

测试建模在腾讯MIG的探索

张桓



- 关于我们

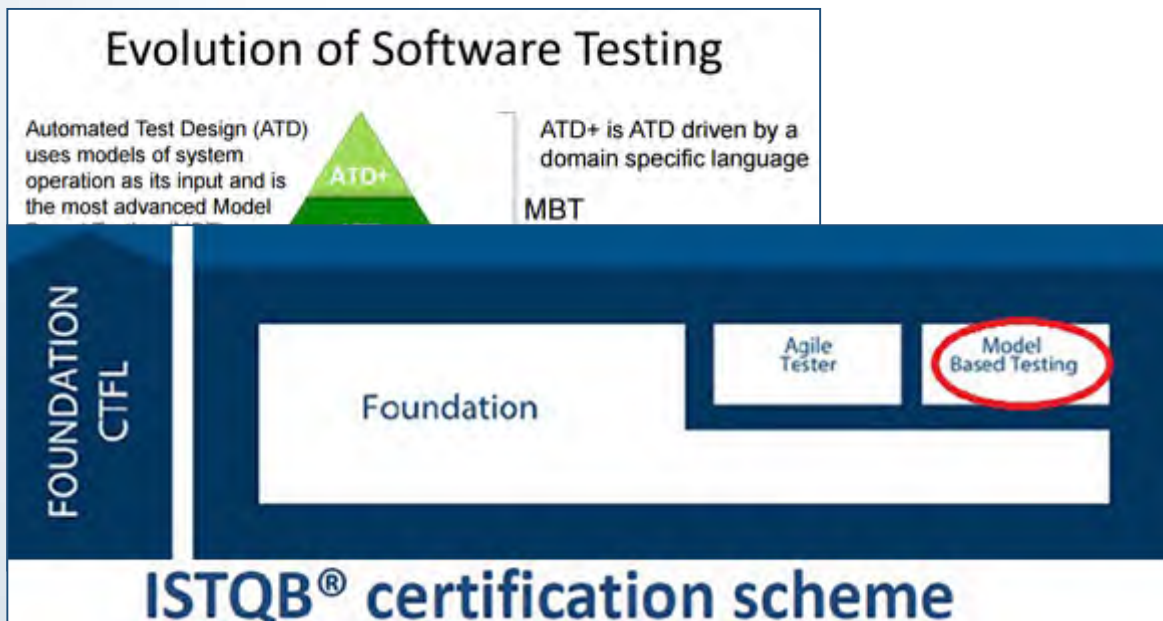


TMQ是什么？腾讯移动品质中心-Tencent Mobile Quality Center 它是腾讯最早专注在移动APP测试的团队，在十余年的时间内承担了近十款业界领先产品测试工作，近七年的android及iOS自动化测试项目经验，为腾讯向移动转型提供了多项质量方案和关键专利。

