CHINA

DDD的庖丁解牛之道

2018 中国·上海

厦门云雾科技 王立

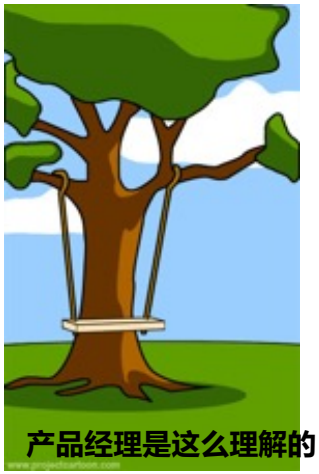
CHINA DEVOPS DAYS

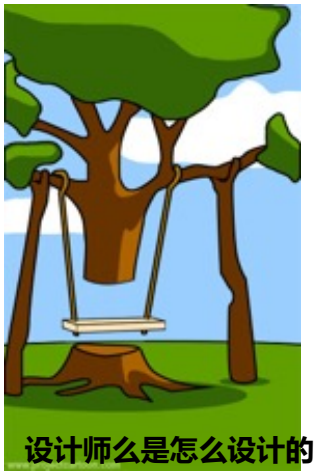
目录

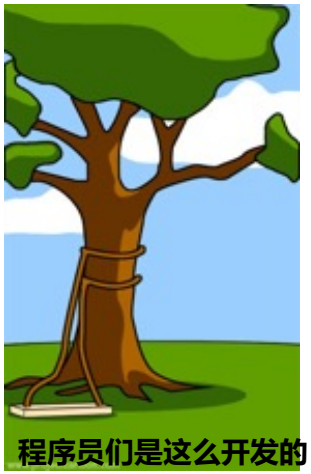
复杂性之源

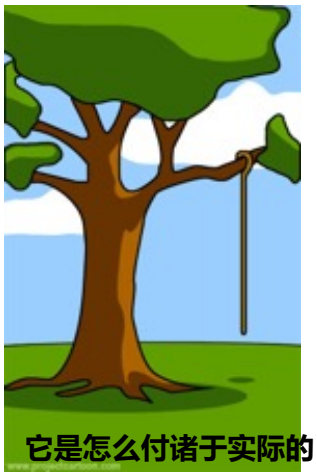
CHINA DEVOPS DAYS

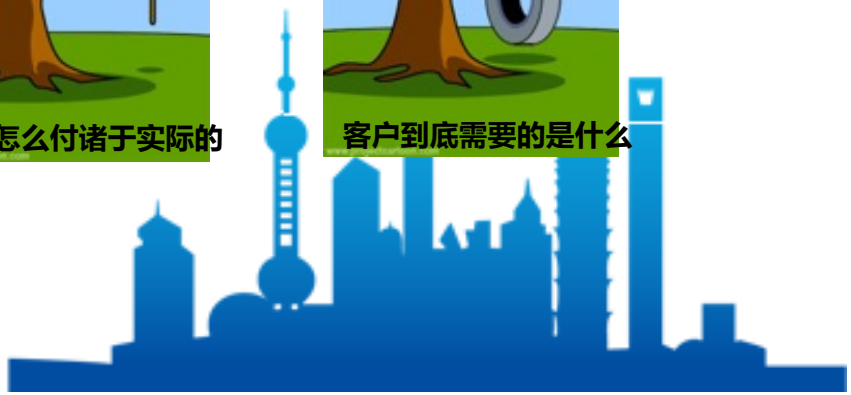
复杂性的源头——需求理解





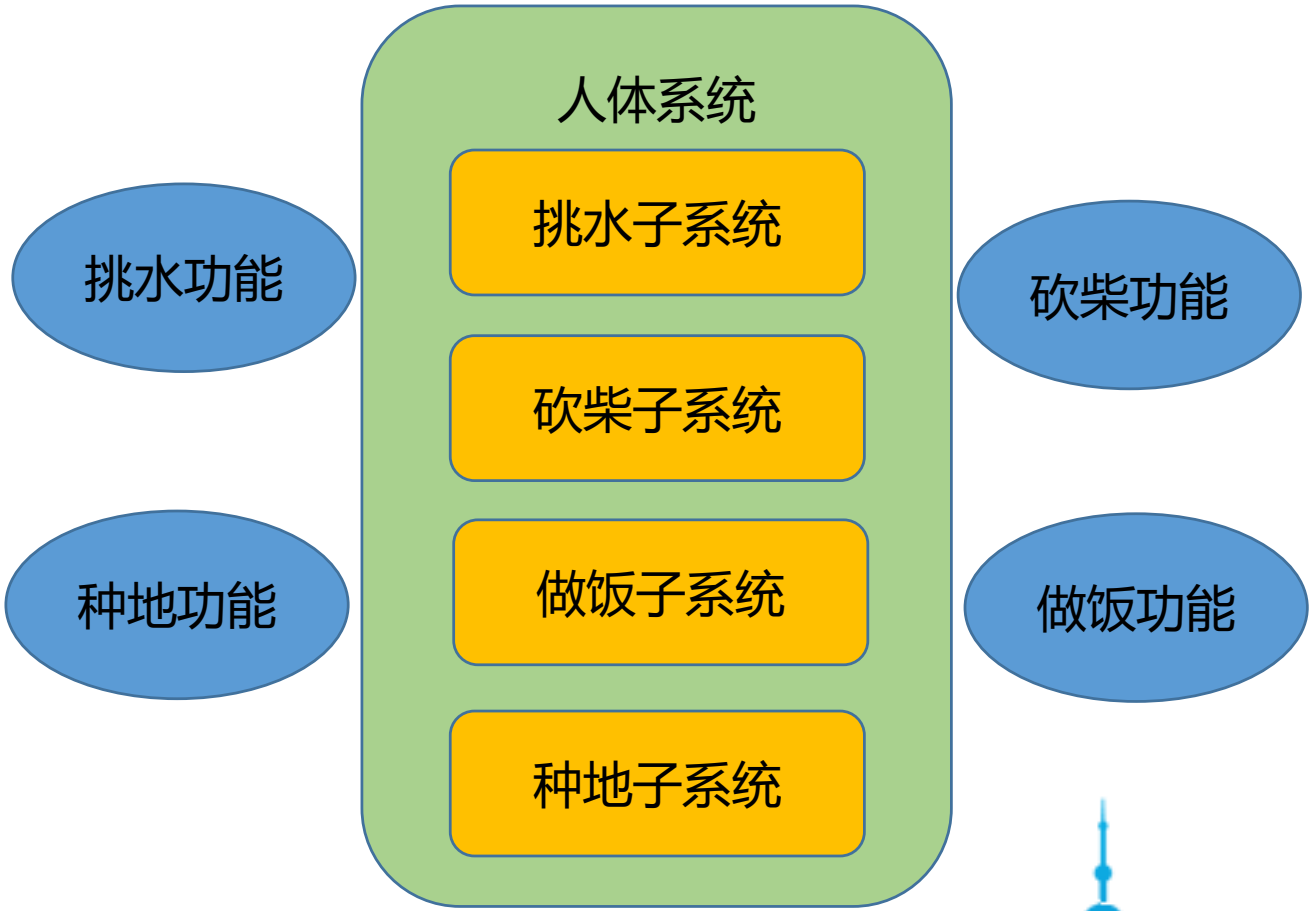






CHINA DEVOPS DAYS

复杂性的源头——系统分析



错误的抽象

CHINA DEVOPS DAYS

复杂性之源——设计实现

解决之道

CHINA DEVOPS DAYS

改善需求沟通

经过过滤和翻译转换的概念，存在遗漏、曲解并且难以察觉，直到上线才暴露

开发　　产品经理

一切不以统一语言为基础的DDD都是伪DDD

CHINA DEVOPS DAYS

改善需求沟通

统一语言实例

需求表达：

当用户验证登录以后，显示一个空的窗口 ✗

当玩家登录以后，显示一个空的游戏控制面板 ✓

CHINA DEVOPS DAYS

改善需求沟通

统一语言实例

命名风格

```
Integer i = new Integer();
String char1 = new String();
public class GameDAO() { }
catch (Exception e)
```

```
String realMeaningOfMyString = new String();

public class ScoreDataLoader() { }

catch (Exception NotLoggedInException)
```

改善需求沟通

统一语言实例

代码风格

```
package tictactoe.client.userInterface;

/**
* Add the string O or X to a cell in the grid.
*/
public class ShowCellGrid{

public static void displayUser (Grid grid, Cell cell) {

        if (!Initialization.flag
        && Initialization.gameStatus.getSequence() == null
        && isEmpty(grid, cell)) {

        Initialization.flag = true;

        String mk= showString(Initialization.gameStatus
        .getCurrentUser().getUserString());

        grid.setHTML(cell.getRowIndex(), cell.getCellIndex(), mk);

        Initialization.gameStatus.getStatus()[cell.getRowIndex()][cell
        .getCellIndex()] = Initialization.gameStatus
        .getCurrentUser();

        GameEnd.checkEnd(Initialization.gameStatus,
        cell.getRowIndex(), cell.getCellIndex());
        }

(...)
}
```

```
package tictactoe.client.userInterface;

/**
* Performs a move in the game.
*/
public class PlayerMove {

/**
* When the player clicks in a cell, the game draws an O or a X on the
* game grid depending on which player's turn it is.
*/
public static void makeMove (GameGrid gameGrid, Cell cell) {

        if (!GameInitialization.waitingMoveFlag
        && GameInitialization.currentGameStatus.getSequenceWinner() ==
        null && isCellEmpty(gameGrid, cell)) {

        GameInitialization.waitingMoveFlag = true;

        String marker =
        showPlayerIcon(GameInitialization.currentGameStatus
        .getCurrentPlayer().getPlayerIcon());

        gameGrid.setHTML(cell.getRowIndex(), cell.getCellIndex(), marker);

        GameInitialization.currentGameStatus.getGameMoves()[cell.getRow
        Index()][cell.getCellIndex()] =
        GameInitialization.currentGameStatus.getCurrentPlayer();

        CheckWinner.checkForWinner(GameInitialization.currentGameStatu
        s, cell.getRowIndex(), cell.getCellIndex());
        }

(...)
}
```

改善需求沟通

确保团队在所有交流活动和代码中坚持使用统一语言

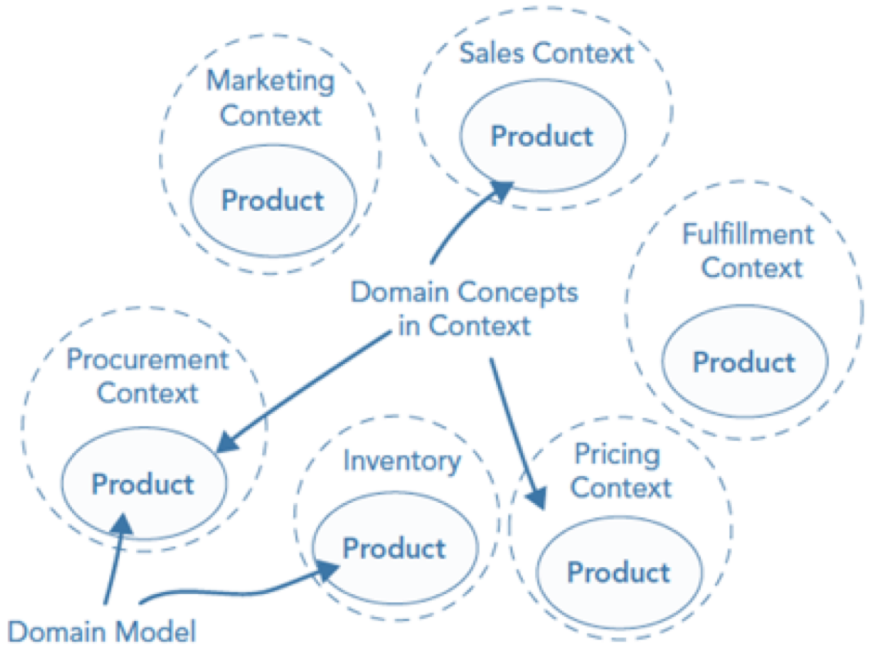
- 在限界上下文中，保持语言的一致性（如口语、图形（如UML图等）、文 字、代码等）。
- 统一语言贯穿于项目的各个环节
  - User Stories
  - Project Meetings
  - Team Emails
  - Instant Messages
  - Schedule Plan
  - Software Documents

