领域驱动设计(DDD)浅谈

FreezeSoul

2011-08-11

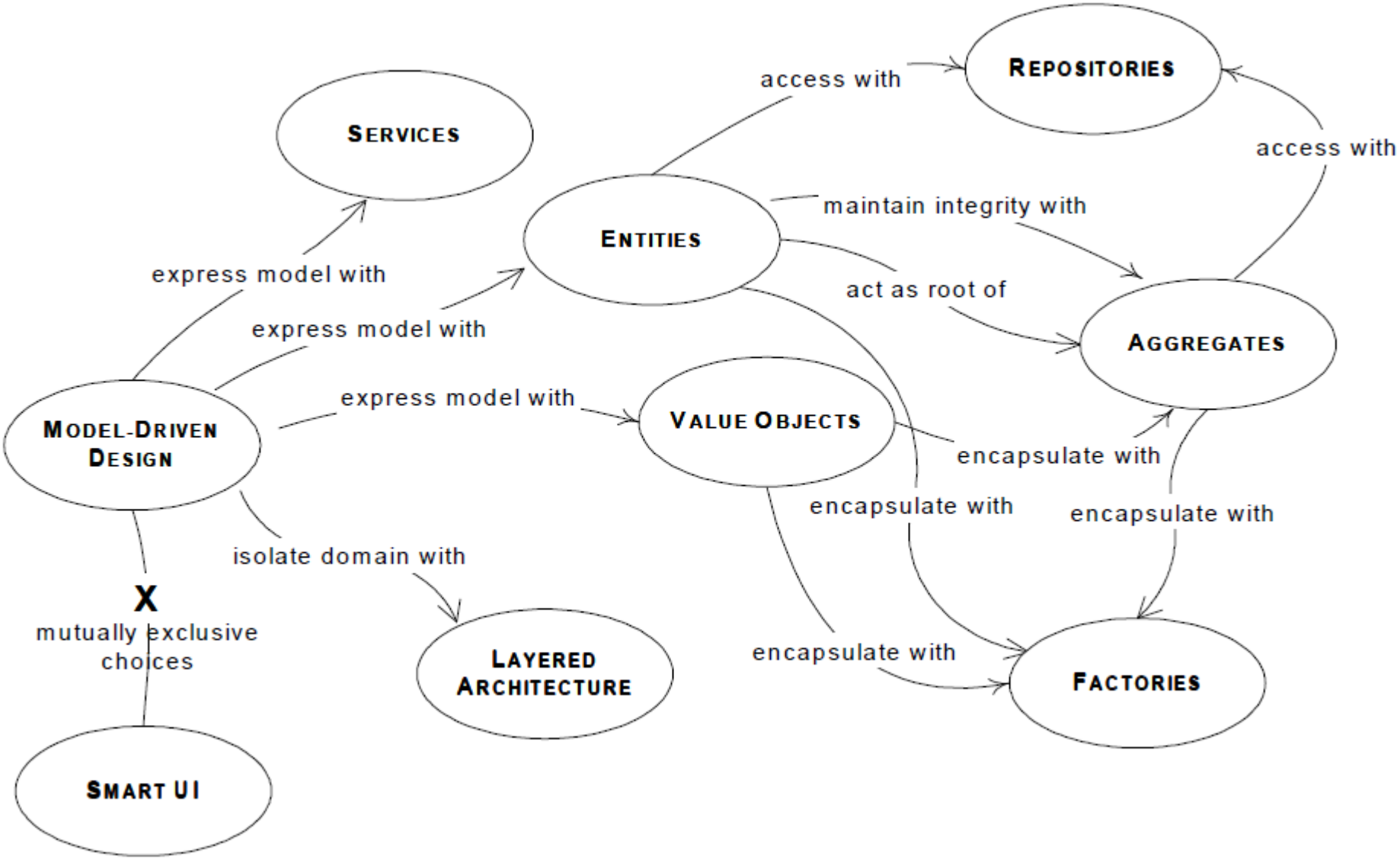
什么是领域驱动设计

- 领域驱动设计的提出是由 Eric Evans在其《领域驱动设计》一书提出的。
- 领域驱动设计(Domain Driven Design)是一种软件开发方法，目的是让软件系统在实现时，准确的基于对真实业务过程的建模，并根据真实业务过程的调整而调整。
- 领域驱动设计标志性特征是理解目标领域并把学到的知识融合到软件中当做首要任务.
- Wiki: http://en.wikipedia.org/wiki/Domain-driven_design