这里有TMQ专家团给您带来的移动测试技术精华，独家原创！拒绝广告，拒绝忽悠！想知道腾讯多款亿级APP的品质秘密么？来TMQ吧，我们不谈虚的，只谈干货！

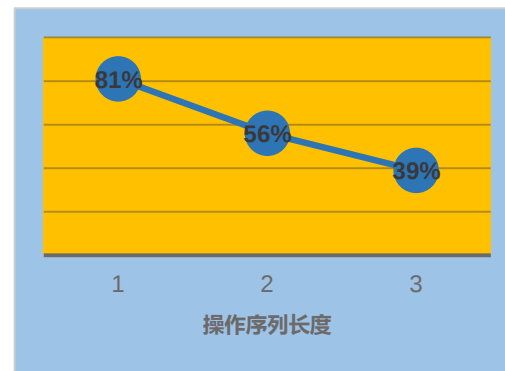
TMQ探索测试建模方法的动机

外因：MBT的应用时机日渐成熟

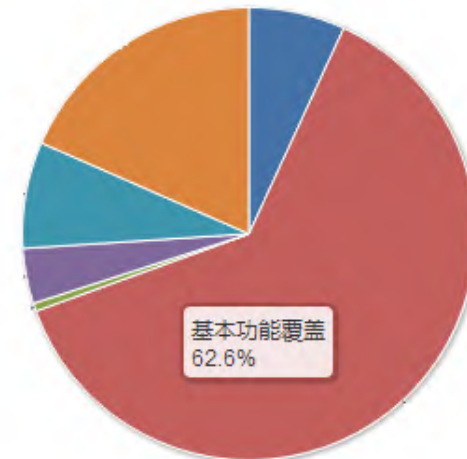


内因：产品质量和测试效率的更高要求

- 更高质量要求 → 更高的测试覆盖率



- 测试左移，防患于未然



MBT：一个可能的有效解决方案

- 基于模型的覆盖率 vs. 测试用例通过率 or 代码行覆盖率
- 模型有助于多角色有效沟通，有助于更加系统全面理解需求，进而于早期发现问题

问题：模型是怎么构建的？

- 产品知识
- 业务领域知识
- 专业知识
- 测试经验
-

需求说明或
真实产品



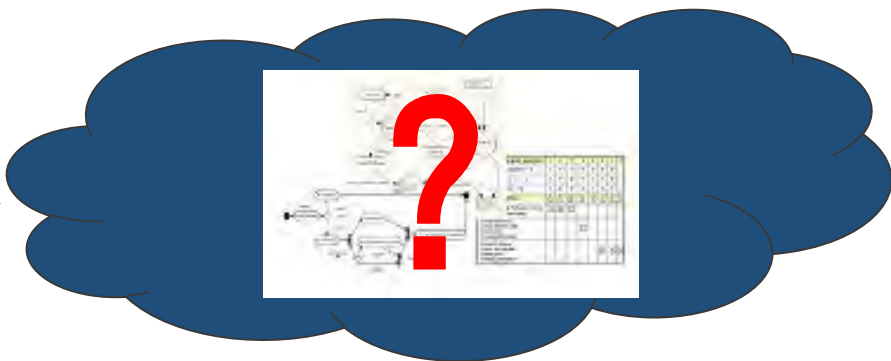
学习
体验



心智模型

外化

测试模型



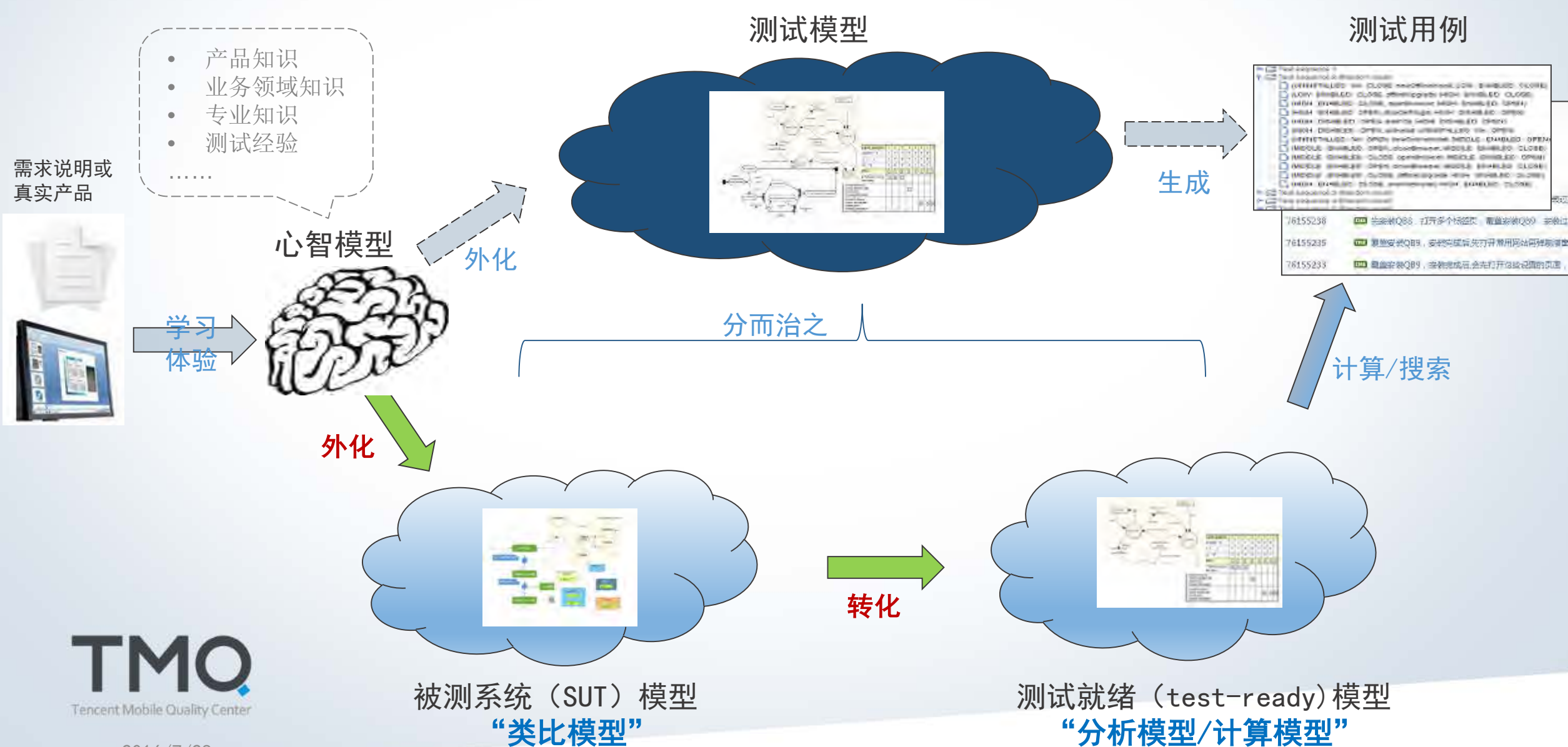
生成

测试用例

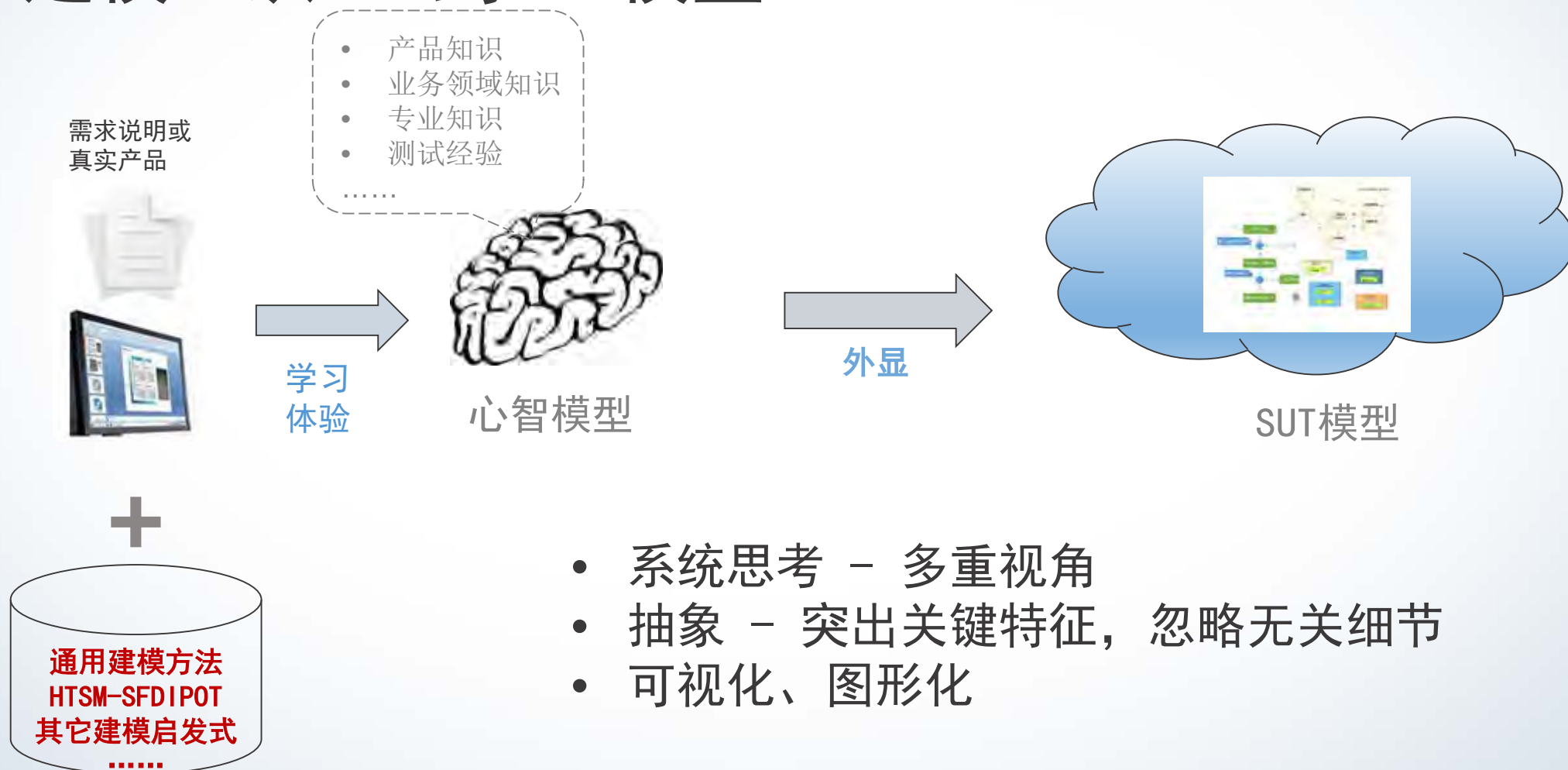


- ? 有没有成熟的建模方法
- ? 用什么语言来建模
- ? 有些什么样的工具
- ? 是否适用于互联网产品测试

方法探索与尝试 - 分而治之