改善需求沟通

划分限界上下文

关注点分离

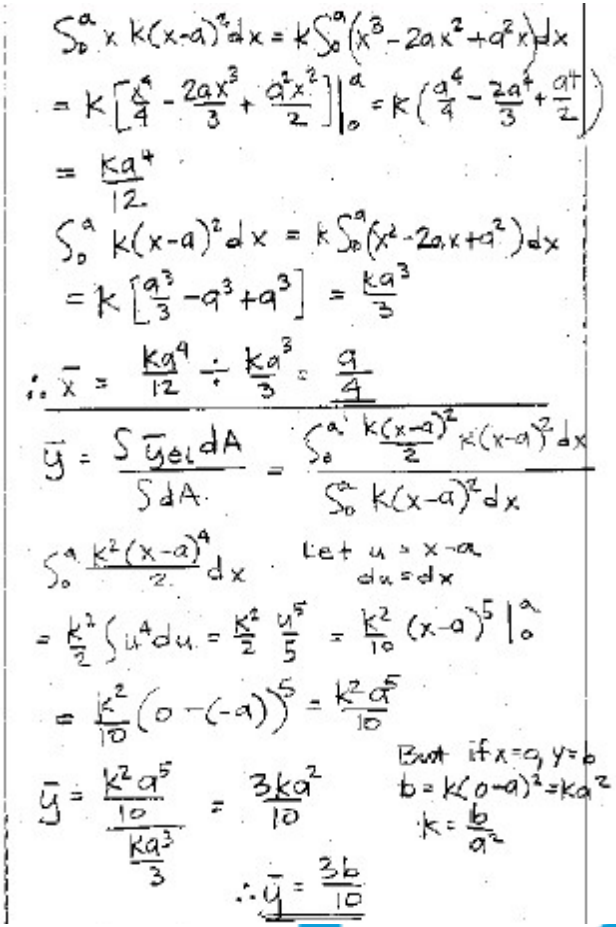
- 从聚合出发： 一个聚合可能是最小颗粒度的限界上下文
- 考虑聚合之间的业务相关性，合并高业务相关性的聚合
- 考虑限界上下文的概念完整性 合并业务价值不明确的限界上下文



如果在不同限界上下文中看到了完全相同的实体，往往意味着模型是错误的。

强化系统分析

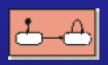
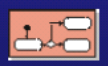
没有统一的数学表达式，很难想象数学领域的发展

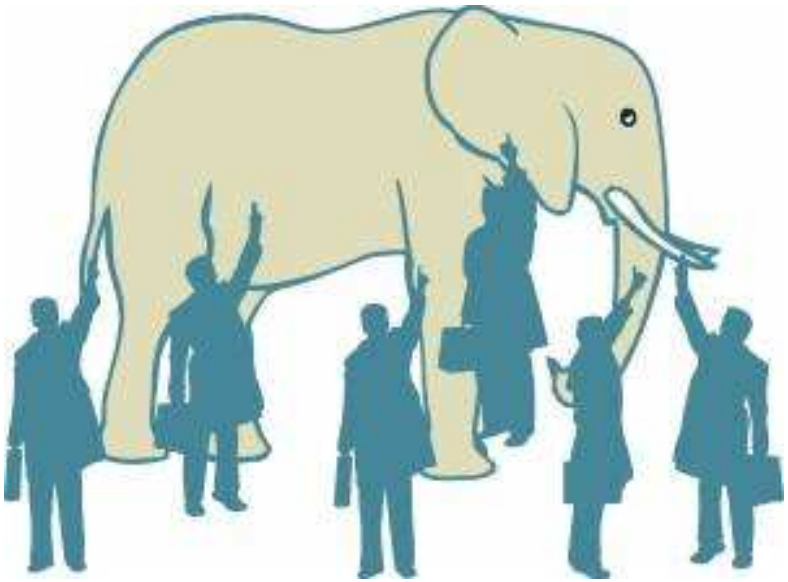


CHINA DEVOPS DAYS

强化系统分析

用UML透视业务的多维度立体模型

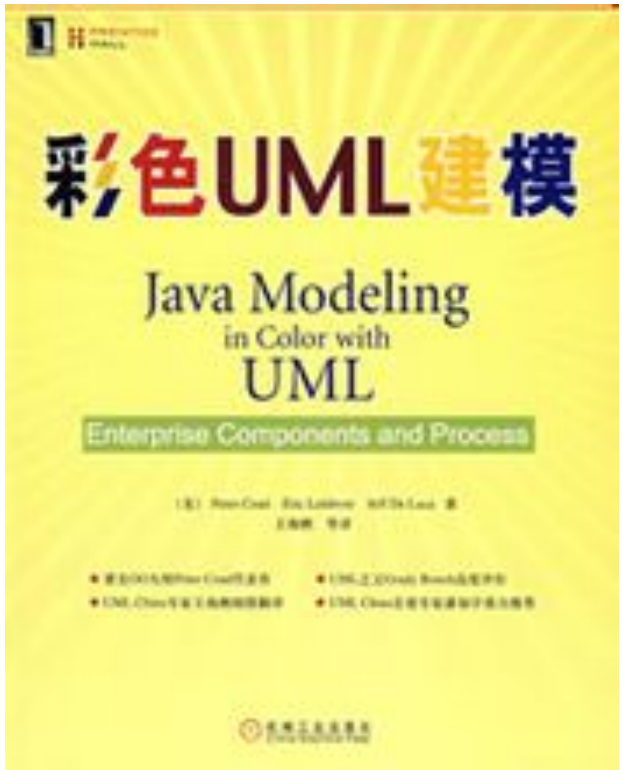
- 顺序图 
  - 最常用。以时间为中心，描述对象间的交互，焦点是消息的时间顺序。
- 协作图 
  - 焦点：收发消息的对象结构组织。利用工具可以由顺序图生成。两者合称“交互图”。
- 状态图 
  - 对一个类的生命循环建模，对复杂的动态行为有用。 
- 活动图
  - 活动到活动之间的控制流

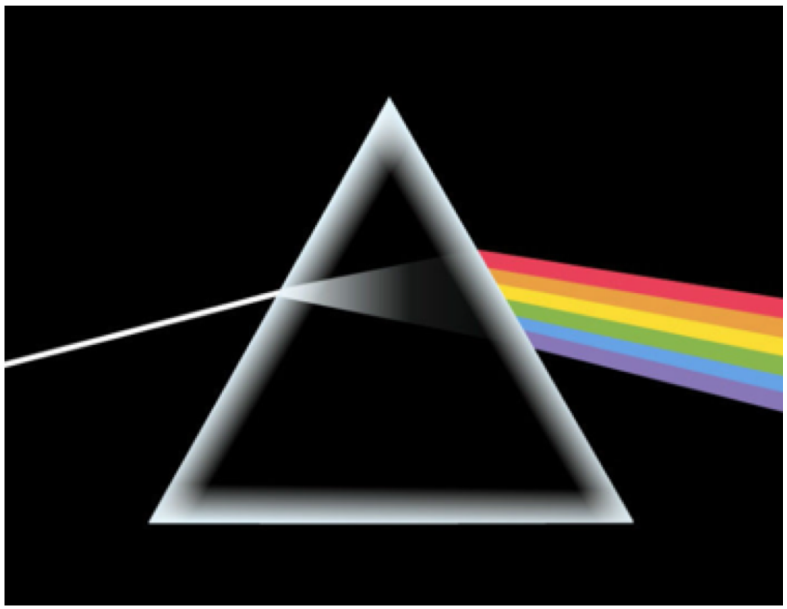


CHINA DEVOPS DAYS

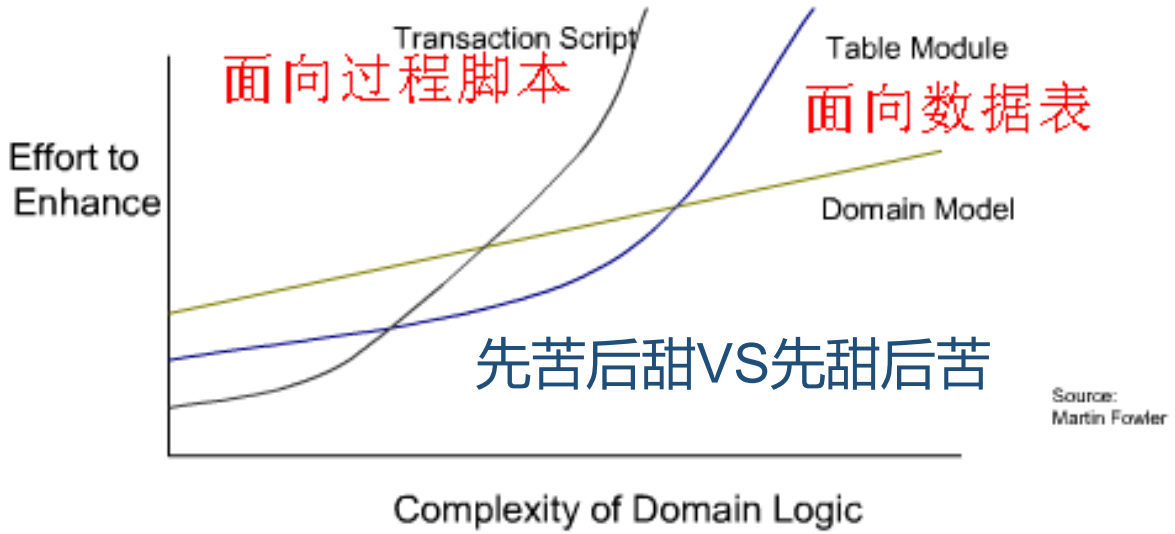
强化系统分析

用四色原型进一步分解业务模型





变革实现方法



choose the right style for the right application/service

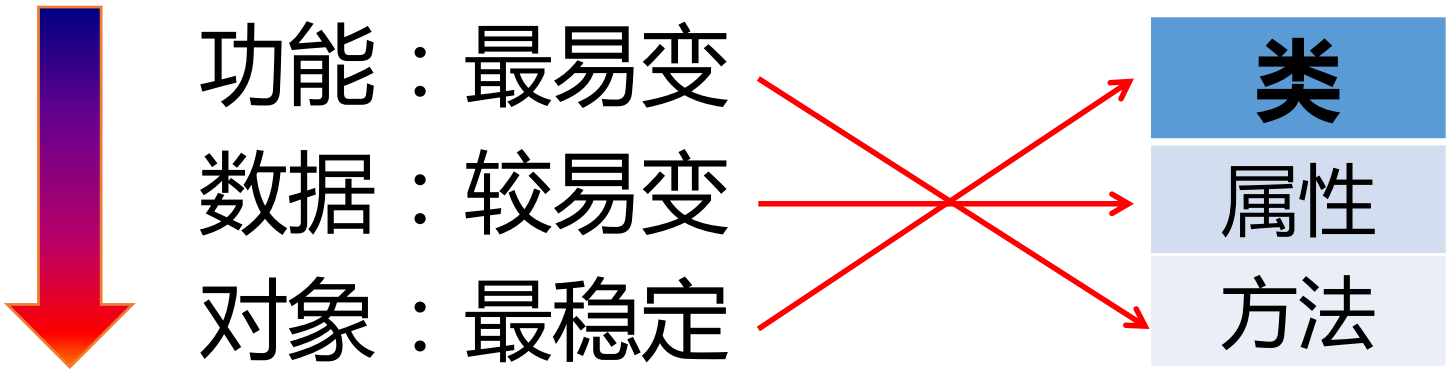
传统架构将业务简单切割成数据和行为，造成系统对业务问题的首肢分离。一旦需求变更，影响的范围边界不清晰，使得维护成本很高。

