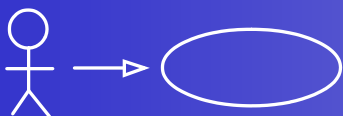


软件需求设计UML全程实作

总览

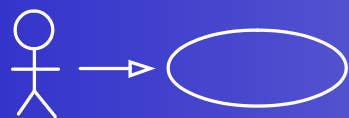
潘加宇



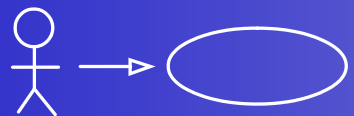
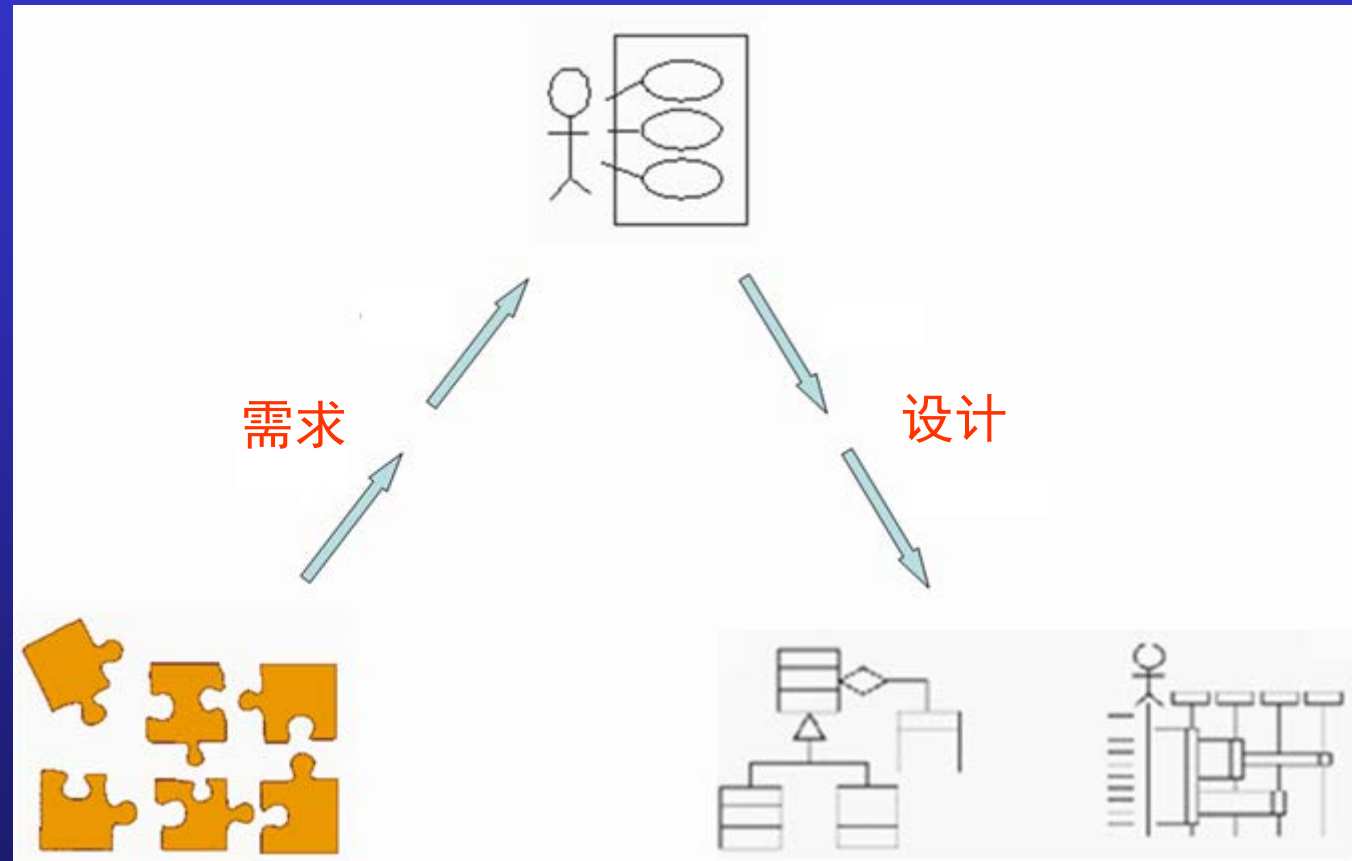
细节时代的来临

- 人人都要吃饭，所以我开饭馆能够赚钱？
- 每个人都要社交，所以开发社交软件就能赚钱？
- 竞争！
- 不再有农民企业家了！

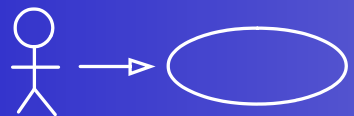
机会驱动、粗放经营的时代开始远去



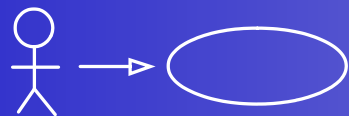
技能



测试一下...



利润=收入-成本

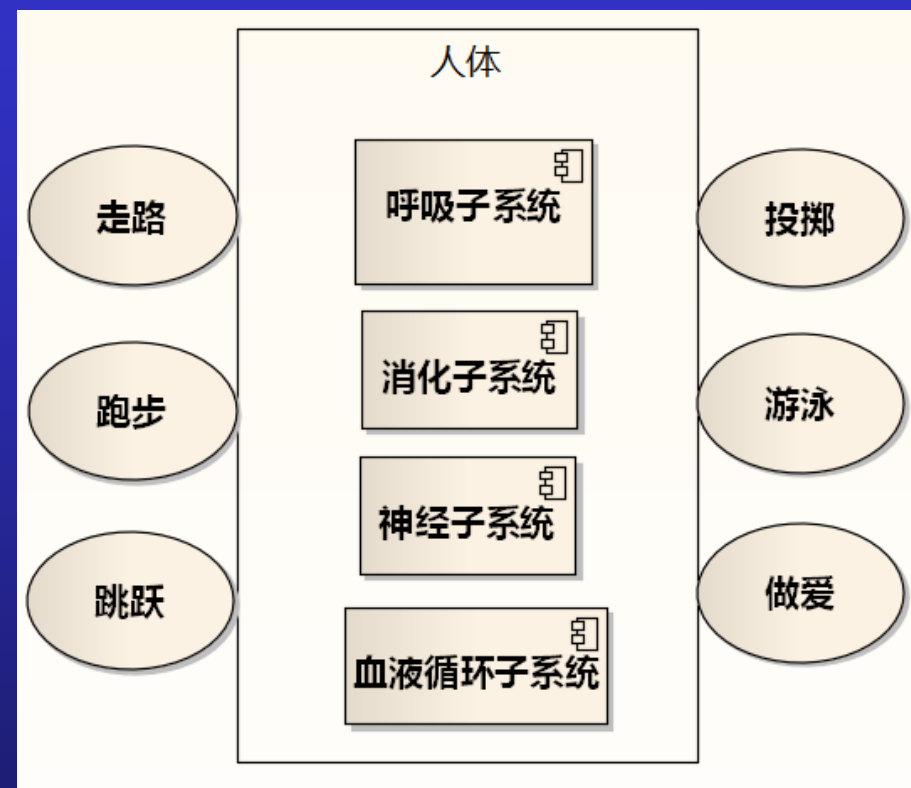


需求和设计

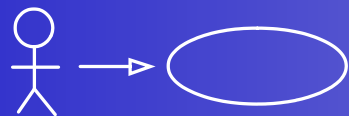
利润 = 需求 - 设计

提升销售...

