

领域特定语言

MEM入学专业讲座

张荷花

2017.9.4

DSL介绍的内容

- 为什么？（Why）
- 是什么？（What）
- 怎么做？（How）
- 怎么用？（Where）

- 为什么？

场景1:

- CRH5型动车组运行中。。。
- 李工: BPS屏显示轴温报警了, 赶紧停车吧~
- 张工: 唉, 已经出现多次误报导致中途停车了, 严重影响了运输秩序。。。
- 李工: 不停不行啊, 万一有危险呢。上次轴温报警没有停车, 网上还有人说是因为领导追求正点率拒绝停车的。。。
- 张工: 停车检修吧。还好我们总结了大量的故障知识和处理办法~
- 李工: 快把那本厚厚的文档拿来, 让我对照着检修。。。

• 翻阅，查找，理解，处理。。。

序号	故障现象		途中处理办法(机械师)	库内处理办法	备注
1	BPS 屏热 轴报 警红 灯亮	TD、LT 屏显 示温 度正 常	<p>1、机械师接到轴温报警通知,在 TD 屏或 LT 屏上查看轴温.并确认 BPS 故障情况.查看 TD 屏或 LT 屏第二页轴温检测系统和 11 页各轴箱检测状态,确认是否出现有切除、故障或离线现象</p> <p>2、在对应车 QEL 柜断开 70Q01 空开 10 秒后闭合,对热轴主机进行复位。(1、3、6、0 车,每台主机对应 2 辆车)</p> <p>3、复位后若故障未消除,通知司机正常运行,机械师注意监控轴温变化,折返站进行断蓄电池复位。</p> <p>4、若断蓄电池复位后故障不能消除,判定为硬件故障,通知司机正常运行,待车回库处理。</p>	<p>1、复位后故障消除,继续以下检查:</p> <p>1.1、断主控端司机室 70Q01 后,检查 CAN 线插头是否有松动。</p> <p>1.2、若 CAN 线插头无异常,需要对 I/O 模块更换。</p> <p>2、如故障不能消失,重复以上检查,若故障依然存在,需对轴报主机的软件进行刷新(慎用)。</p> <p>3、用轴温检测软件测试系统各部件功能状态。</p>	<p>1、点温位置为轴箱端盖上部中间位置。</p> <p>2、随车机械师将检查内容告知司机</p> <p>3、下同</p>

• 翻阅, 查找, 理解, 处理。。。

1	BPS 屏热 轴报 警红 灯亮	单轴 温度 显示 负数 恒定 值时	<p>1. 查看 TD 屏或 LT 屏第 11 页各轴箱检测状态。若出现切除、故障或离线。做 70Q01 空开复位。</p> <p>2. 故障未消除, 通知司机正常运行, 折返终到进行点温。若温度正常, 通知司机正常运行。</p>	<p>1、断主控端司机室 70Q01 后, 测量传感器阻值 (4 6, 7 9 针); 阻值在 1000 欧姆左右为正常。若发现异常说明传感器故障, 更换传感器。</p> <p>2、若判定传感器正常, 继续检查传感器与分线盒线路是否有异常, 对不良部位进行更换。</p> <p>3、用轴温检测软件测试系统各部件功能状态。</p>	
		多轴 温度 显示 负数 恒定 值时	<p>1、随车机械师做 70Q01 空开复位, 并通知司机正常运行, 到达前方图定站后, 对故障轴所在同一转向架轴端点温。若点温无异常, 通知司机正常运行。</p> <p>2、到达折返站后, 做断蓄电池复位, 若复位后故障没有消除, 对故障轴所在同一转向架轴点温, 若点温正常, 通知司机正常运行。</p>	<p>1、一般为 SUT 盒内部故障, 利用轴温报警软件检测信号传输是否有较大波动, 确认故障 SUT 盒位置, 并且更换故障 SUT 盒</p> <p>2、单车或 2 个车显示负数检查:</p> <p>2.1、优先检查转向架上部分线盒。</p> <p>2.2、若分线盒未发现异常, 故障车对应的热轴检测主机可能故障, 更换主机。</p> <p>3、用轴温检测软件测试系统各部件功能状态。</p>	
		监控 屏轴 温显 示跳 变 (3S 内温 度值	<p>1、若 TD 屏显示某车单轴温度跳变, 随车机械师做 70Q01 空开复位。</p> <p>2、若故障未消除, 无需停车, 随车机械师通知司机继续运行, 加强途中监控。</p> <p>3、动车组到达图定折返站后, 随车机械师通知司机进行断蓄电池复位, 确定 BPS 报警情况。如继续报警, 下车对故障车单车全部</p>	<p>1、用软件检测 5 分钟的温度变量, 判断通讯故障位置。</p> <p>2、检查接线传感器线是否有松动。</p> <p>3、检查 SUT 盒状态, 打开检查 SUT 盒内部。</p> <p>4、用轴温检测软件测试系统各部件功能状态。</p>	

- CRH5型动车组故障处理中。。。
- 李工：唉，现在都是阿尔法狗时代了，不能自动判断误报和处理么？
- 张工：这厚厚的文档，那也得变成“狗”能理解的规则才行啊。。。
- 李工：那有合适的技术工具把这些“故障知识”都有效表示出来么？
- 张工：。。。

场景2:



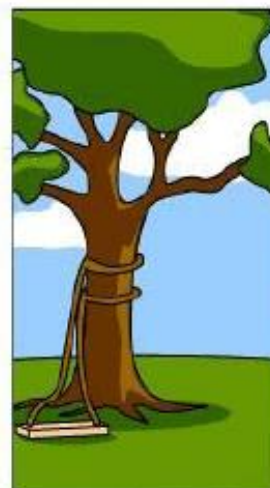
How the customer explained it



How the Project Leader understood it



How the Analyst designed it



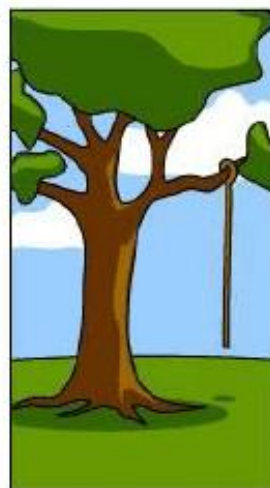
How the Programmer wrote it



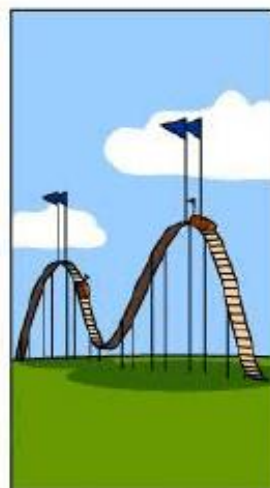
How the Business Consultant described it



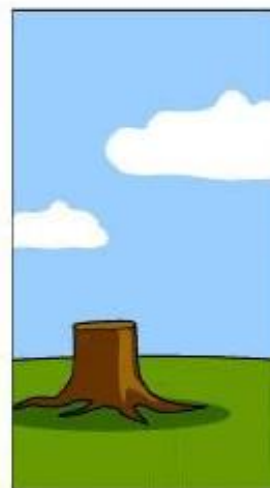
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Gap:人与机器之间、人与人之间

- 神器：领域特定语言DSL
- 改善开发人员与领域专家的交流
 - Validation & Verification (Boehm, 1979)
 - Validation: Are we building the right product?
 - Verification: Are we building the product right?
 - 软件项目最困难、也是最容易导致项目失败的根源就是与客户以及软件用户的交流
 - DSL提供了一种简洁且精确的语言，促进交流
- 有效表示“领域知识”，作为人与机器交流的纽带。

- 是什么？

1. 领域特定语言的概念

- 英文：Domain-specific language
- 缩写：DSL
- 中文翻译：领域语言、特定领域语言、领域专用语言、领域建模语言等
- 问题：对于DSL，大家如何理解？能否举出一些例子？

DSL or not?

中华人民共和国国歌

田汉 词
聂耳 曲

1=G $\frac{2}{4}$

$\underline{5}$ | $\underline{1}$ $\underline{1}$ | $\underline{1}$ $\underline{1}$ $\underline{5}$ $\underline{6}$ $\underline{7}$ | $\underline{1}$ $\underline{1}$ | $\underline{0}$ $\underline{3}$ $\underline{1}$ $\underline{2}$ $\underline{3}$ | $\underline{5}$ $\underline{5}$ $\underline{5}$ |
 起 来! 不 愿 做 奴 隶 的 人 们! 把 我 们 的 血 肉,

 $\underline{3}$ $\underline{3}$ $\underline{1}$ $\underline{3}$ | $\underline{5}$ $\underline{3}$ $\underline{2}$ | $\underline{2}$ - | $\overset{\text{>}}{6}$ $\overset{\text{>}}{5}$ | $\underline{2}$ $\underline{3}$ | $\underline{5}$ $\underline{3}$ $\underline{5}$ | $\underline{3}$ $\underline{2}$ $\underline{3}$ $\underline{1}$ $\underline{2}$ | $\underline{3}$ $\underline{0}$ |
 筑 成 我 们 新 的 长 城! 中 华 民 族 到 了 最 危 险 的 时 候,

 $\underline{5}$ $\underline{6}$ $\underline{1}$ $\underline{1}$ | $\underline{3}$ $\underline{3}$ $\underline{5}$ $\underline{5}$ | $\underline{2}$ $\underline{2}$ $\underline{2}$ $\underline{6}$ $\underline{6}$ | $\underline{2}$ $\underline{5}$ | $\underline{1}$ $\underline{1}$ | $\underline{3}$ $\underline{3}$ | $\underline{5}$ - | $\underline{5}$ - |
 每 个 人 被 迫 着 发 出 最 后 的 吼 声! 起 来! 起 来! 起 来!

 $\underline{1}$ $\underline{3}$ $\underline{5}$ $\underline{5}$ | $\underline{6}$ $\underline{5}$ | $\underline{3}$ $\underline{1}$ $\underline{5}$ $\underline{5}$ $\underline{5}$ | $\underline{3}$ $\underline{0}$ $\underline{1}$ $\underline{0}$ | $\underline{5}$ $\underline{1}$ |
 我 们 万 众 一 心, 冒 着 敌 人 的 炮 火 前 进,

 $\underline{3}$ $\underline{1}$ $\underline{5}$ $\underline{5}$ $\underline{5}$ | $\underline{3}$ $\underline{0}$ $\underline{1}$ $\underline{0}$ | $\overset{\text{>}}{5}$ $\overset{\text{>}}{1}$ | $\overset{\text{>}}{5}$ $\overset{\text{>}}{1}$ | $\overset{\text{>}}{5}$ $\overset{\text{>}}{1}$ | $\overset{\text{>}}{1}$ $\underline{0}$ ||
 冒 着 敌 人 的 炮 火 前 进! 前 进! 前 进! 进!

被遗忘的时光

--蔡琴--

陈宏铭曲
吴曦改编

Andante

The musical score for "被遗忘的时光" (Time Forgotten) is presented in a standard musical notation format. It is in the key of D major (indicated by two sharps) and 4/4 time. The tempo is marked "Andante". The score is written for piano, with a melody in the right hand and a bass line in the left hand. The melody features several triplets and slurs, indicating a flowing, lyrical quality. Dynamic markings include "rit." (ritardando) and "a tempo". The score is arranged by Wu Xi, based on a composition by Chen Hongming.

DSL or not?



- 麦当劳、星巴克：
 - “热巧”
 - “小美式”
 - “低卡”

【然并卵=然而并没有什么卵用】



释义：指一些事物看上去很复杂很高端却没有实质性的效果，或花了时间、精力去做一件事情后却得不到理想的收益。多用来表达无奈、调侃之情。

DSL or not ?



T_EX

L_AT_EX

- 定义
 - 划清边界，澄清概念
 - 不好的定义：a language for a specific domain

DSL的定义

- 一种专注于**某一特定领域**并具有**有限表达能力**的计算机**编程语言**。
 - 计算机编程语言：人来用，可被计算机所执行
 - 语言特性：流畅性。不仅包含独立短语，还应包含其组合方式
 - 有限表达能力：仅包含支持一个领域的最小特性，DSL不能用于构造一个完整的软件系统，而是系统的一个方面
 - 专注于领域：DSL的有限表达能力仅在专注于一个小的领域时才有意义

广为人知的DSL语言

DSL	用途
SQL	用于关系数据库的查询和操作
Ant, Make	用于Build软件系统
YACC, Bison, ANTLR	用于生成语法解析器
HTML	网页标注语言
EXCEL	电子表格处理
VHDL	硬件描述语言
Mathematica	数学计算语言
○ ○ ○	○ ○ ○

DSL的分类

- 文本DSL
 - 外置式DSL
 - 内置式DSL
- 非文本的DSL
 - 领域特定建模语言 DSML (Domain specific modeling language)

外置式DSL（External DSL）

- 有时被称为“小语言”
 - Little language
- 外置式DSL的特点
 - 完全独立于应用的主语言（通用语言如C、Java）
 - 具有自己定制的语法
 - 与通用语言的语法不同
 - 也可使用其他语言的语法如XML
 - 基于外置式DSL语言编写的脚本（程序）通常使用文本解析技术进行解析
- 举例？

- 示例1：SQL语言
 - 用于数据库查询
 - 例： *select * from ct where times between 5 and 7*
 - 例： *select * from ct order by times desc*
- 示例2：正则表达式
 - 用于字符串匹配
 - 例： [A-Za-z] 匹配任意单词字符，不区分大小写
 - 例： \d{3}-\d{8} 匹配具有3位区号的电话号码
 - 在Java中的应用
 - String s = "010-12345678";
 - System.out.println(s.matches("\\d{3}-\\d{8}"));

• 示例3: XML语言

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Copyright w3school.com.cn -->
- <note>
  <to>George</to>
  <from>John</from>
  <heading>Reminder</heading>
  <body>Don't forget the meeting this weekend!</body>
</note>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited with XML Spy v2007 (http://www.altova.com) -->
- <breakfast_menu>
  - <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>two of our famous Belgian Waffles with plenty of real maple syrup</description>
    <calories>650</calories>
  </food>
  + <food>
  + <food>
  - <food>
    <name>French Toast</name>
    <price>$4.50</price>
    <description>thick slices made from our homemade sourdough bread</description>
    <calories>600</calories>
  </food>
  + <food>
</breakfast_menu>
```

内置式DSL（Internal DSL）

- 有时被称为“嵌入式语言”
 - Embedded language
- 内置式语言的特点
 - 顾名思义，内置在某一通用语言中
 - 是通用语言如C的一种特殊写法
 - 用内置式DSL编写的程序事实上也是有效的通用语言程序；以特别的方式用了语言的一个子集
- 示例？
 - Lisp的macro机制支持内置式DSL的定义；
 - Rails是基于Ruby的内置式语言

`(function-name arg1 arg2 arg3)`

Lisp的macro

- 示例：任务表

```
<todo name = "housework">
  <item priority = "high">Clean the hose</item>
  <item priority = "medium">Wash the dishes</item>
  <item priority = "medium">Buy more soap</item>
</todo>
```

S-表达式

```
(todo "housework"
  (item (priority high) "Clean the house")
  (item (priority medium) "Wash the dishes")
  (item (priority medium) "Buy more soap"))
```

- 写一个任务表的管理程序, 把任务表数据存到一组文件里, 当程序启动时, 从文件读取这些数据并显示给用户。

Java：解析XML文件, 从中得出任务表数据, 然后写代码遍历XML树, 再转换为Java的数据结构, 最后再把数据展示给用户

```
(defmacro item (priority note)
  `(block
    (print stdout tab "Priority: " ~(head (tail priority)) endl)
    (print stdout tab "Note: " ~note endl endl)))
```

- Rails

- Ruby是一个通用的编程语言（the programming language used to manipulate the framework）
- Rails是Ruby on Rails的简称，是一个开源框架用于开发基于数据库的Web应用，基于MVC模式实现。
- Rails使用Ruby实现的，Ruby是Rails的宿主语言

Editing the Controller

```
def sayit  
  render :text => "<h2>Hello World!</h2>"  
end
```


基于Java的内置式DSL

- Miss Grant 密室的例子