CHINA DEVOPS DAYS

变革实现方法

上升到对象层面是必然

需求变化时：



CHINA DEVOPS DAYS

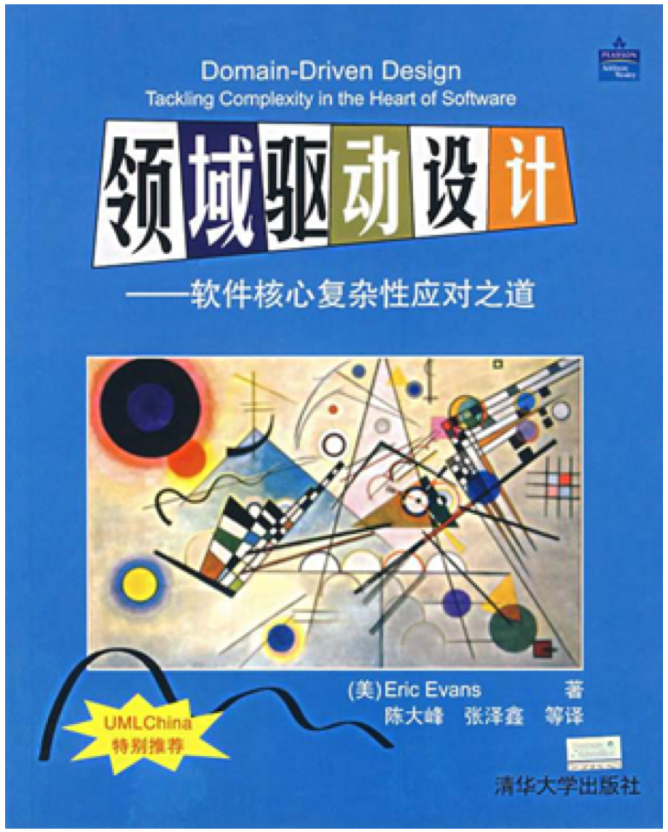
变革设计思路

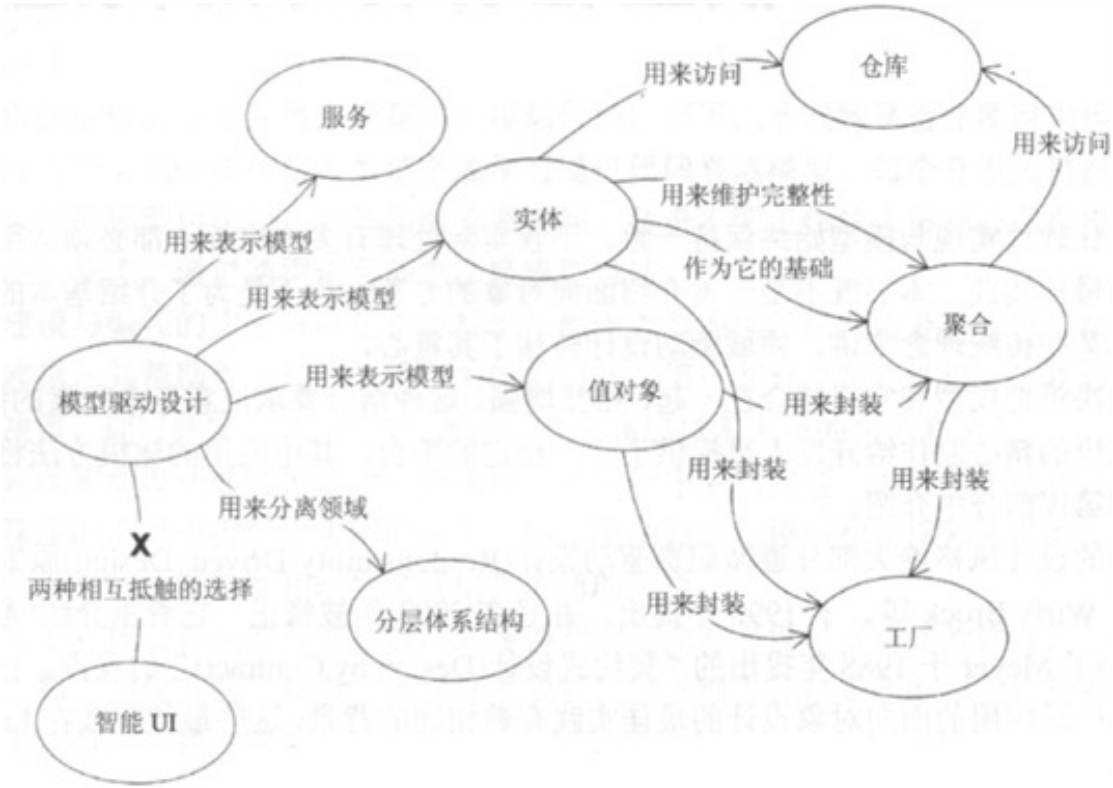
用较稳定的把不稳定的包起来

CHINA DEVOPS DAYS

变革实现方法

DDD内功心法 ——不是设计风格，也不是架构模型，它是一种思想方法

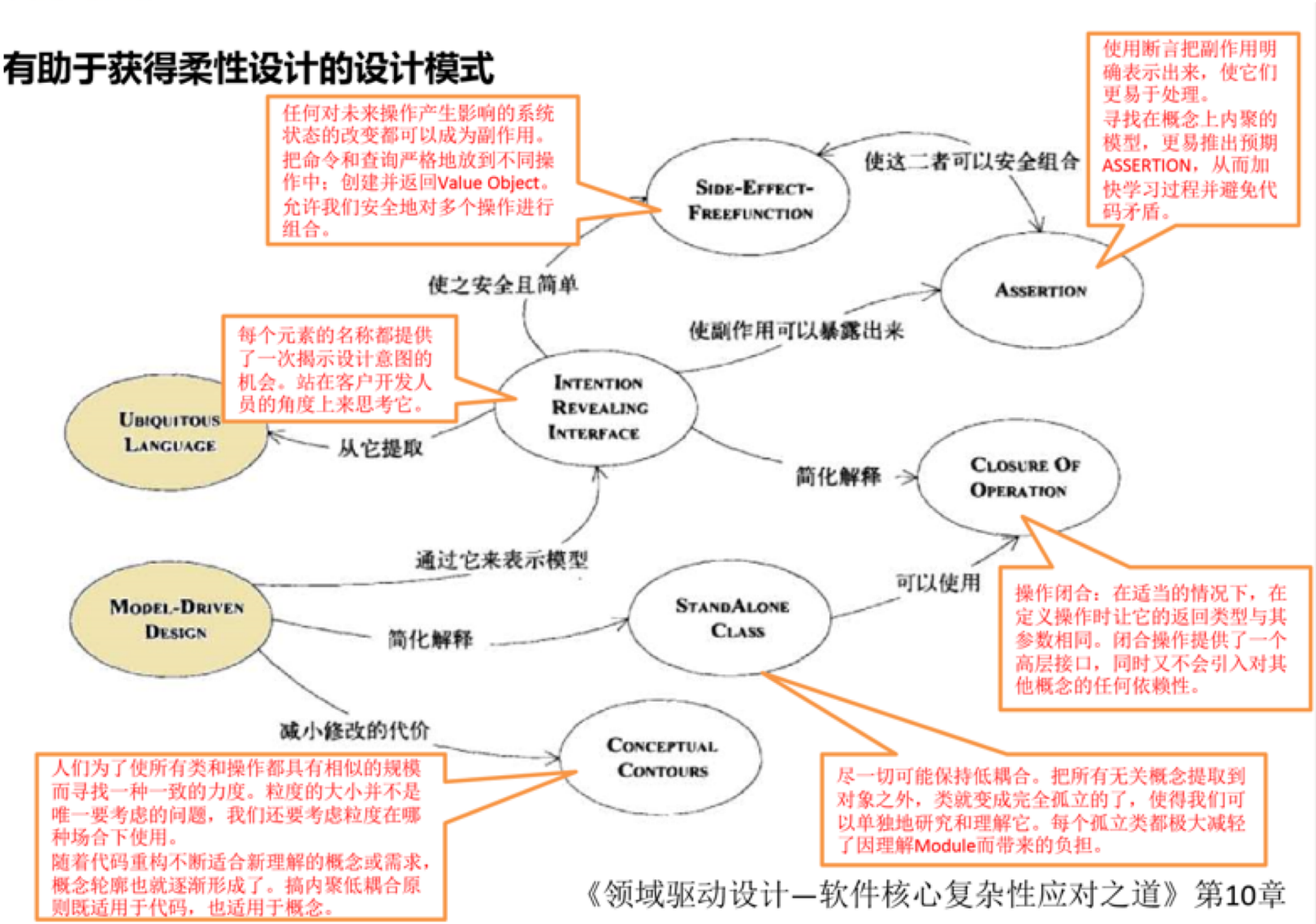




CHINA DEVOPS DAYS

变革实现方法

有助于获得柔性设计的设计模式



《领域驱动设计—软件核心复杂性应对之道》第10章

实践案例

CHINA DEVOPS DAYS

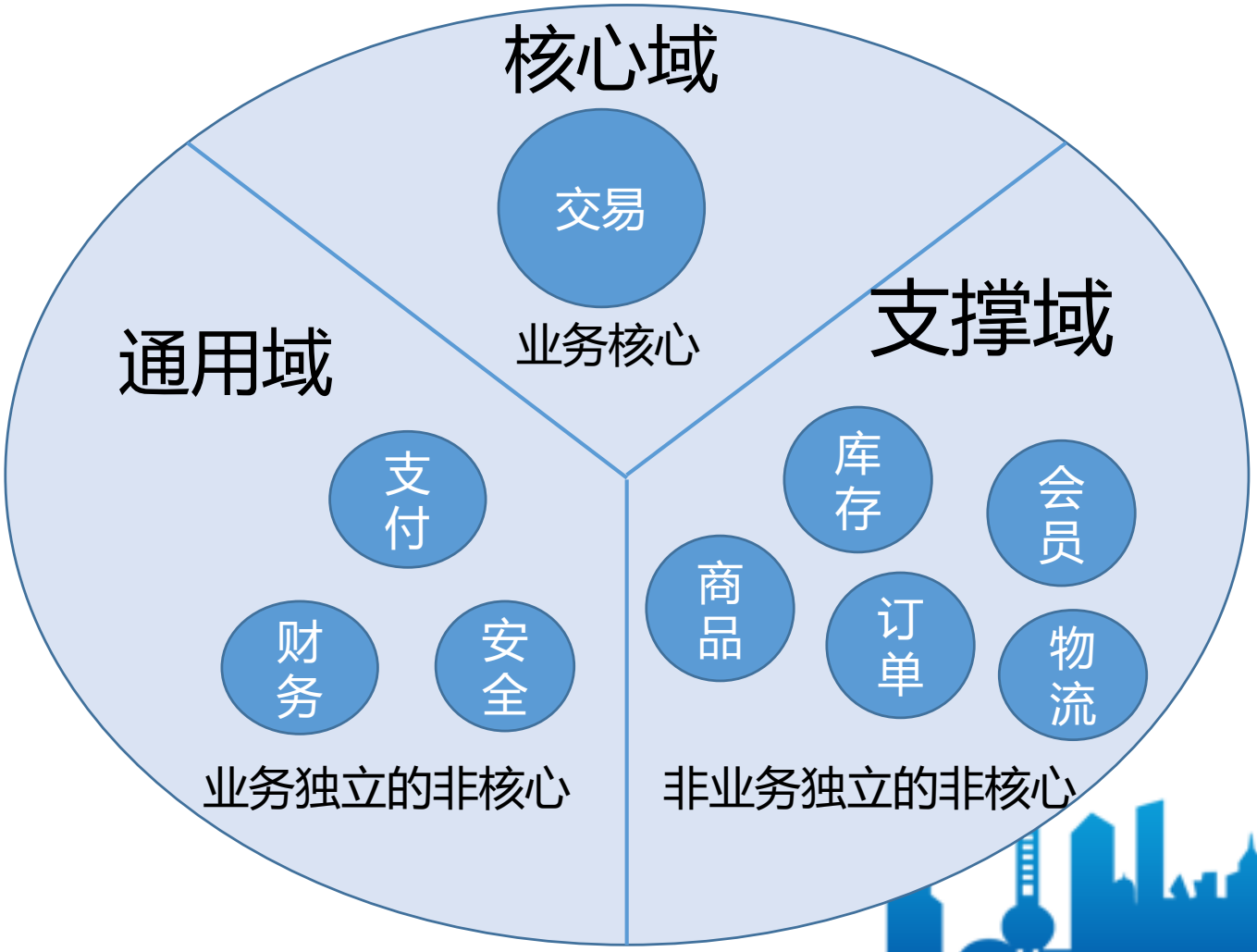
解牛分步策略

- 第一步：领域划分和限界上下文
- 第二步：在事件风暴中应用统一语言
- 第三步：用四色原型分析业务
- 第四步：从分析模型到领域驱动设计

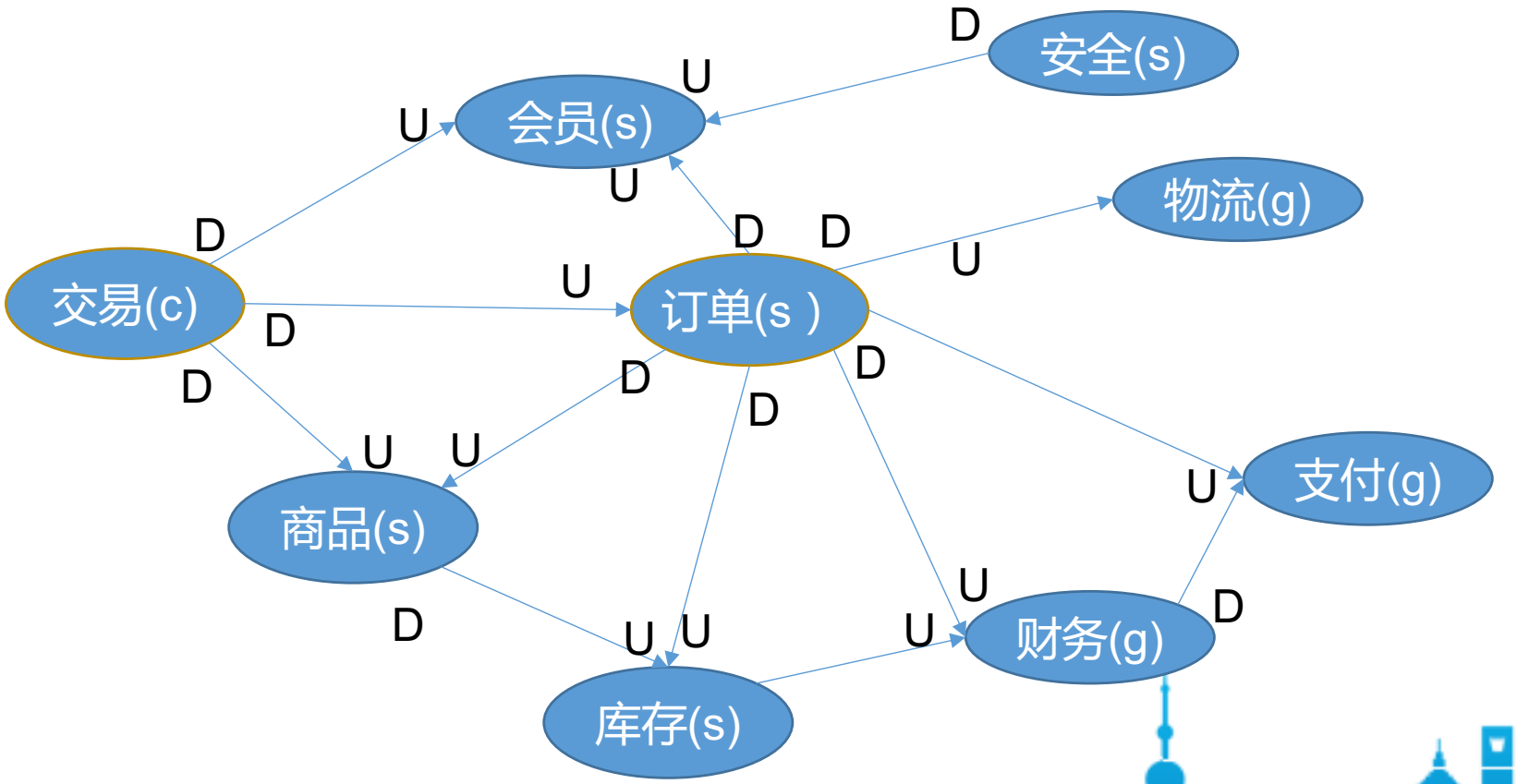
第一步：领域划分和限界上下文

CHINA DEVOPS DAYS

在线交易系统的领域划分



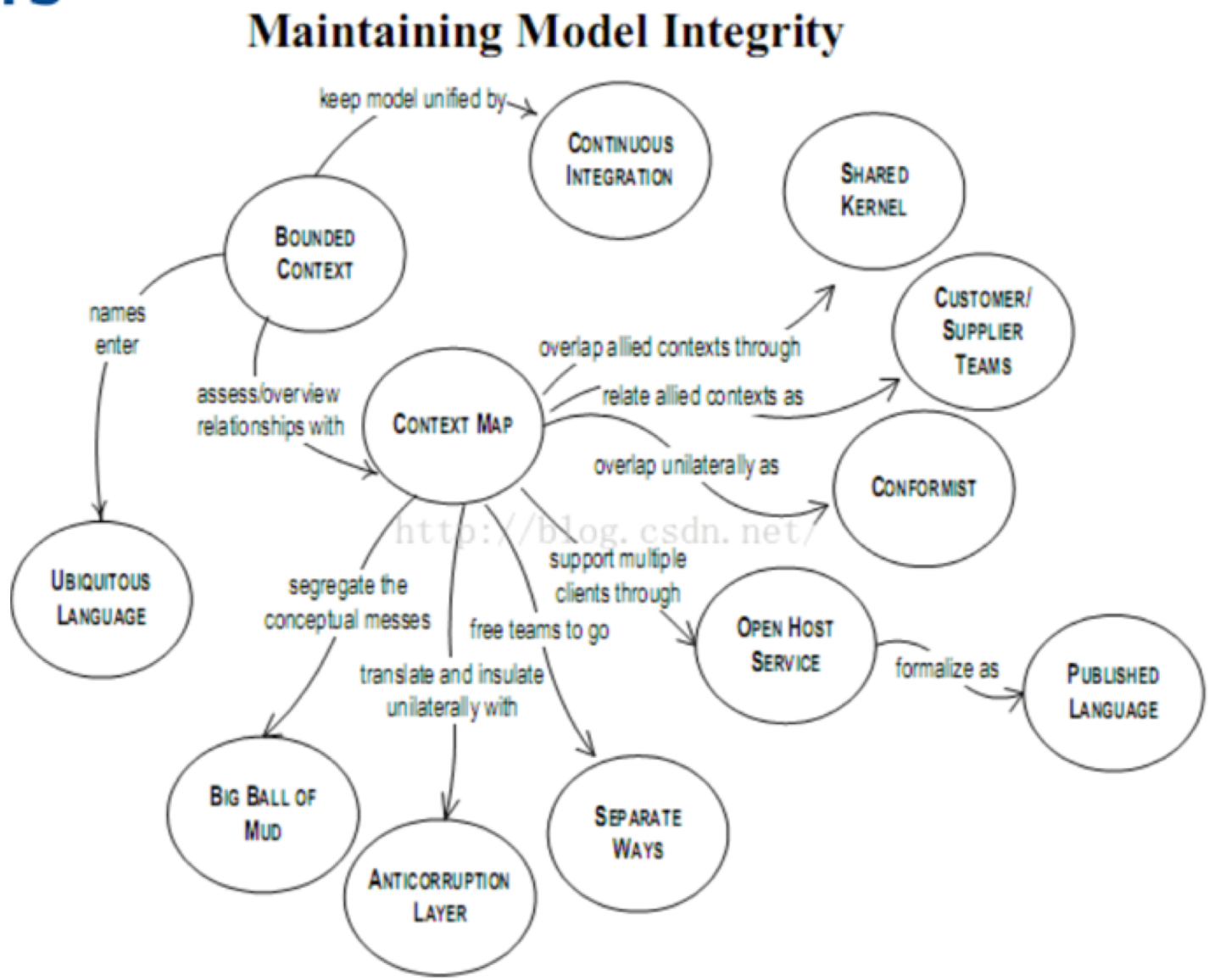
在线交易系统的限界上下文划分



上下文与领域原则上一一对应

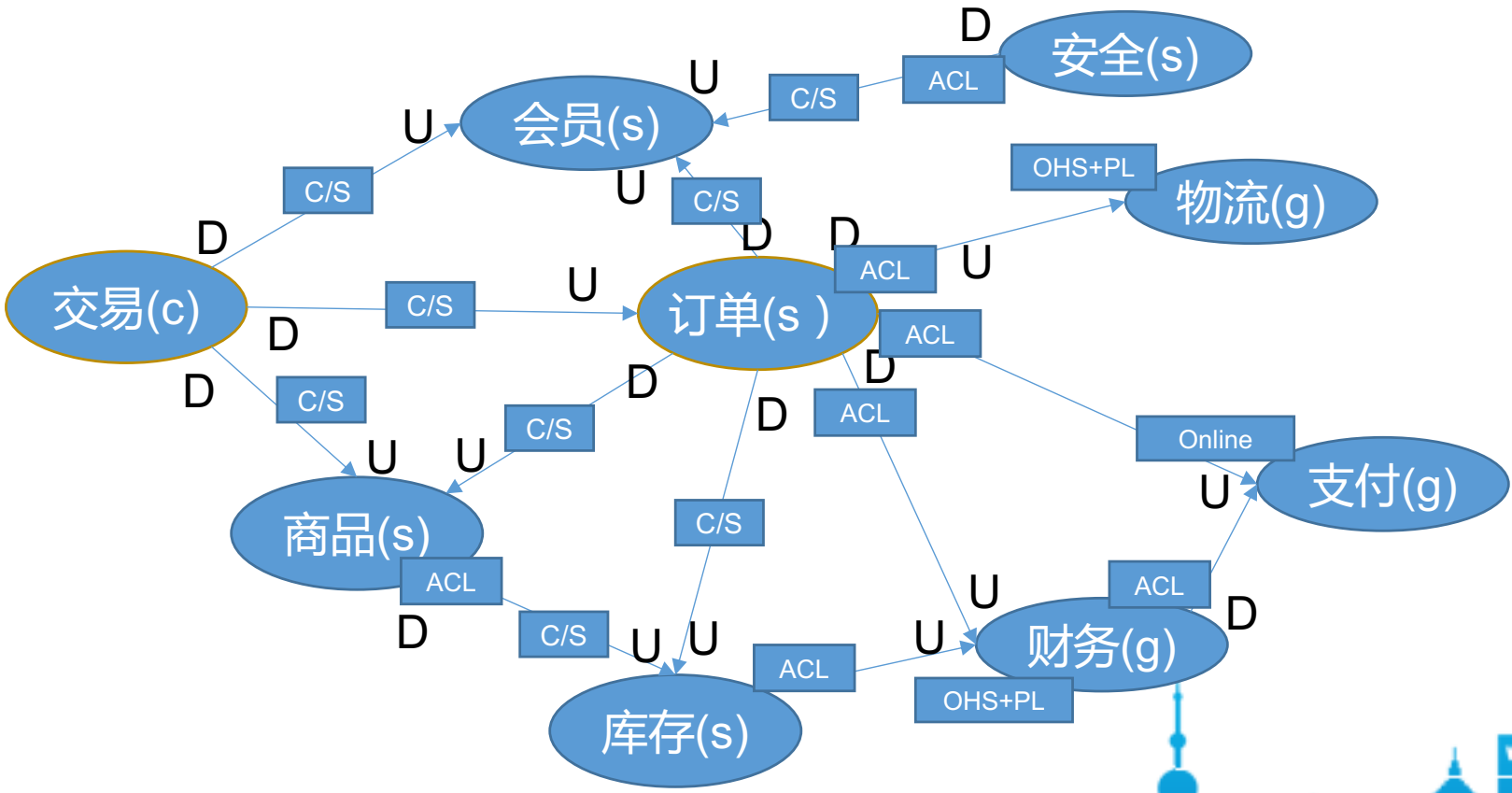
CHINA DEVOPS DAYS

上下文映射类型



CHINA DEVOPS DAYS

上下文映射图