降低成本...



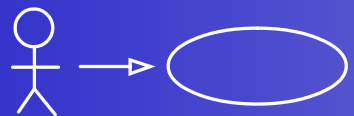
需求设计不分，利润缩水



需求和设计

需求	设计
卖的视角	做的视角
具体	抽象
产品当项目做	项目当产品做

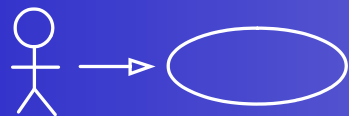
不同的视角



需求和设计



如何【低成本】做出【好卖】的各款产品



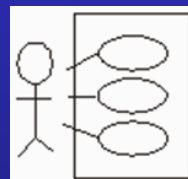
建模 workflow

➤ 组织要解决什么问题——业务建模

➤ 为了解决组织的问题，待开发系统应提供什么功能和性能——需求

这就是建模！

SRS



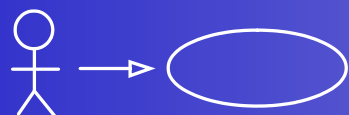
无意识→有意识
隐式知识→显式知识

提升销售

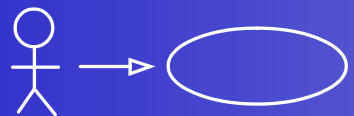
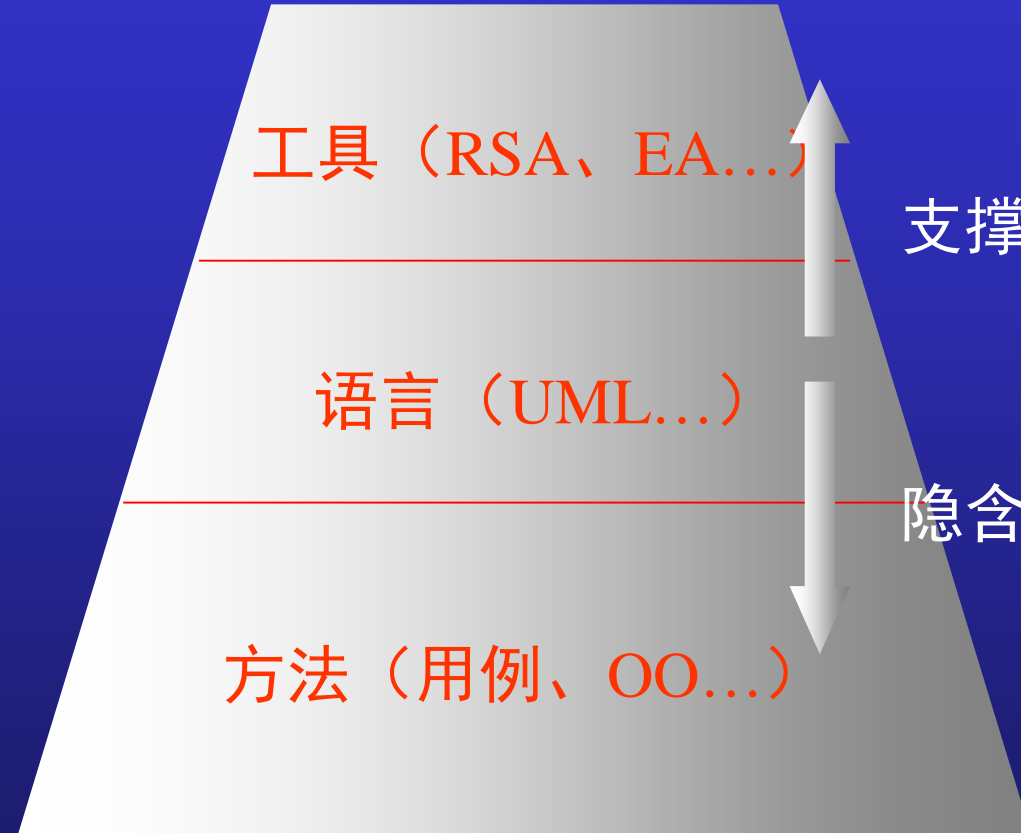
➤ 为了提供功能，系统内部应该有什么样的核心机制——分析

➤ 为了满足性能，系统的核心机制如何用选定平台实现——设计

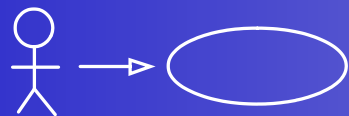
降低成本



建模知识体系



测试一下...

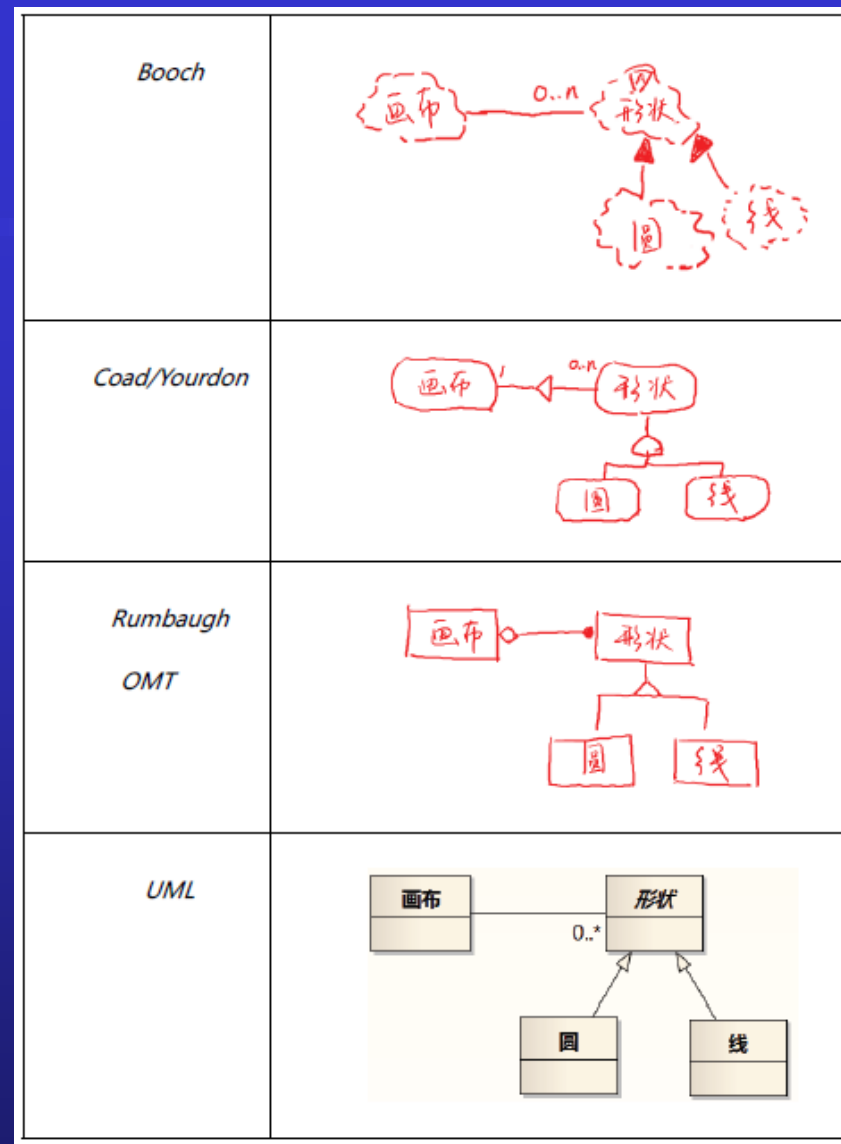


UML

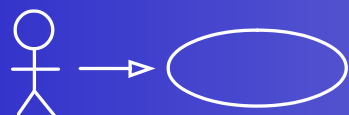
<i>Wirfs-Brock</i>	Responsibility（责任）
<i>Booch</i>	Operation（操作）
<i>Coad/Yourdon</i>	Service（服务）
<i>Stroustrup</i>	Function（函数）
×××××	Method（方法）