SUT建模：从SUT到SUT模型

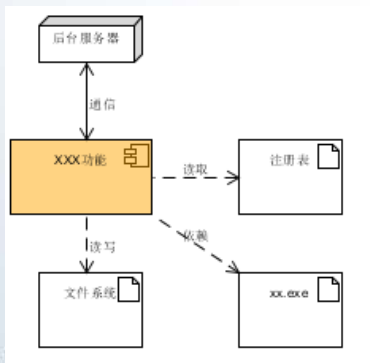


- 系统思考 - 多重视角
- 抽象 - 突出关键特征，忽略无关细节
- 可视化、图形化

SUT建模 - 多视角“观察”

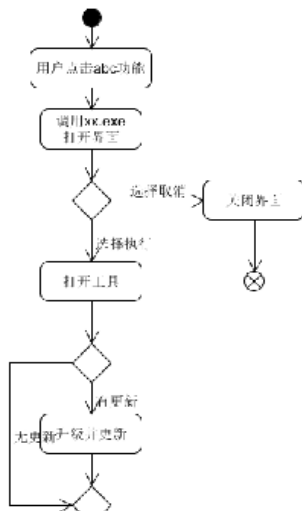
系统上下文

- 描述目标系统与环境和其它系统的依赖关系
- UML或自由格式，框+线条、箭头
- 关注点：外部依赖（环境及其它内外部系统）



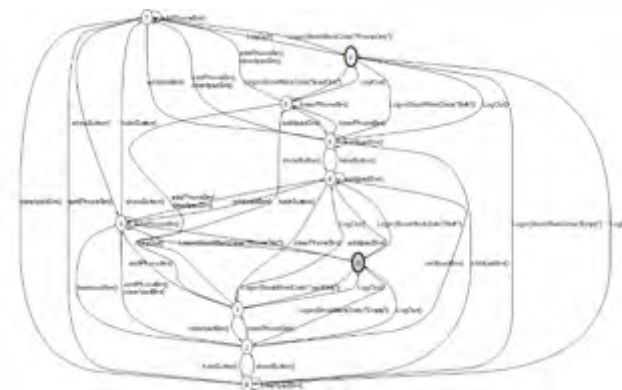
业务流程

- （层次化）描述业务流程或应用场景
- UML活动图+决策表
- 关注点：流程特征、活动间关系，数据流向，业务规则



系统行为

- 描述被测系统行为特征
- 状态机
- 关注点：系统对内外部事件的响应，系统状态的变化

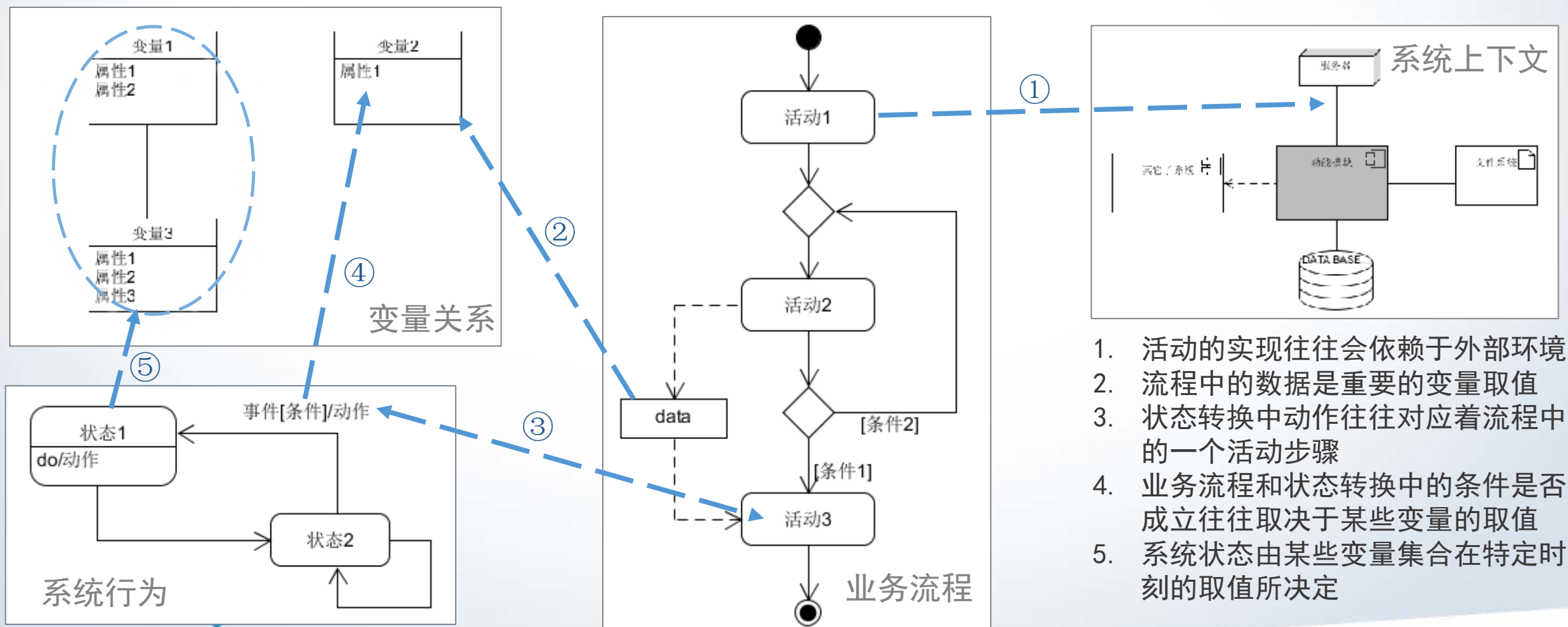


变量关系

- 描述相关变量、变量属性及变量间关系
- UML类图、思维导图等
- 关注点：变量、变量属性、变量间关系、属性及关系的约束限制等



SUT模型各种视图的关系



1. 活动的实现往往会依赖于外部环境
2. 流程中的数据是重要的变量取值
3. 状态转换中动作往往对应着流程中的一个活动步骤
4. 业务流程和状态转换中的条件是否成立往往取决于某些变量的取值
5. 系统状态由某些变量集合在特定时刻的取值所决定