第二步：在事件风暴中应用统一语言

概念：事件风暴

Event Storming是一种领域建模的实践，最初由Alberto Brandolini 开发

- 领域专家介绍业务
- 参与者可以任意提问
- 参与者根据自己对业务的理解，将
- 领域事件写在橙色即时贴上
- 每个即时贴一个事件
- 事件采用“xx已xx”的格式，如“订单已创建”

CHINA DEVOPS DAYS

寻找事件

从业务目标出发，按时间顺序寻找引发的事件，表示法：“XX已YY”。

交易核心域的部分事件如下：

订单已创建 → 订单已支付 → 仓储已发货 → 订单已签收

寻找事件背后Command（决策）

寻找**引发领域事件的Command（决策）**

CHINA DEVOPS DAYS

探索Command的产生、处理并生成领域事件的过程

类同于用例回合

Command的产生：谁？基于什么信息？发出了什么Command ？

1、**客户**基于**产品**信息，发出了**下单**的Command

3、系统基于**订单已生成事件**，产生了**锁定库存**Command

5、第三方支付基于客户的订单支付结果，产生了**支付成功**Command

Command的处理： 系统根据什么信息处理Command？产生（更新）了什么结果？生成了什么领域事件？

2、交易系统根据**下单**的Command和**库存**的*可售*数额，产生了**订单**，生成了**订单已生成事件**

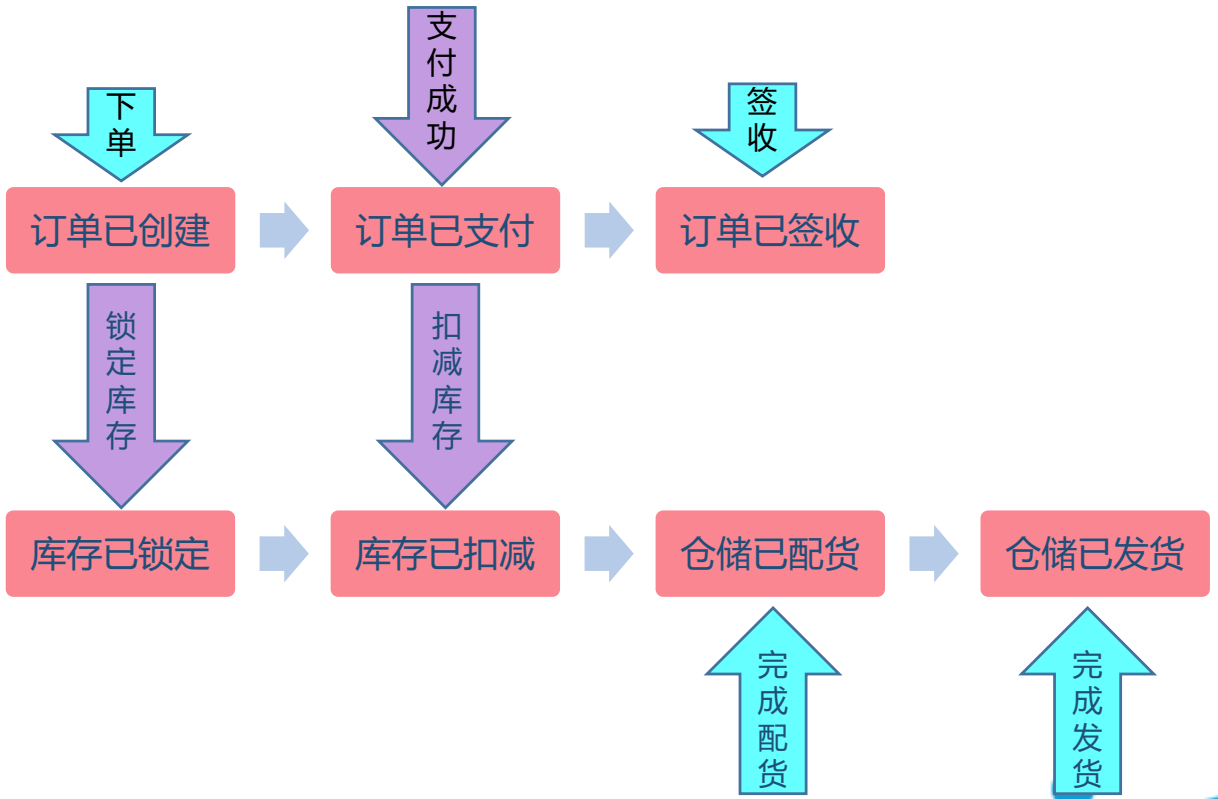
4、系统根据**锁定库存**Command ，更新了**库存**的*可售*数额和*已售未付款*数额，生成了**库存已锁定事件**