```
idle
    .actions(unlockDoor, lockPanel)
    .transition(doorClosed).to(active)
    ;

active
    .transition(drawerOpened).to(waitingForLight)
    .transition(lightOn).to(waitingForDrawer)
    ;

waitingForLight
    .transition(lightOn).to(unlockedPanel)
    ;

waitingForDrawer
    .transition(drawerOpened).to(unlockedPanel)
    ;

unlockedPanel
```

- 金融产品交易的例子

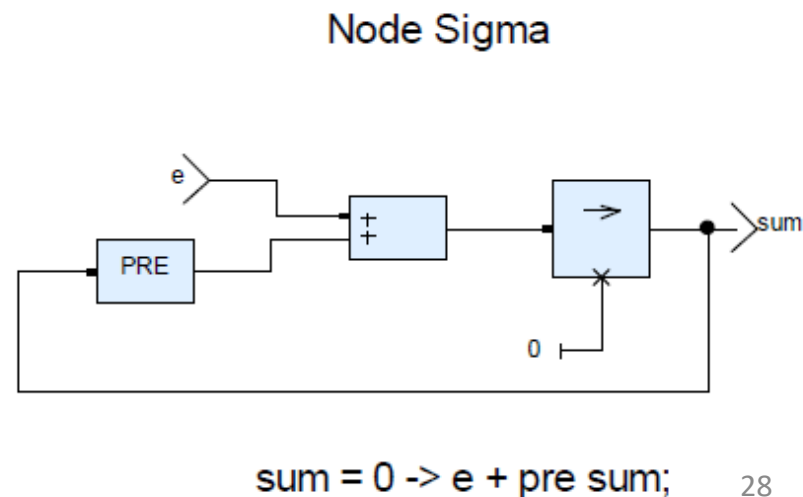
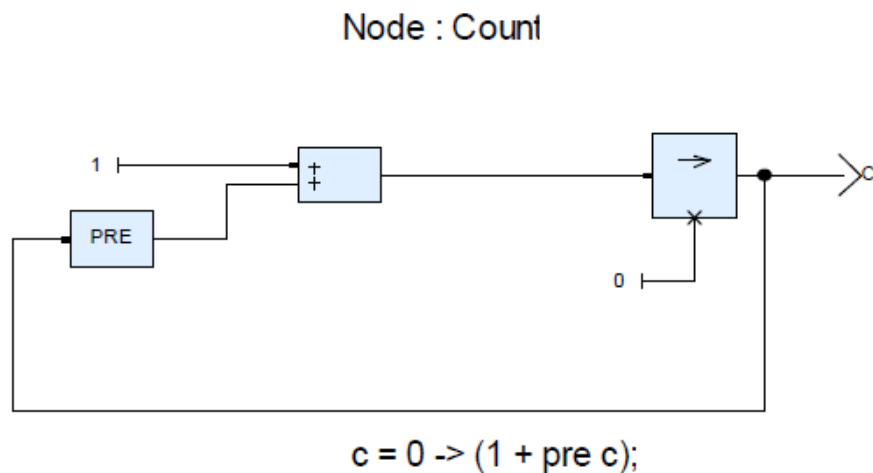
```
Order o =  
    new Order.Builder()  
        .buy(100, "IBM")  
        .atLimitPrice(300)  
        .allOrNone()  
        .valueAs(new StandardOrderValuer())  
        .build();
```

DSL的分类

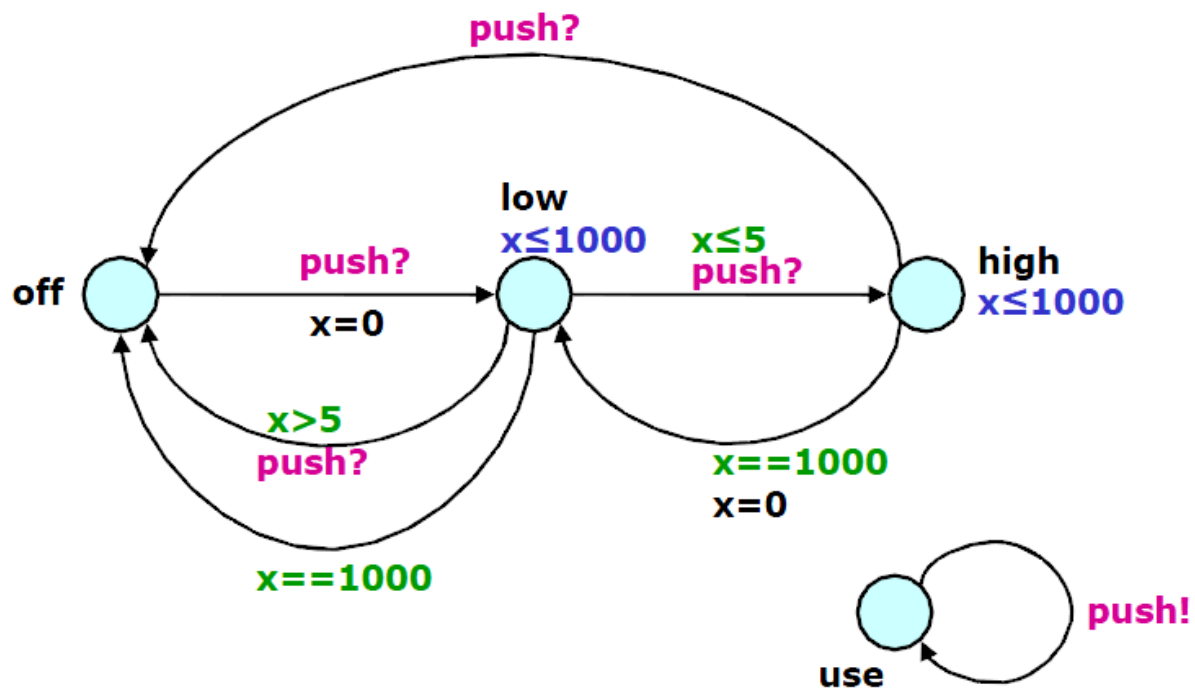
- 文本DSL
 - 外置式DSL
 - 内置式DSL
- 非文本的DSL
 - 领域特定建模语言
DSML (Domain specific modeling language)
- 完整语言 vs. 片段语言

领域特定建模语言（DSML）

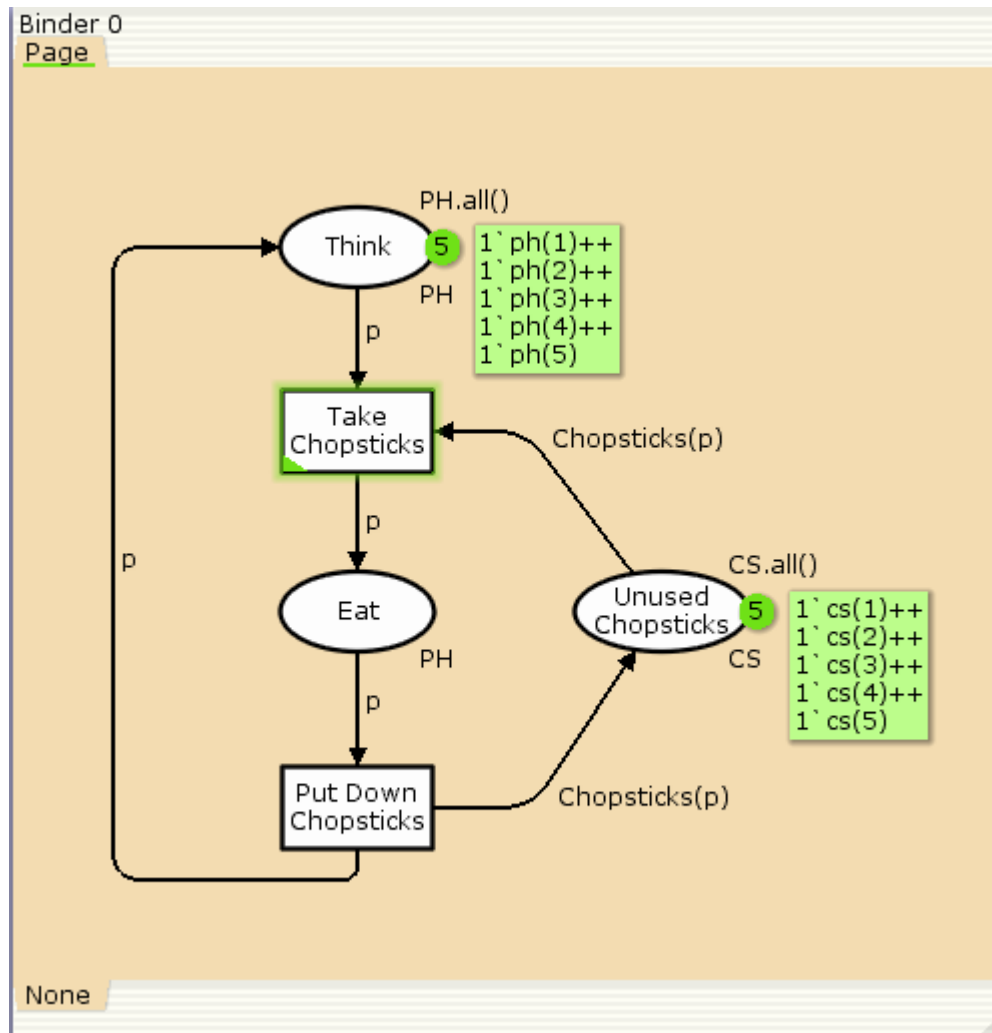
- 图形化的DSL，一般被称为modeling language
 - 图形化的语言
 - 以非文本的形式出现
 - 可执行
 - 示例：反应式系统同步编程Lustre语言（SCADE工具）



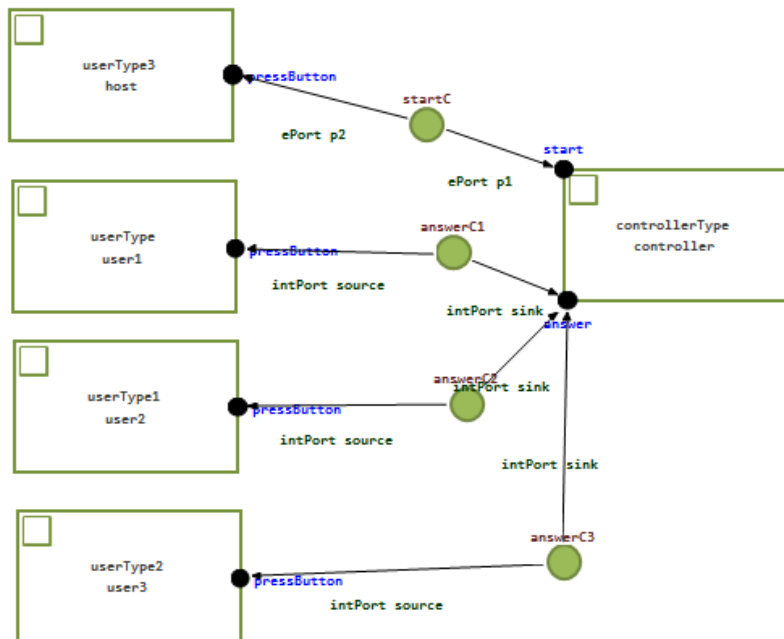
时间自动机语言



有色Petri网建模语言

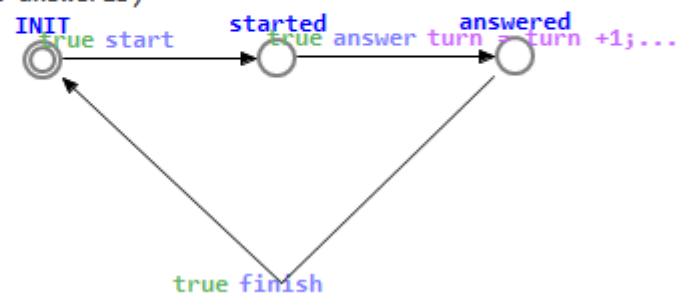


BIP语言（文本与图形化）



```

int turn = 0
int answerID = 0
int score1 = 0
int score2 = 0
int score3 = 0
export ePort start()
export intPort answer(int answerID)
ePort finish()
    
```



```

compound type SystemType
  component userType user1
  component userType1 user2
  component userType2 user3
  component controllerType controller
  component userType3 host
    
```

atomic type userType3

```

export port ePort pressButton
    
```

```

place INIT
    
```

```

initial to INIT do {
}
    
```

```

on pressButton from INIT to INIT provided true do {}
    
```

end

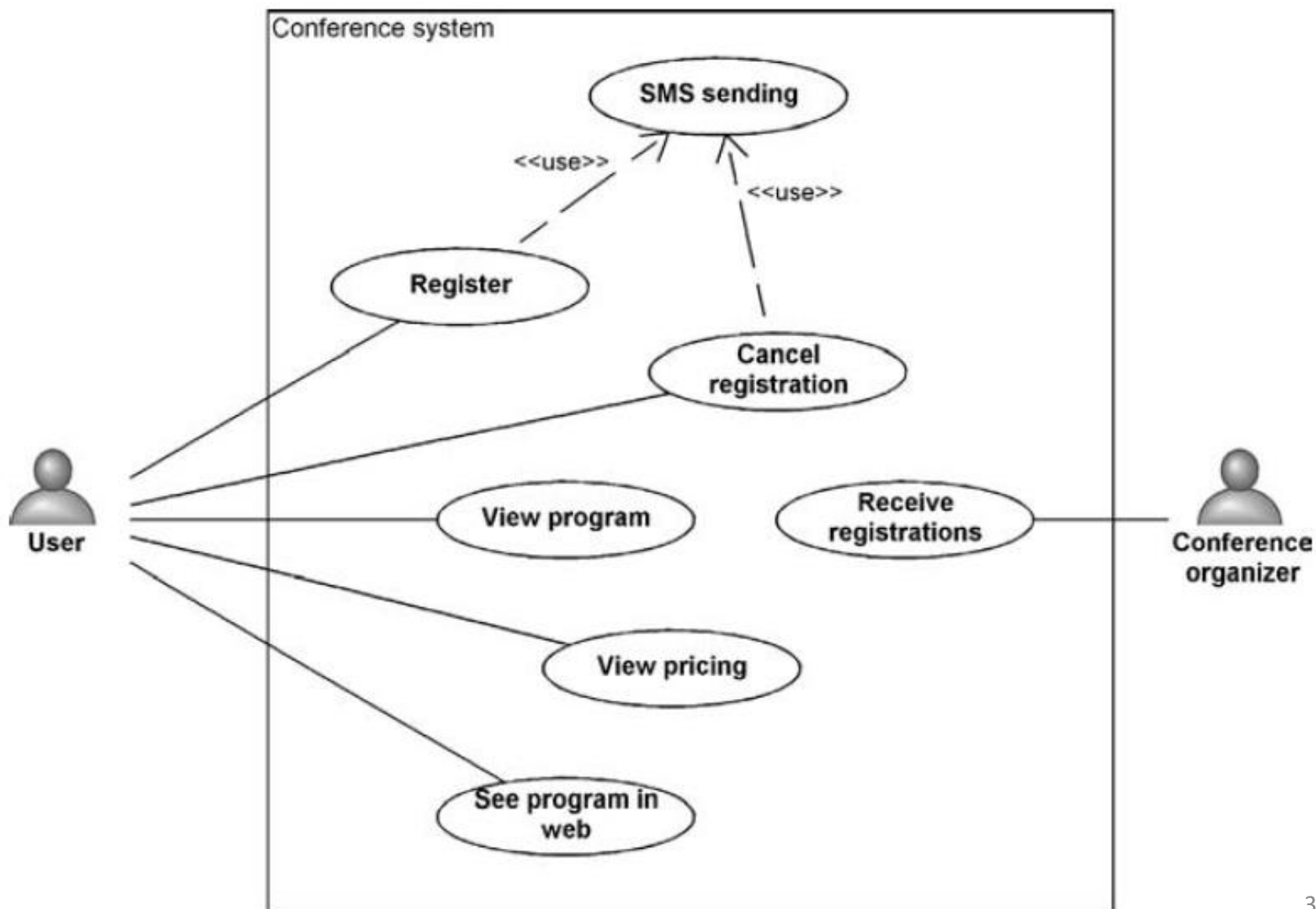
```

connector rendezvous _startC(controller.start, host.pressButton)
connector intrendezvous2 _answerC1(user1.pressButton, controller.answer)
connector intrendezvous2 _answerC2(user2.pressButton, controller.answer)
connector intrendezvous2 _answerC3(user3.pressButton, controller.answer)
    