90年代初，有一定影响方法学有50多种

统一软件开发表示法



吃饭
恰饭
食饭
呷哺



$$\begin{aligned} \int_0^a x k(x-a)^2 dx &= k \int_0^a (x^3 - 2ax^2 + a^2x) dx \\ &= k \left[\frac{x^4}{4} - 2ax^3 + \frac{a^2x^2}{2} \right]_0^a = k \left(\frac{a^4}{4} - \frac{2a^4}{3} + \frac{a^4}{2} \right) \\ &= \frac{ka^4}{12} \quad \text{隐含着数学共识} \\ \int_0^a k(x-a)^2 dx &= k \int_0^a (x^2 - 2ax + a^2) dx \\ &= k \left[\frac{x^3}{3} - 2ax^2 + a^2x \right]_0^a = \frac{ka^3}{3} \\ \therefore \bar{x} &= \frac{ka^4}{12} \div \frac{ka^3}{3} = \frac{a}{4} \\ \bar{y} &= \frac{\int_0^a \bar{y} dA}{\int_0^a dA} = \frac{\int_0^a \frac{k(x-a)^2}{2} k(x-a)^2 dx}{\int_0^a k(x-a)^2 dx} \\ \int_0^a \frac{k^2(x-a)^4}{2} dx & \quad \text{Let } u = x-a \\ & \quad du = dx \\ &= \frac{k^2}{2} \int_0^a u^4 du = \frac{k^2}{2} \frac{u^5}{5} = \frac{k^2}{10} (x-a)^5 \Big|_0^a \\ &= \frac{k^2}{10} (0 - (-a)^5) = \frac{k^2 a^5}{10} \\ \bar{y} &= \frac{\frac{k^2 a^5}{10}}{\frac{ka^3}{3}} = \frac{3ka^2}{10} \quad \text{But if } x=a, y=b \\ & \quad b = k(0-a)^2 = ka^2 \\ & \quad k = \frac{b}{a^2} \\ \therefore \bar{y} &= \frac{3b}{10} \end{aligned}$$

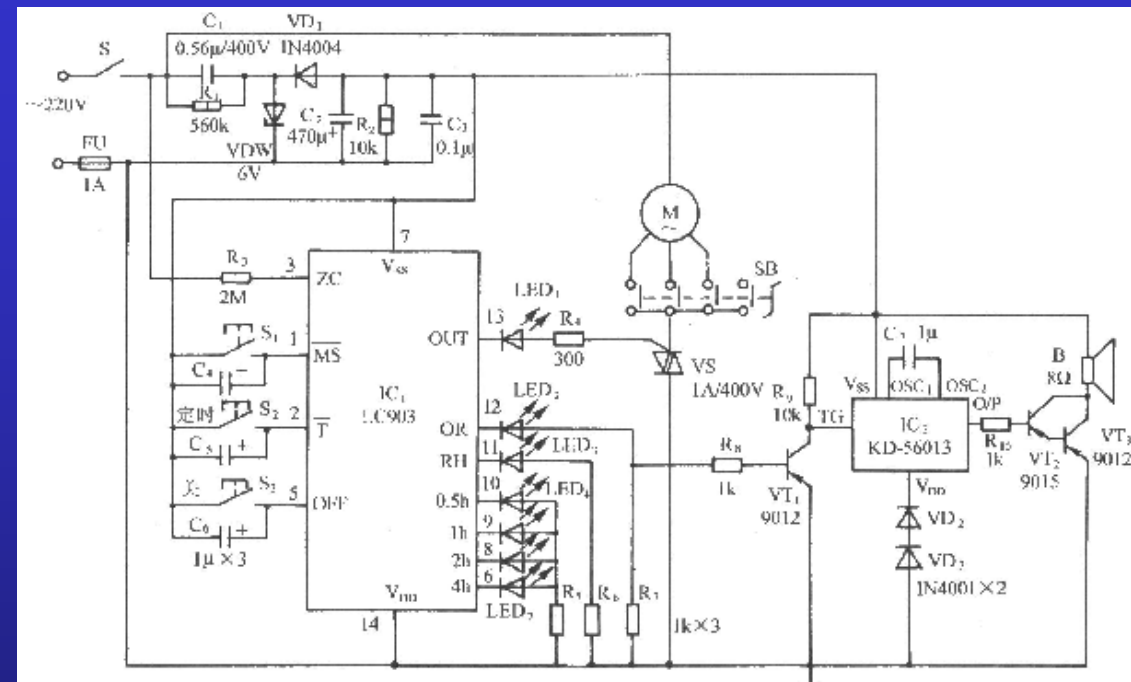
2

Sonata IV
Johan Agrell (1701-1765)

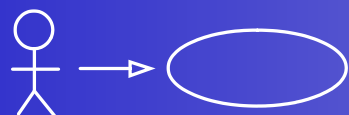
Allegro

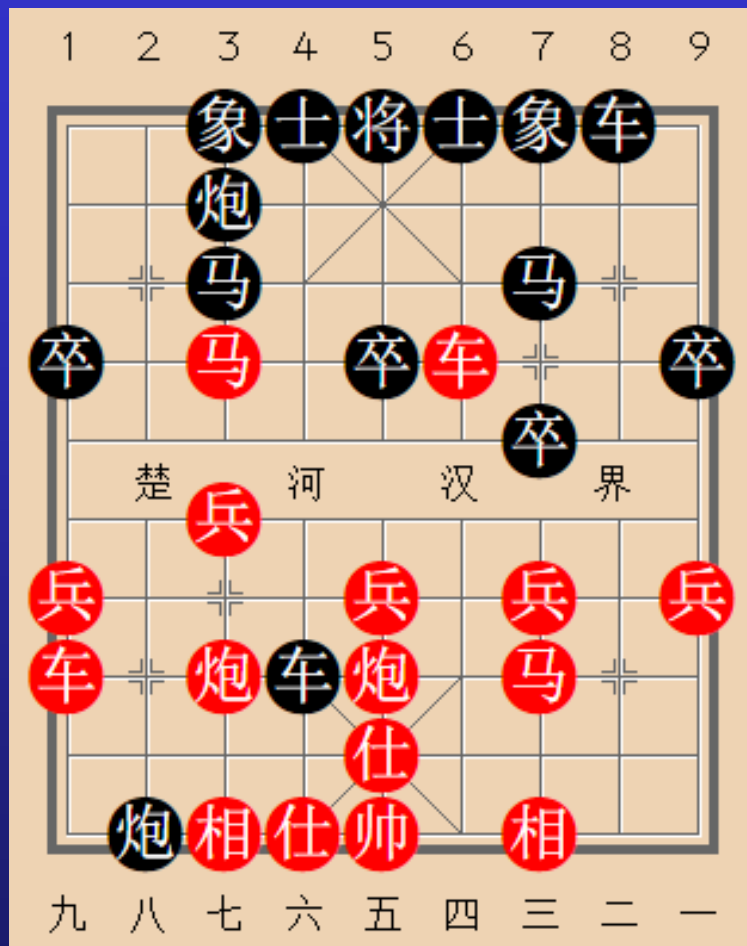
隐含着基本乐理

©1997-99 Johan Tefreman Non-commercial copying welcome. Revisions: 1.12 <http://www.lybrate.lu.se/runeberg/agrell>