建模语言和工具的选择

- 为什么选择UML？

- 熟悉程度高
- 能满足大部分SUT建模要求
- 通用语言，便于跨角色交流
- 半形式化

- 工具选择

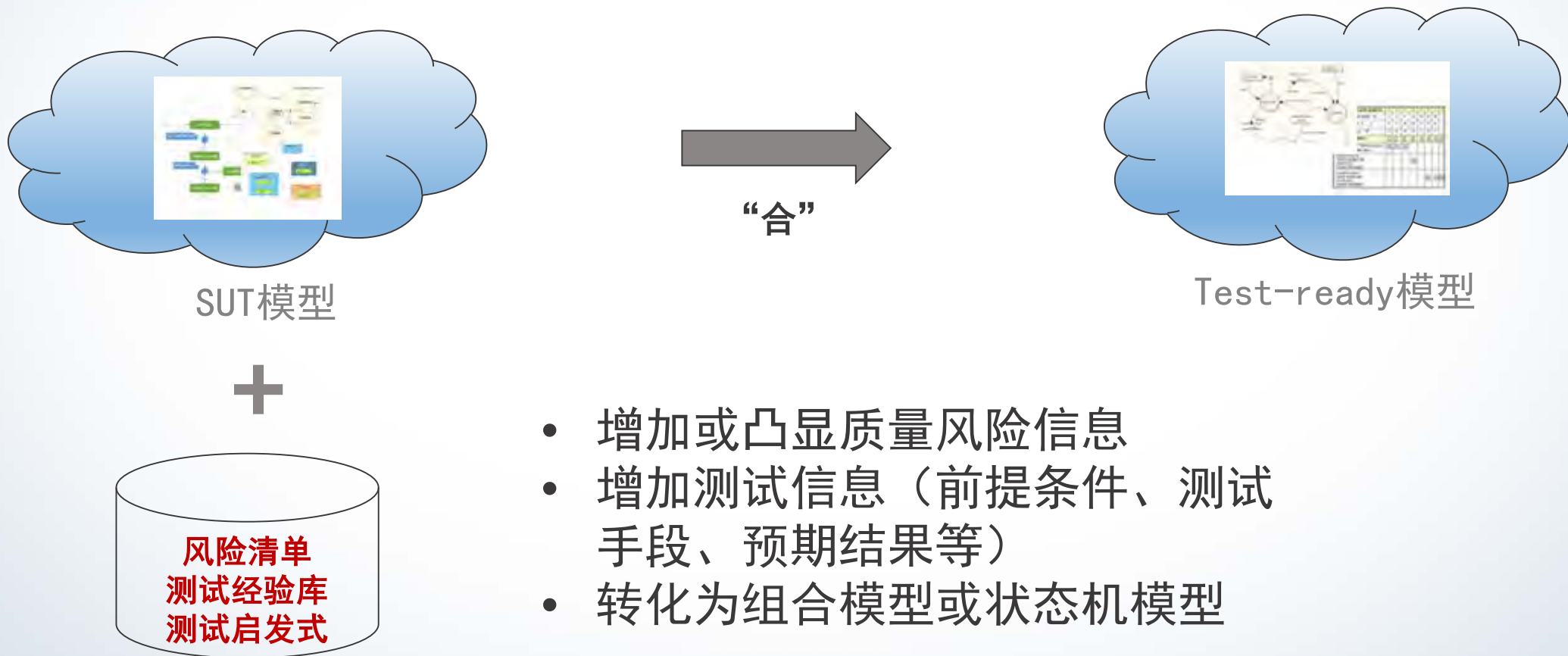
- 符合UML2.0以上规范
- 开源
- 最好能支持XMI转换

- 不足之处：

- 通用语言 vs. 测试专用语言
 - UTP, TDL……
- 过于复杂
 - 选择需要的子集
- 不够用：
 - 业务规则用决策表/树描述
 - 任务间关系借鉴ConcurTaskTree

