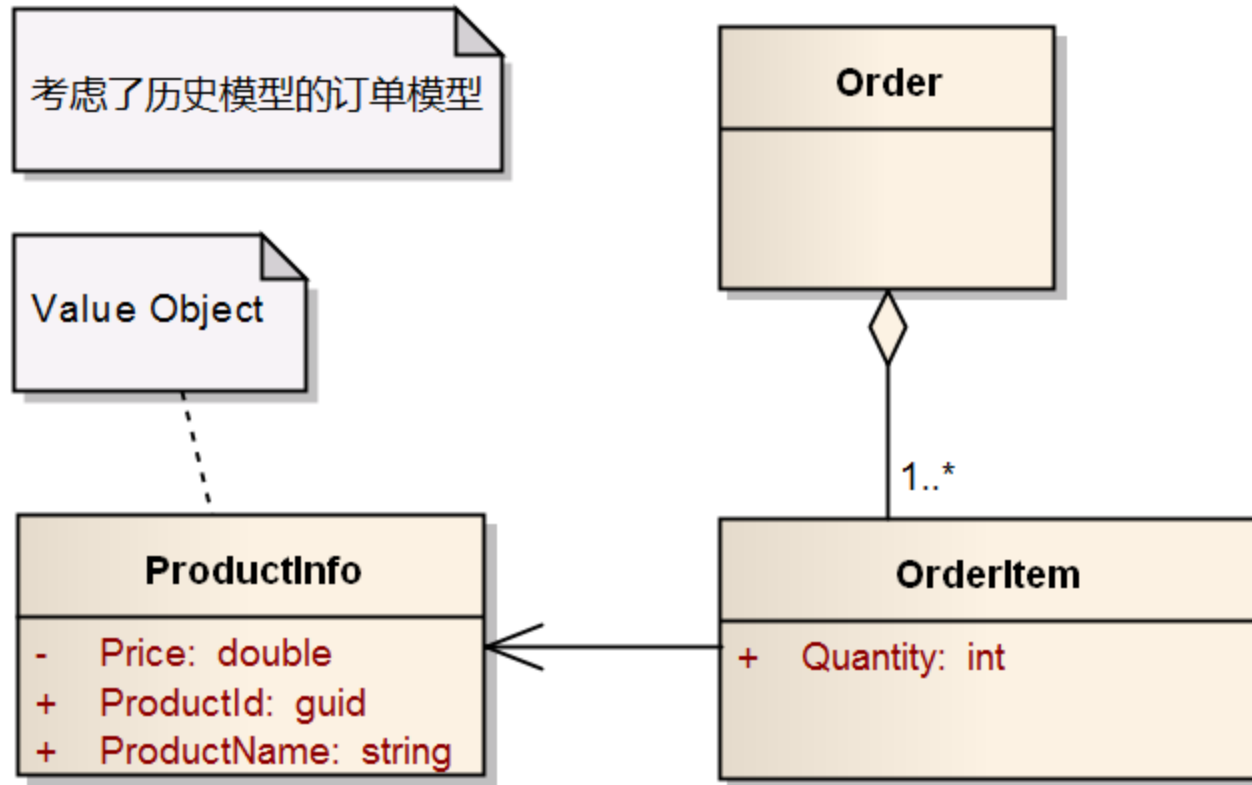
DDD案例 – 借书场景领域模型



DDD案例 – 订单模型