```

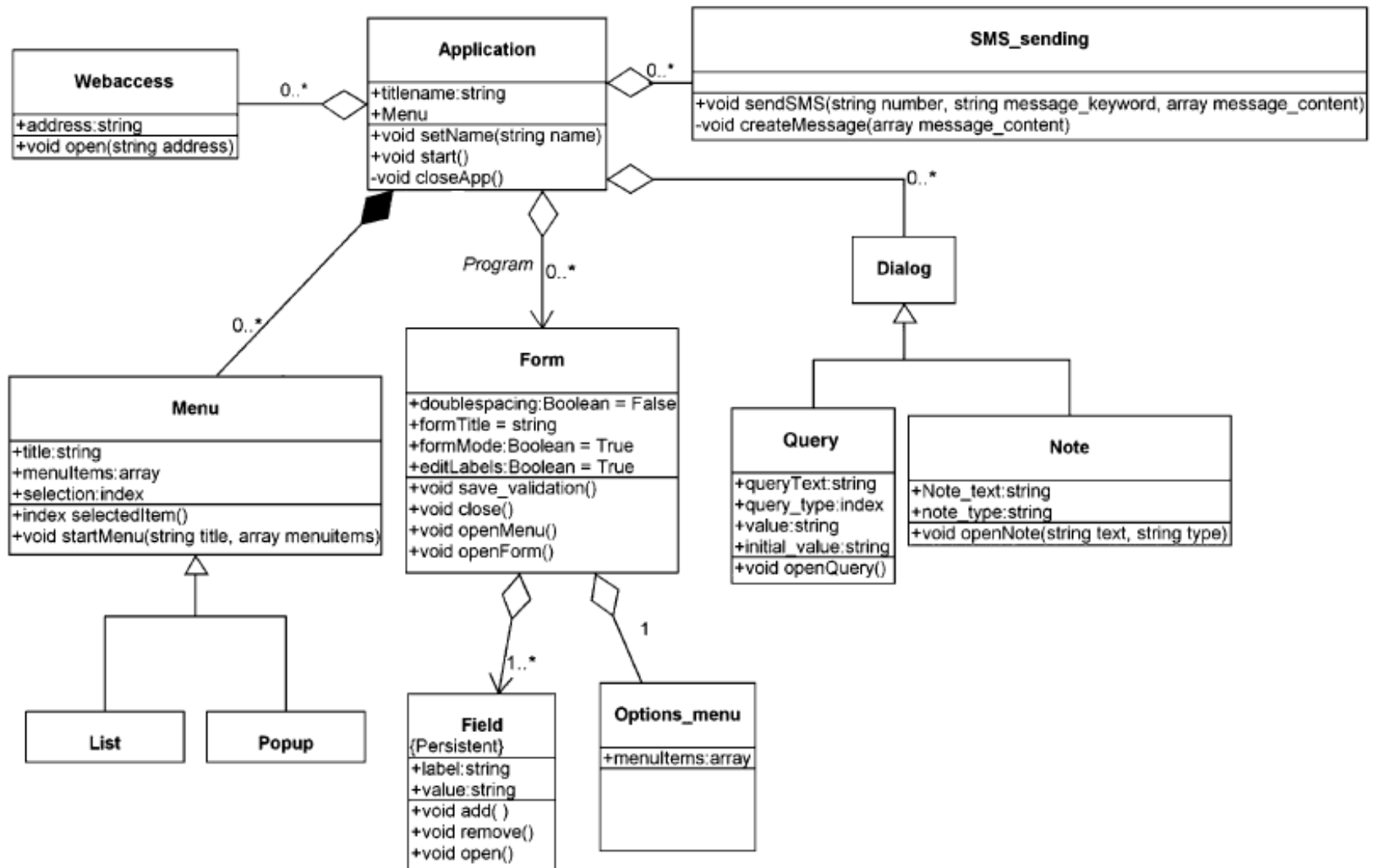
是UML吗？

- 不是。。。
- 示例：会议注册的手机应用
 - 用户可使用SMS来注册会议
 - 选择支付方式
 - 查看会议日程和speaker信息
 - 在Web网页浏览会议日程

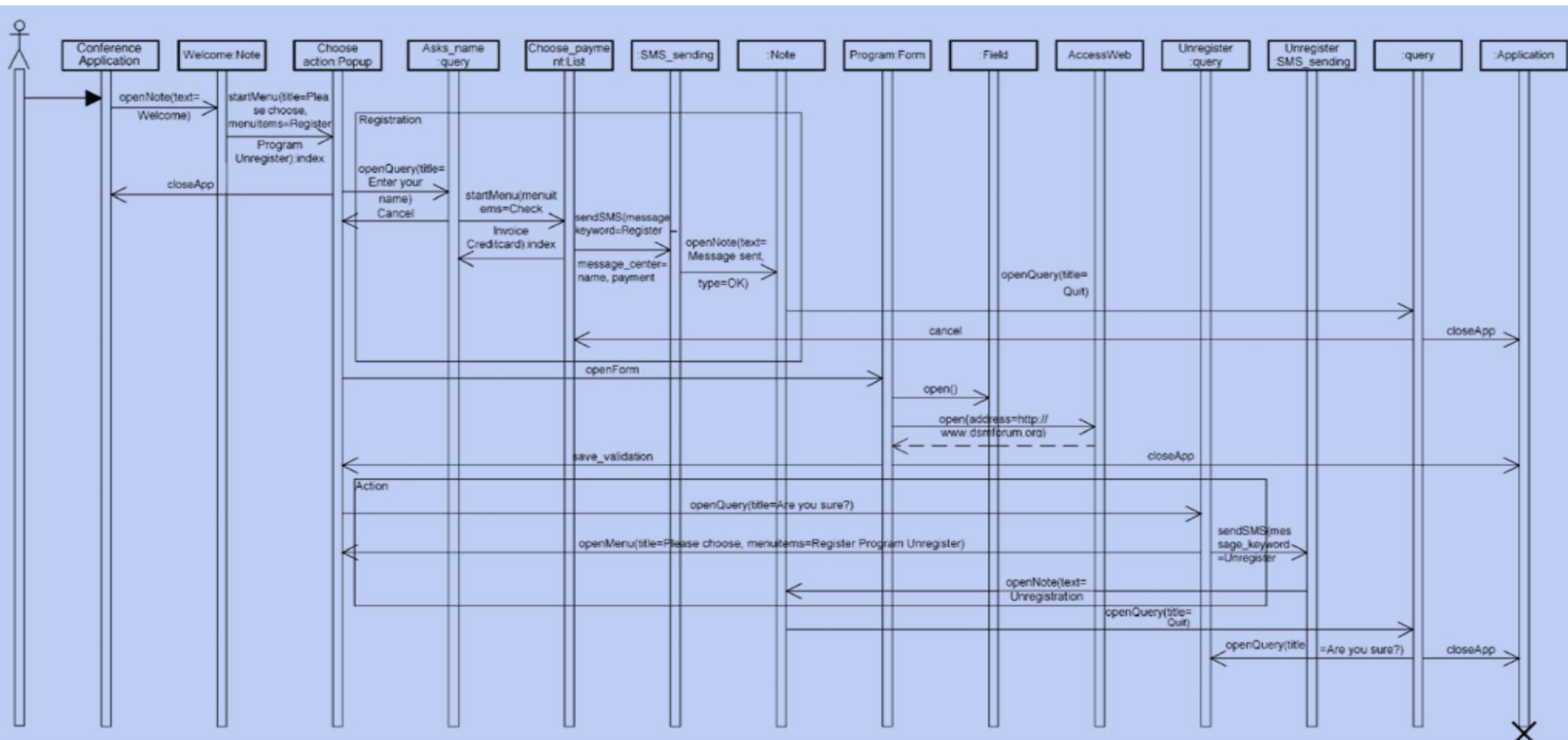
UML建模 - 用例图



UML建模 - 类图



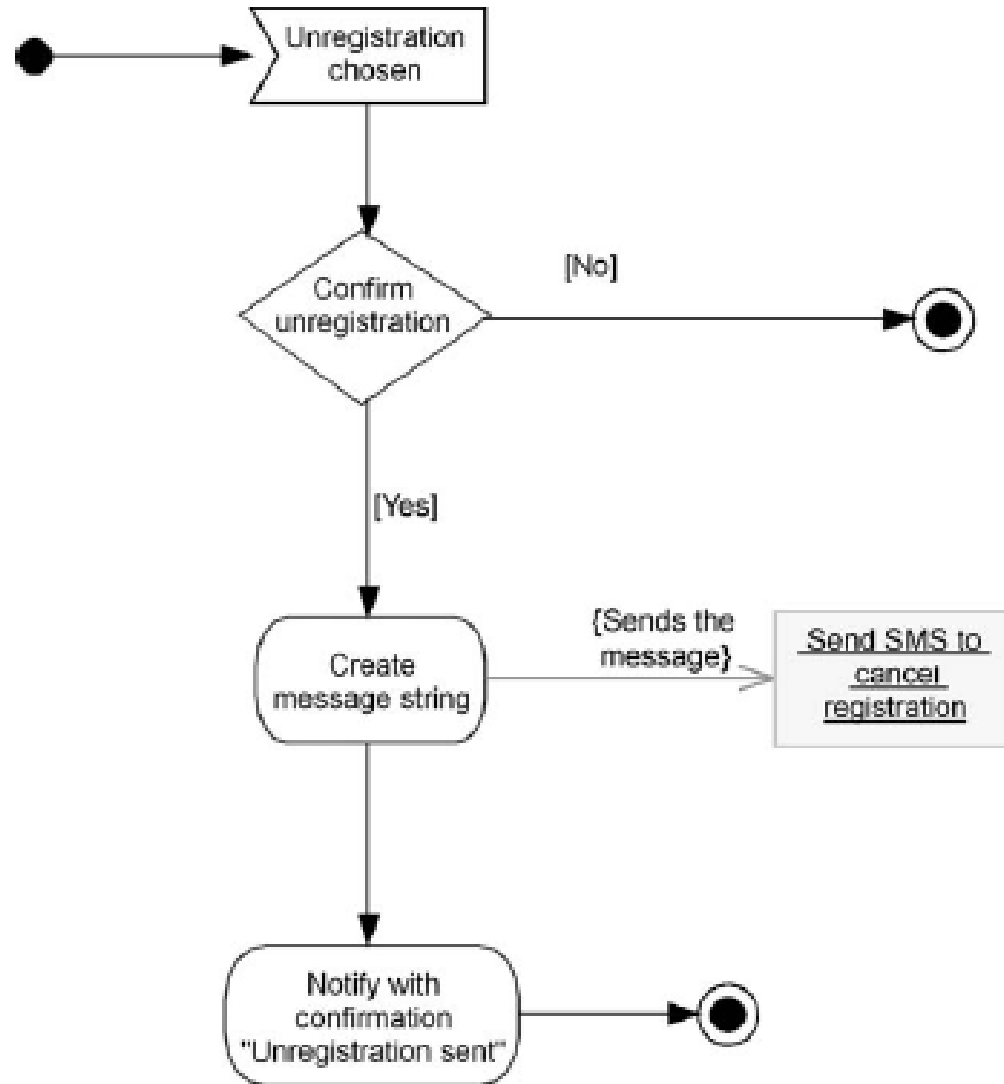
UML建模 - 顺序图



问题：顺序图不能表示选择

UML建模 – 活动图

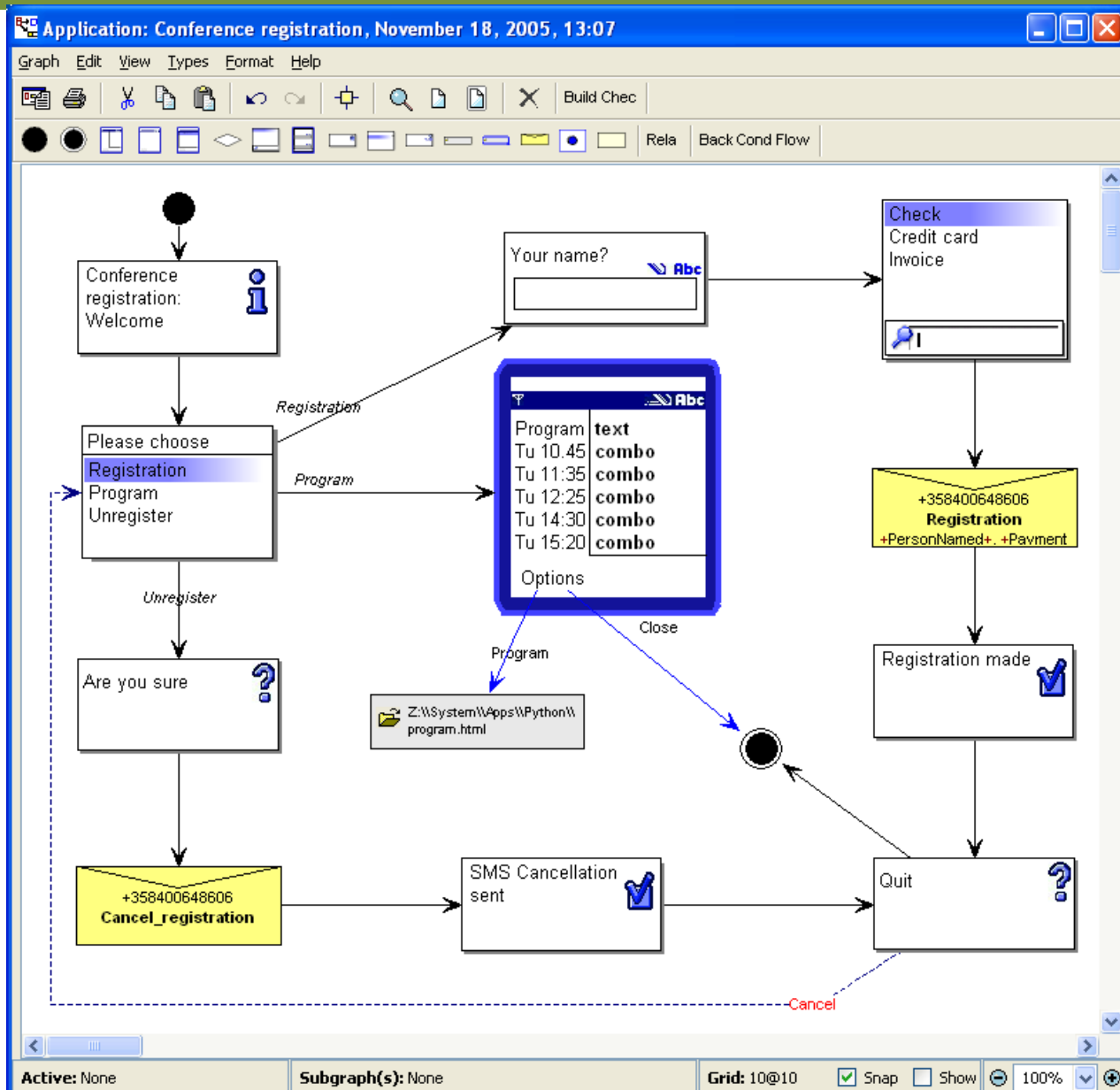
- 使用活动图建模 unregister 的选择情况



UML建模总结

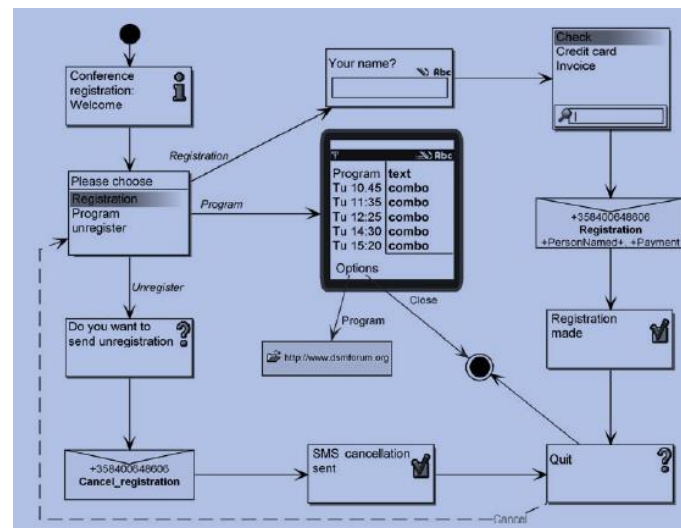
- 建模过程复杂
 - 1. 用例图
 - 2. 类图或顺序图或状态图
 - 3. 手工添加代码内容
- 特点
 - 先建模，后手工编码
 - 设计模型或与代码分离，或用于生成部分代码
 - 开发后期模型不能自动对照代码更新，维护成本高
- 完结问题
 - UML fever

DSL解决方案



DSL解决方案

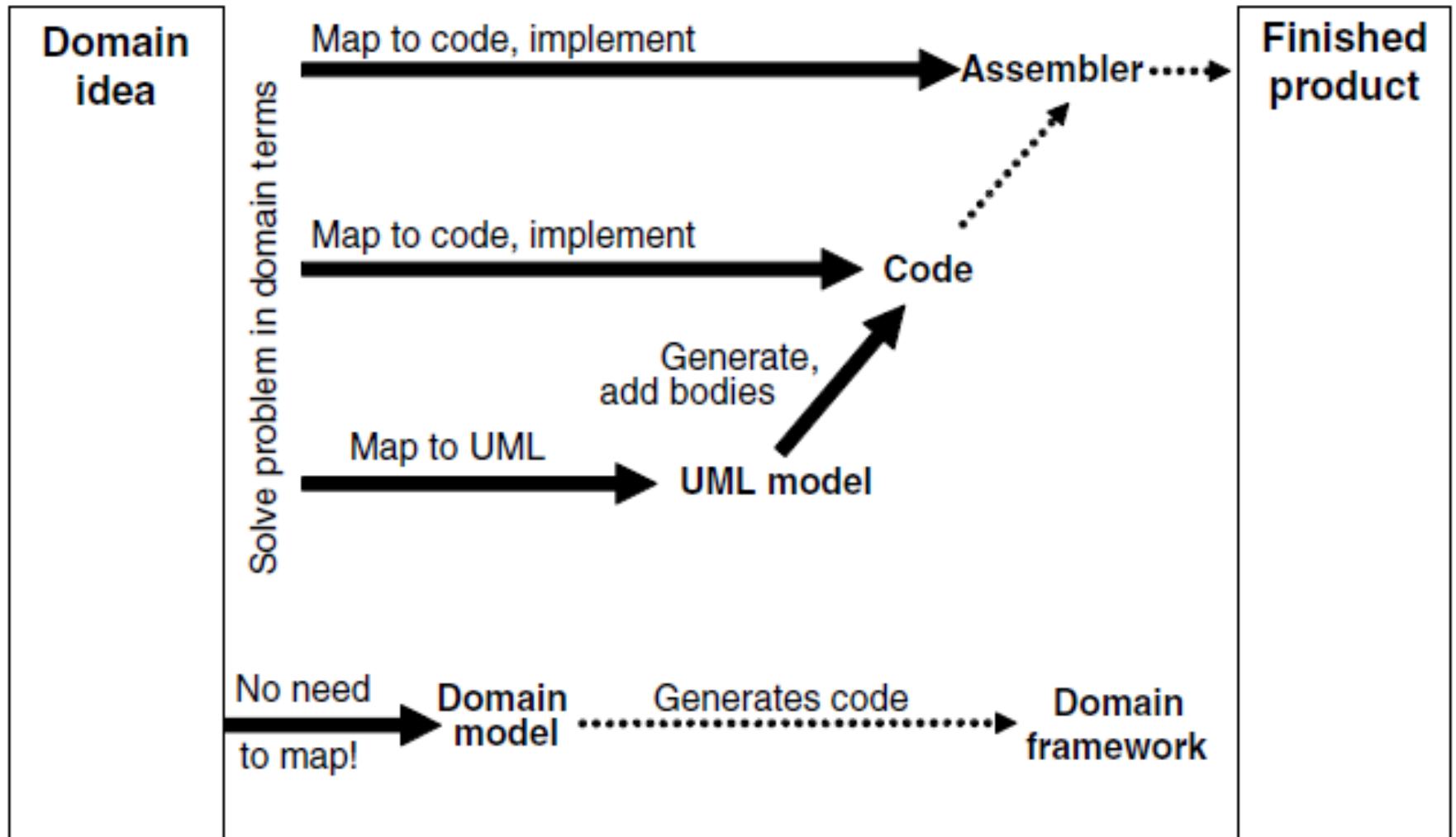
- 直接使用领域概念建模
 - 信息、下拉菜单、SMS、表格、查询等
- 引入领域规则以阻止非法设计
 - 限制SMS之后只能结束或回到主菜单
- 自动生成合适的完整代码
- 完结：当应用描述清楚了即建模结束



什么是DSL

- 两个内容
 - 通过直接使用问题域的概念与规则提升抽象层次
 - 自动生成指定的语言代码或其他形式的最终产品

比较示意图



代码与模型的关系

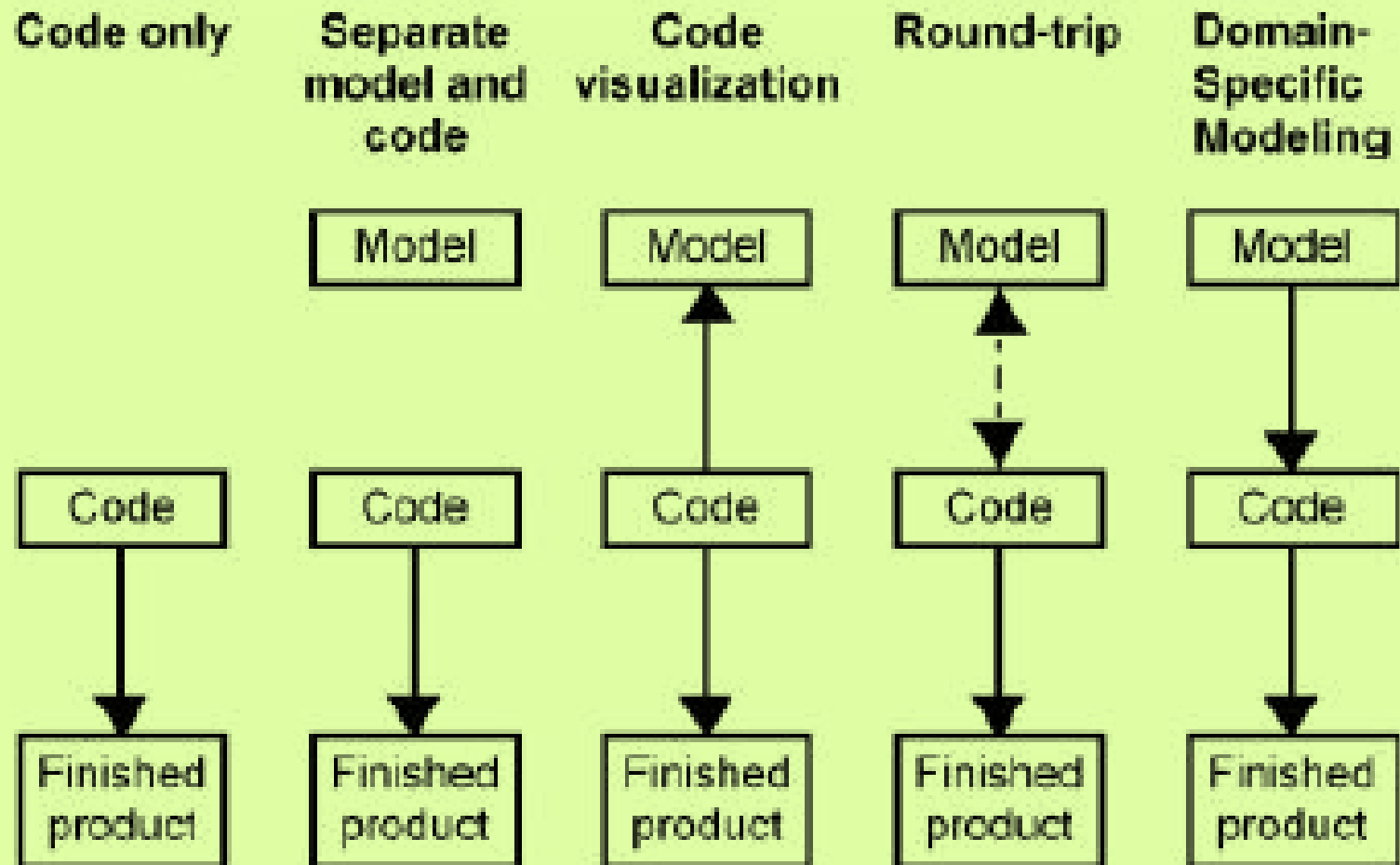


FIGURE 1.1 Aligning code and models

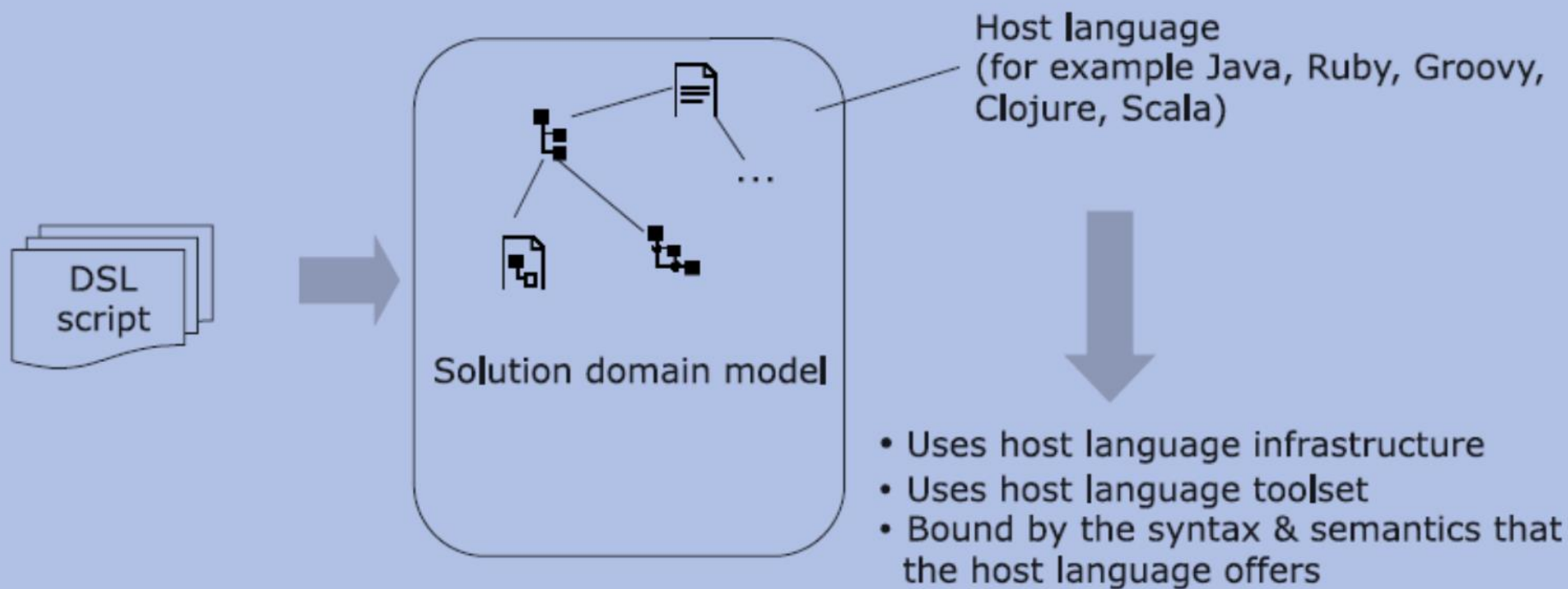
- 怎么做？

DSL的分类

- 文本DSL
 - 外置式DSL
 - 内置式DSL
- 非文本的DSL
 - 领域特定建模语言 DSML (Domain specific modeling language)

内置式DSL如何实现？

- 内置式DSL的解析



- 使用已有的一个宿主语言来实现DSL，从而可以重用宿主语言的已有处理机制。

示例：简单电脑配置

- 电脑：
 - 处理器：
 - 2核，速度2500， i386系列
 - 硬盘1
 - 150G
 - 硬盘2
 - 75G， 7200转， SATA接口
- 一般的API （以Java为例）

思考：如何描述？

一般API对电脑装配专家在理解上的障碍？

参数的语义不明确！