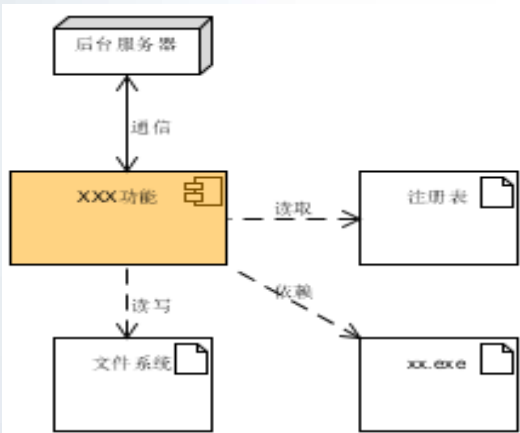
经验：绘制草图时，使用纸笔或白板最高效

TRM建模：从SUT模型到Test-ready模型

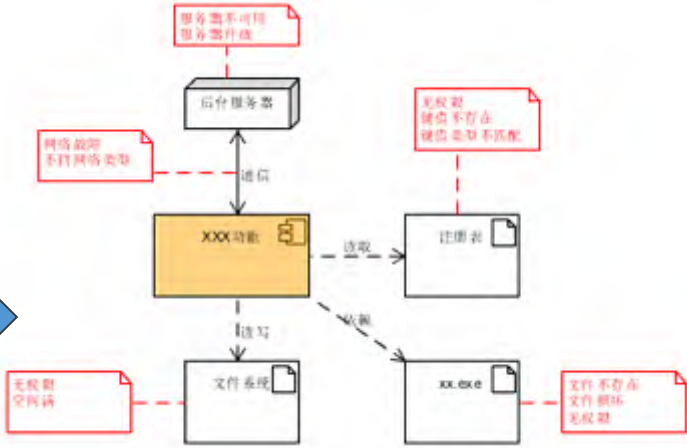


TRM建模 - 基于风险扩展SUT模型（示例一）

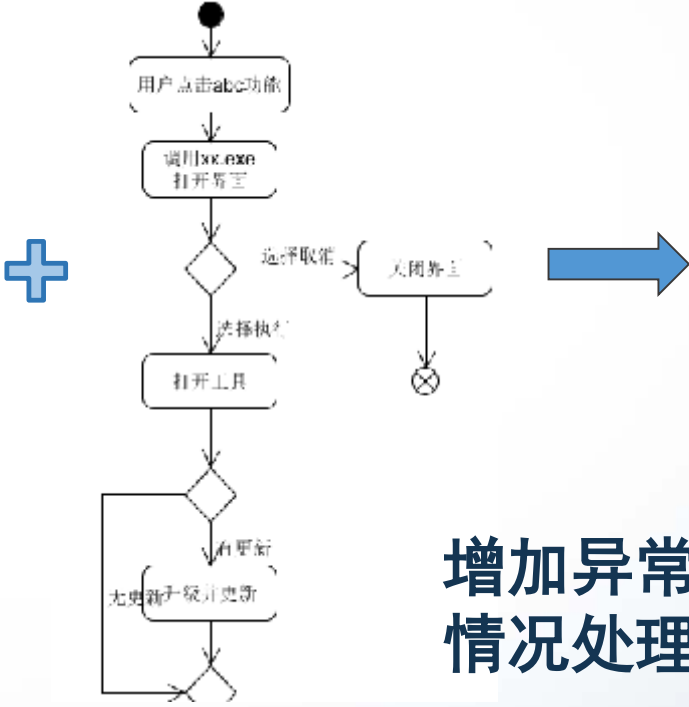
系统上下文图



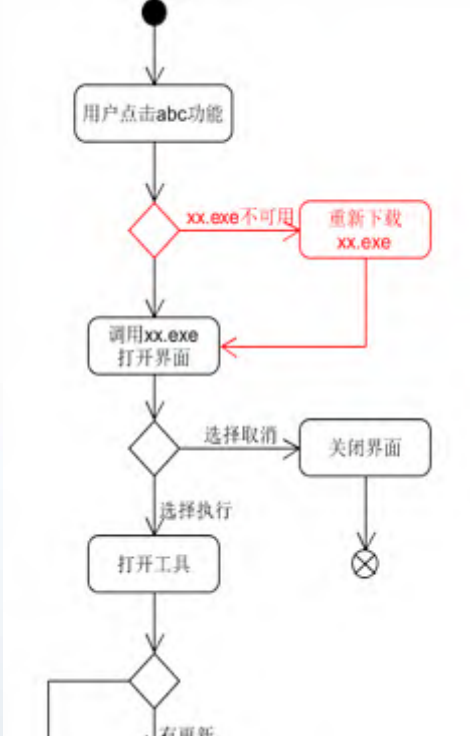
带风险标识的系统上下文图



业务流程图



扩展后的业务流程图



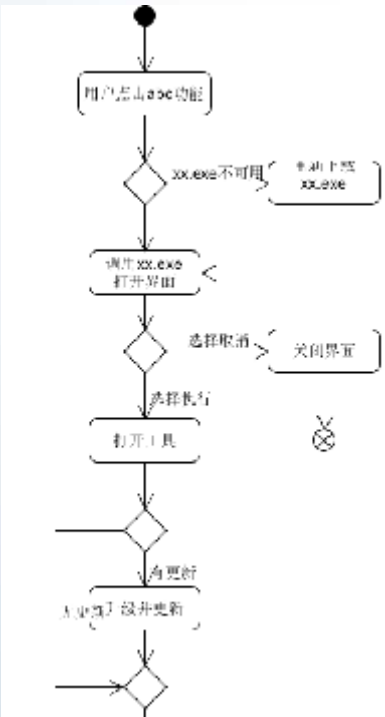
标识风险信息



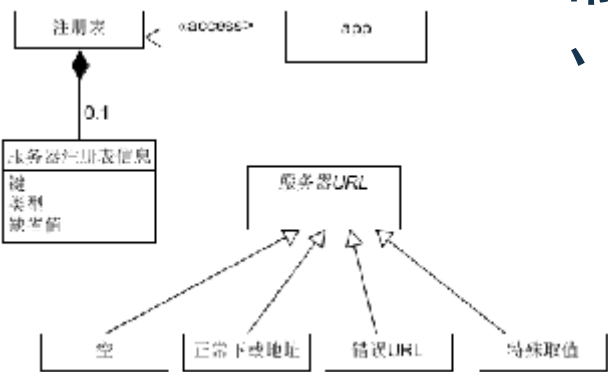
增加异常情况处理

TRM建模 - 基于风险扩展SUT模型（示例二）

业务流程图

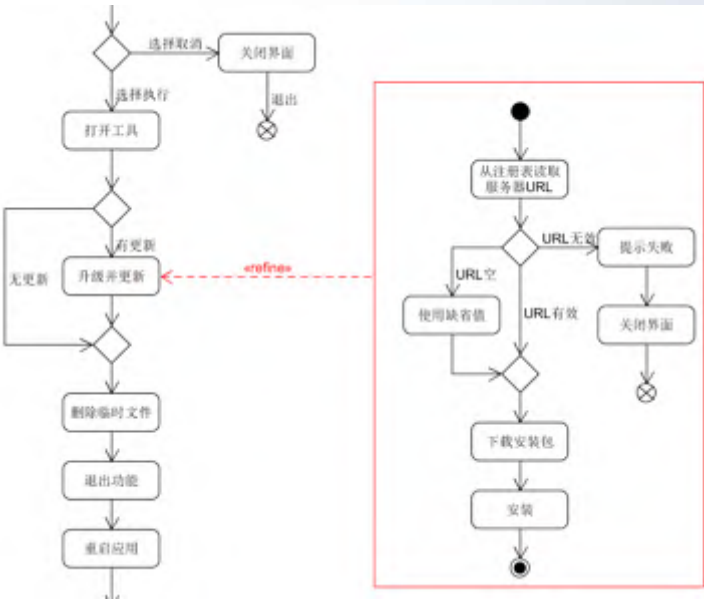


变量关系图



增加特殊值、异常值处理，扩展、完善业务流程

扩展后的业务流程图

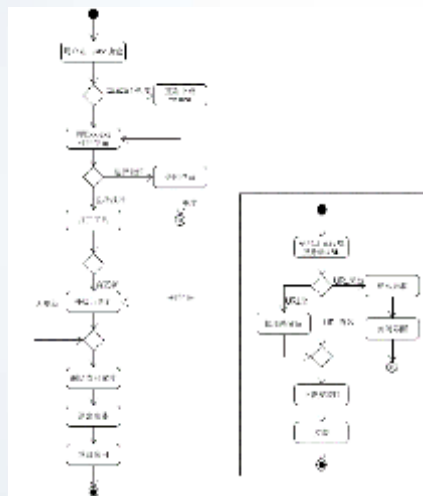


整理变量关系

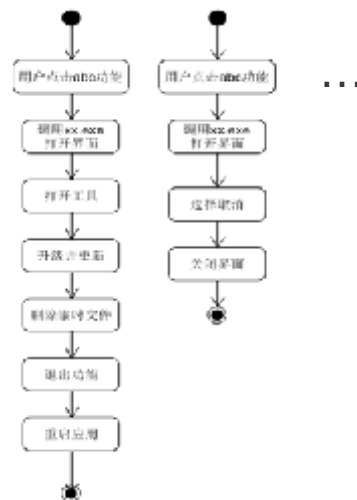
变量信息、域分析结果

TRM建模 - 模型转换 (组合模型)