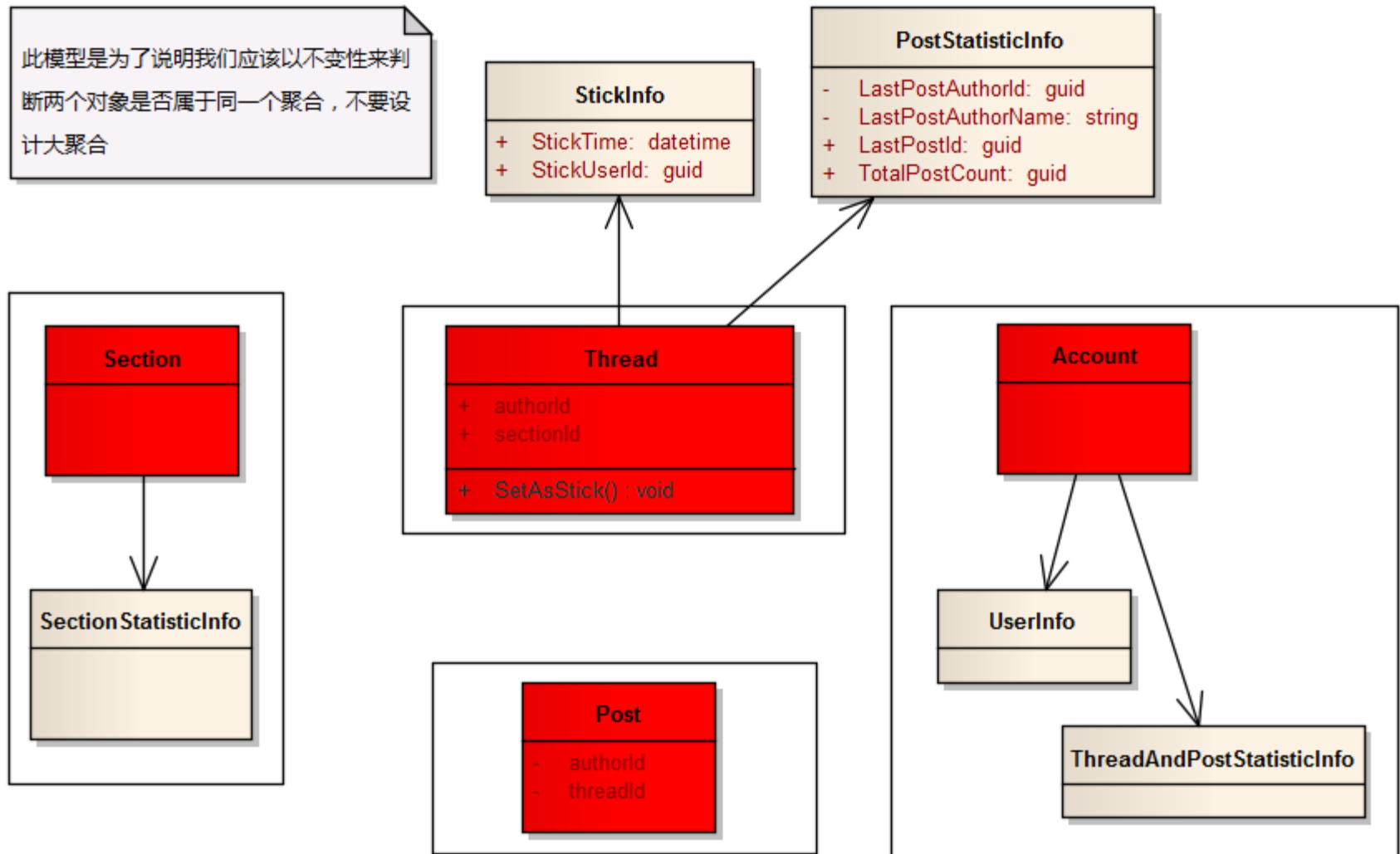


DDD案例 – 订单模型