```
Processor p = new Processor(2, 2500, Processor.Type.i386);  
Disk d1 = new Disk(150, Disk.UNKNOWN_SPEED, null);  
Disk d2 = new Disk(75, 7200, Disk.Interface.SATA);  
return new Computer(p, d1, d2);
```

DSLs的表现形式

Function Sequence

```
computer();  
  processor();  
    cores(2);  
    speed(2500);  
    i386();  
  disk();  
    size(150);  
  disk();  
    size(75);  
    speed(7200);  
    sata();
```

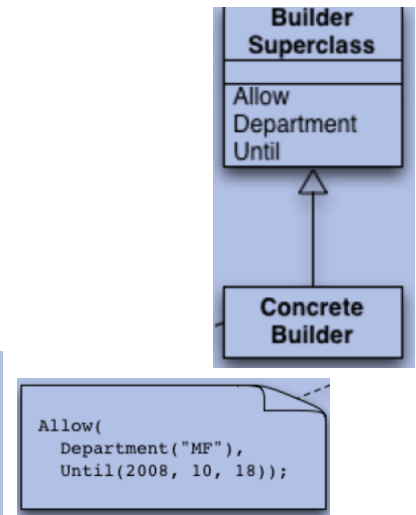
Nested Function

```
computer(  
  processor(  
    cores(2),  
    speed(2500),  
    i386  
  ),  
  disk(  
    size(150)  
  ),  
  disk(  
    size(75),  
    speed(7200),  
    SATA  
  )  
);
```

Method Chaining

```
computer()  
  .processor()  
    .cores(2)  
    .speed(2500)  
    .i386()  
  .disk()  
    .size(150)  
  .disk()  
    .size(75)  
    .speed(7200)  
    .sata()  
  .end();
```

Object Scoping



DSLs的表现形式

Nested Closure

```
computer do
  processor do
    cores 2
    i386
    speed 2.2
  end
  disk do
    size 150
  end
  disk do
    size 75
    speed 7200
    sata
  end
end
```

Literal List

```
martin.follows(
  "WardCunningham",
  "bigballofmud",
  "KentBeck",
  "neal4d");
```

Literal Map