扩展后的业务流程图



转换为串行流程



分离行为主体
添加测试信息

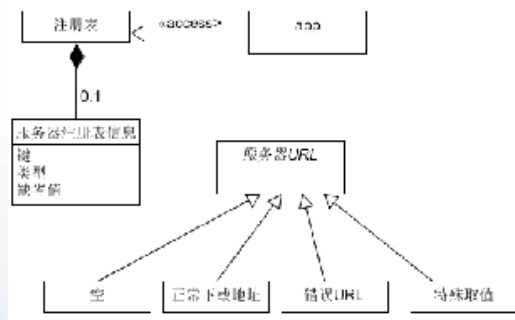


转换为决策表



事件	Case1	Case2	Case3	Case4	Case5	Case6
安装方式	静默安装	√	√	√	√	√
安装包	指定包	√	√	√	√	√
是否静默安装	是	√	√	√	√	√
TargetUrl (注册表)	有	url为: url为下url地址无				√
TargetVersion (注册表)	有	√	√	√	√	version无
升级方式	静默升级 (自动升级)	√	√	√	√	√
升级方式	手动升级	√	√	√	√	√
升级方式	静默升级 (手动静默升级方式)	√	√	√	√	√
升级方式	静默升级 (手动静默升级方式)	√	√	√	√	√
升级方式	静默升级 (手动静默升级方式)	√	√	√	√	√

(借助PICT、因果图等测试设计工具和方法)



变量关系图



遍历算法

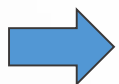
经验库

启发式

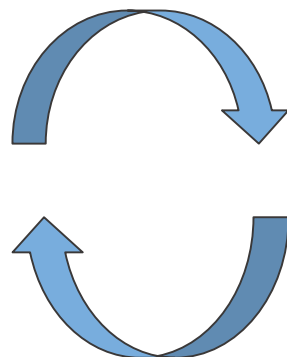
TRM建模 - 模型转换 (状态机模型)



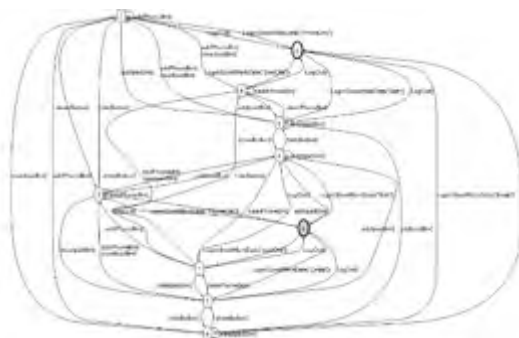
识别系统状态
和行为



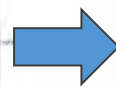
生成状态机模型



反馈



遍历



```
TestSuite(  
  testCase(  
    Login(bookmarkData("empty")),  
    addPhoneNum(),  
    addPadam(),  
    Logout(),  
    Login(bookmarkData("PhoneOnly")),  
    hideButton(),  
    clearPhoneNum(),  
    clearPhoneNum(),  
    showButton(),  
    hideButton(),  
    addPhoneNum(),  
    Logout(),  
    Login(bookmarkData("empty")),  
    Logout(),  
    Login(bookmarkData("PhoneOnly")),  
    addPhoneNum(),  
    addPadam(),  
    clearPhoneNum(),  
    Logout(),  
  )  
)
```

Online

Offline

自动化测试框架
(关键字驱动)