6、系统根据**支付成功**Command ，产生了**支付记录**，更新了**订单支付*状态***，生成了**订单已支付事件**

形成**统一语言**！！！

CHINA DEVOPS DAYS

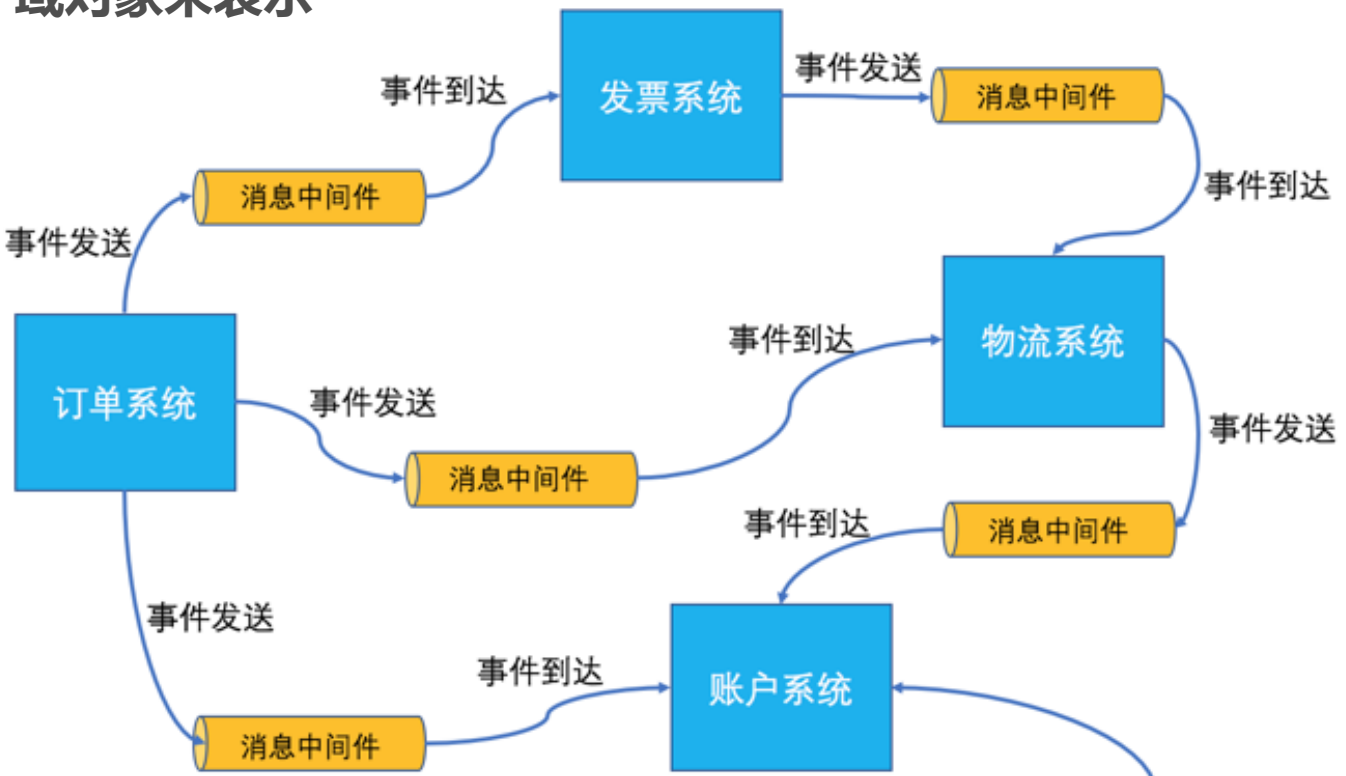
寻找更多Command和事件



CHINA DEVOPS DAYS

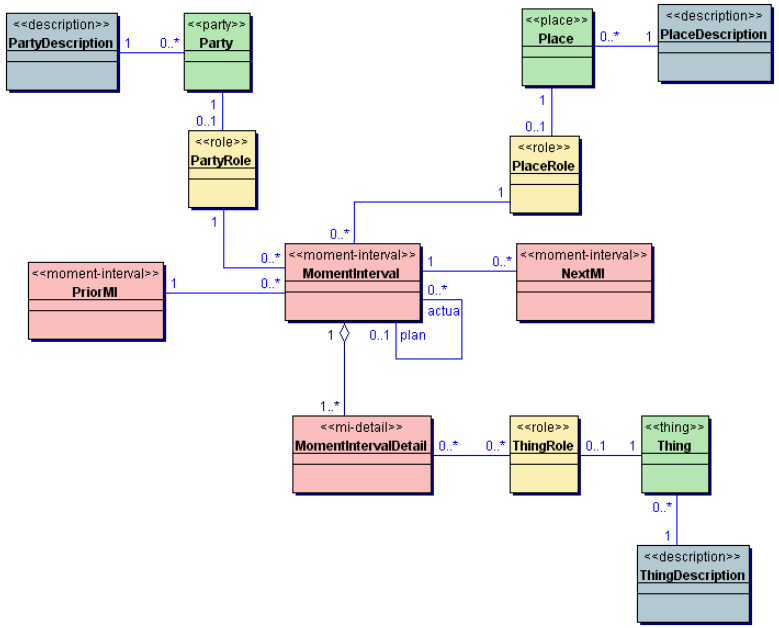
领域事件

将领域中发生的活动建模成一系列的离散事件，每个事件都用领域对象来表示



第三步：用四色原型分析业务

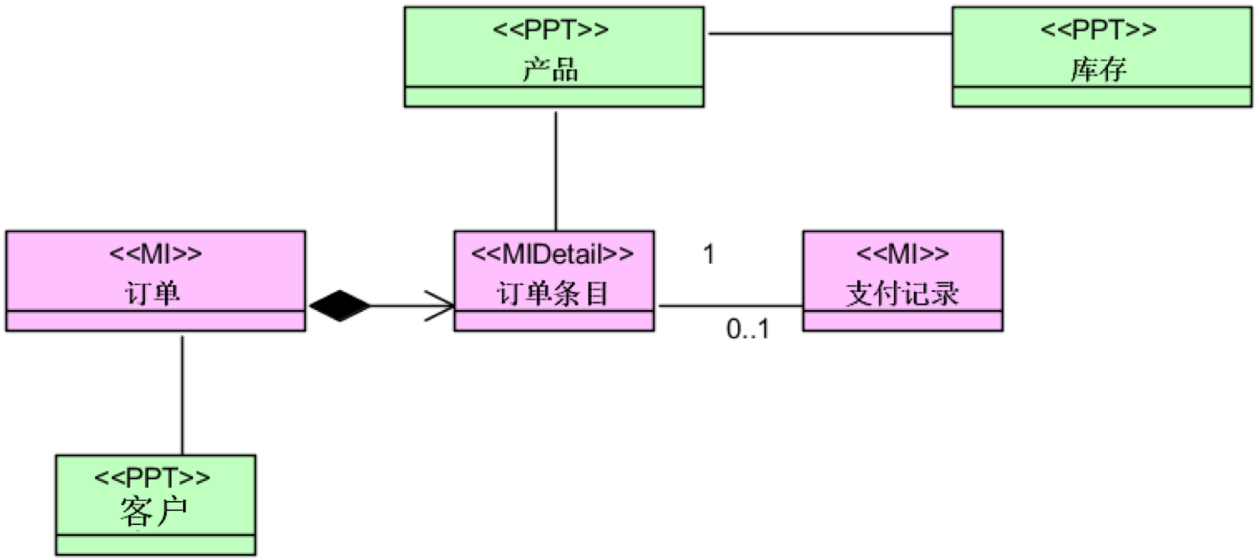
四色是领域模型的元模型



- Part、Place、Thing：简称PPT，用淡绿色表示；
- Description：简称Desc，用淡蓝色表示，主要用来对PPT、ROLE和MI进行描述。
- Role：用淡黄色表示，主要表示PPT在某个场景下扮演的角色。
- MomentInterval：简称MI，用淡红色表示，主要表示在一刻或一段时间内发生的一件事情。
- MomentInteval Detail：简称MIDetail，用淡红色表示，主要表示MI的明细。