在基本共识上沟通





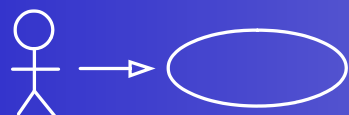
➤ 民工下棋，看似热闹，思考无深度

➤ 无知无畏，到处都是正确答案

➤ 职业初段：只有两三处值得考虑

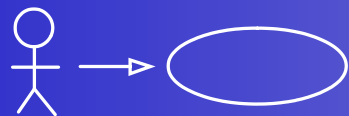
➤ 平静，思考有深度

提高沟通效率和深度



- “高手”喜欢画“草图”：来，我给大家讲讲！
- “优点”：
 - 项目要依赖于“我”头脑中的隐式知识
 - 怎么画都对，解释权归“我”所有
- 以形式的粗陋 遮掩 内容的粗陋
 - 因为用马粪纸写作，所以有理由写错别字，犯语法错误？
 - 因为在沙滩上用树枝演算，所以有理由乱用公式定理？

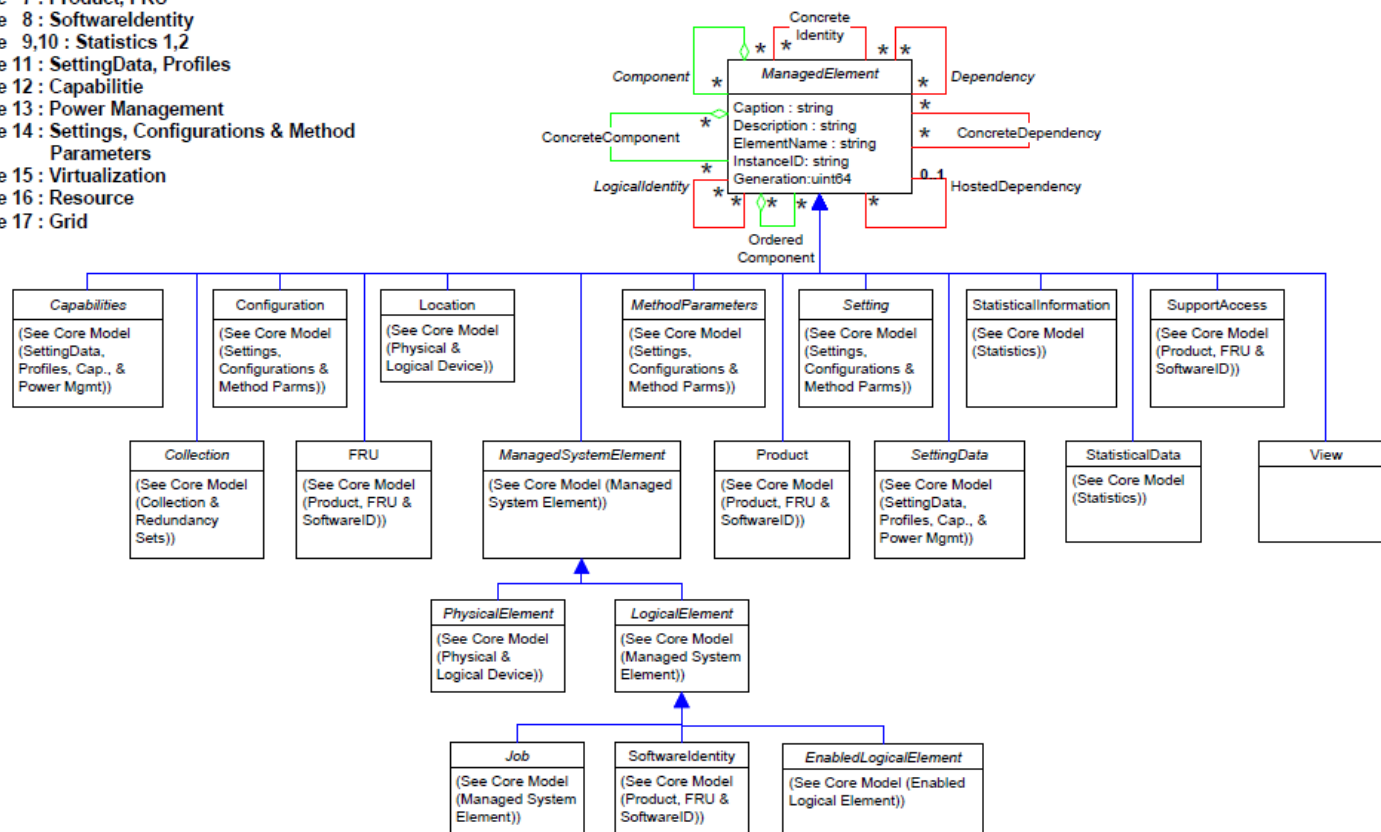
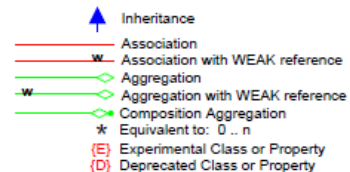
强制露出脓包



Title : Core Specification 2.48.0
 Filename: CIM_Core.vsd
 Author : DMTF SysDev-wg
 Date : 7 Oct 2016

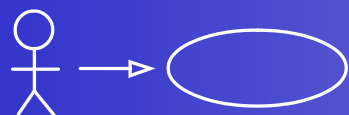
Page 1 : Overview
 Page 2 : ManagedSystemElement
 Page 3 : EnabledLogicalElement
 Page 4 : Physical Elements and Logical
 Devices (Including StorageExtents)
 Page 5 : Collections/RedundancySet
 Page 6 : Redundant Groups
 Page 7 : Product, FRU
 Page 8 : SoftwareIdentity
 Page 9,10 : Statistics 1,2
 Page 11 : SettingData, Profiles
 Page 12 : Capabilities
 Page 13 : Power Management
 Page 14 : Settings, Configurations & Method
 Parameters
 Page 15 : Virtualization
 Page 16 : Resource
 Page 17 : Grid

Page 18 : Opaque Data
 Page 19 : Fingerprint & Launch
 Page 20 : Views
 Page 21,22 : Association Hierarchy 1,2
 Page 23,24 : Dependency Hierarchy 1,2
 Page 25 : Aggregation Hierarchy
 Page 26 : Abstract Associations