- 定义状态空间
- 定义状态转换条件
- 定义转换动作

- 动作转换为触发操作

难点：MBT工具的选择

- 对工具的期望

- 容易理解和使用，文档和实例丰富
- 开源：成本、可扩展性考量

- 现实

- 基于转换的工具基本上只支持FSM/EFSM
- 缺乏支持业务流程模型的工具

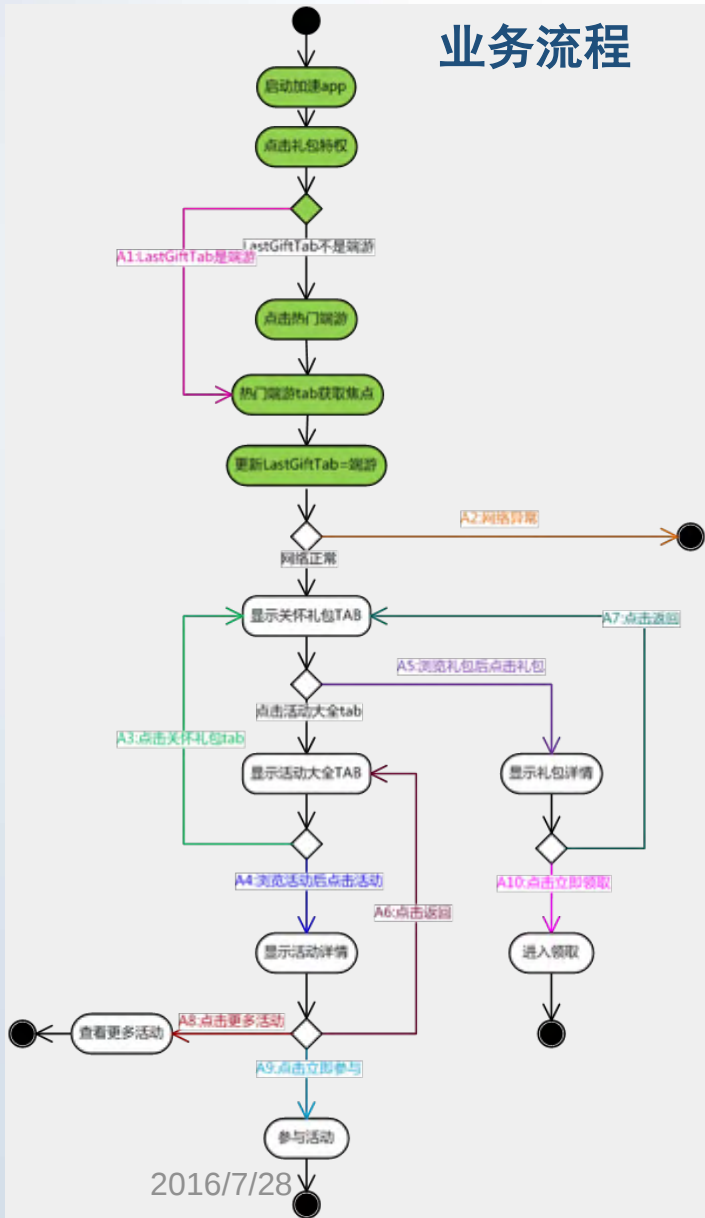
- 实践

- 选用了Nmodel及类似的Pymodel和ModelJunit
- 方法迁就工具：MBT的应用规模和用途受限
- 由业务流程模型生成测试用例还是依赖人工

开发高效好用的工具将成为下一阶段的主要目标

实践案例一：游戏合作项目端游业务

业务流程



2016/7/28

用例no	用例步骤	关注点	串行子流程
TF1	A1,A2	断网情况页面正常提示，且注册表LastGiftTab正确更新	
TF2	A4,A8	正常打开默认浏览器跳转更多活动	
TF3			
TF4			
TF5			
TF6			

变量分析		条
礼包详情	礼包规则行数-有效等价类	[1,5]
		>=6
	礼包规则行数-无效等价类	<1
	礼包规则行数-无效边界值	0
		1
		5
礼包种类个数	礼包种类个数-有效等价类	[1,3]
		[4,6]
		>7
	礼包种类个数-无效等价类	<1
		无异常
		无异常

决策表		TF4 (礼包规则Tab-进入礼包Tab-更新礼包Tab-网络异常)										TF5
用例no	用例步骤	关注点	条件	a	b	c	d	e	f	g	h	i
入口	礼包特权	礼包特权	礼包特权	X								
	礼包特权	礼包特权	礼包特权		X							
	礼包特权	礼包特权	礼包特权			X						
	礼包特权	礼包特权	礼包特权				X					
礼包详情	礼包详情	礼包详情	礼包详情	X						X		
	礼包详情	礼包详情	礼包详情		X							
	礼包详情	礼包详情	礼包详情			X						
	礼包详情	礼包详情	礼包详情				X					
	礼包详情	礼包详情	礼包详情					X				
	礼包详情	礼包详情	礼包详情						X			

实践案例二：闲贝项目

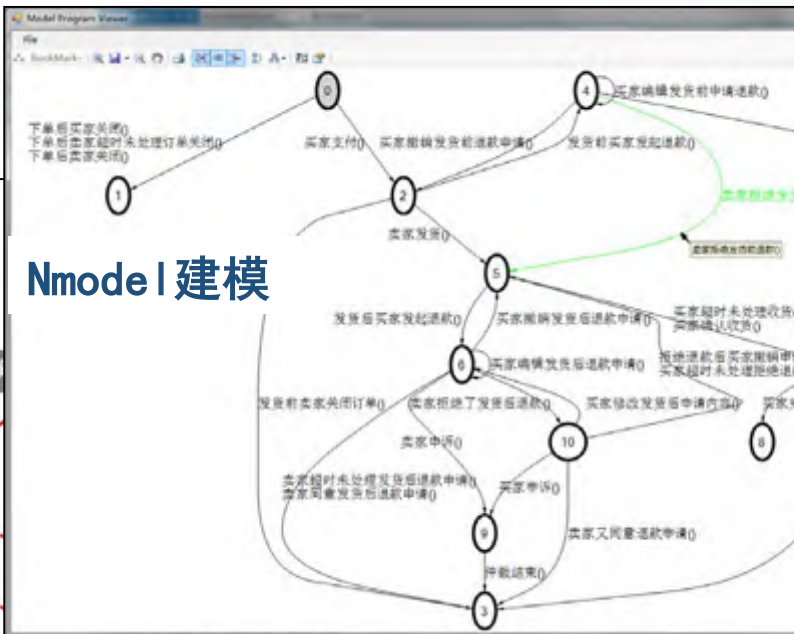
订单状态机

任何状态都可以被管理员置为已冻结，冻结后由管理员来操作此订单的状态

已冻结



NmodeI 建模



```
TestSuite(  
  TestCase(  
    买家支付(),  
    发货前卖家关闭订单()  
  ),  
  TestCase(  
    买家支付(),  
    发货前买家发起退款(),  
    卖家拒绝发货前退款(),  
    买家确认收货()  
  ),  
  TestCase(  
    买家支付(),  
    卖家发货(),  
    发货后买家发起退款(),  
    卖家拒绝了发货后退款(),  
    买家修改发货后申请内容(),  
    卖家超时未处理发货后退款申请()  
  ),  
  TestCase(  
    买家支付(),  
    发货前买家发起退款(),  
    卖家同意发货前退款申请()  
  ),  
  TestCase(  
    买家支付()  
  )  
)
```

测试项	测试点	条件	完善后的用例
未付款	买家在卖家关闭订单后，支付订单	1. 买家拍下物品，未付款 2. 买家/卖家进入订单详情页	1. 卖家关闭订单 2. 买家支付
待发货	买家在卖家发货后，申请退款	1. 买家已支付，待卖家发货 2. 买家/卖家进入订单详情页	1. 卖家完成发货操作 2. 买家申请退款
发货前退款	买家在卖家拒绝退款申请后，撤销退款申请	1. 买家已支付，待卖家发货 2. 买家申请退款 3. 买家/卖家进入退款处理页	1. 卖家拒绝退款申请 2. 买家撤销退款申请
	买家在卖家同意退款申请后，撤销退款申请	1. 买家已支付，待卖家发货 2. 买家申请退款 3. 买家/卖家进入退款处理页	1. 卖家同意退款申请（或卖家超时未处理） 2. 买家撤销退款申请
		1. 买家已支付，待卖家发货	

实践收获

- 来自实践者的声音：

- “加强了对产品的理解……大大提升了测试覆盖率。”
- “需求全面深入了解……覆盖范围全面，发现错误的效果好……维护工作量减少”
- “对测试对象分析的更加透彻……提高测试效率。”
-

- 主要收获：

- 测试“左”移：需求评审环节加强，测试早期参与程度提高
- 以学习为目的的建模方法收到的预期的效果
- 建模学习的同时系统梳理了测试设计方面的知识
- 积累了一些建模案例总结、学习心得和经验文档
- 理解了测试模型的概念和用途，为MBT的引入打下了基础