```
computer(
  processor(
    :cores => 2,
    :type => :i386),
  disk(
    :size => 150),
  disk(
    :size => 75,
    :speed => 7200,
    :interface => :sata))
```


如何实现

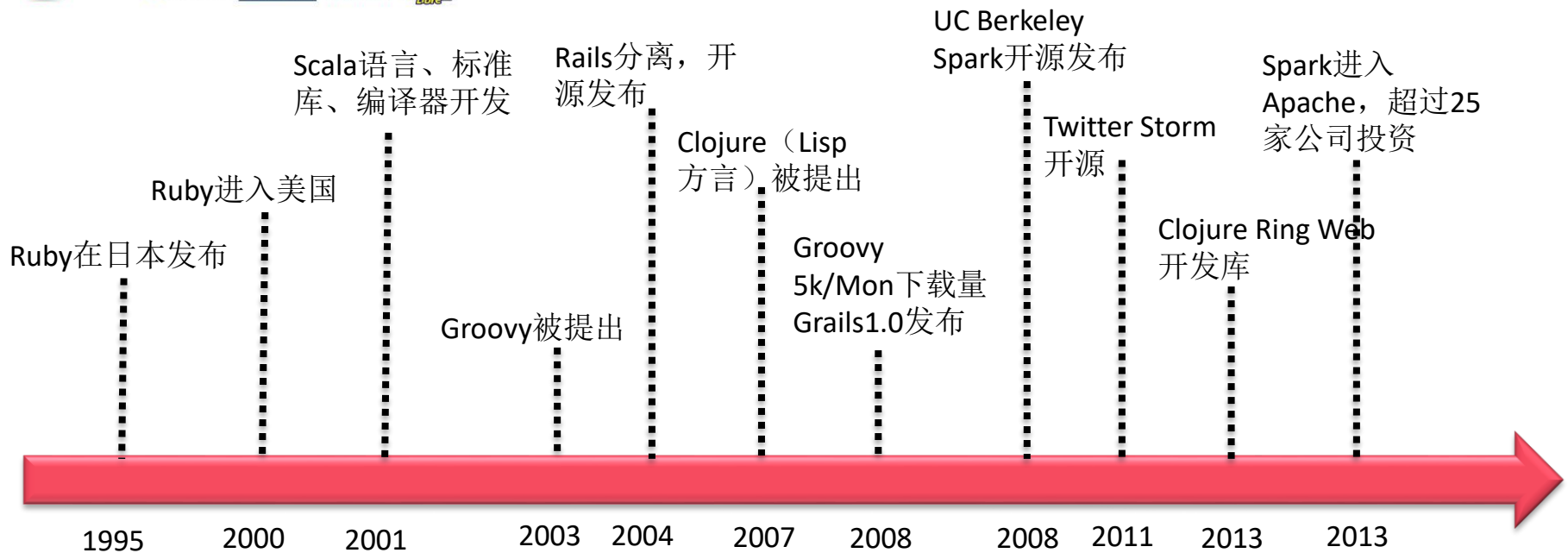
方法名	实现方法	方法名	实现方法
Expression Builder	常采用Composite模式 有semantic model支撑	Literal List	一种列表数据结构
Function Sequence	需要Context Variables保存状态 使用全局方法	Literal Map	一种map数据结构
Nested Function	反应语言的自然层次结构 调用顺序为先参数后外部	Dynamic Reception	定制的缺失错误处理(动态语言)
Method Chaining	用返回值连接方法 用递进式接口强制方法链上的顺序	Annotation	定义: 附加数据到 Semantic Model上 处理: 执行Semantic Model
Object Scoping	面向对象的继承	Class Symbol Table	Expression Builder定义字段 反射查找
Closure	一段代码片段可以被当作对象 语言本身提供的特性支持	Textual Polishing	正则表达式支持下的语言 文本修改替换
Nested Closure	显式地执行闭包内容 Ruby语言的instance evaluation	Literal Extension	语言支持: C#的扩展方法, Ruby的开放类

新型语言： 内置式DSL的宿主

- Groovy: 动态类型，面向对象，语法和java非常相似，扩展了语言编程能力
- Scala: 面向对象+函数式特性，比java高级的类型系统
- Clojure: Lisp动态版方言，函数式+并发特性，复杂宏系统，
- Ruby: Ruby 是一种面向对象、命令式、函数式、动态的通用语言
- 新型语言的诞生，一方面提供了比传统语言（C++、Java）更强的编程能力和特性，另一方面也大大促进DSL的发展。大量的内置式DSL基于这些语言被开发、应用。



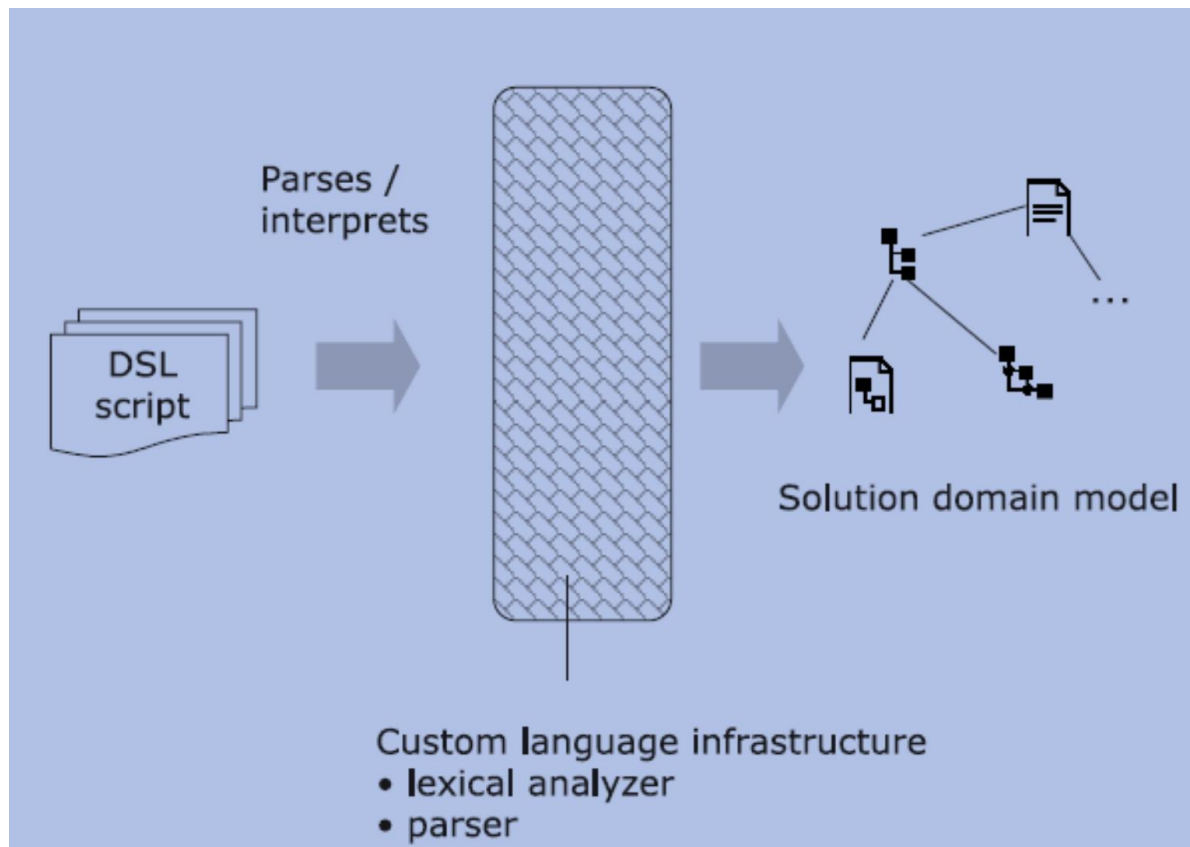
Sina, LinkedIn, Twitter, Novell, FourSquare, Sony, Siemens, Thatcham, OPower, GridGain, AppJet, Reaktor



外置式DSL如何实现？

DSL	用途
SQL	用于关系数据库的查询和操作
Ant, Make	用于Build软件系统
YACC, Bison, ANTLR	用于生成语法解析器
HTML	网页标注语言
EXCEL	电子表格处理
VHDL	硬件描述语言
Mathematica	数学计算语言
○ ○ ○	○ ○ ○

- 外置式DSL的解析



- 对于外置式DSL，需要自己开发语言的处理机制。包括词法解析、语法解析、代码生成等。

- 外置式DSL技术
 - 词法解析
 - Regex Table Lexer
 - 语法解析算法
 - *Recursive Descent Parser*
 - *Parser Combinator*
 - *Parser Generator*
 - 输出策略
 - *Embedded Interpretation*
 - *Embedded Translation*
 - *Tree Construction*

语言工作台

- 语言工作台
 - 用于设计和构建DSL的特殊IDE
 - 一般包含DSL的编辑器、编译器
 - 示例：
 - GMF <http://www.eclipse.org/modeling/gmf/> 是 Eclipse项目中快速开发图形界面的工具包
 - Visual Studio DSL 工具，微软针对特定领域开发而专门设计的，包含在Visual Studio SDK中
 - MetaEdit+ <http://www.metacase.com/> 图形化的DSL工具
 - MPS: <http://www.jetbrains.com/mps/> JetBrains公司的DSL工具,通过这个平台可以直接定义规则，生成代码
 - Xtext: <http://www.eclipse.org/Xtext/> 基于Eclipse的一个简单高效的DSL开发环境，和java语言良好融合。能够生成属于DSL的编程语言环境。

示例：家谱、族谱



《三国演义》 刘备

汉献帝尊称其为“皇叔”
更以

“中山靖王之后，孝景帝玄
孙”自居

- “孝景皇帝生十四子。第七子乃中山靖王刘胜。胜生陆城亭侯刘贞。贞生沛侯刘昂。昂生漳侯刘禄。禄生沂水侯刘恋。恋生钦阳侯刘英。英生安国侯刘建。建生广陵侯刘哀。哀生胶水侯刘宪。宪生祖邑侯刘舒。舒生祁阳侯刘谊。谊生原泽侯刘必。必生颍川侯刘达。达生丰灵侯刘不疑。不疑生济川侯刘惠。惠生东郡范令刘雄。雄生刘弘。弘不仕。刘备乃刘弘之子也”

七世

行一 寬 字洞公 廣性謹 娶馬氏 勤苦自甘 生二子 休信 休龍

八世

行一 休信 字允若 邑庠生 娶徐氏 无子 继娶李氏 生三子 休龍 休行 休民 生一子 休 继娶氏 无子

九世

行一 測 字澤南 娶孟氏 生二子 元修 元立
二 潛 娶郭氏 生一子 元榮
三 潤 娶孟氏 生二子 元相 元松
行一 作 娶康氏 生一子 元松

十世

行一 椿 娶白氏 生三子 繼榮 繼煥 繼煥 出继族兄
二 元立 字本若 娶郭氏 生五子 繼煥 繼煥 繼煥 繼煥 繼煥
行一 元榮 娶李氏 生二子 繼煥 繼煥 出继族兄 繼煥 繼煥

十一世

行一 燧 娶李氏 无子
二 榮 娶任氏 生一子 建全
三 燧 娶王氏 无子
行一 燧 娶張氏 生三子 建貴 建升 建正
二 燧 娶蕭氏 生三子 建煥 建祥 建升 建煥 出继族兄 字廣而嗣
三 燧 娶張氏 生一子 建才
行一 燧 娶黃氏 生一子 建坊
行一 燧 娶王氏 生三子 建貴 建泰 建忠
行一 燧 娶郭氏 无子 过林姓之子 建勇 而嗣
行一 燧 娶李氏 生一子 建忠

Family Tree

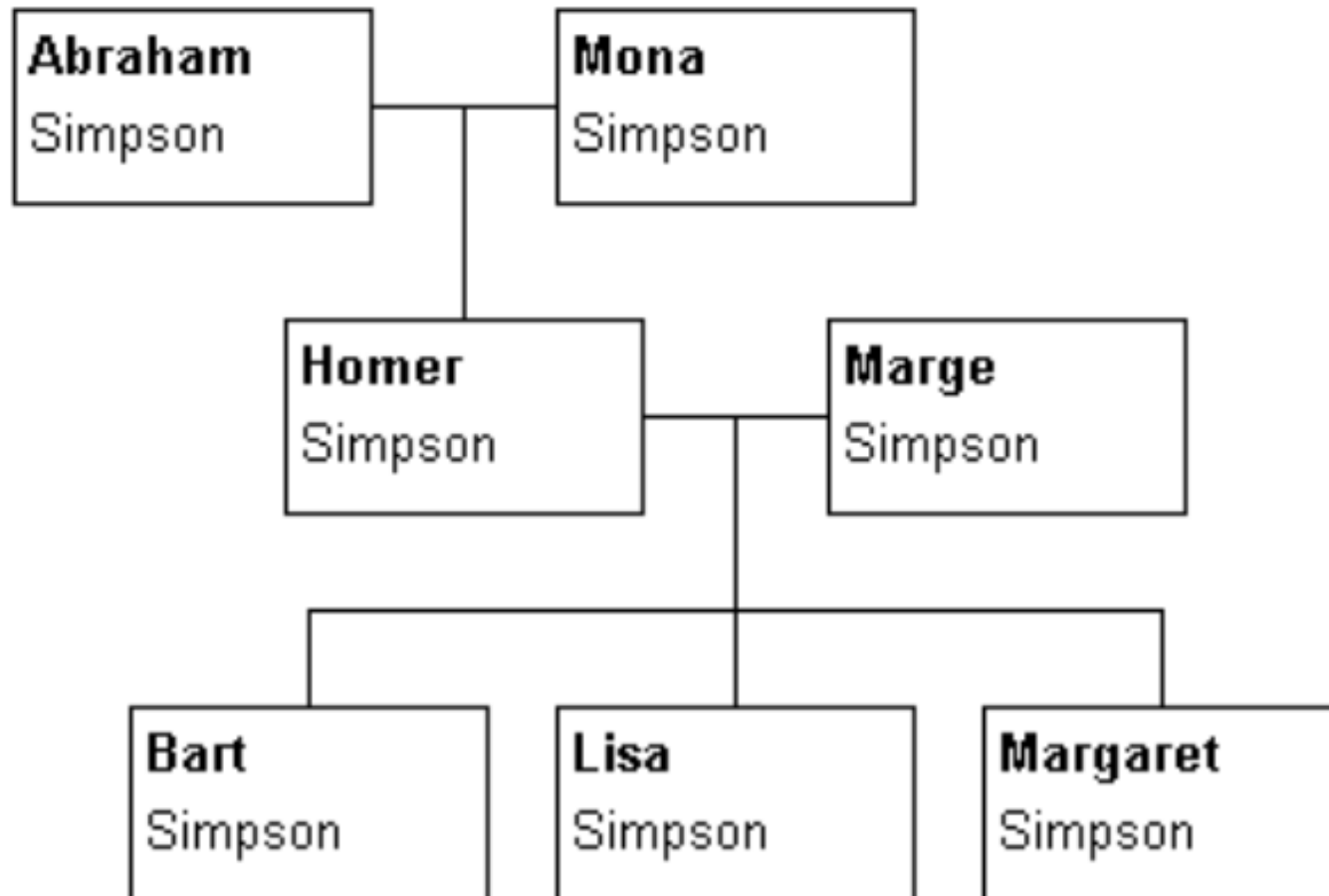


Figure 1-1. Sketch of a graphical notation for the Family Tree.

如何设计Family Tree?

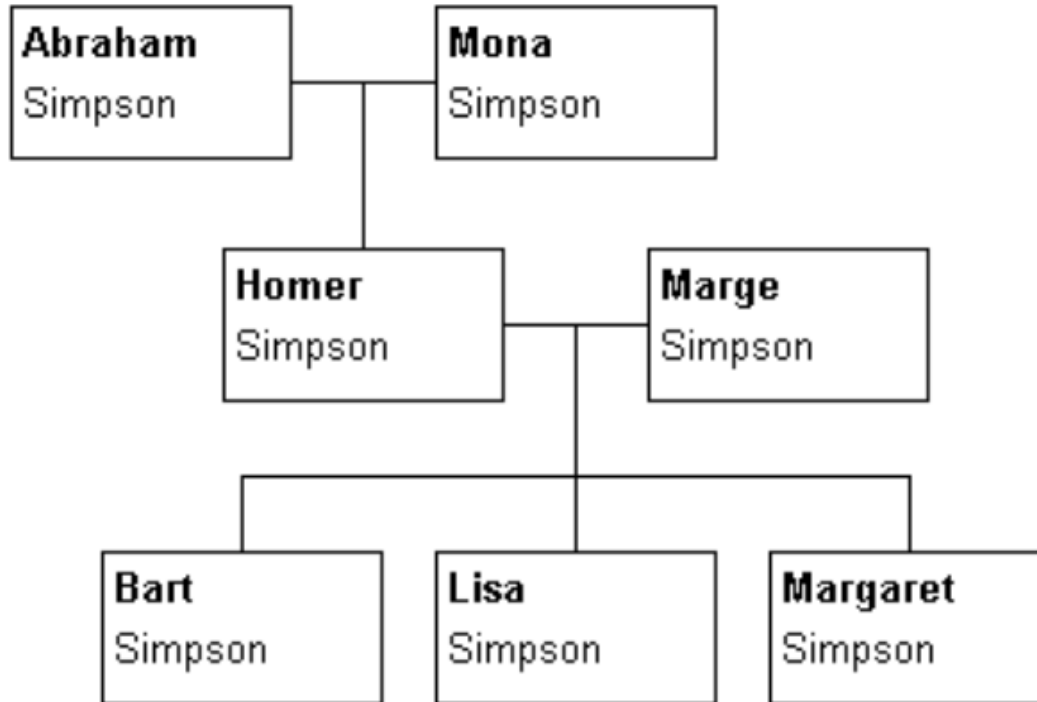


Figure 1-1. Sketch of a graphical notation for the Family Tree.

- 讨论：如何实现一个Familytree的工具？
 - 包含哪些内容？
 - 图形
 - 对象、关系、角色、属性
 - ...

如何设计Family Tree?

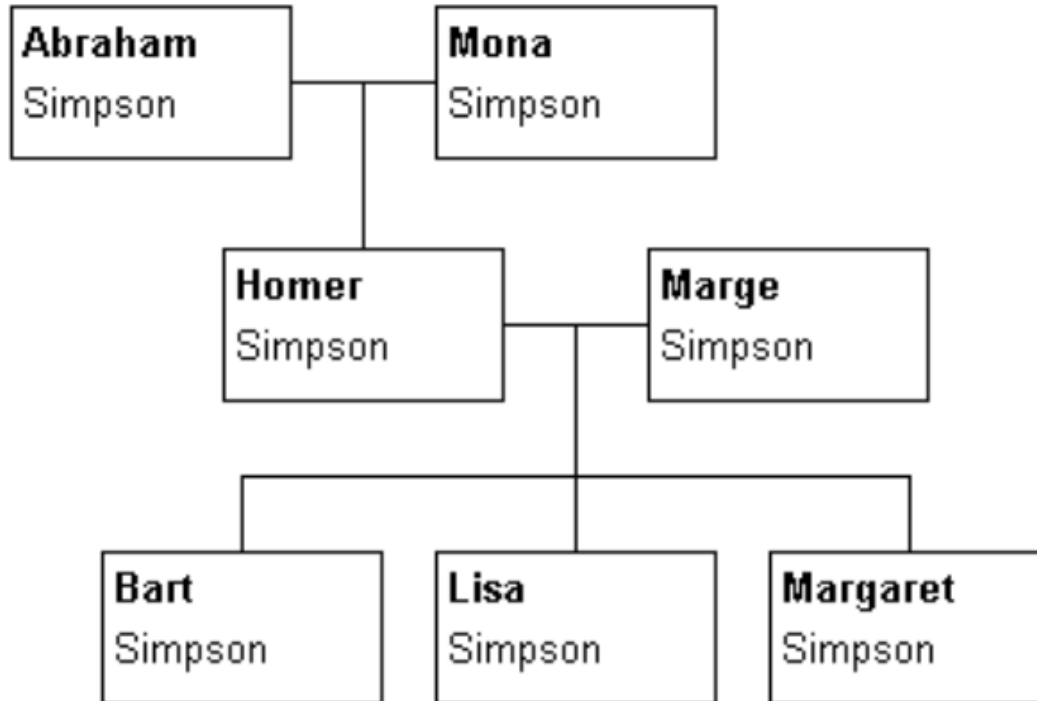


Figure 1-1. Sketch of a graphical notation for the Family Tree.

- 讨论：如何实现一个Familytree的工具？
 - 图形中存在哪些约束？
 - 一夫一妻制
 - 男左女右
 - 属性值约束
 - ...

如何设计Family Tree?

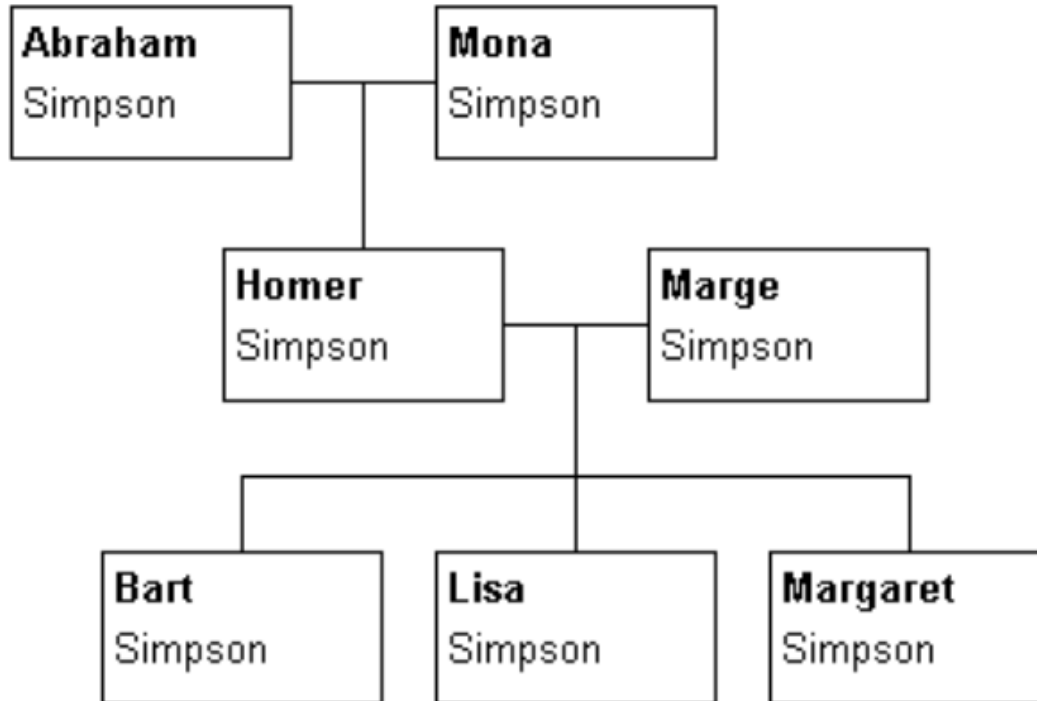


Figure 1-1. Sketch of a graphical notation for the Family Tree.

- 讨论：如何实现一个Familytree的工具？
 - 如何实现？
 - 固化实现：图形+代码
 - DSM：元模型，模型检索生成输出

一种元模型定义：GOPPRR

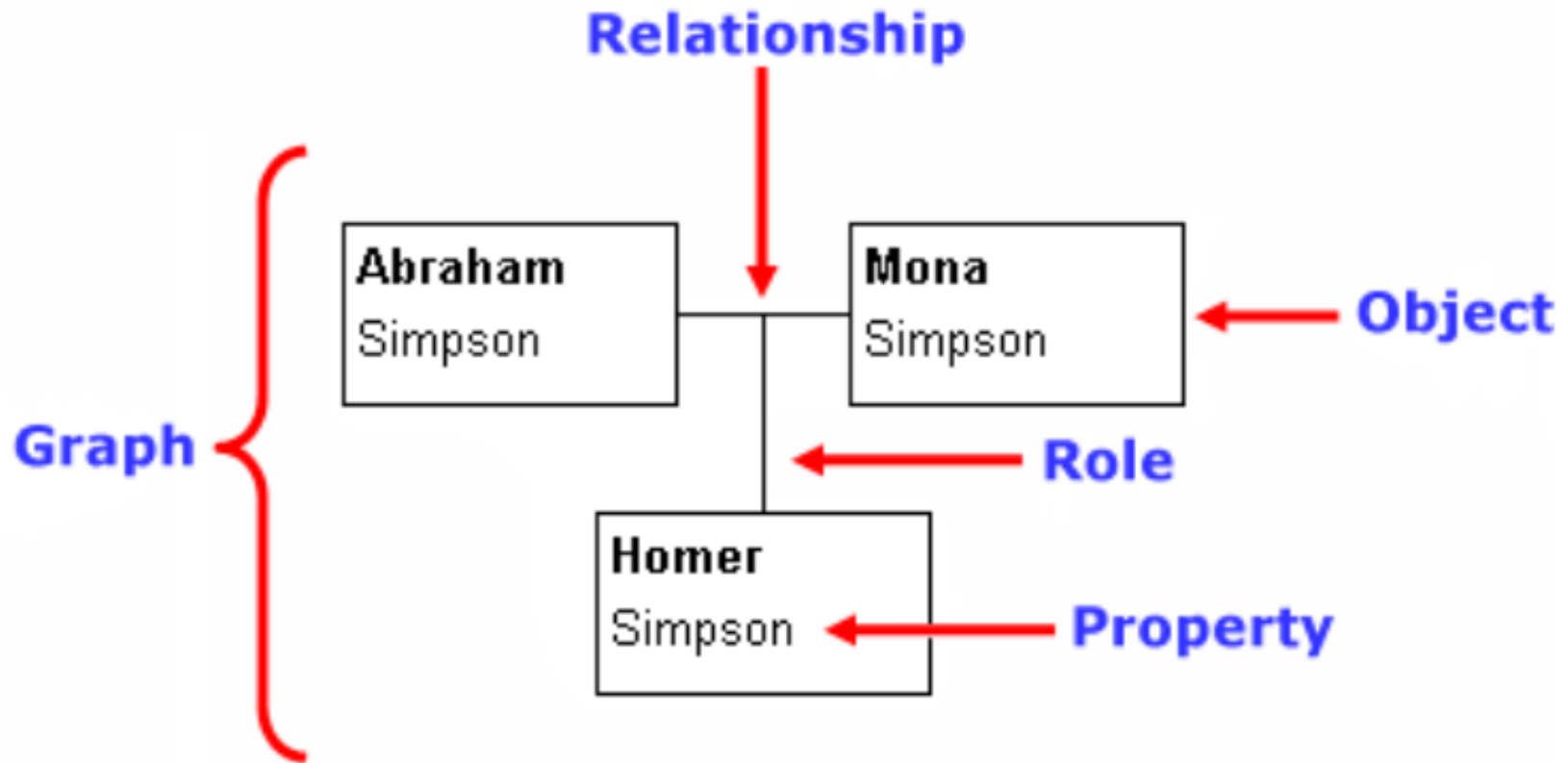


Figure 1-2. GOPPRR metatypes as found in Family Tree.

- 缺少的一个P？ Port
- 可选，表示连接的边界

DSL工具

- MetaEdit +
 - <http://www.metacase.com/>
 - 1991 MetaEdit, research prototype
