

基于模型的接口自动化测试实践

北京新媒传信科技有限公司

薛亚斌 @土司阿哈

缺陷根因分析：测试60%的漏测是因为**异常数据覆**

盖不全（server端的缺陷分析）

数据越界

空指针异常

数据无法遍历



分析：为什么在当前阶段暴露呢？

内部因素：以前主要解决模拟客户端测试，主要基于业务场景测试，而现阶段更多要关注接口本身测试

外部因素：考核方式、合作模式变化



分析：现阶段遇到问题

不同类型测试差异：基于业务场景和基于接口测试差异

测试覆盖：测试过程本身是一个抽样检测的过程？如何用最少的用例覆盖最多的场景

思维方式：人员能力



思考

**自动生成
测试用例 ? ? ?**



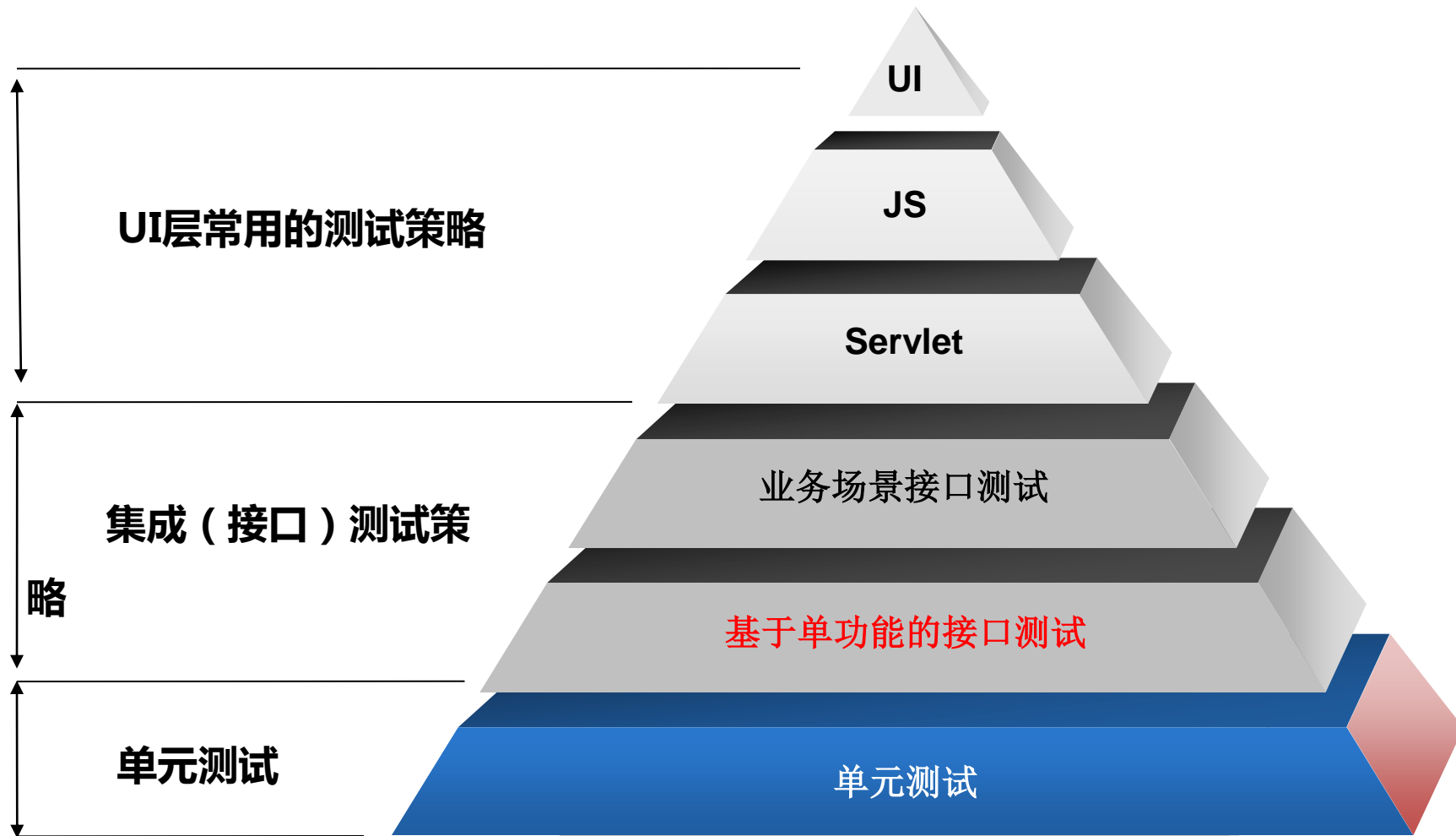
关于接口测试??