问题：测试建模如何有效应用于互联网产品测试？

- “方法挺好的，但是太重了，没有时间这样做”
 - 提高熟练程度，使用工具提高效率
 - 结合测试策略决定资源投入（Google AGC）
 - 方法可根据实际情况裁剪
- “使用了建模方法，还是出现重要的遗漏”
 - 主要原因是经验不足，建模视角单一
 - 增加模型评审环节
 - 与探索式测试结合；探索新方法
- “需求一直在变，模型很快就过时”
 - 维护模型比维护测试用例成本低
 - 建模过程促进学习交流
 - 敏捷建模、“推迟决策”

下一步的探索

• 实践方向

- 方法的持续优化改进
- 经验库构建
- 建模工具开发
- 更多方法和工具的尝试
 - 与探索式测试结合
 - MBT + BDD
 -

• 研究方向

- 测试领域专用语言
- 与Usage Model的结合
- 基于实际运行信息的自动建模
- 更多的实用算法

谢谢



腾讯移动品质中心官网：
多年经验总结+纯干货型分享



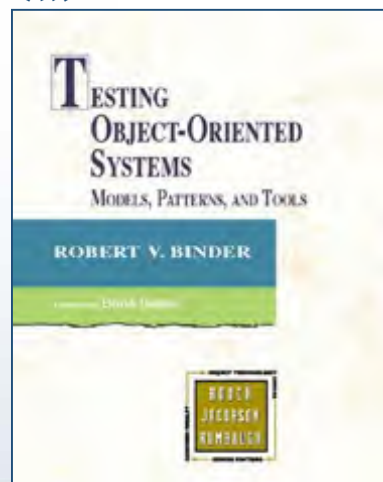
Backup: 模型在测试领域并非新概念

分析学派的观点

测试必须是

- 系统的 (systematic) : 确保每个目标组合被测试
- 聚焦的 (focused) : 聚焦于高风险
- 自动化的 (automated) : 产生和运行大量一致和可重复的测试。

基于**模型**的测试满足上述三点。



测试是基于模型的



上下文驱动学派的观点

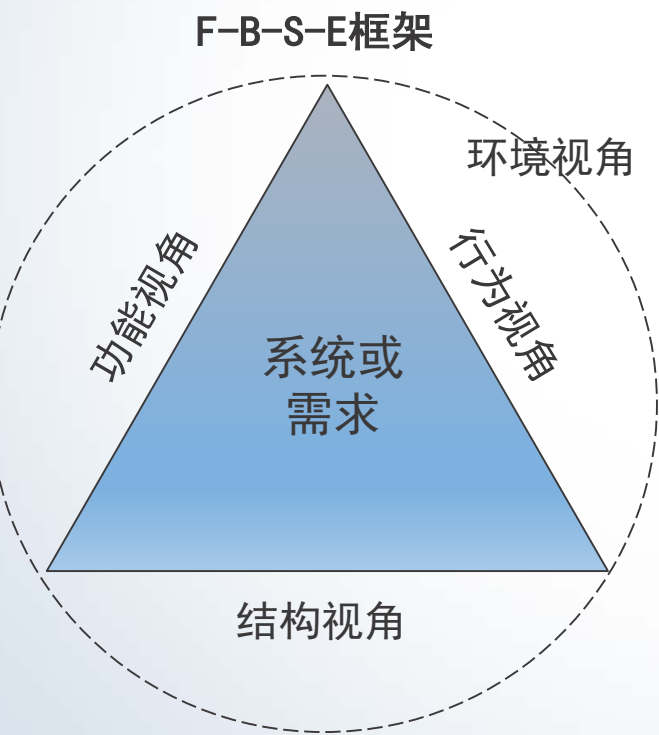
Lesson 25: All testing is based on models

“测试员在设计测试时，头脑中可能会有一个想像的图景，也可能有功能清单或某种图表。测试员会有谁是谁用户、用户关心什么的一些概念。所有这些都是**模型**。”



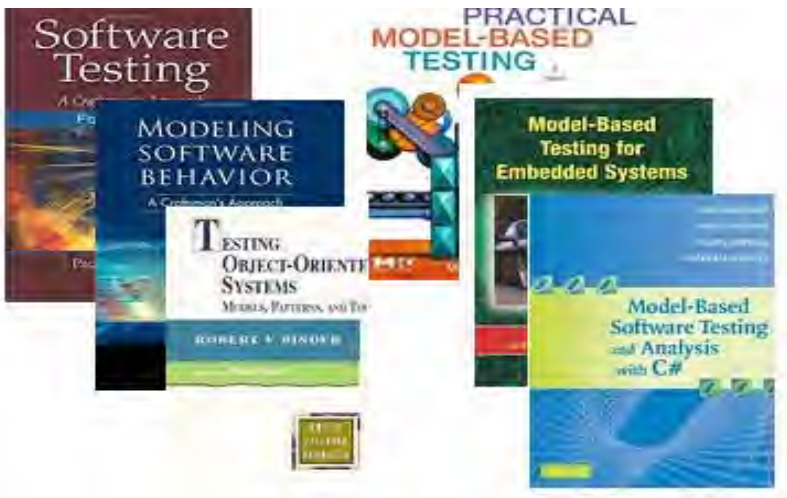
Backup: 方法来源

通用系统分析建模方法



2016/7/28

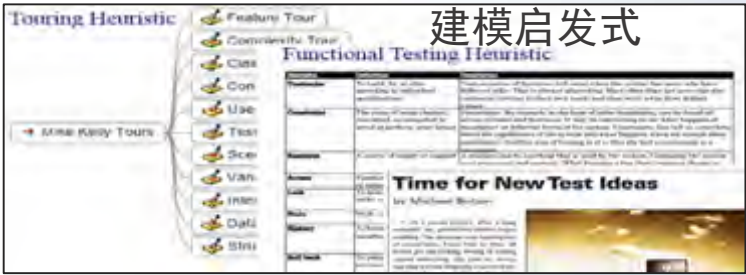
MBT及分析学派



UML与面向对象分析



上下文驱动学派



论文

.....

Backup: 实施经验

- 管理层支持
 - 明确重要性、制定KPI、提供资源
- 成立跨产品的虚拟学习小组
 - 以项目形式运作，定期交流讨论
 - 输出经验文档和培训
- 选择pilot项目试点
- 战术层面：从见效快的点入手
 - 引入需求分析工具消除歧义，完善逻辑结构
 - 从熟悉的内容开始：流程图到活动图，思维导图到决策表
 - 从容易理解的部分开始：测试设计与模型
 - 通过工具在实践中学习：PICT，Nmodel等