 - 1993 MetaEdit, commercial version, 图形化定义建模语言的第一个工具
 - 1995 发布MetaEdit+, 开始支持多用户、多建模语言和多模型集成
- MetaEdit+ 5.0
 - Visual studio extension
 - Eclipse plug-in
 - Simulink集成

Family tree建模语言的元模型

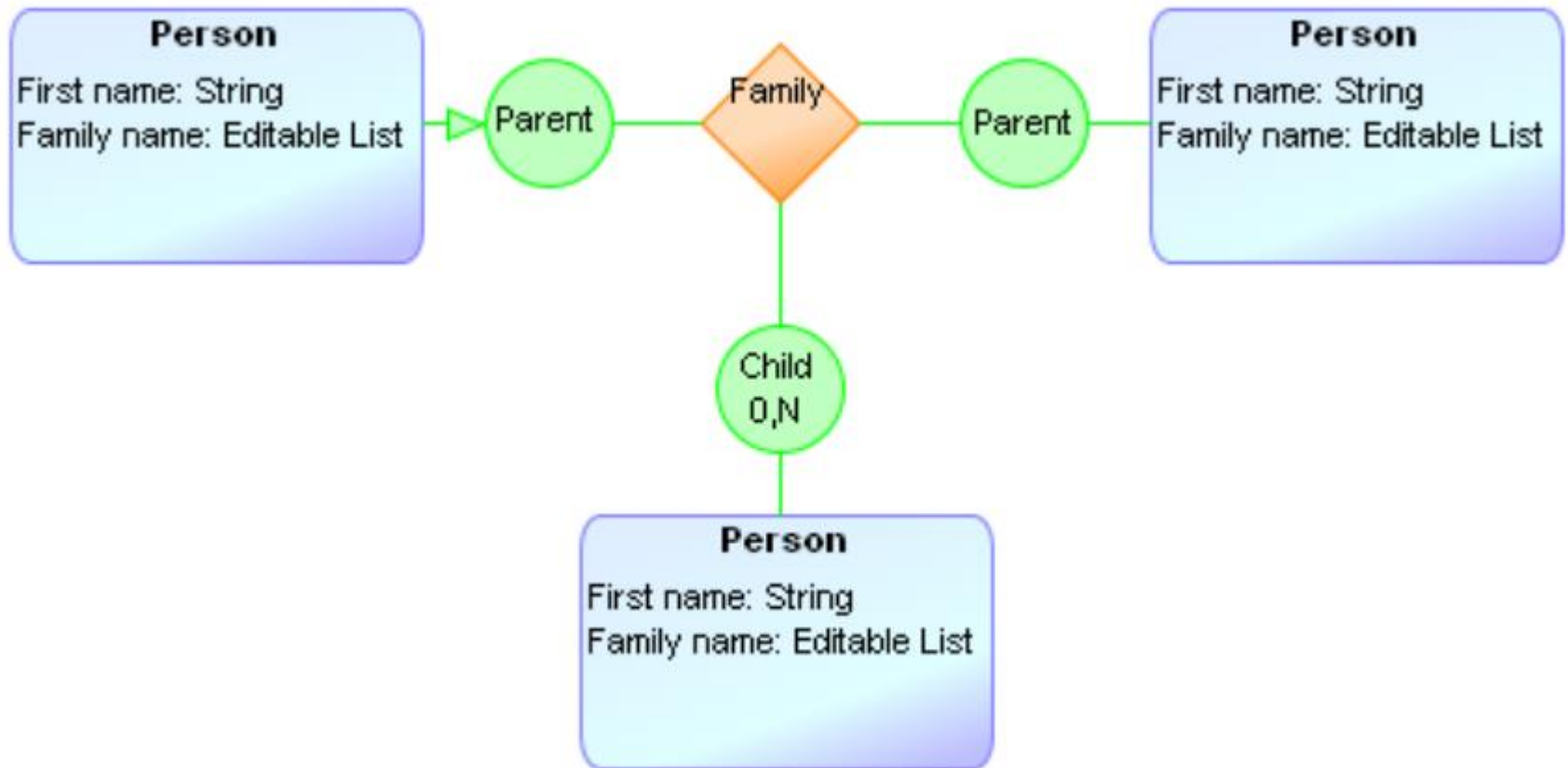
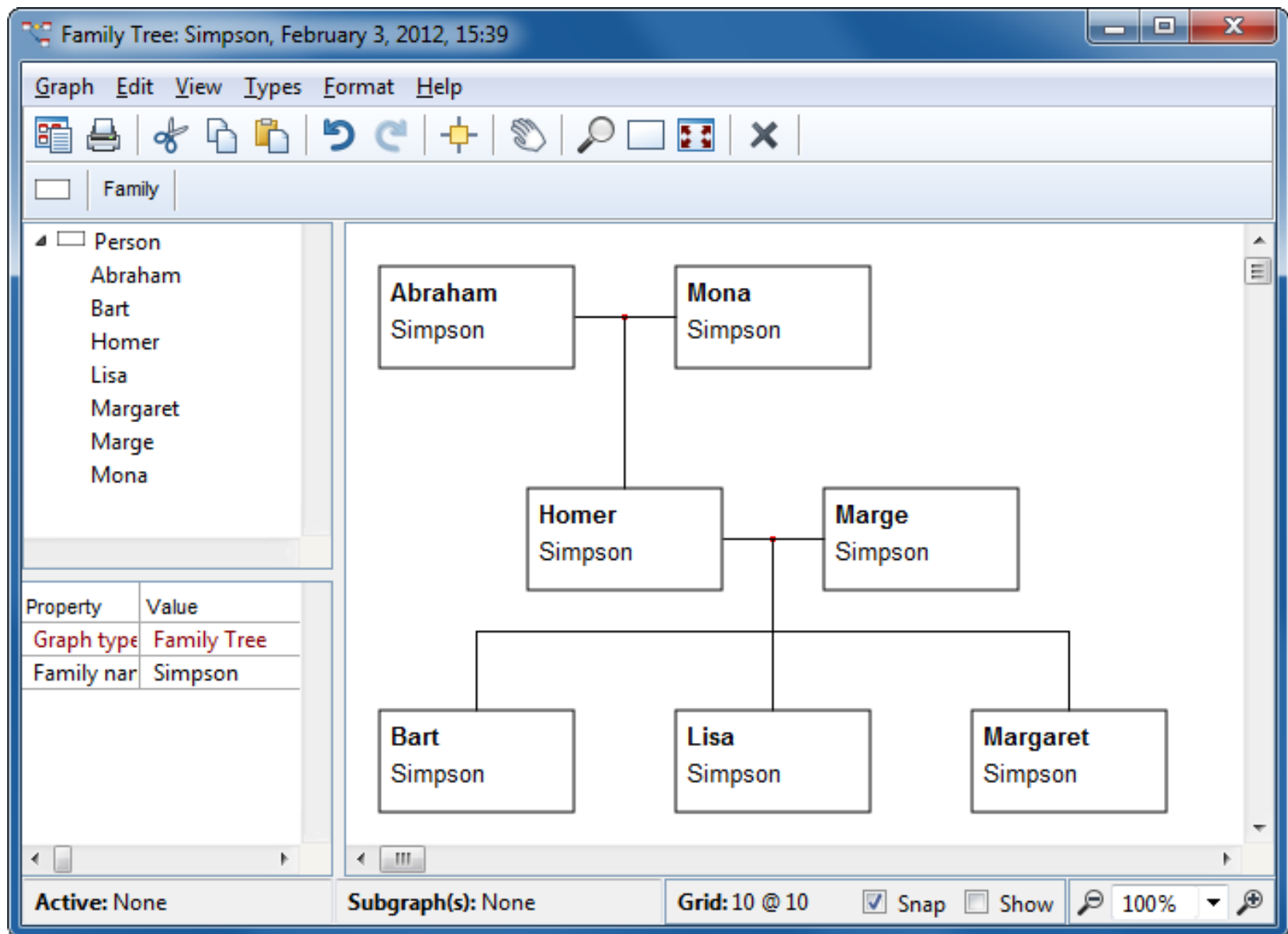
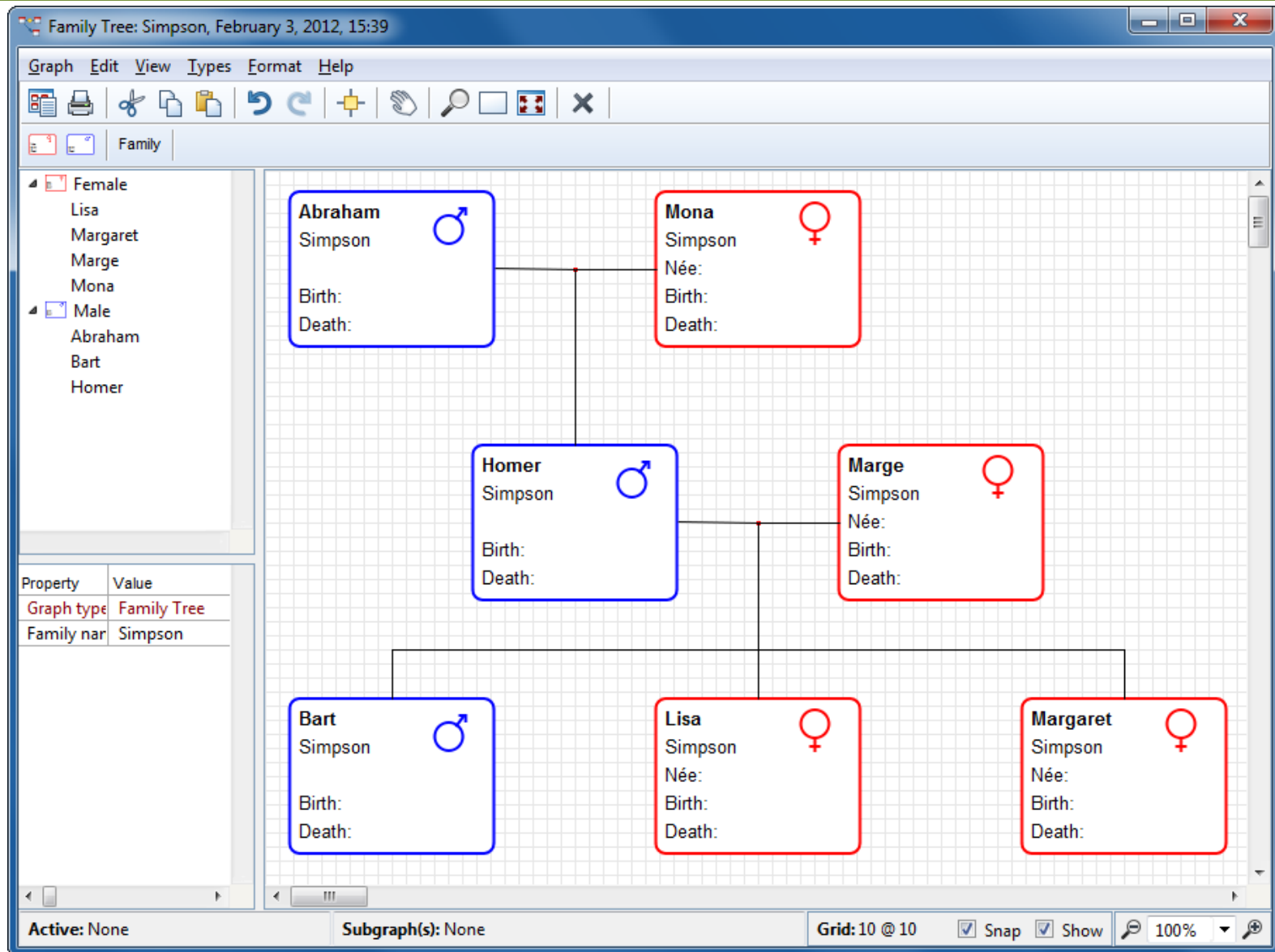


Figure 1-3. Metamodel for the Family Tree modeling language.

创建的DSML

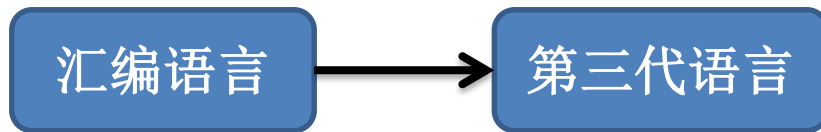


改善的DSML



更好的抽象层次

- 开发者通过“抽象”来提升生产力



- 而今，传统编程语言和建模语言却对生产力贡献相对较小
 - Java / UML
- DSL：进一步提高抽象层次，直接使用问题域中的概念和规则



- 怎么用？

DSLs 的应用案例

案例1: 开发MIS / ERP 软件产品

拓步ERP平台—热键旧版DEMO企业版 - [MRP运算结果综合查询]

文件(F) 编辑(E) 业务(B) 窗口(W) 帮助(H)

拓步ERP平台

主控面板 主生产计划维护 脱期生产工单维护 MRP运算结果综合查询

功能平台

- 采购管理系统
- 销售分销系统
- 门店零售系统
- 库存管理系统
- 质量管理系统
- 外协管理系统
- 条码管理系统
- 生产制造平台
 - 主生产计划系统
 - 物料需求计划系统
 - MPS计划查询
 - 客户订单查询
 - 客户订单排产
 - 主生产计划维护
 - MRP计划运算
 - 脱期生产工单维护
 - 脱期采购订单维护
 - MRP运算
 - MRP运算结果综合查询
 - 运算异常数据查询
 - 运算信息产品正查
 - 运算信息物料反查
 - MRP时段查询

- 标准功能
- 用户功能
- 我的收藏
- 我的报表

预览 打印 刷新 查询 帮助 关闭

	物料编号	低位码	物料名称	规格	单位	物料类型	物料类别	来源码	可用库存量	总计划量	最小批量	批量增量	安全库
1	0009001	1	测试物料		米	普通	临时散料	采购	1,989.	2,011.	1.	1.	
2	0100100020000	1	01001创新半成品	银色上盖(按键)+透明壳	个	普通	鼠标裸机	自制	180.	44,290.	1.	1.	50.
3	0100100020400	1	01001创新鼠标	米色上盖+蓝色按键+蓝色	个	普通	鼠标裸机	自制	101.	0.	1.	1.	
4	0100520020000	1	创新鼠标本体	USB带灯透明蓝装饰条	个	普通	鼠标裸机	自制	1.	0.	1.	1.	
5	050003	1	测试设备半成品		公斤	普通	清洗剂	自制	0.	0.	1.	1.	
6	05AC000001	1	海藻清洗剂		克	普通	清洗剂	采购	0.	0.	1.	1.	
7	05AC000002	2	海藻清洗剂		克	普通	清洗剂	采购	910.	0.	1.	1.	

综合计划 毛需求查询 预计入库 库存明细 预留明细 库存汇总 预计失效 计划明细 时段汇总

	来源类型	单据号	单据状态	预计入库数量	预计入库日期	目的仓库	生产批次	源单号
1	采购在途	P006074030	发放	100.	2012-04-13	43		
2		P007034049	计划	10.	2012-04-13	43		
3		P009024082	计划	1,000.	2012-04-13	43		
4		P009064083	计划	10.	2012-04-13	43		
5	采购在检	PRE1002001	发放	50.	2010-02-03	43		P010024091
6		PRE1105001	执行	1,000.	2011-05-07	43		
7	采购申请	PR09040026	发放	100.	2012-04-13	43		
8		PR09060028	N	10.	2012-04-13	43		
9		PR11100037	发放	1.	2012-04-13	43		
10		PR12020001	计划	100.	2012-04-13	43		
11	计划工单	W008040900001	计划	1,000.	2012-04-13	43		
12		W012032600001	计划	200.	2012-04-13	43		

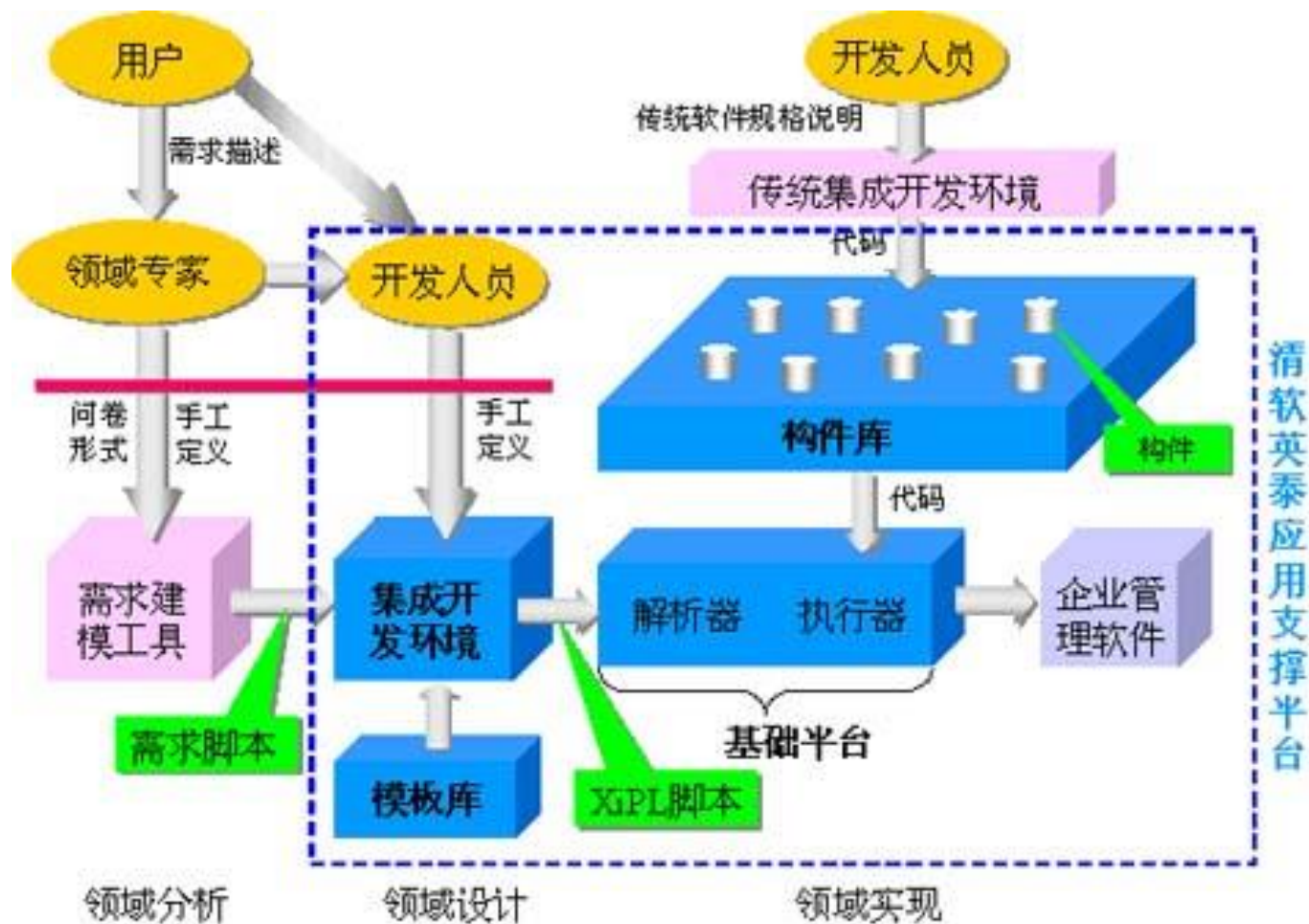
查询条件: (公司范围: 深圳市拓步软件技术有限公司)

本次操作费时 3秒

你有 28 份新邮件! 你有 0 件审核事务!

IMLKI深圳市拓步软件技术有限公司 期间: 2012.4 用户: 超级用户 2012-04-28 11:15:14

基于XiPL脚本的领域开发框架



用XiPL脚本实现

江苏长江电器股份有限公司ERP系统 v1.0 - 脚本培训演示

退出 数据 保存 可写 只读 部门 查询

树状对象举例

- 根节点 (000)
 - 一级节点A (010)
 - 二级节点AA (011)**
 - 二级节点AB (012)
 - 二级节点AC (013)
 - 一级节点B (020)
 - 一级节点C (030)

职员编码: ZY001 职员姓名: 职员A 性别: 男

部门: 电子商务部 薪资: 1,500.00 职称: 助理工程师

入职日期: 2003-09-05 资料文件名: AutoUpdate.exe

序号	职员编码	职员姓名	性别	部门	薪资	职称
1	ZY001	职员A	男	电子商务部	1,500.00	助理工程师
2	ZY002	职员B	女	研发中心	9,000.11	高级工程师

第一页 / 第二页

所属部门: PDM部门 薪资下限: 0.00 薪资上限: 10,000.00

入职日期下限: 1999-01-01 入职日期上限: 2004-12-31

XiPL脚本

- 脚本分为4类
 - 1. 表单脚本
 - 定义界面中**控件**的类型、位置，定义**数据的来源、去向**等
 - 2. 窗体脚本
 - 定义界面中的**按钮、菜单、状态栏**等的类型、位置，定义界面所要**实现的逻辑功能**
 - 3. 主窗体脚本
 - 用于定义主窗体的**名称、图标资源和菜单资源**
 - 4. 皮肤脚本
 - 定义应用程序中使用到的**图片资源**，界面颜色风格，**菜单、按钮、区域的颜色风格**

用XiPL脚本实现

江苏长江电器股份有限公司ERP系统 v1.0 - 脚本培训演示

退出 数据 保存 可写 只读 部门 查询

树状对象举例

根节点 (000)
 一级节点A (010)
 二级节点AA (011)
 二级节点AB (012)
 二级节点AC (013)
 一级节点B (020)
 一级节点C (030)

职员编码 ZY001 职员姓名

部门 电子商务部 薪资

入职日期 2003-09-05 资料文件名

序号	职员编码	职员姓名	性别
1	ZY001	职员A	男
2	ZY002	职员B	女

XiMainFormScript.ms XiFaceScript.ks wJbpxys.fs sJbp

1 主窗体：江苏长江电器股份有限公司ERP系统

2 图标 = XiApp

3 菜单表 = TxMenu

4 块尾

5

所属部门 PDM部门 薪资下限

入职日期下限 1999-01-01 入职日期上限

XiMainFormScript.ms XiFaceScript.ks wJbpxys.fs

1 皮肤列表：FaceList

2 当前皮肤 = CurrentFace

3 皮肤：CurrentFace

4 登录图片 = M_Login

5 主窗体图片 = m_btl

6 菜单背景色 = \$00E7928F

7 菜单加亮色 = \$00A03838

8 菜单字体颜色 = \$00FFFFFF

9 工具栏颜色 = \$00E1D5C9

10 背景色 = \$00F2F3F1

11 只读区域字体色 = \$00000000

12 只读区域背景色 = \$00F2F3F1

13 编辑区域字体色 = \$00000000