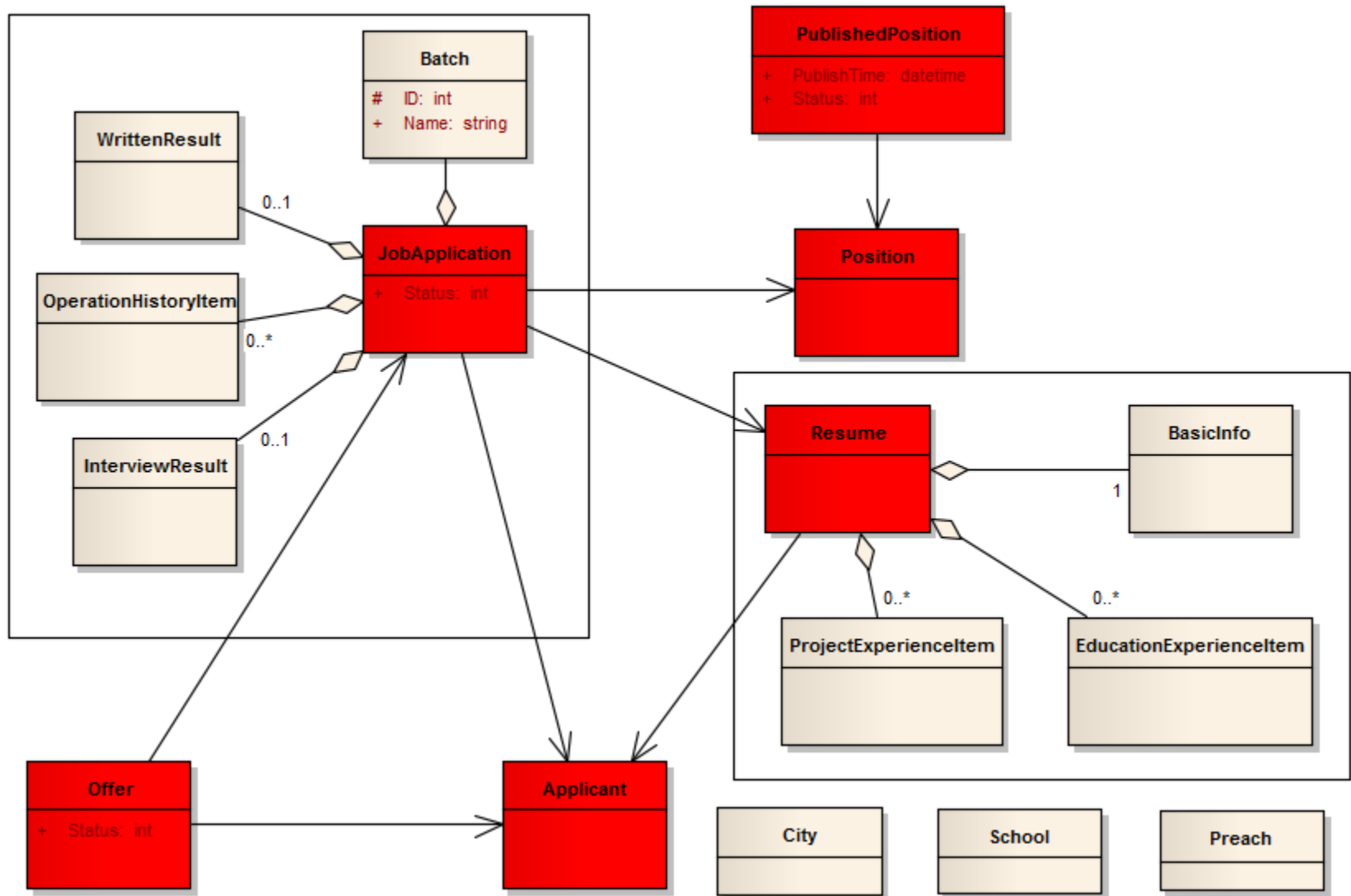


DDD案例 – 论坛模型