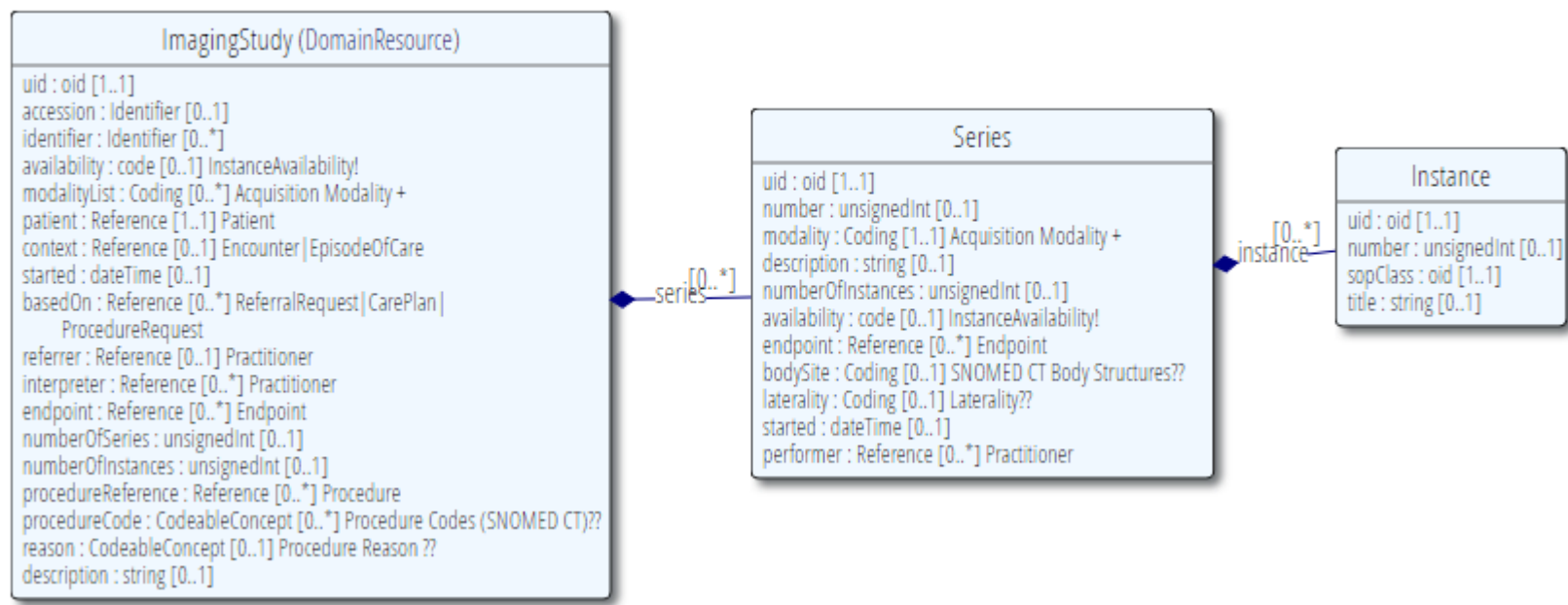


通用信息模型 (CIM)

制造业



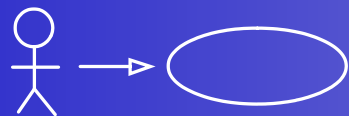
UML Diagram (Legend)

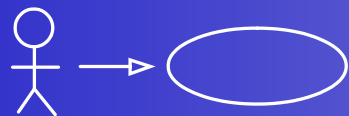


FHIR

快速医疗互操作性资源

医疗卫生





Unified Modeling Language (UML) Version 2.5

<http://www.omg.org/spec/UML/2.5/>

ISO/IEC 19505-2:2012

Information technology -- Object Management Group
Unified Modeling Language (OMG UML) -- Part 2:
Superstructure

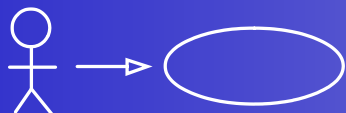
www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=52854

当前页面：国家标准频道首页 > 搜索 > 标准数据库分类检索 > 中国标准文献分类法

根据您所提交的搜索条件“统一建模语言”的搜索结果

- GB/T 28174.4-2011
[统一建模语言\(UML\) 第4部分：图交换](#)
- GB/T 28174.3-2011
[统一建模语言\(UML\) 第3部分：对象约束语言\(OCL\)](#)
- GB/T 28174.2-2011
[统一建模语言\(UML\) 第2部分：上层结构](#)
- GB/T 28174.1-2011
[统一建模语言\(UML\) 第1部分：基础结构](#)