14 编辑区域背景色 = \$00FFFFFF

15 表格标题背景色 = \$00ECDAC6

16 必填区域标题色 = \$00B85F00

17 普通区域标题色 = \$00000000

18 焦点颜色 = \$00E7AE8C

19 指向颜色 = \$0031B2FF

20 边框颜色 = \$00733C00

21 页选中色 = \$00FFF5F0

22 一级菜单选中 = M_yjxz

23 一级菜单普通 = M_yjyd

24 一级菜单指向 = M_yjsq

25 块尾

26 块尾

树状对

江苏长江电器股份有限公司ERP

退出 数据 保存 可写

树状对象举例

根节点 (000)

一级节点A (010)

二级节点AA (011)

二级节点AB (012)

二级节点AC (013)

一级节点B (020)

一级节点C (030)

表单 : sJbpxys

版本 = 1.0.0.1

打印模板 =

页数 = 2

页名 = 第一页, 第二页

页左边界 = 210

页上边界 = 100

页宽度 = 558

页高度 = 270

状态 =

名称 =

树状对象 : tJbpxys

//中文名称 : 树状对象

根节点名称 = 树状对象举例

数据库表 = TDTreeNode

记录ID = tn_id

本地ID = fID

编码 = tn_bm

本地编码 = fBm

父节点 = tn_sjid

本地父节点 = fSjID

分类ID =

本地分类ID =

分类编码 =

末级标识 =

本地末级标识 =

排序字段 = fID

树节点序号 = fSjdxh

正常来源 = [S?TDTreeNode?tn_id]

选择方式 = 1

审计名称 =

提示 = 树状对象举例

区域显示格式 = [L:tJbpxys:fMc] ([L:tJbpxys:fBm])

页 = 0

左边界 = 5

上边界 = 5

宽度 = 200

高度 = 430

切换顺序 = 0

区域 : fID

类型 = i

正常来源 = [A?tn_id]

权限许可 =

块尾

区域 : fSjID

类型 = i

正常来源 = [A?tn_sjid]

权限许可 =

块尾

区域 : fMc

类型 = s(60)

正常来源 = [A?tn_mc]

权限许可 =

块尾

区域 : fBm

类型 = s(32)

正常来源 = [A?tn_bm]

权限许可 =

块尾

块尾

多记录对象

一级节点A (010)
 二级节点AA (011)
 二级节点AB (012)
 二级节点AC (013)
 一级节点B (020)
 一级节点C (030)

部门: 电子商务部 薪资: 1,500.00 职称: 助理工程师
 入职日期: 2003-09-05 资料文件名: AutoUpdate.exe

序号	职员编码	职员姓名	性别	部门	薪资	职称
1	ZY001	职员A	男	电子商务部	1,500.00	助理工程师
2	ZY002	职员B	女	研发中心	9,000.11	高级工程师

多记录对象 : mJbpxys

区域 : fID

区域 : fxB

中文名称 = 多记录对象演示

数据库表 = TJzy

记录ID = zy_id

本地ID = fID

序号字段 =

本地序号 =

名称字段 =

本次名称 =

编码字段 =

本地编码 =

提示区域 = fBm

检查区域 =

关联主表 =

关联区域 =

关联ID =

帐目 =

帐目ID =

帐目关联 =

帐目方式 =

正常来源 = [S?Tjzy?zy_id??left outer j

on zy_bmid = bm_id where zy_sjid=[L:tJ

转录来源 =

选择方式 = 1

审计名称 =

页 = 1

行指示器 = 1

左边界 = 0

上边界 = 0

宽度 = 558

高度 = 250

切换顺序 = 0

是否分页 = 否

每页行数 = 8

分段获取 = 否

标题 = 序号

类型 = i

允许为空 = 是

正常来源 = [A?zy_id]

转录来源 =

提交去向 =

审核去向 =

帐目去向 =

输入方式 = 直接

输入掩码 =

按钮图标 =

提交约束 =

修改约束 =

审核约束 =

默认值 =

允许输入 = @0

隐藏条件 = @0

权限许可 =

宽度 = 50

提示 =

显示格式 =

标题 = 性别

类型 = i

允许为空 = 是

正常来源 = [A?zy_xb]

转录来源 =

提交去向 =

审核去向 =

帐目去向 =

输入方式 = 选择

输入掩码 =

按钮图标 = btn_xlxx

提交约束 =

修改约束 =

审核约束 =

默认值 =

允许输入 = @1

隐藏条件 = @0

权限许可 =

宽度 = 50

提示 =

显示格式 = (女,男)

块尾

块尾

江苏长江电器股份有限公司ERP系统 v1.0 - 脚本培训演示

退出 数据 保存 可写 只读 部门 查询

树状对象举例

- 根节点 (000)
 - 一级节点A (010)
 - 二级节点AA (011)
 - 二级节点AB (012)
 - 二级节点AC (013)
 - 一级节点B (020)
 - 一级节点C (030)

职员编码: ZY001 职员姓名: 职员A 性别: 男

部门: 电子商务部 薪资: 1,500.00 职称: 助理工程师

入职日期: 2003-09-05 资料文件名: AutoUpdate.exe

序号	职员编码	职员姓名	性别	部门	薪资	职称
1	ZY001	职员A	男	电子商务部	1,500.00	助理工程师

单记录对象 : bJbpxys

区域 : fID

中文名称 = 单记录对

数据库表 = TJzy

记录ID = zy_id

本地ID = fID

名称 = zy_xm

本地名称 = fXm

编码 = zy_bm

本地编码 = fBm

状态 =

检查区域 =

本地状态 =

关联主表 =

关联区域 =

关联名称 =

关联编码 =

关联状态 =

上级区域 =

类别区域 =

标题 = ID

类型 = i

允许为空 = 是

正常来源 = [S:TJzy?zy_id,zy_xm,zy_bm,zy_xb,zy_zc,zy_bmid
zy_xz,zy_rzrq,zy_zlwjm,zy_zlwj?zy_id=[L:mJbpxys:fID]]

转录来源 =

提交去向 =

审核去向 =

帐目去向 =

输入方式 = 直接

输入掩码 =

按钮图标 =

提交约束 =

修改约束 =

审核约束 =

默认值 = 0

允许输入 = @1

隐藏条件 = @1

单记录

区域 : fBmQ

标题 = 所属部门

类型 = s (32)

允许为空 = 是

正常来源 =

转录来源 =

提交去向 =

审核去向 =

帐目去向 =

输入方式 = 直接 + 选择

输入掩码 =

按钮图标 = btn_xz

提交约束 =

修改约束 =

审核约束 =

默认值 =

允许输入 = @1

隐藏条件 = @0

权限许可 =

组 = gBottom

区域 : fBm

标题 = 职员编码

类型 = s (32)

允许为空 = 否

正常来源 = [B?zy_bm]

转录来源 =

提交去向 = zy_bm

审核去向 =

帐目去向 =

输入方式 = 直接

输入掩码 =

按钮图标 =

提交约束 =

修改约束 =

审核约束 =

默认值 =

允许输入 = @1

隐藏条件 = @0

权限许可 =

组 = gTop

系统 v1.0 - 脚本培训演示

部门 查询

职员编码 ZY001 职员姓名 职员A

部门 电子商务部 薪资 1,500.00

入职日期 2003-09-05 资料文件名 AutoUpdate.exe

序号	职员编码	职员姓名	性别	部门
1	ZY001	职员A	男	电子商务部
2	ZY002	职员B	女	研发中心

第一页 / 第一页 /

所属部门 PDM部门 薪资下限 0.00 薪资上限 10,000.00

入职日期下限 1999-01-01 入职日期上限 2004-12-31

窗体脚本



按钮 : btnExit

图标 = btn tc

文本 = 退出

提示 = 退出当前窗口

事件 = CloseForm

块尾