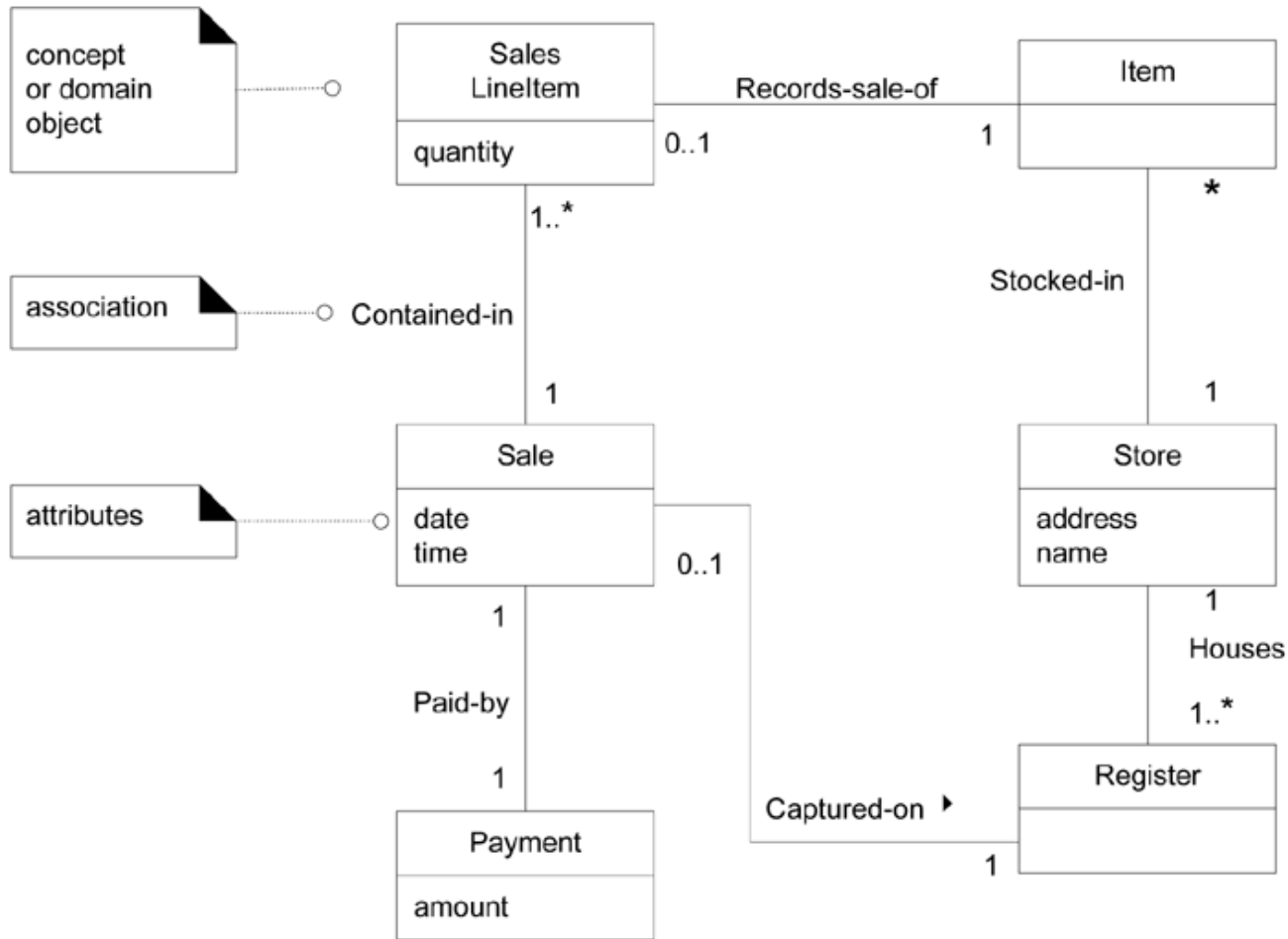
此模型是为了说明我们应该以不变性来判断两个对象是否属于同一个聚合，不要设计大聚合