www.chinagb.org/ChineseStandardShow-197675.html



UML元素：14种图

➤ 用例图

➤ 类图

➤ 对象图

➤ 组件图

➤ 部署图

➤ 包图

➤ 组合结构图

➤ 序列图

➤ 通信图

➤ 状态机图

➤ 活动图

➤ 交互概述图

➤ 时间图

➤ 扩展机制图

结

构

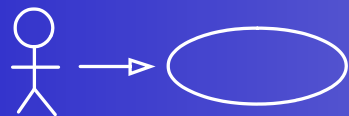
行

为

当前版本：2.5



挑着用的工具箱



建模 workflow

*业务建模

愿景

业务用例图

现状业务序列图

改进业务序列图

*需求

系统用例图

系统用例规约

*分析

分析类图

分析序列图

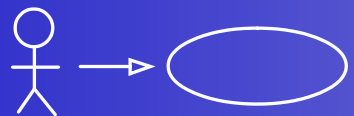
分析状态机图

*设计

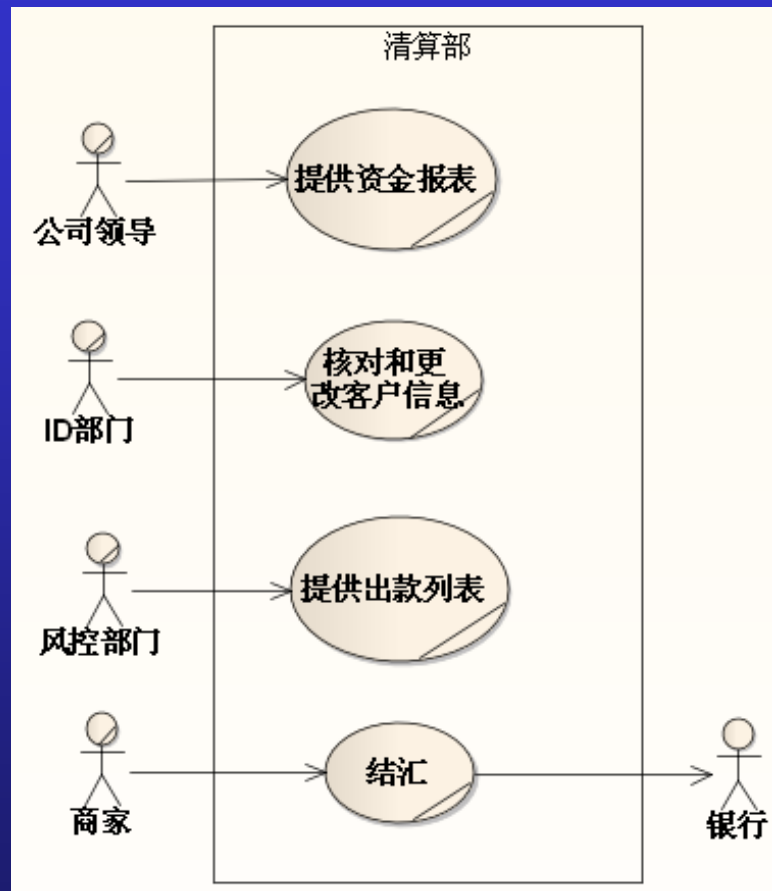
建立数据层

精化业务层

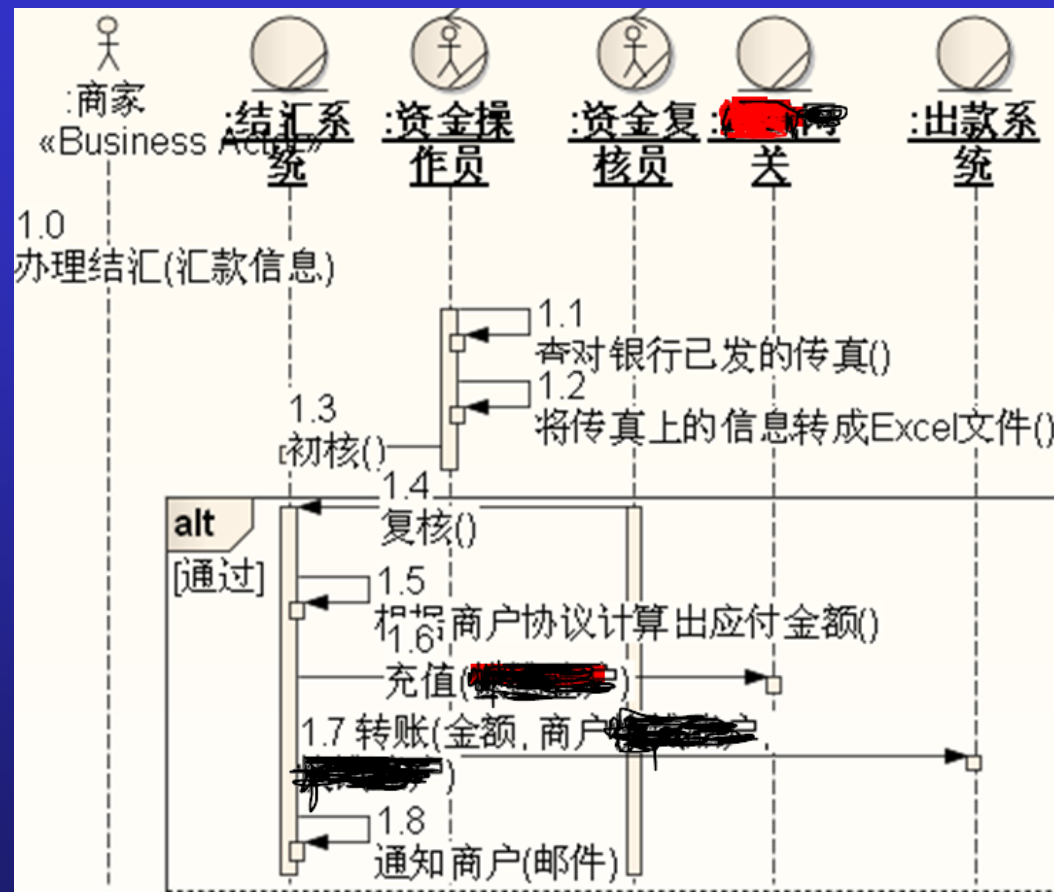
精化表示层



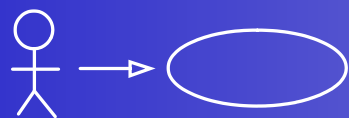
业务建模工件



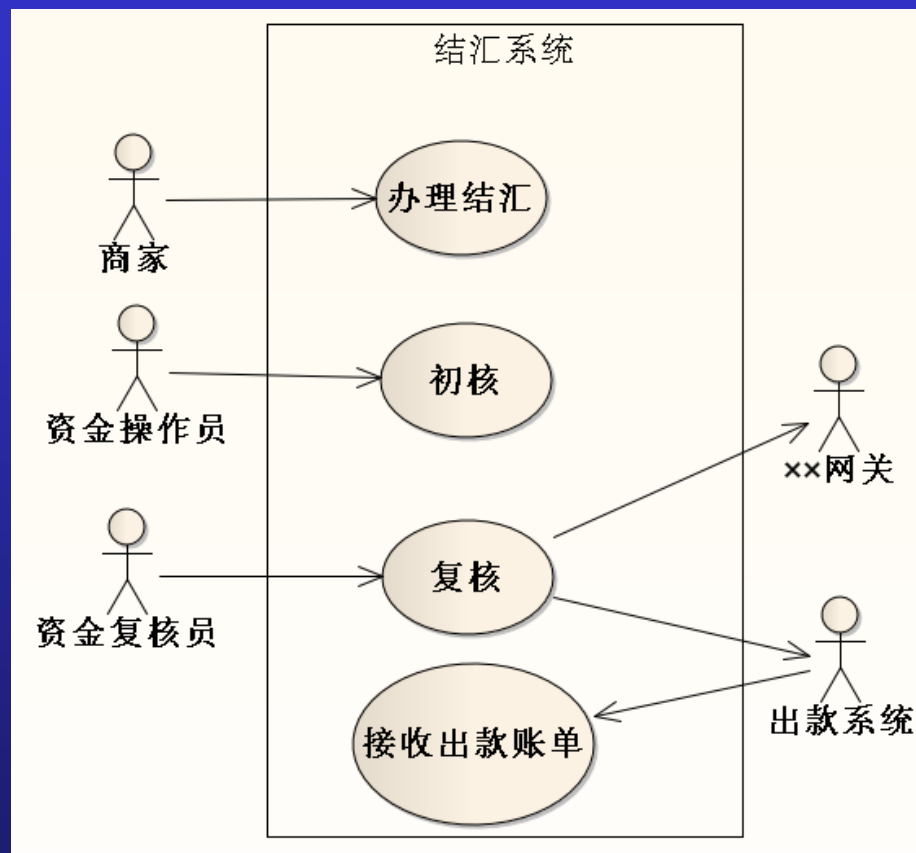
业务用例图



业务序列图



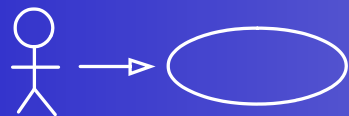
需求工件



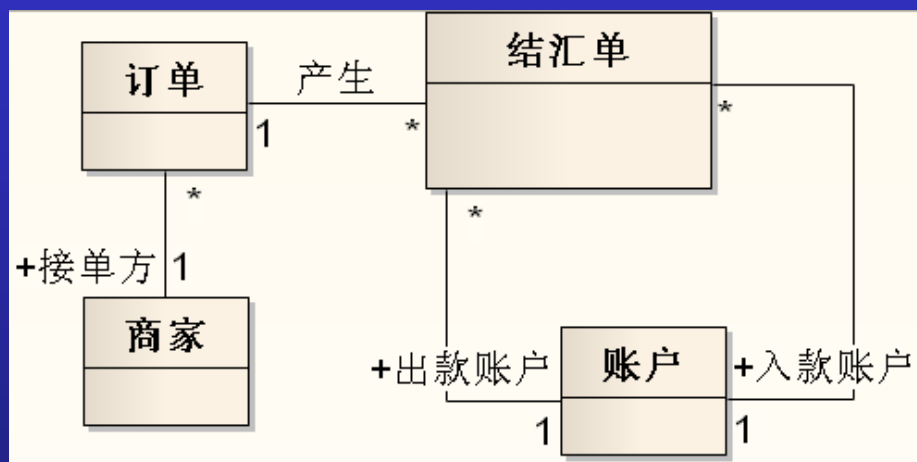
系统用例图

用例编号	UC1	用例名: 提交故障
执行者		
客户 (主)		
前置条件		
客户已获得红苹果权限		
后置条件		
系统已经记录故障信息		
涉众利益 (必须从涉众中来)		
客户 (1) ——担心不能及时解决, 得不到跟踪... 希望方便		
销售代表 (2) ——担心恶意的提交故障, 希望客户能清楚		
基本路径		
1. 客户选择故障类别		
2. 系统显示该类别说明		
3. 客户提交故障的描述文字		
4. 系统验证输入的内容符合要求		
5. 系统保存故障信息		
6. 系统提示故障已保存, 以及要处理销售代表信息		
扩展		
4a. 不符合要求:		
4a1. 系统提示输入内容存在的问题		
5a. 保存失败:		
5a1. 系统提示故障保存失败, 联系谁谁谁		
字段列表		
5. 故障类别+故障描述+客户身份+提交时间		