接口测试：“组件与组件”、“系统与系统”、“组件与系统”之间的测试?? **测试的重点是要检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。**

策略：是否可以把接口测试分为，基于业务场景测试和基于单接口测试??



基于金字塔模型测试策略再分析



数据分析：数据类型分析

字符类型	范围
Byte	一个字节 (-2^7 — 2^7-1)
Short	两个字节 (-2^{15} — $2^{15}-1$)
Int	四个字节 (-2^{31} — $2^{31}-1$)
Long	八个字节 (-2^{63} — $2^{63}-1$)
Float	四个字节 ($1.4E-45$ ~ $3.4E+38$, $-1.4E-45$ ~ $-3.4E+38$)
Double	八个字节 ($4.9E-324$ ~ $1.7E+308$, $-4.9E-324$ ~ $-1.7E+308$)
Char	一个字节 (Char 0— $2^{16}-1$) , 是unicode编码
Boolean	一个字节 , 其取值只有两个 , true和false

数据分析：业务约束分析

业务数据约束：

在业务数据中存在互斥、组合约束、类型约束等

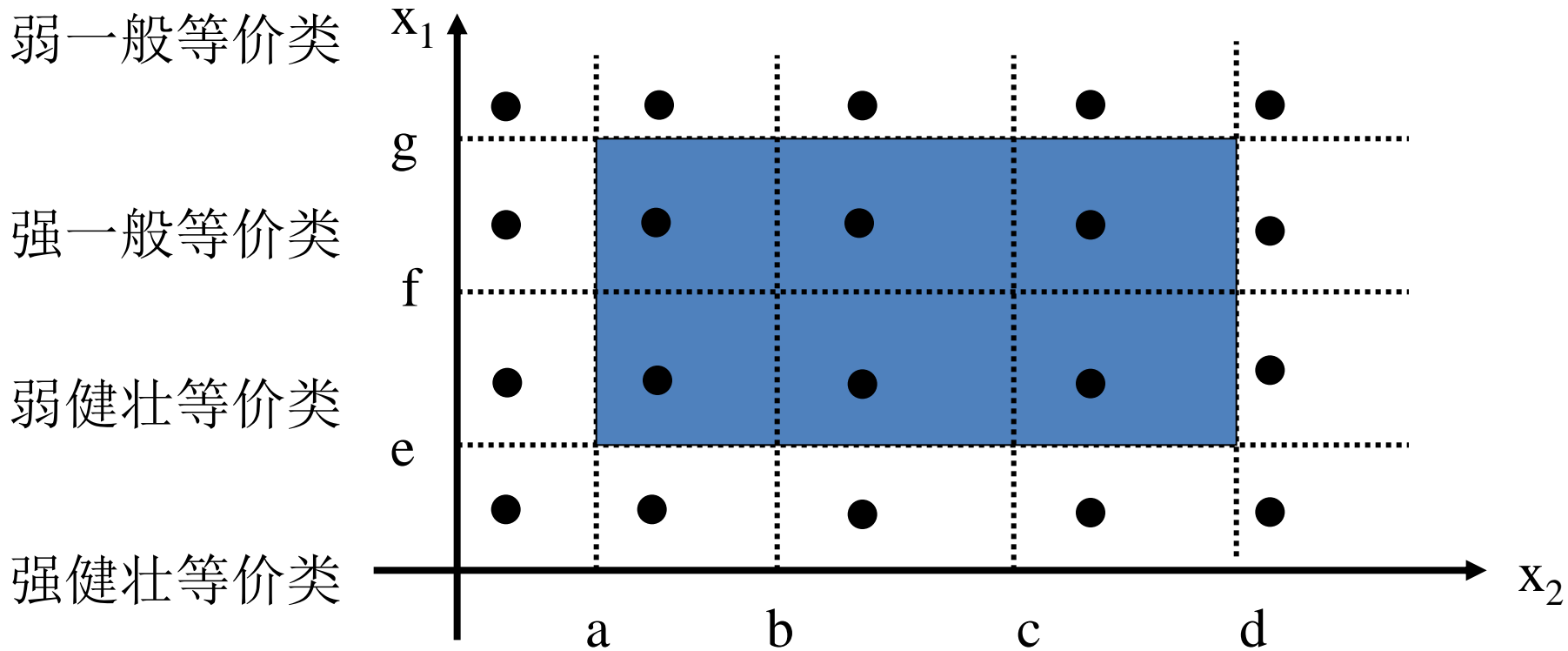
结果约束：

业务数据中定义的数据值对结果的影响



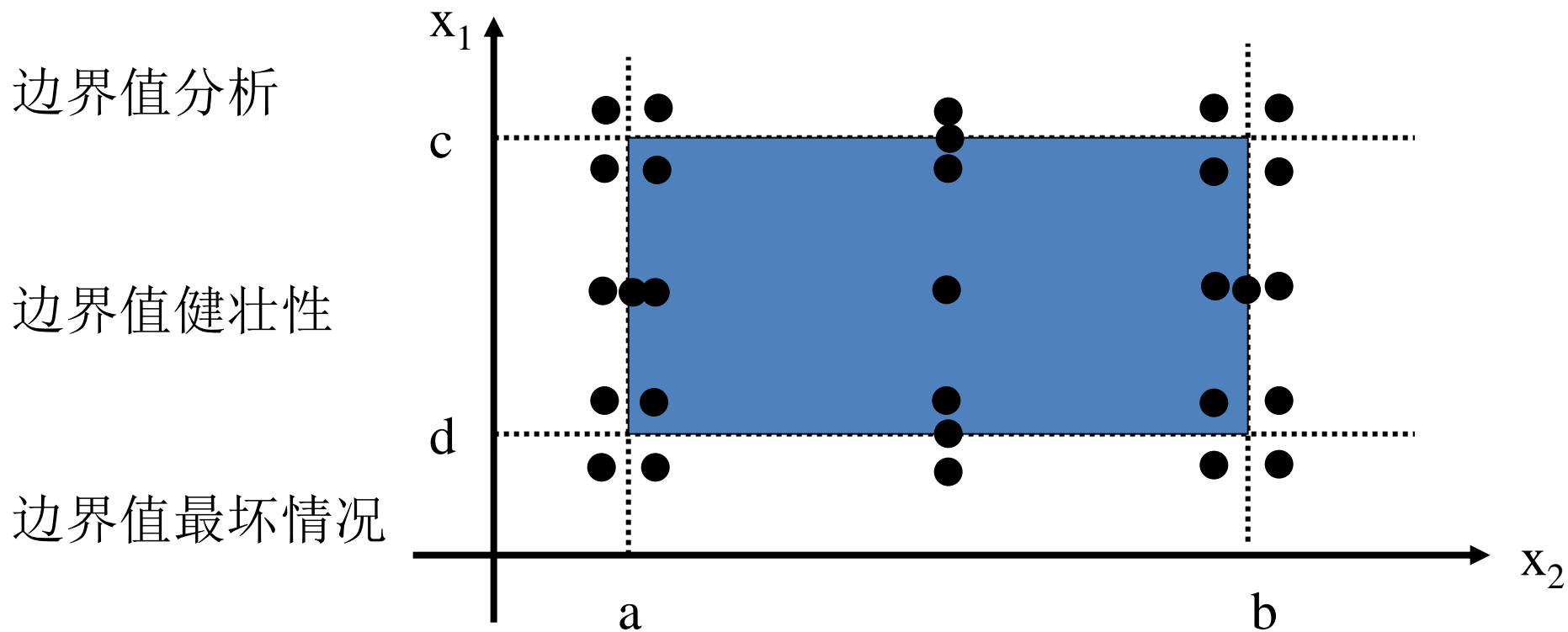
数据分析：数据生成范围分析

基于等价类划分的策略分析



数据分析：数据生成范围分析

基于边界值划分的策略分析



数据分析：数据范围覆盖分析

多输入参数组合的范围分析

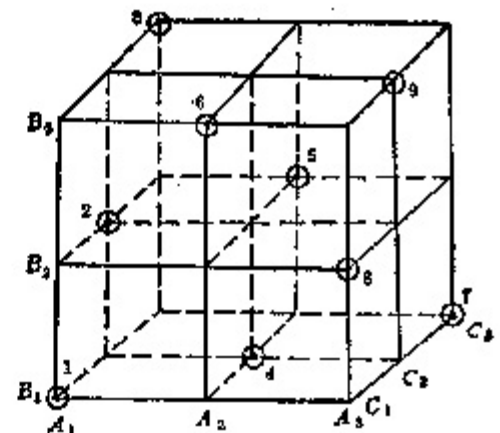
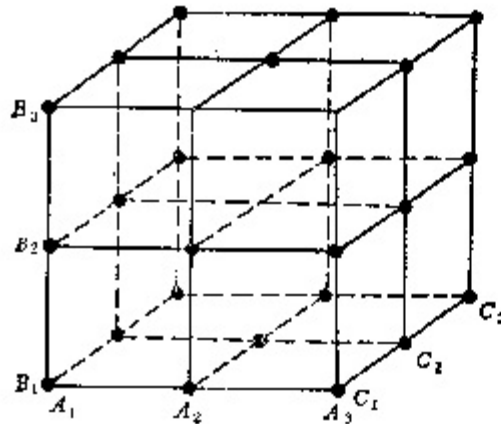
对于**多输入参数组合类的测试方法**目前业界流行

两种方法：

OATS (Orthogonal Array Testing Strategy) ,

即**正交表法**；

Pairwise/All-Pairs Testing , 即**配对测试法**。



数据分析：数据范围覆盖分析

多输入参数组合的范围分析