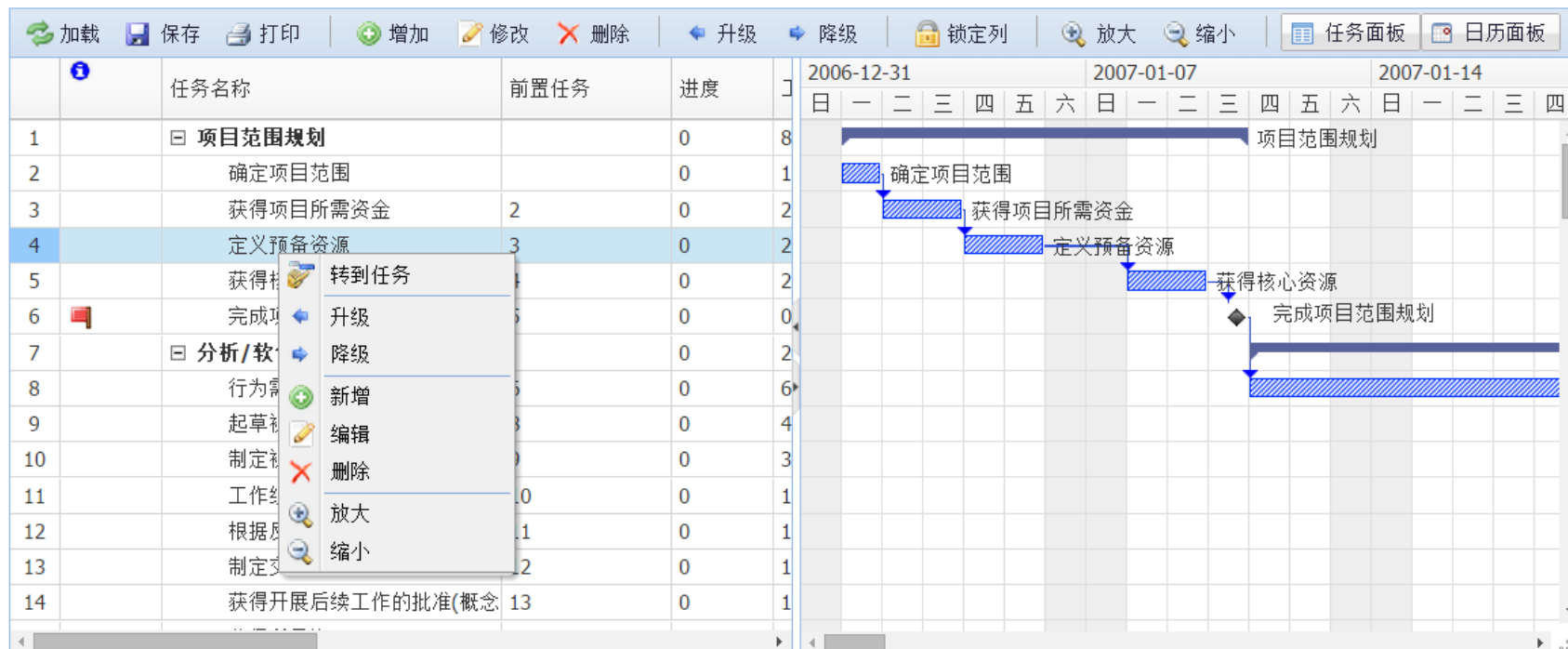
DDD案例 – 校招系统领域模型



DDD - POS机系统领域模型



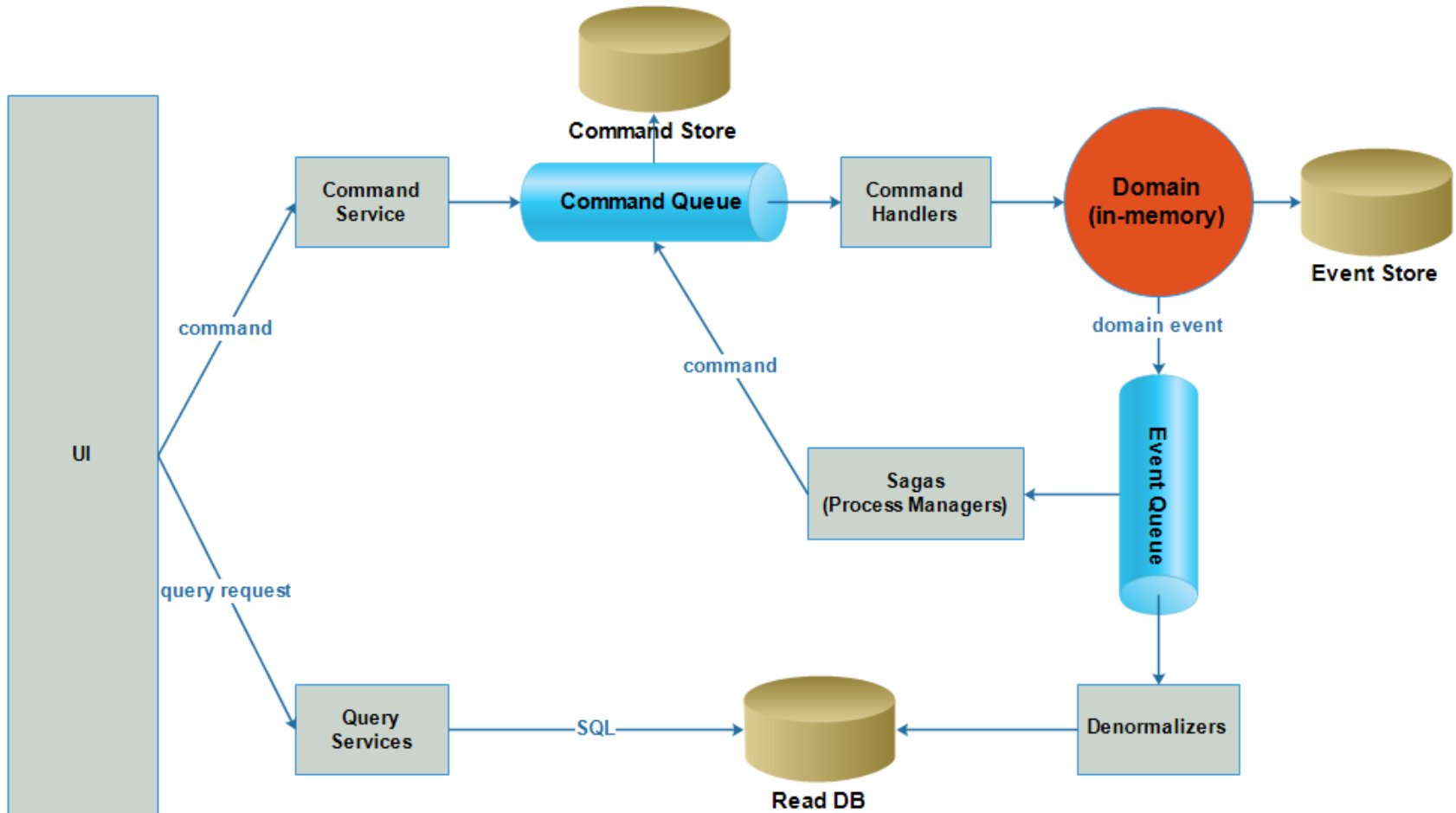