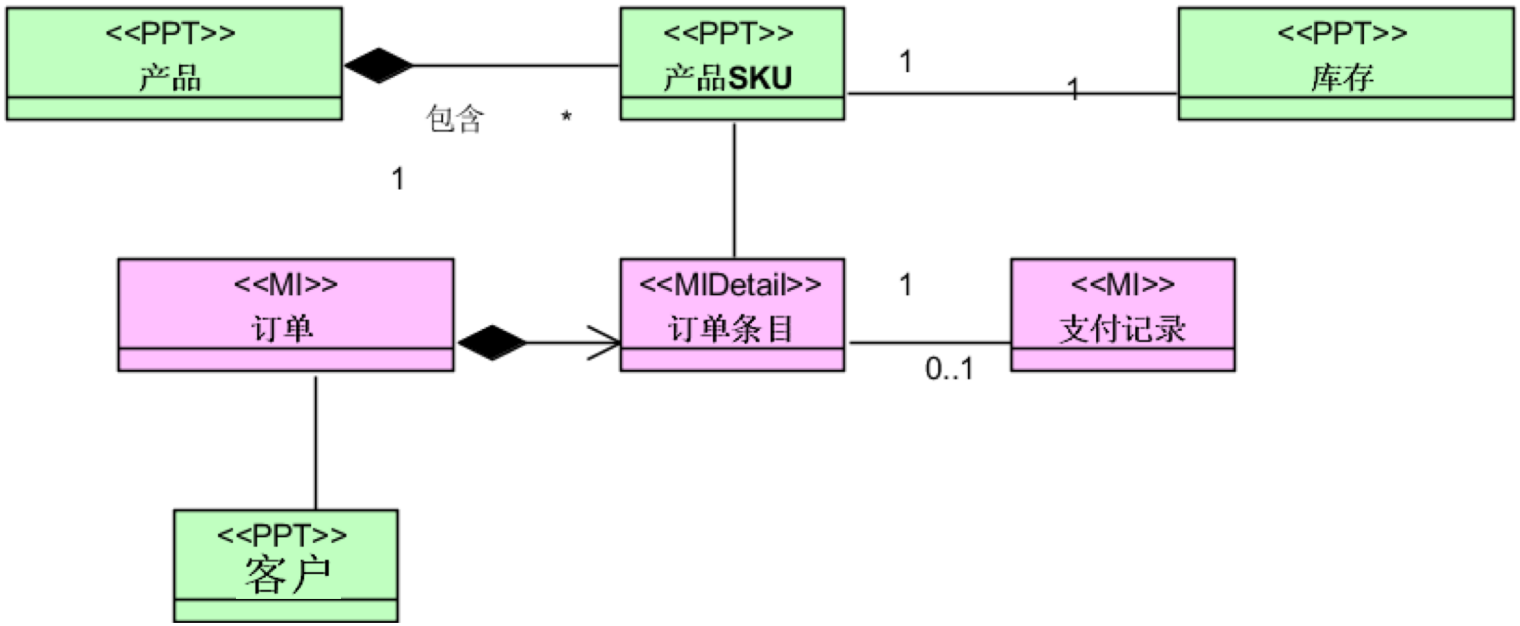
CHINA DEVOPS DAYS

初步梳理对象之间的关系



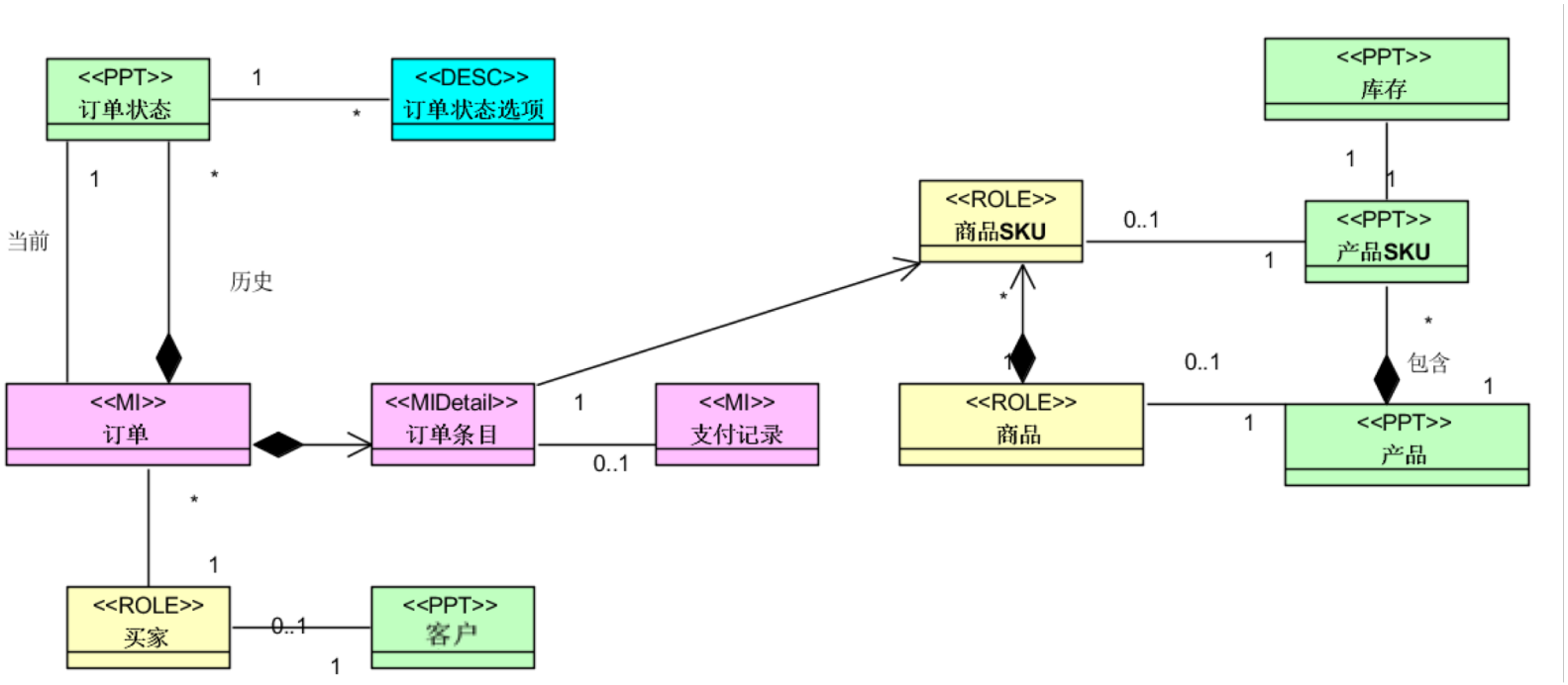
CHINA DEVOPS DAYS

加深理解对象之间关系



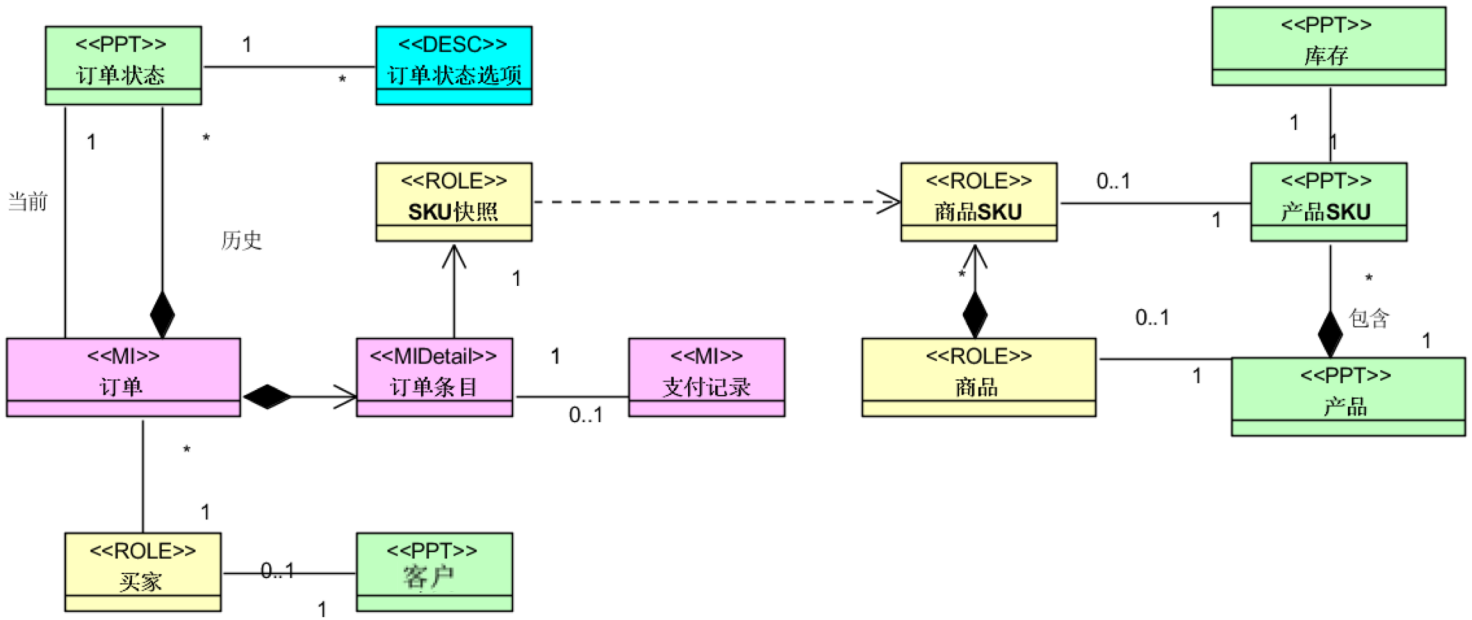
CHINA DEVOPS DAYS

加入ROLE和DESC



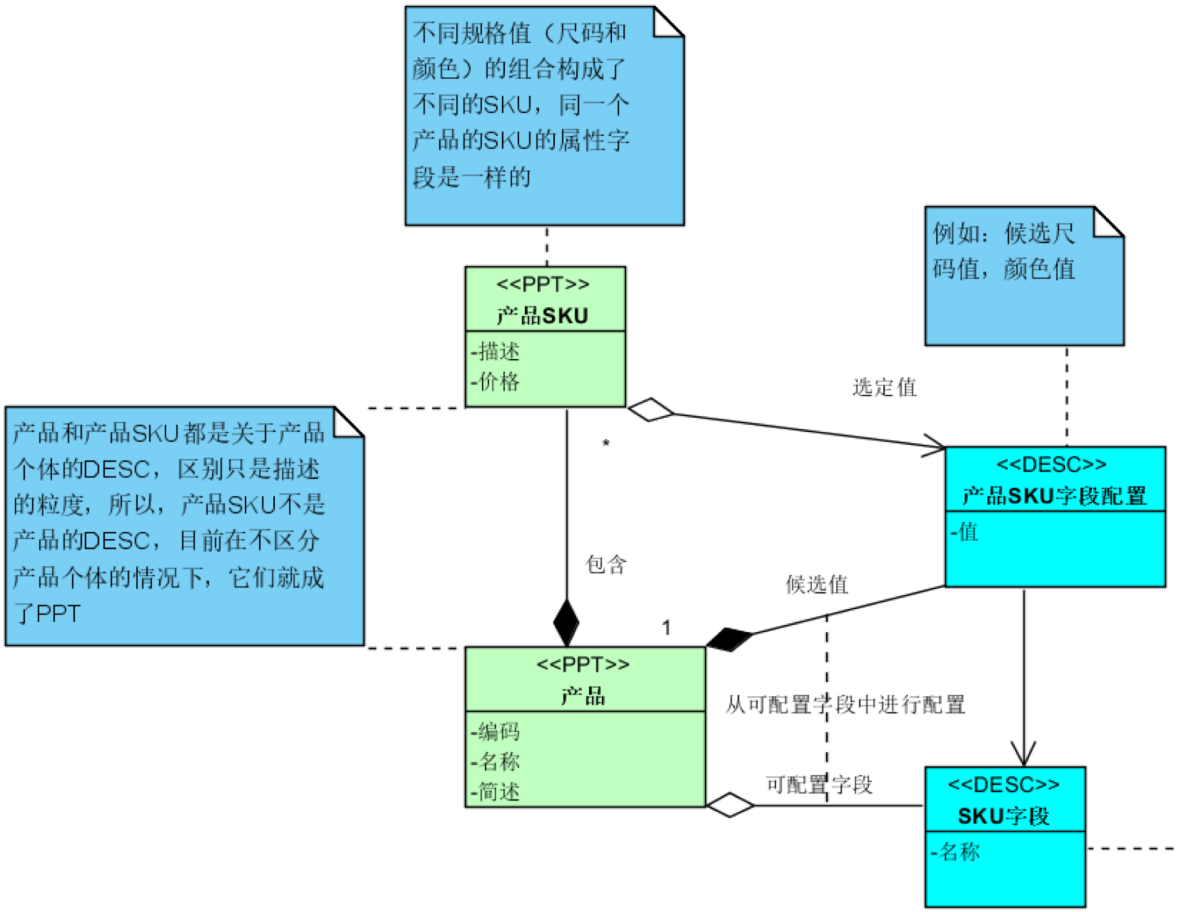
CHINA DEVOPS DAYS

“逐步”建立有效的模型



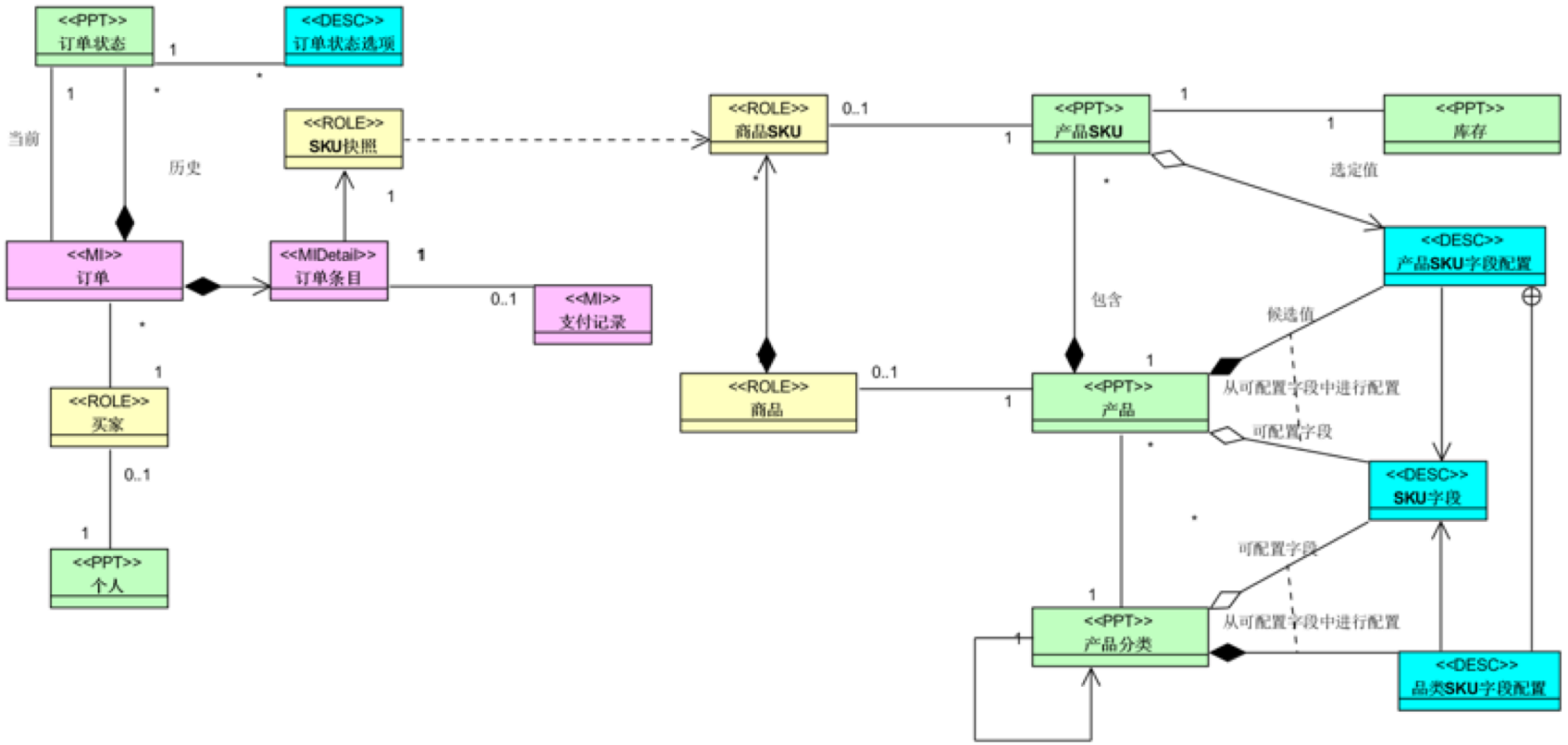
CHINA DEVOPS DAYS

精炼领域模型



CHINA DEVOPS DAYS

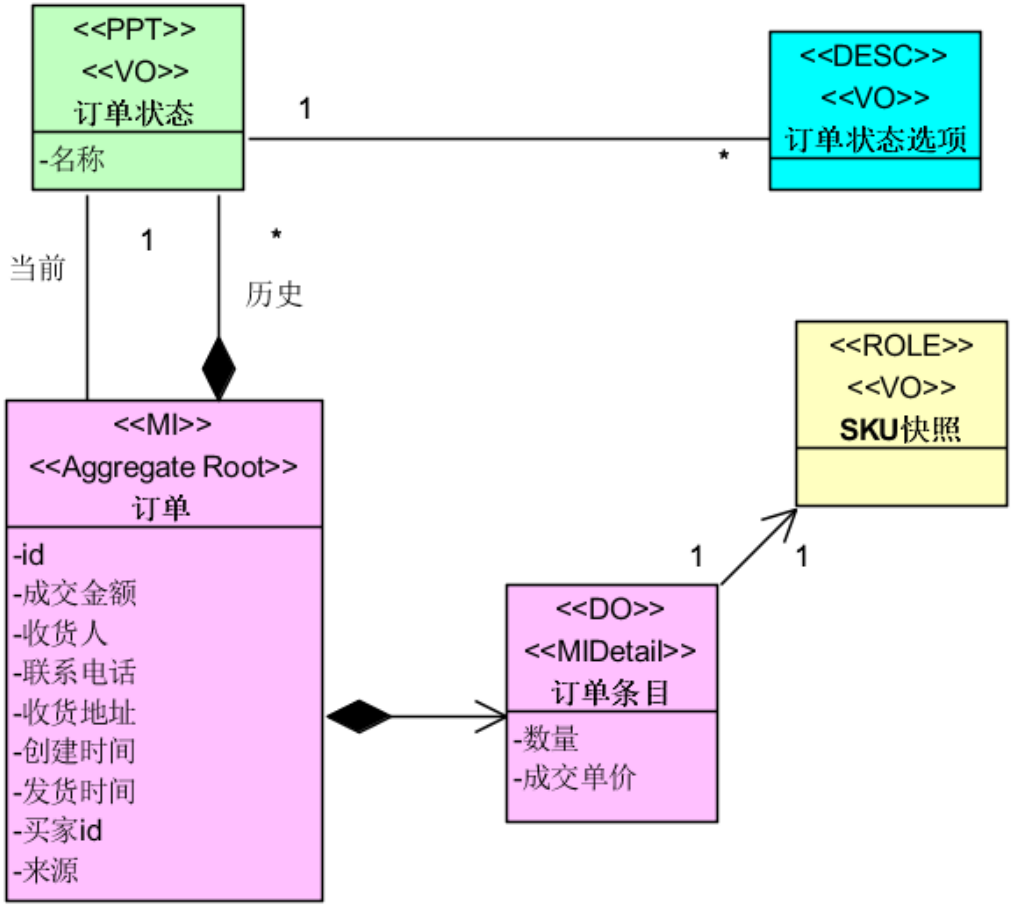
交易系统分析模型（部分）



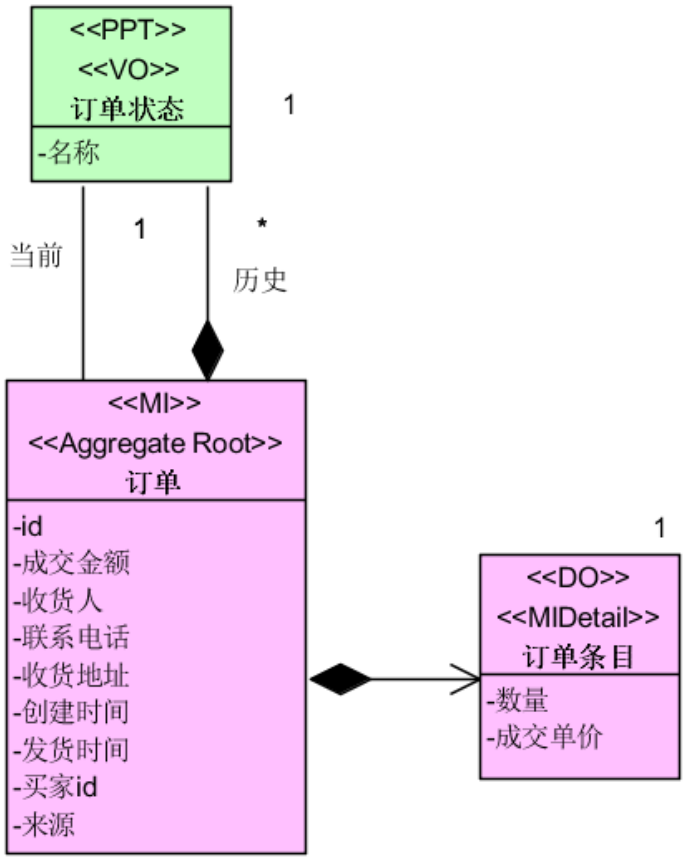