Pairwise方法是源于对传统的正交分析方法进行优化后得到的产物，它的理论来源于如下的数学统计（如表2-1示）：

表2-1 N维深度所引发缺陷的数目与百分比

维数	浏览器（194Bugs）		服务模块（171Bugs）	
	百分比	累计百分比	百分比	累计百分比
1	28.6	28.6	41.7	41.7
2	47.5	76.1	28.6	70.3
3	18.9	95.0	19.0	89.3
4	2.2	97.2	7.1	96.4
5	2.2	99.4	0.0	96.4
6	0.6	100.0	3.6	100.0

从上表可以看出，约35%的缺陷是单个因子影响下产生，约38%是2个因子交互作用影响产生，约19%是3个因子交互作用影响下产生。累计约73%的缺陷是由1个或2个因子的作用引起的。因此，Pairwise就是基于覆盖所有2因子的交互作用产生的用例集合性价比最高而产生的。

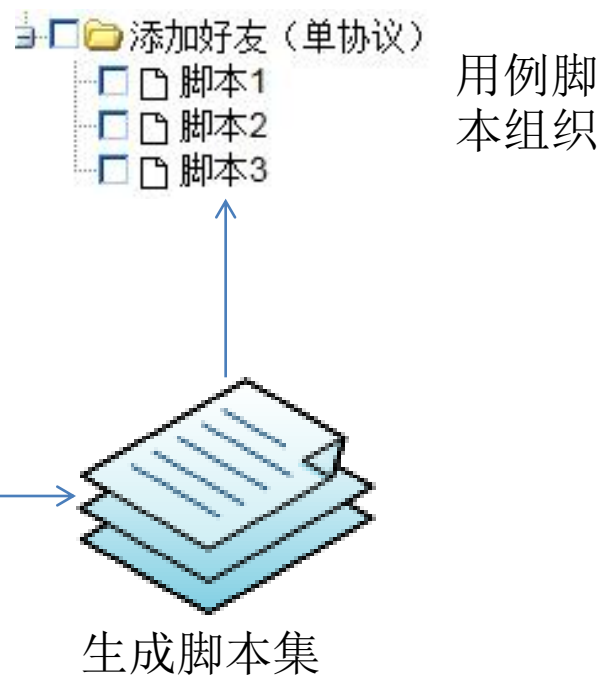
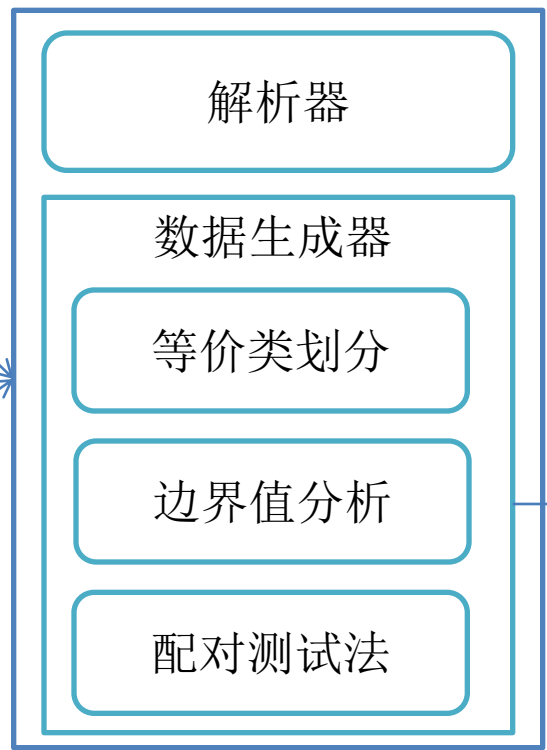
方案：数据智能生成方案