DDD案例 – 项目计划管理



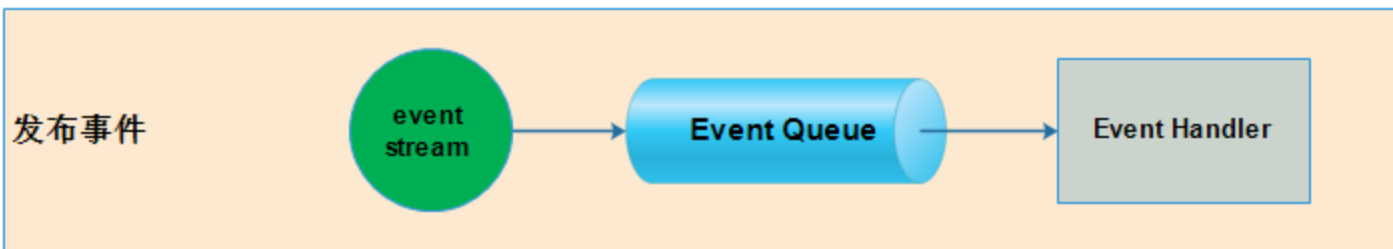
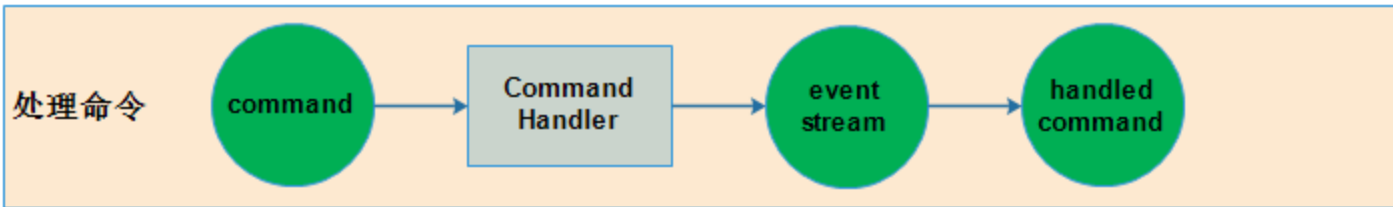
ENODE – 简介