第四步：从分析模型到领域驱动设计

CHINA DEVOPS DAYS

概念：聚合/聚合根/实体/值对象



实体VS值对象

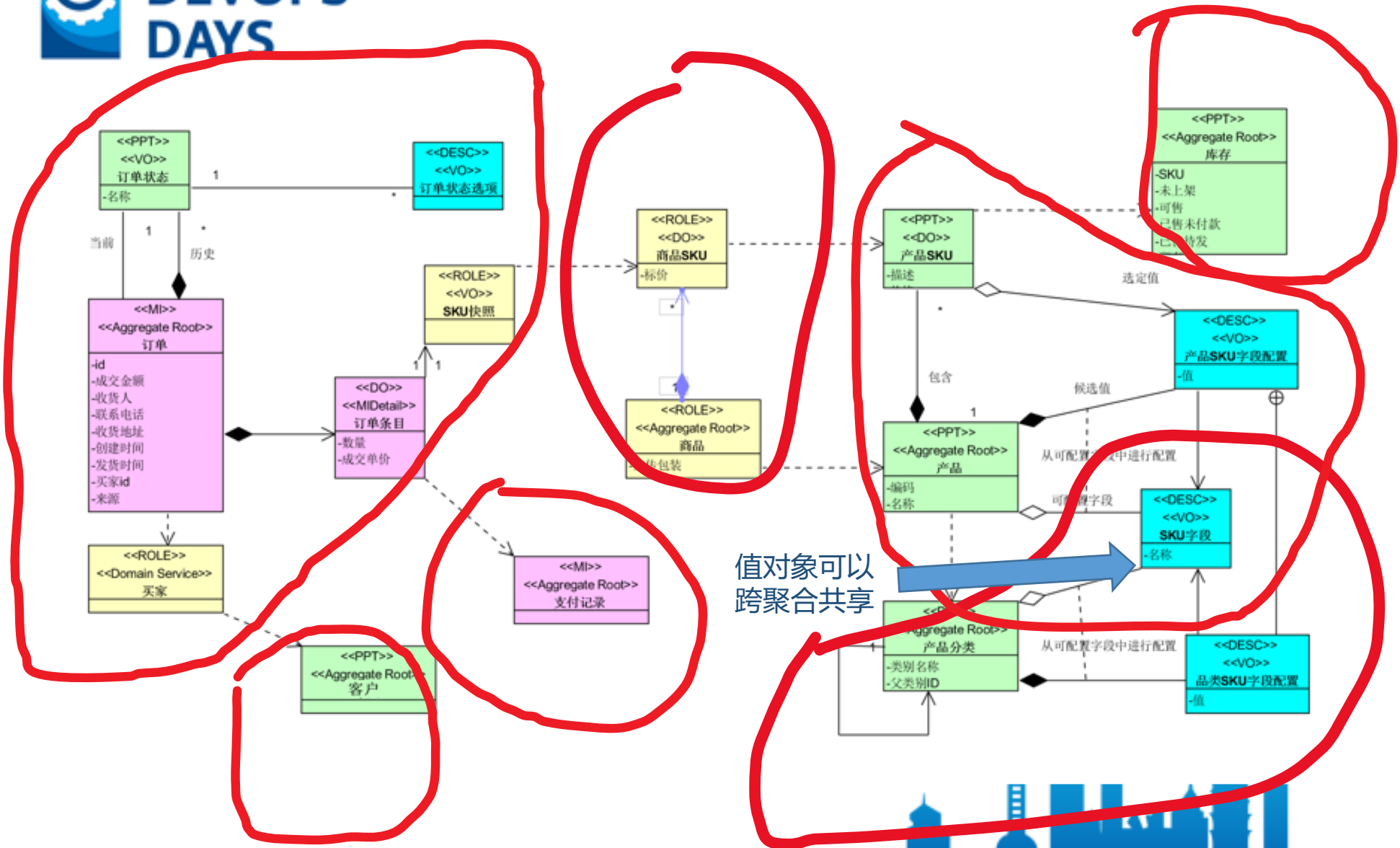


值对像：地址、电话、状态、收货人

实体:买家、订单条目

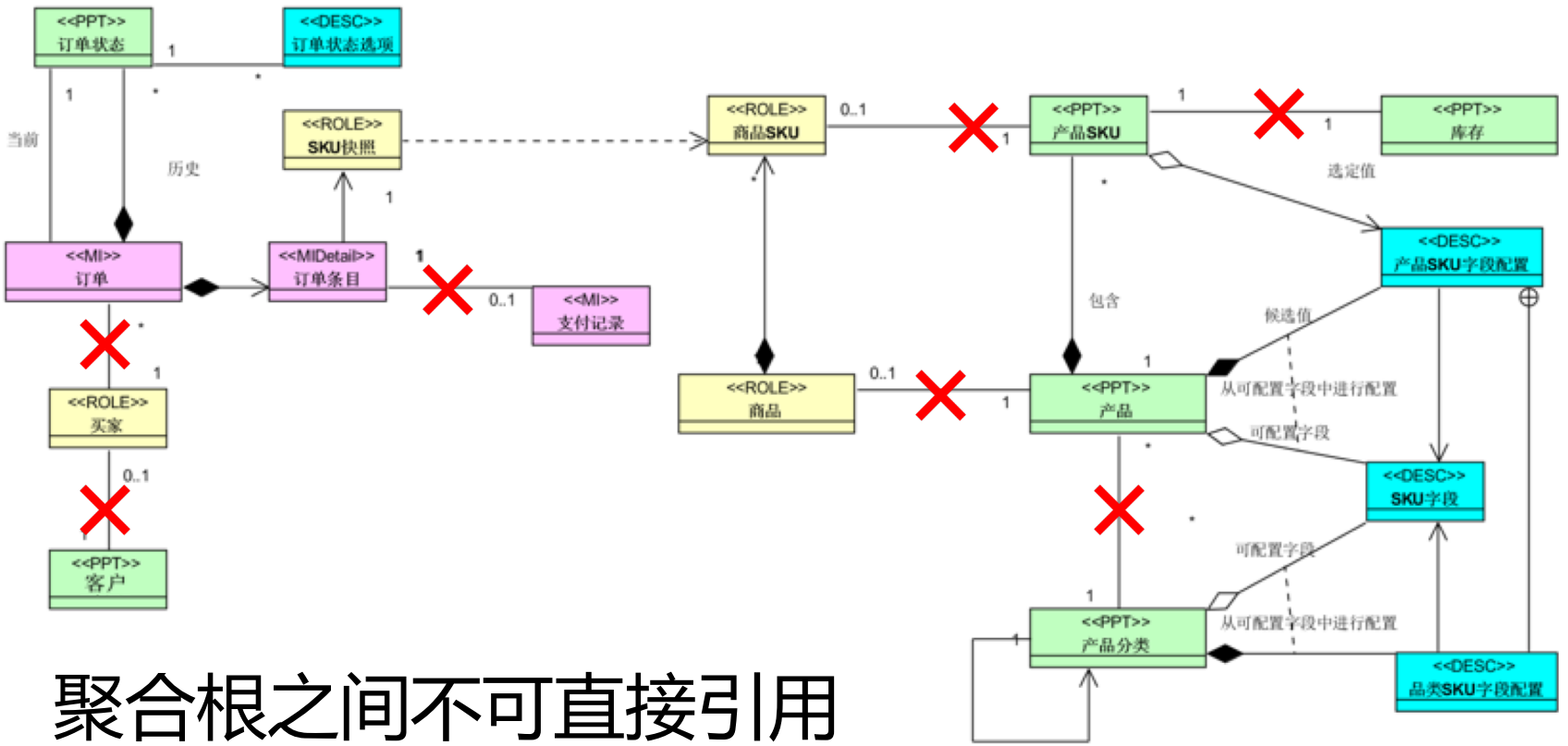
CHINA DEVOPS DAYS

划分聚合



降低实现模型的关联数量

分析模型的关联不等于设计模型的关联，设计模型可以实现为查询

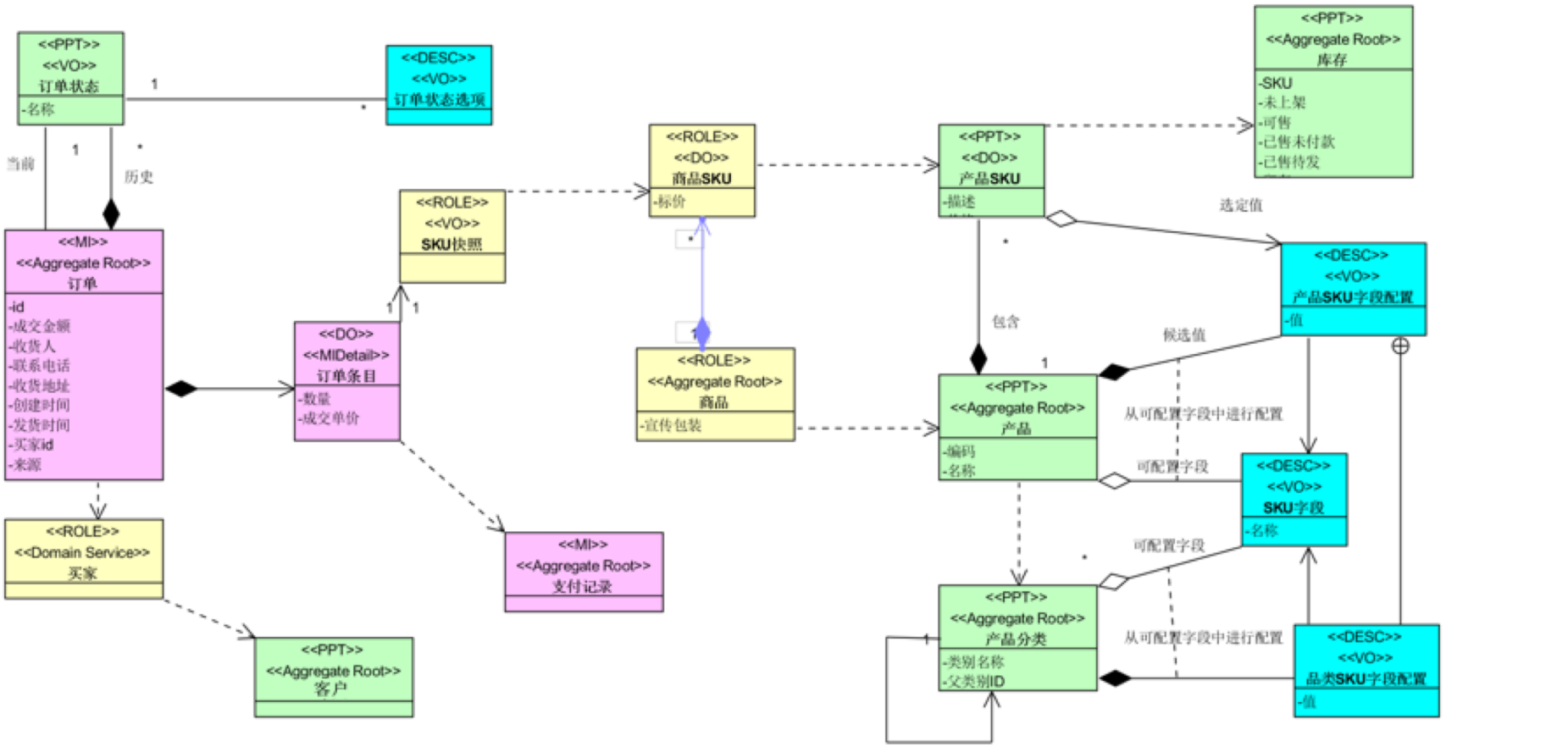


聚合根之间不可直接引用

CHINA DEVOPS DAYS

从四色原型到DDD实现

分析模型不等于设计模型，但需要降低二者的表示差异性。

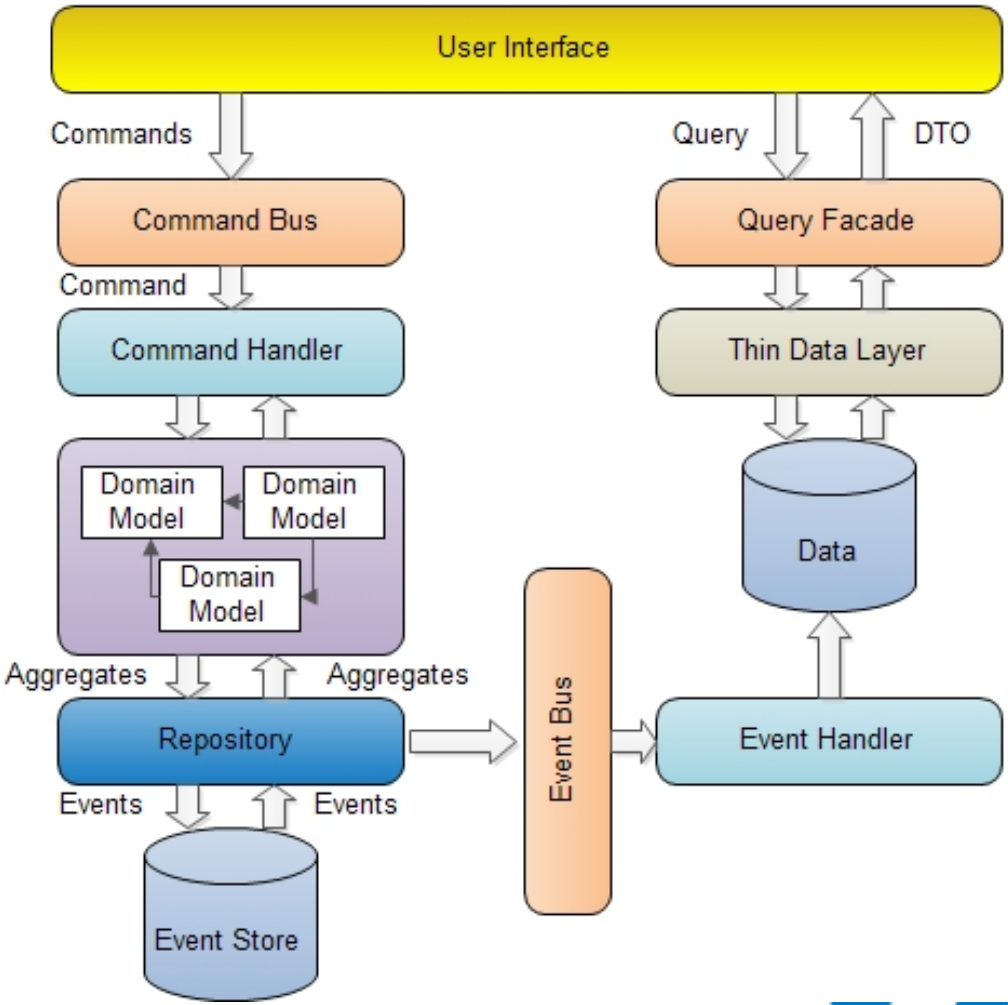


聚合与四色原型

- Part、Place、Thing：可以实现为DO或聚合根，也可实现为枚举值成为VO；