数据自动生成过程

数据规则定义

解析生成

生成测试脚本集



案例：数据规则定义

```

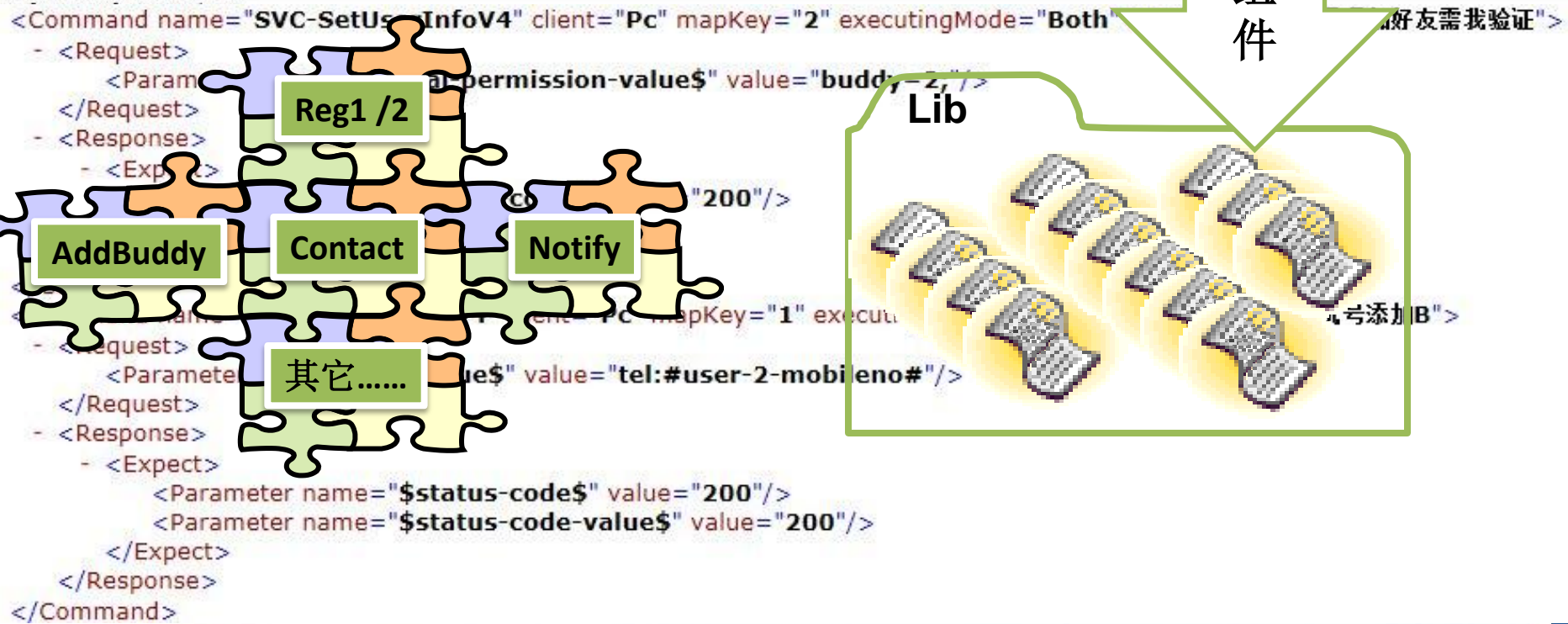
<TestCase>
- <Command author="xueyabin" grade="1" name="ContactRequestV5ReqArgs">
  - <Request>
    <Parameter name="userId" scope="#user-1-userId#" weight="4" not-null="true" type="int"/>
    <Parameter name="result" scope="0,1,2" type="int" verify="true"/>
    <!-- 当result的值为0或2时，不再带有其它属性。 -->
    <Parameter name="localName" scope="(0,16]" weight="2" type="String" verify="true"/>
    <Parameter name="buddyLists" not-null="true" type="String"/>
    <Parameter name="onlineNotify" scope="0,1" type="int" default="0"/>
    <Parameter name="exposeMobileNo" scope="0,1" type="int" default="0"/>
    <Parameter name="exposeName" scope="0,1" type="int" default="0"/>
    <Parameter name="exposeBasicPresence" scope="0,1" type="int" default="1"/>
    <Parameter name="acceptInstantMessage" scope="0,1" type="int" default="1"/>
    <Parameter name="reason" scope="(0,30]" type="String"/>
  </Request>
  - <Response>
    <Parameter name="statusCode" scope="200" type=""/>
  </Response>
  - <Rule>
    <Group order="2" para="userId,localName"/>
    - <IF name="result" operator="in" value="0,2">
      <Then name="localName,exposeName" operator="=" value=",0"/>
      <!-- <Then name="localName,buddyLists,onlineNotify,exposeMobileNo,exposeName,exposeBasicPre
    </IF>
    <!-- <Condition1 rule="[result] in {0,2}"> <Then name="localName,buddyLists,onlineNotify,exposeMol
      operator="="></Then> </Condition1> -->
    </Rule>
</Command>
</TestCase>
  