业务规则		

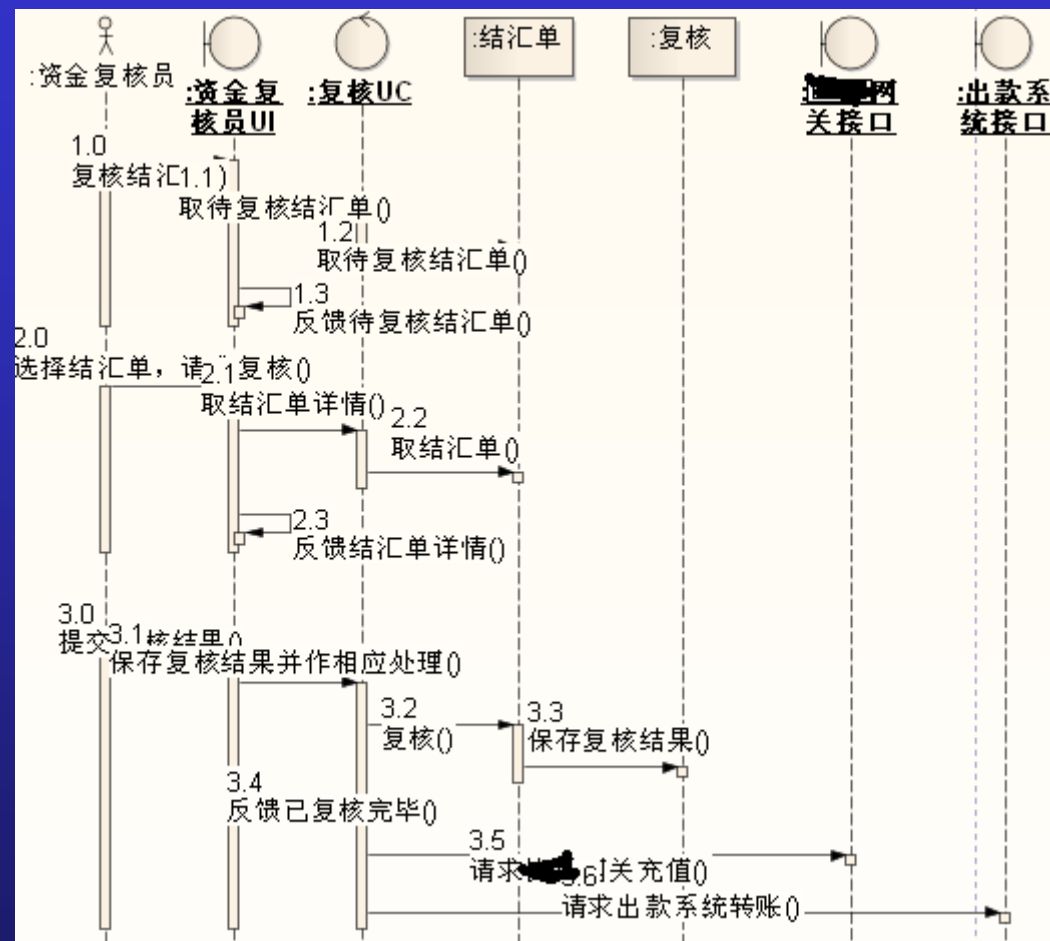
用例规约



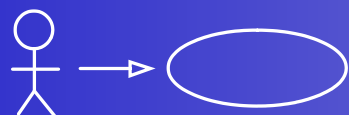
分析工件



分析类图



分析序列图



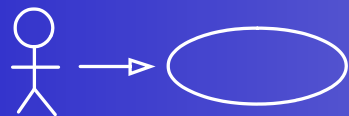
设计工件

```
public Cargo(TrackingId trackingId, RouteSpecification routeSpecification)
{
    if (trackingId == null)
    {
        throw new ArgumentNullException("trackingId");
    }
    if (routeSpecification == null)
    {
        throw new ArgumentNullException("routeSpecification");
    }
    _handlingEvents = new List<HandlingEvent>();

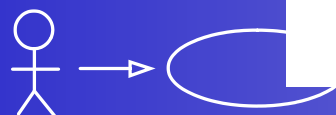
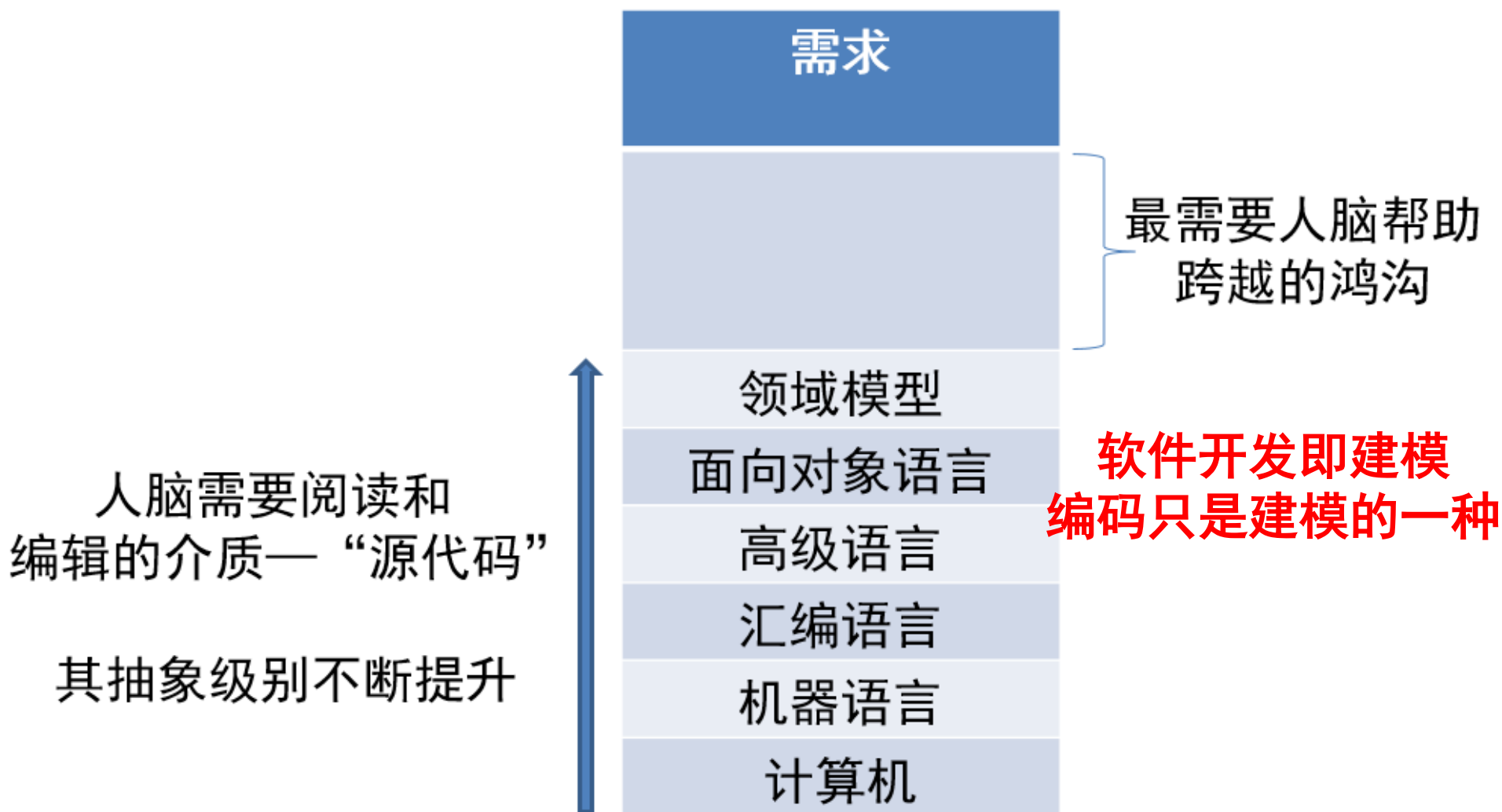
    TrackingId = trackingId;
    _routeSpecification = routeSpecification;
    Delivery delivery = Delivery.DerivedFrom(_routeSpecification, _itinerary, _lastHandlingEvent);
    DomainEvents.Raise(new CargoRegisteredEvent(this, _routeSpecification, delivery));
}

/// <summary>
/// Specifies a new route for this cargo.
/// </summary>
/// <param name="destination">New destination.</param>
public virtual void SpecifyNewRoute(Location.Location destination)
{
    if (destination == null)
```