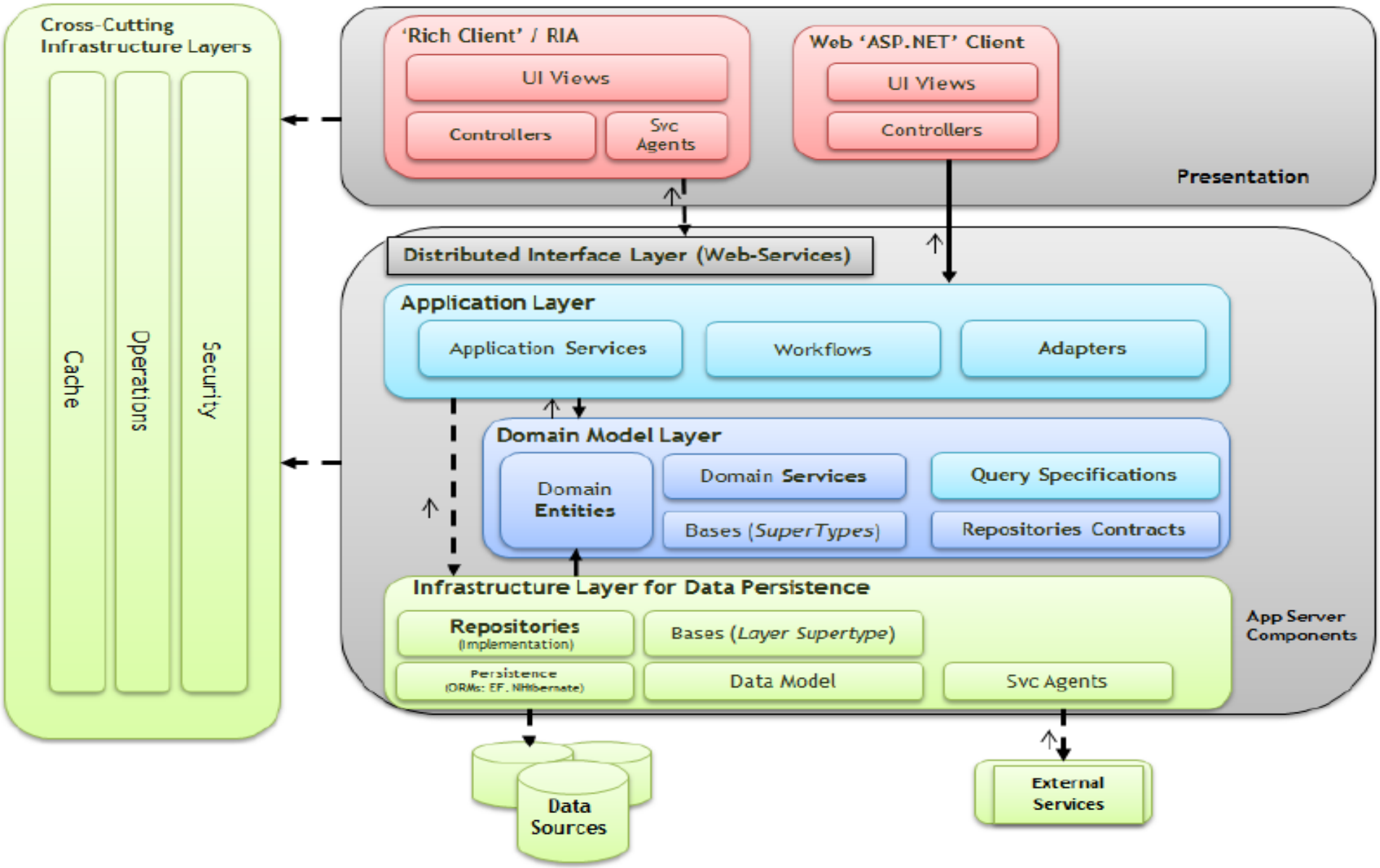
领域驱动设计的适合场景

- 并非所有的软件开发需求都适合
- 领域驱动设计适合于一个需求不断变化、持续发展的软件项目
- 对于简单的项目,Smart UI(anti pattern)同样适合
- 统一过程、敏捷开发、驱动领域设计3者的关系