```

案例：脚本组织方式

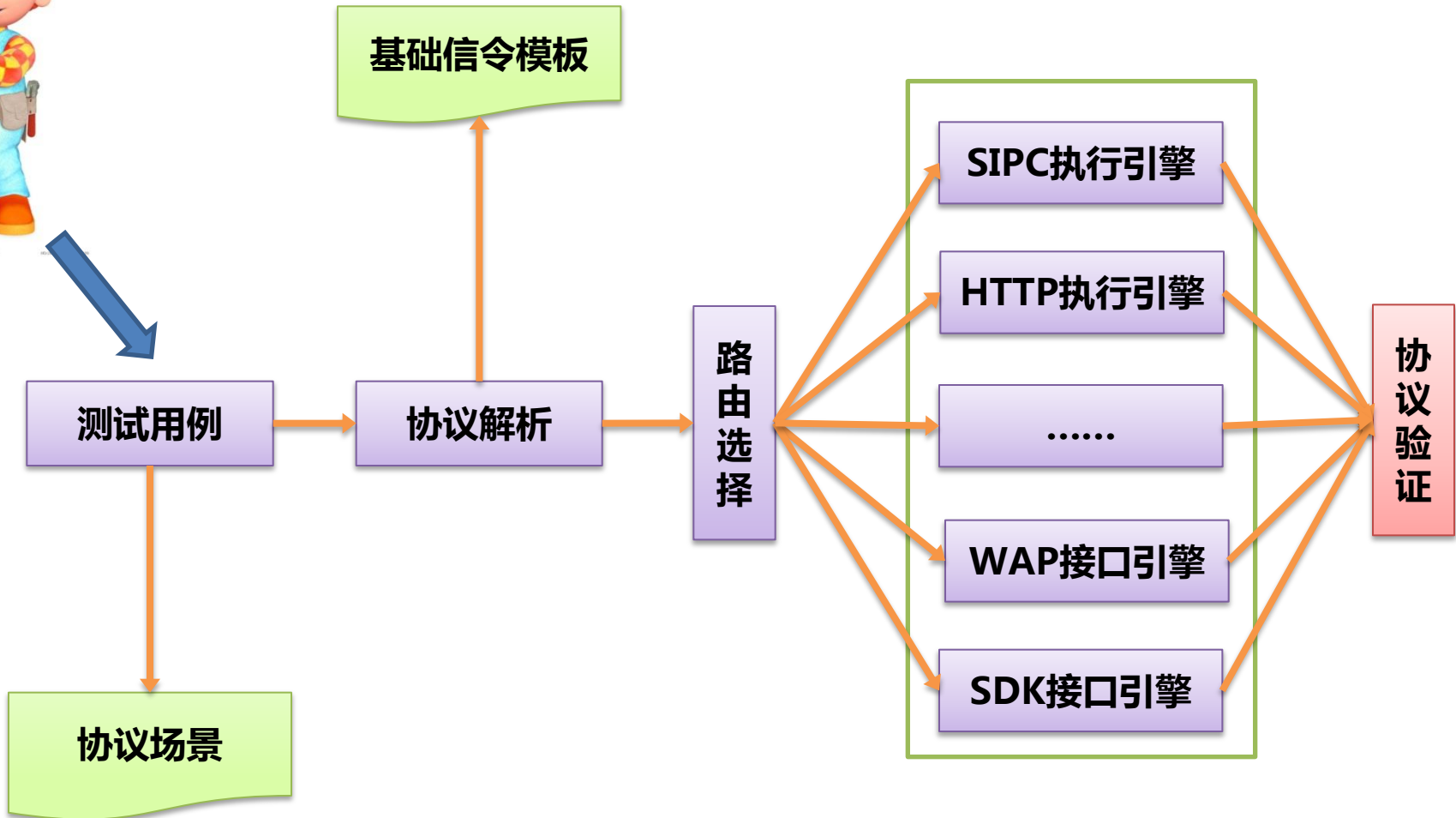
- 添加好友
- addBuddyCase1
- addBuddyCase10
- addBuddyCase11
- addBuddyCase12
- addBuddyCase13
- addBuddyCase14
- addBuddyCase15
- addBuddyCase16
- addBuddyCase17
- addBuddyCase18
- addBuddyCase19
- addBuddyCase2
- addBuddyCase20
- addBuddyCase21
- addBuddyCase3
- addBuddyCase4
- addBuddyCase5
- addBuddyCase6
- addBuddyCase7
- addBuddyCase8
- addBuddyCase9

```
- <Prepared>
  - <Users>
    <User isLogin="1" client="Map" value="1" id="1"/>
    <User isLogin="1" client="Map" value="2" id="2"/>
  </Users>
  - <inil>
    - <Functions>
      <Fun name="Relation" client="Map" type="4" users="1,2"/>
    </Functions>
  </inil>
</Prepared>
+ <Command name="AddBuddyV5ReqArgs" type="Both" mapKey="1" nickName="添加好?
- <Command name="ContactRequestV5ReqArgs" type="Both" mapKey="2" nickName="
  - <Request>
    <Parameter exposeMobileNo="0"/>
    <Parameter result="1"/>
    <Parameter localName="X"/>
    <Parameter acceptInstantMessage="1"/>
    <Parameter exposeBasicPresence="1"/>
    <Parameter exposeName="0"/>
    <Parameter onlineNotify="0"/>
    <Parameter reason="e"/>
    <Parameter userId="#user-1-userId#"/>
  </Request>
  - <Response>
    - <Expect>
      <Parameter name="statusCode" value="200"/>
      <Parameter result="1"/>
      <Parameter localName="X"/>
    </Expect>
  </Response>
</Command>
```


案例：场景级业务生成过程



案例：自动化测试用例执行过程



参考资料

接口自动化平台的实现和策略

下载地址一：<http://vdisk.weibo.com/s/zIYPKCGFIGuBv>

下载地址二：<http://vdisk.weibo.com/s/zIYPKCGFIxh83>

接口测试分析与实施

下载地址一：<http://vdisk.weibo.com/s/zIYPKCGFIxhiY>

分层自动化测试

下载地址：<http://vdisk.weibo.com/s/zIYPKCGFIxh7f>

讨论提问



讨论!