- **ENode**是一个用于开发**DDD+CQRS+ES+EDA**架构风格的应用开发框架
- 领域模型常驻内存, (in-memory)
- 分布式、高性能、幂等支持、可扩展
- 架构层面尽量消除并发
- 面向最终一致性

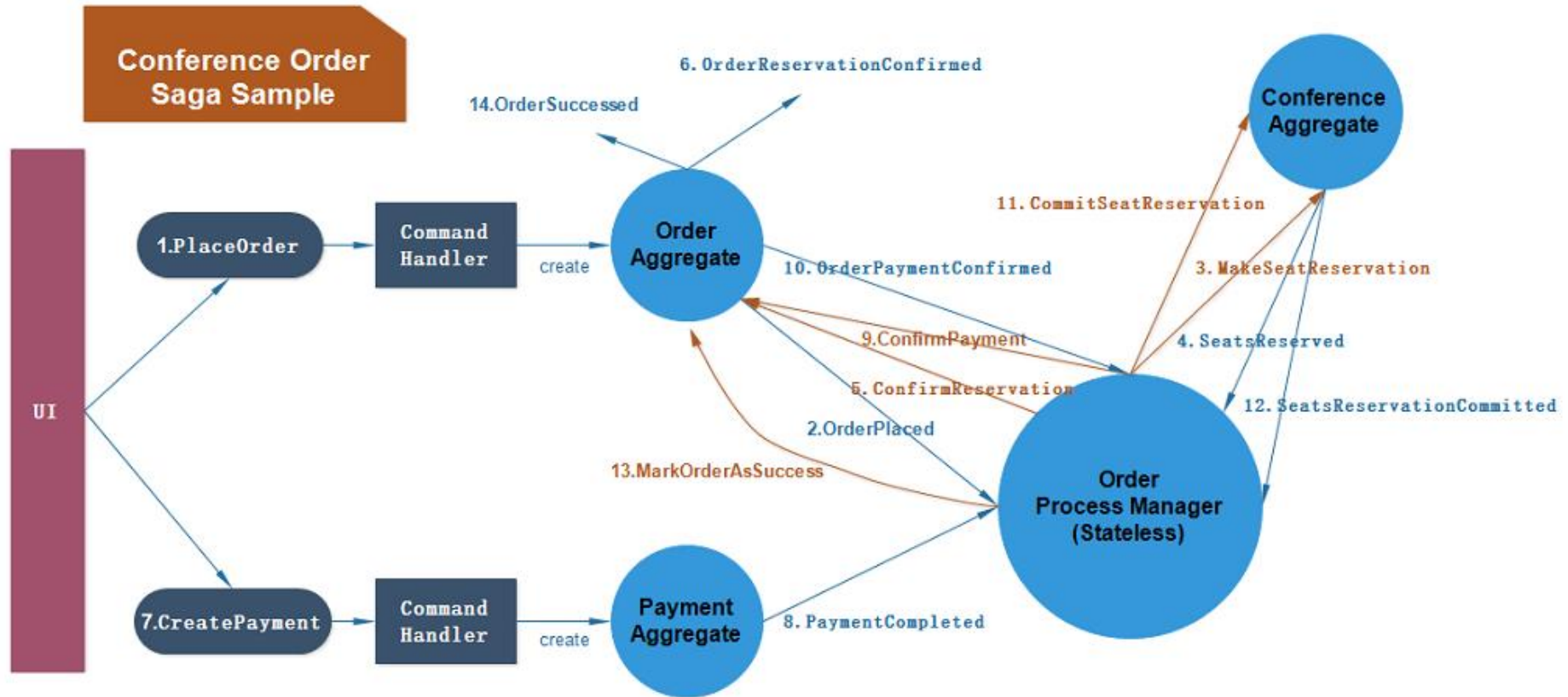
ENODE – 架构



ENODE – Command处理流程



ENODE – Conference案例



Thanks!