领域驱动的基本构成要素



领域驱动设计的基本软件架构



领域驱动设计的基本软件架构说明

- 用户界面层（表示层）Presentation Layer
- 应用层Application Layer
- 领域层（模型层）Domain Model Layer
- 基础设施层Infrastructure Layer

领域驱动设计的核心语言

- 通用语言（Ubiquitous Language）——领域模型
- 一旦决定使用领域驱动开发，需要建立通用语言术语表。（目的是将其作为一种交流语言）

领域驱动的基本构成要素

- Entity（实体）
- Value Object（值对象）
- Service（服务）
- Aggregate（聚合根）
- Factory（工厂）
- Repository（仓储）
- ~~Module（模块/包）~~
- ~~Context（上下文）~~

1.实体Entity

- 以标识作为基本特征的对象，并且协调其对象的操作来完成自己的职责，我们称之为实体。
- Entity的最基本职责是确保连续性，以便使其行为更清楚及可预测。
- 只添加在领域概念中至关重要的行为，和这些行为所必需的属性，以及标识特征。
- 与Value Object区别

2.值对象Value Object

- 用于描述领域的某个方面而本身没有概念标识的对象称为Value Object。
- 不可变，简单，不用分配任何标识。只关心这个模型元素的属性。
- 在不同场景下可能Value Object与Entity是可切换的

3.聚合Aggregate

- 聚合是相关的一组领域对象的集合（包括实体或值对象等），这组领域对象联合起来表述一个完整的领域概念。
- 每个Aggregate都有一个根（root）和一个边界（boundary）。
- 在不同的领域边界或上下文中，同一事物是否作为聚合的标准可能不同。

4.服务Service

- 一些领域概念不适合被建模为对象，它的一些领域行为既不属于某个Entity也不属于某个Value Object职责，这时应该添加一种作为独立接口的操作，称之为Service，它是没有状态的
- 确保“操作”存在于Ubiquitous Language的术语表中，它也是领域模型的一部分
- 此Service非彼Service（应用层Service、领域层Service）

5.工厂Factory

- 为了避免复杂Entity/ Aggregate的创建，或避免暴露过多的内部结构。
- 在创建复杂领域对象时应把其作为整体，满足所有规则。
- 与Repository区别，不包含数据访问及持久代码，只包含创建复杂领域对象的业务规则及逻辑。

6.仓储Repository

- 领域模型中所有与数据库打交道的操作。
- 解决Entity不应知道持久细节以及性能问题。
- 与Factory的区别，不包含任何领域逻辑，只包含数据访问逻辑。

驱动领域开发之柔性设计

- 1.Intention-Revealing Interfaces (释意接口)
- 2.Side-Effect-Free Function（无副作用的函数）
- 3.Assertion（断言）
- 4.Conceptual Contour（概念轮廓）
- 5.Standalone Class（孤立的类）
- 6.Closure Of Operation（闭合操作）
- 更多话题：保持模型的完整性、精炼等

Domain-specific language

- DSL(特定领域语言)是一种特化的，面向问题的语言。
- 编程语言不是面向特定的问题域而是一般问题域。
- 更关注我们要做什么，而不是怎么做的问题。
- DSL不处理编程语言中所涉及到的技术问题。
- 比如：UML、正则表达式、CSS、TSQL、LINQ、WF、jQuery、中文编程？？？

Entity Framework

- 并非单纯的Orm工具
- 多种使用方式，如：数据库优先、模型优先、Code First
- 结合Linq，WCF等技术

Demo

- **http://microsoftnlayerapp.codeplex.com/**

讨论

- 对于我们的系统是否适合领域驱动方式的开发？
- 如何做？