用文本表达模型（代码就是设计）

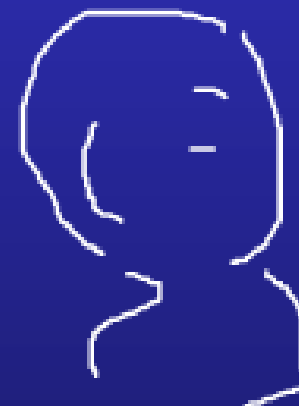


建模 workflow



建模 workflow

编辑工具	称呼
IDE	代码
文档工具	文档

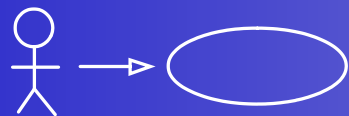


被滥用的“设计”、“文档”



建模 workflow

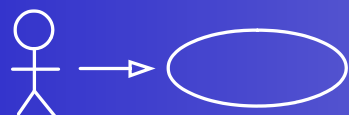
workflow	思考内容	UML元素	推荐使用
业务建模	组织内系统之间	用例图、类图、序列图、活动图	用例图、序列图、类图
需求	系统边界	用例图、序列图、活动图、状态机图	用例图、文本
分析	系统内核心域	类图、序列图、状态机图	类图、序列图、状态机图
设计	系统内域之间	类图、序列图、状态机图、组件图、部署图	用文本表达模型（所谓代码）



UML工具

工具	优点	缺点
Rational Software Architect 9.6	功能强大 企业开发、互联网开发	个头大 价格高
Rational Rhapsody 8.3	功能强大 实时系统开发 设计级调试	价格高 业务建模和需求较弱
Enterprise Architect 13.5	功能较强大 个头小、价格适中 操作方便	代码生成能力尚待提高
StarUML2 2.8.1	不交费也能用	功能稍弱

汇总 <http://www.umlchina.com/Tools/Newindex1.htm>

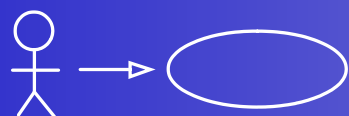


答疑记录

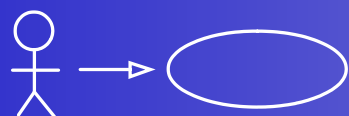
① www.umlchina.com/qa/Index1.htm

业务建模

- [应急办事件管理系统的愿景目标](#) (2017/10/30)
- [业务流程能代替愿景吗](#) (2017/10/1)
- [A组织研究猫的行走路径，目标组织是A组织](#) (2017/9/25)
- [输液药房和配置中心作为业务工人出现在业务序列图中合适吗](#) (2017/8/28)
- [业务对象是否需要将AABB插件显露出来](#) (2017/8/16)
- [先定位目标机构还是机构范围？](#) (2017/7/9)
- ["记录日志"子过程有很多交互，应不应该表达出来？](#) (2017/5/3)
- [《软件方法》第2章开头达明一派歌词的含义](#) (2017/3/29)
- [机场净空障碍物管理的研究对象](#) (2017/2/22)



UMLChina书籍



<http://www.umlchina.com>

建模带来竞争优势

第二版预计
2018年3月出版

软件方法

上册

业务建模和需求



潘加宇 著

清华大学出版社

建模带来竞争优势

预计2018年出版

软件方法（下）

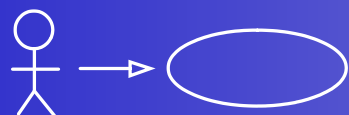
分析和设计



潘加宇 著

www.umlchina.com/book/softmeth.htm

挑一个错误付给您5.12元报酬





斐力庇第斯从马拉松跑回雅典报信，虽然已是满身血迹、精疲力尽，但他知道：没有出现在雅典人民面前，前面的路程都是白费。

学到的知识如果不能最终【用】于您自己的项目之中，也同样是极大的浪费。而这最后一段路最是艰难。

UMLChina 聚焦最后一公里，所提供服务全部与您自己的项目密切结合，帮您走完最艰难的一段路。

简历

1974年生

1989-1994 中国科技大学

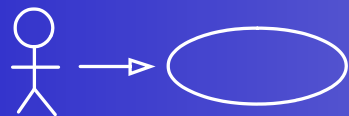
1994-1997 北京大学

1997-1998 清华大学

1998-2002 程序员

2002-至今 UMLChina

270+



答疑方式



QQ
1493943028



微信
umlchinapan



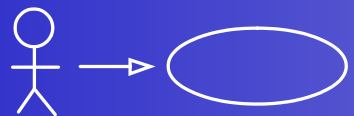
QQ群
647242431



微信公众号
umlchinagroup1



微博
UMLChina潘加宇



答疑方式

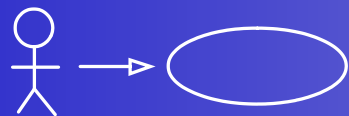


幻灯

<http://www.umlchina.com/training/slide.htm>

课上答对问题有奖金
金额从2.56元到40.96元

请扫我微信以便当场转账
加微信时烦告知姓名



<http://www.umlchina.com>