- Description：可以实现为DO或聚合根，实现为枚举类型可以成为VO。
- Role：如果Role有自己的状态需要被记录，可以实现为DO或聚合根，且推荐采用关联而不是继承自PPT ；没有状态要被记录，可以实现为有状态领域服务。
- MomentInterval：可以实现为聚合根（记录状态）或领域服务（不记录状态）。
- MomentInteval Detail：可以实现为DO（记录状态）或领域服务（不记录状态） ，包含在MI内。

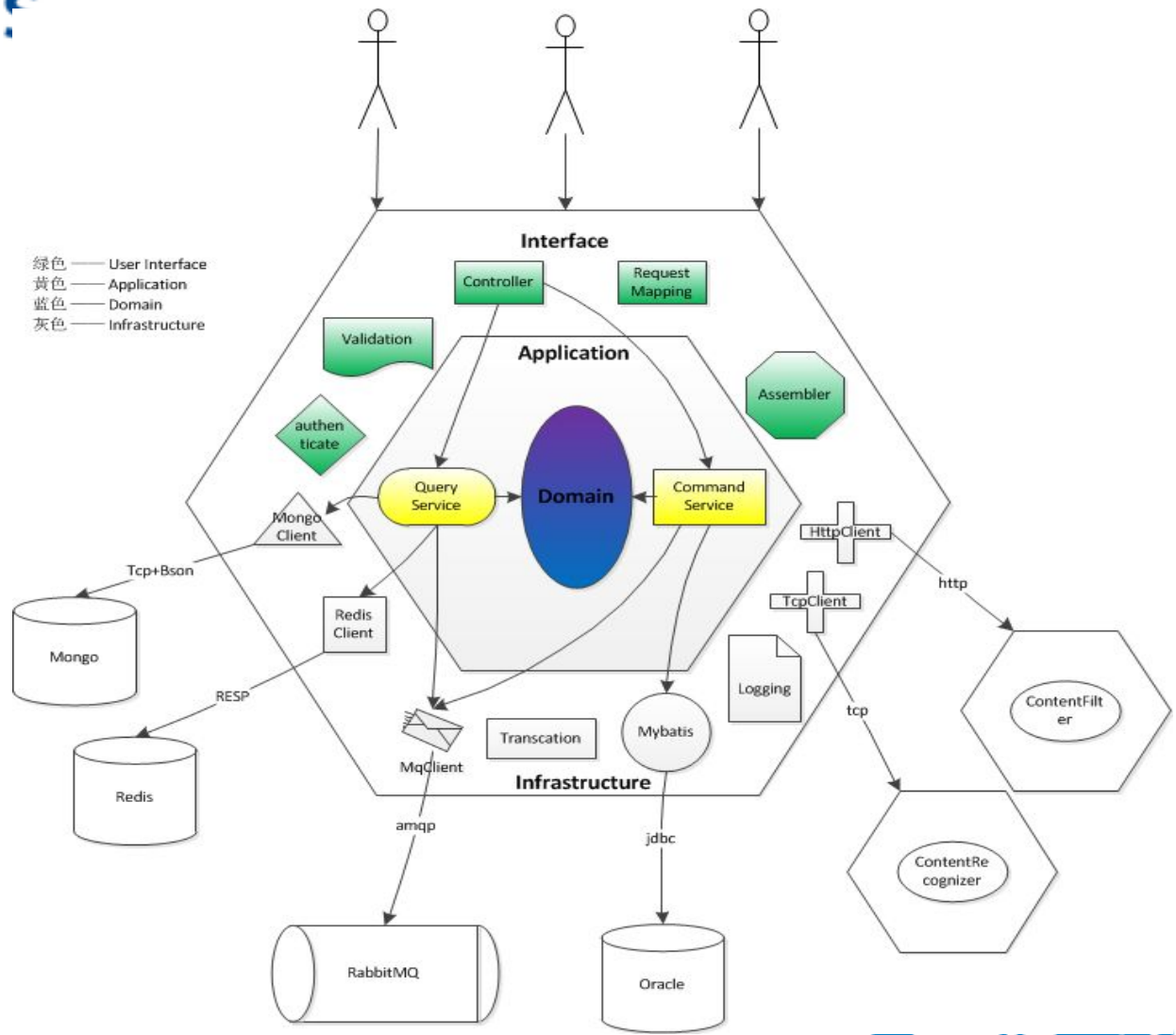
CHINA DEVOPS DAYS

聚合与CQRS



CHINA DEVOPS DAYS

DDD与六边形架构

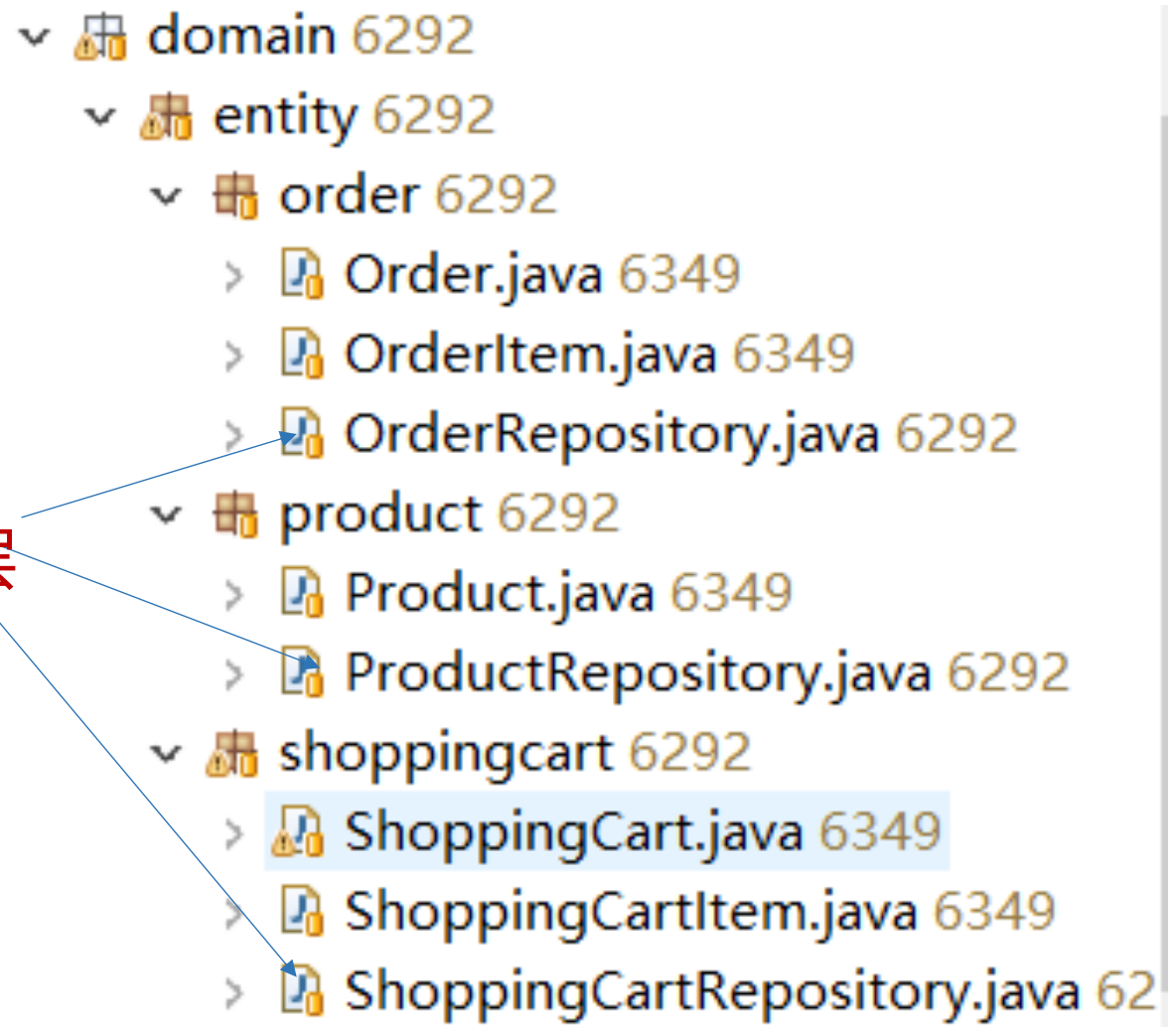


CHINA DEVOPS DAYS

聚合的持久化

为了便于讲解，将代码做了一些简化，去掉了sku，直接用pruduct代替

仓储接口处于领域层



CHINA DEVOPS DAYS

DDD知识地图

CHINA DEVOPS DAYS

谢 谢 ！