```
// 功能：导入树状对象的数据
```

事件 : LoadTreeObjData

条件 -

状态 // 功能：单记录对象性别区域的下拉按钮单击事件

事件 : bJbpxys.fXb.OnClick

条件 =

状态许可 =

后续状态 =

热键 =

///-----

// 功能：下拉出性别选择

///-----

动作 : ListSex

条件 =

动作 = .SelectBytesProc, bJbpxys, fXb

块尾

块尾

按钮 : btnLoadData

图标 = btn cx

文本 = 数据

提示 = 导入树状对象的数

事件 = LoadTreeObjData

块

块尾

块尾

按钮 : btnLoadData

图标 = btn bc

文本 = 保存

提示 = 保存单记录对象的数据

事件 = SaveMastData

块尾

- XiPL脚本是一种典型的DSL

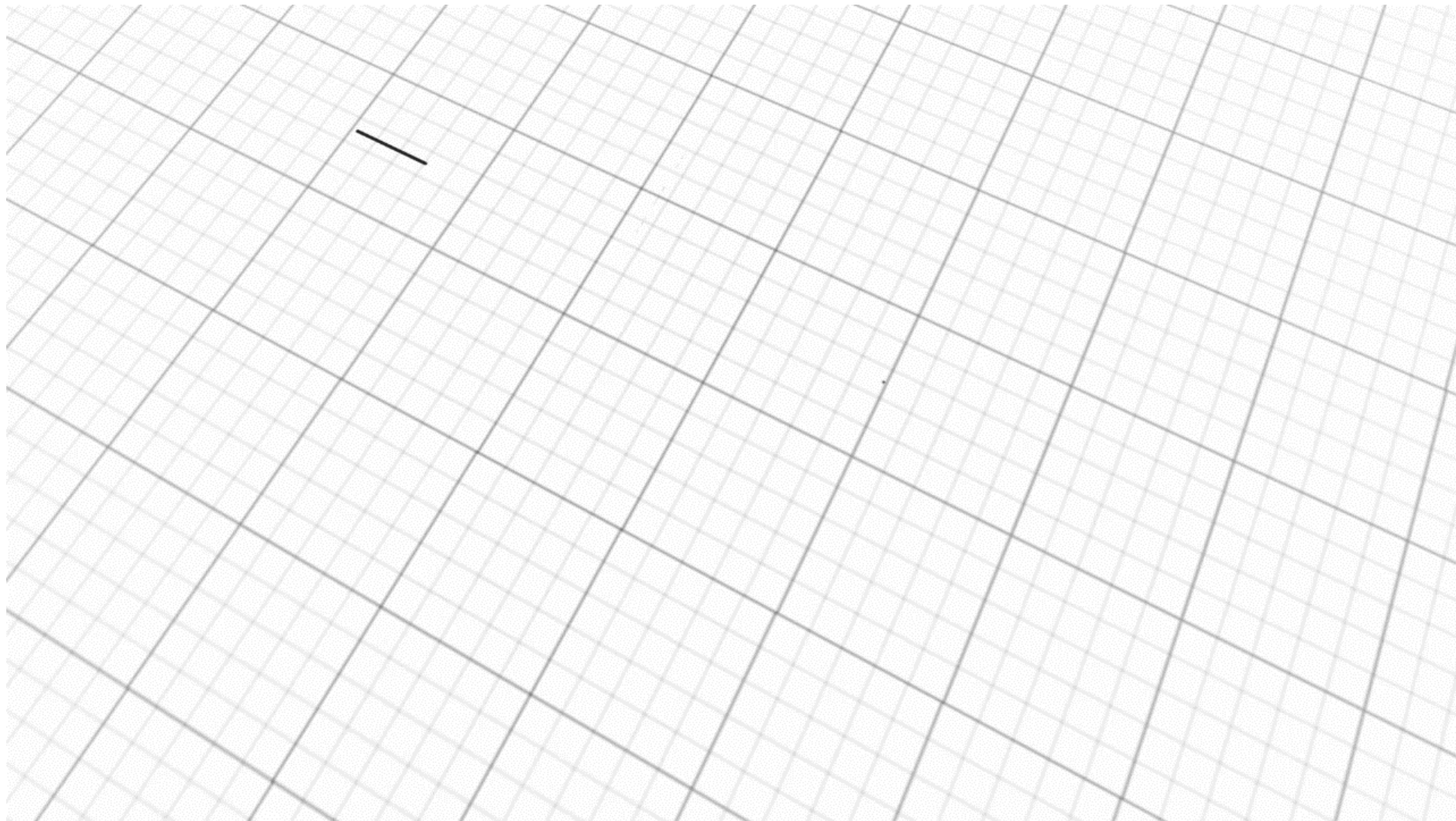
XiPL脚本-效果

- 使用XiPL脚本给公司带来的好处
 - 高中生或中专生，培训1周，即可完成基本的MIS系统开发工作
 - 降低人力成本
 - 大大减少了程序的bug
- 使用XiPL脚本带来的问题
 - 对于一些特殊的场合，XiPL脚本的表达能力不够
 - 需要底层平台暴露出更多的内容，以完成对应的工作。
 - XiPL脚本的更新问题
 - 为满足特殊场合增加的内容，使得XiPL脚本变得复杂。。。

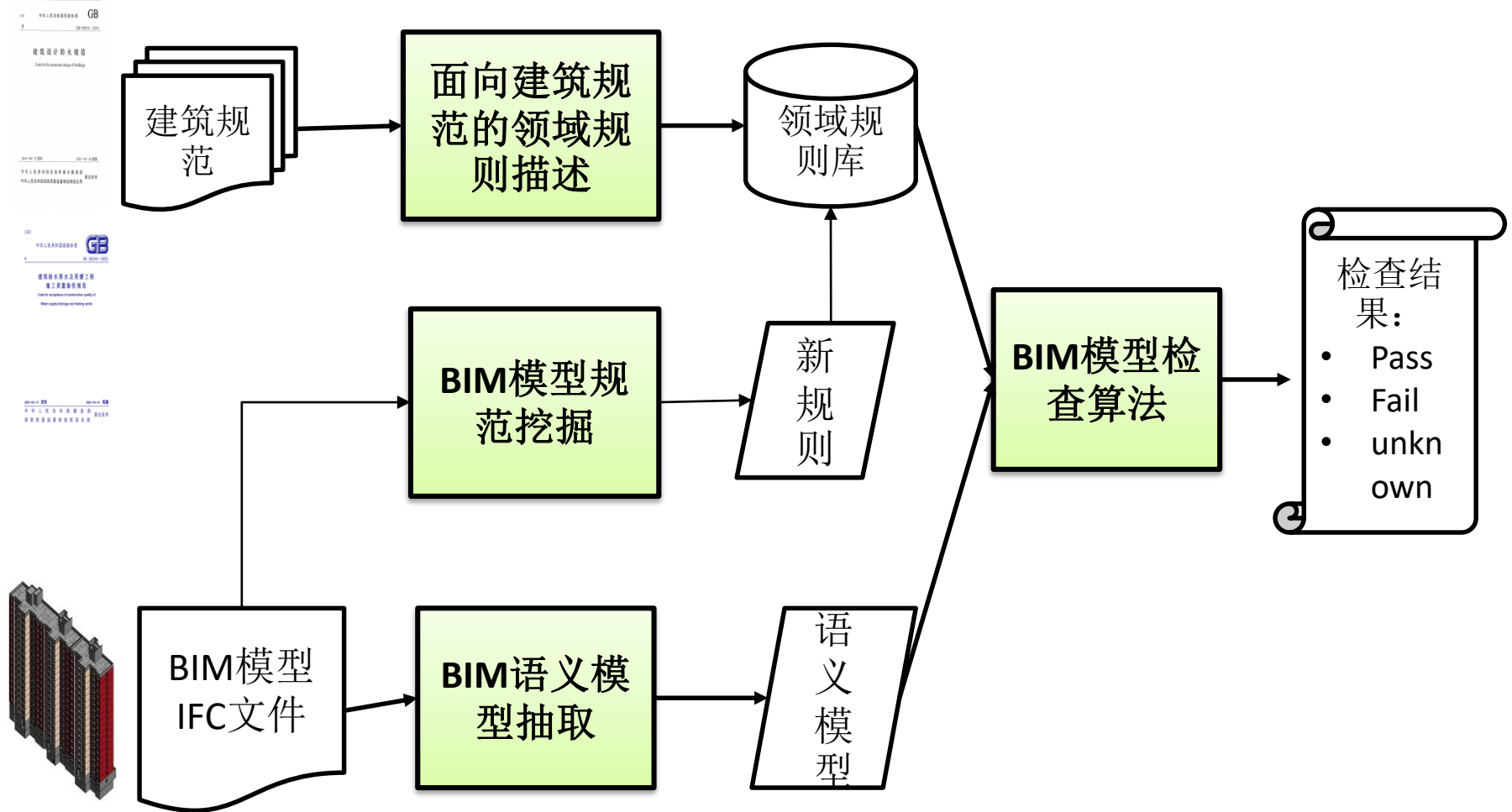
XiPL脚本总结

- XiPL脚本语言特点
 - 文本形式编写，自动生成界面；
 - 封装了业务逻辑的实现细节（C语言、Delphi、Java）；封装了复杂的数据库访问SQL语句，提供简单的业务原语
- 基于XiPL脚本的开发特点
 - 大量的拷贝粘贴
 - 业务层面进行描述
 - 简单且快速

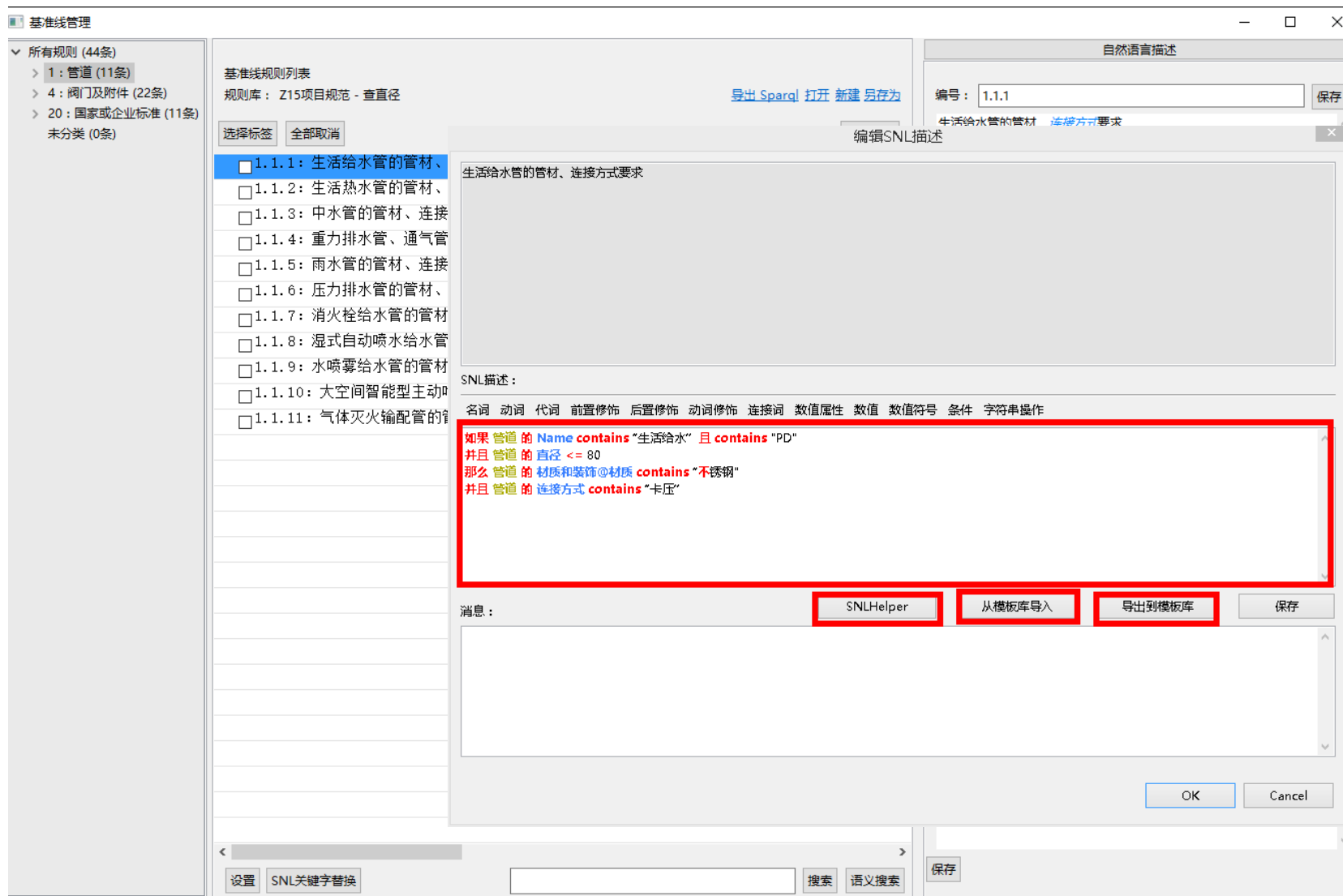
案例2：建筑领域规范描述和检查



BimChecker框架



建筑领域规范描述语言SNL



1. 防火规范条款6.4.10

- 6.4.10: 疏散走道在防火分区处应设置常开甲级防火门。
- DSL描述:
 - 如果 建筑 有 疏散走道 并且 防火分区 边界有 墙 并且 墙 处于 疏散走道 并且 墙 有 门 并且 门 处于 疏散走道 那么 门 的 Name contains “甲级”
 - 如果 建筑 有 过道 并且 防火分区 边界有 墙 并且 墙 处于 过道 并且 墙 有 门 并且 门 处于 过道 那么 门 的 Name contains “甲级”

防火规范：条款6.4.10

- 模型：LTKA-LJ-1-AR-旅检楼-B1-20161024.rvt
- 规范符合性检查

CheckResult

截取图片 导出检查报告

检查结果汇总

规范共包含条款1条；自动检查条款1条；其中模型适用的条款1条。

通过率 0% (0条/1条)；不通过率 100% (1条/1条)；不符合要求的构件总数目为1个。

-----检查结果分类统计-----

4硬性错误：通过率 0% (0条/1条)；不通过率 100% (1条/1条)。

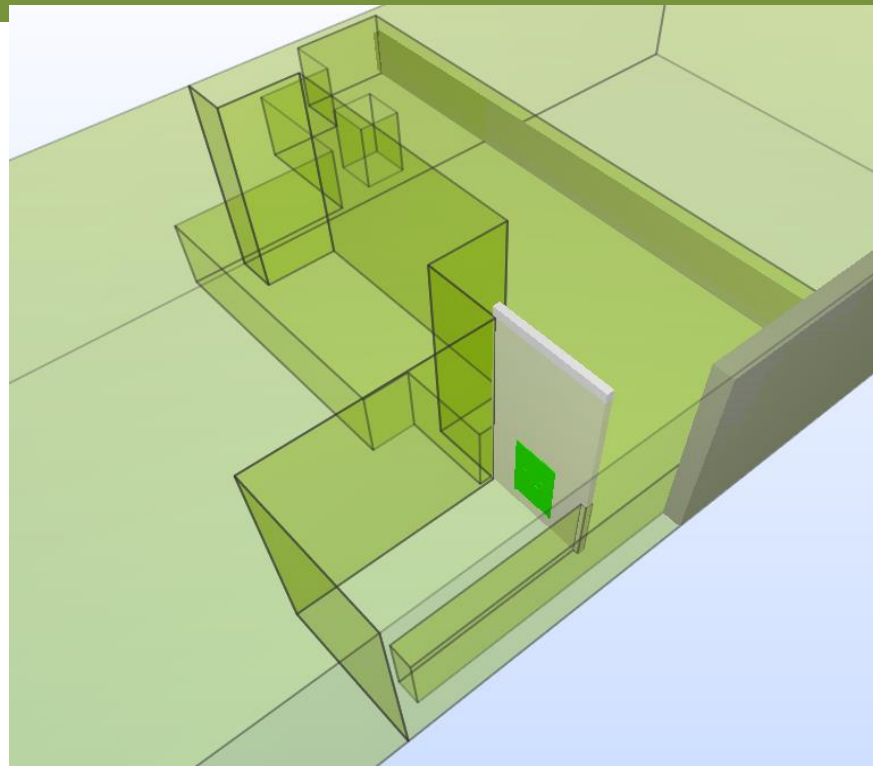
条款编号	条款内容	检查结果	错误类型	错误个数	错误清单	轴网信息	错
6.4.10	疏散走道在防火分区处应设置常开甲级防火门	未通过	4硬性错误	1个 门: 1个;	双扇防火门 门 J-乙级 防火门1300× 2300mm 1407281		不 火 道 那 级

检查结果：经过模型检查工具的自动检查，发现了莲塘口岸-B1模型在过道和防火分区相交汇的门的防火等级不符合国标要求，根据《建筑设计防火规范》要求，门的防火等级应为甲级，查出1个门的名称里包含的防火等级是乙级。

防火规范：条款6.4.10

示例：双扇防火门 J-乙级防火门-
1300X2300是走道在防火分区处的门，但是它的防火等级是乙级，不满足规范的要求。

Info	
Door. -2.82	
Hyperlinks	
Identification	Location
Quantities	Material
Relations	Classification
Property	Value
Model	LTKA-LJ-1-AR-旅检楼-B1-20161024
Discipline	Architectural
Name	双扇防火门: J-乙级防火门1300×2300mm:1407281
Type	J-乙级防火门1300×2300mm
Description	
Operation	Double Door Single Swing
Layer	A-DOOR-___-OTLN
System	
Building Envelope	True
Geometry	Boundary Representation
Application	Autodesk Revit 2015 (CHS)
GUID	2Qsd20w0b7QBFVFFOH1L7
BATID	1407281



2. 防火规范条款8.5.4

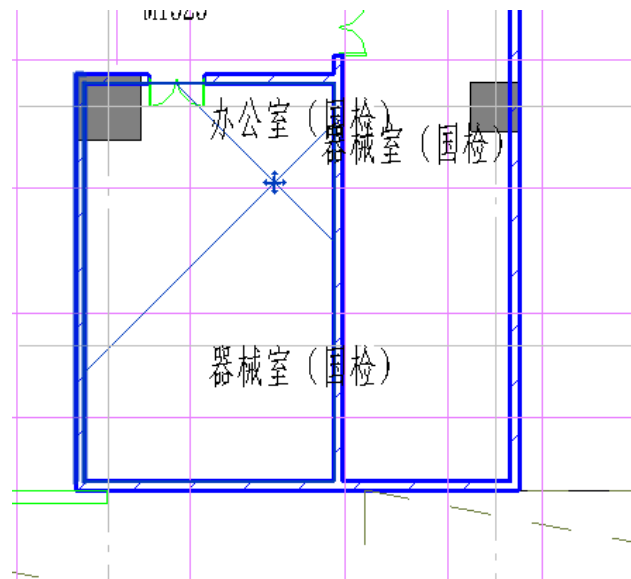
- 8.5.4: 地下或半地下建筑（室）、地上建筑内的无窗房间，当总建筑面积大于200m²或一个房间建筑面积大于50m²，且经常有人停留或可燃物较多时，应设置排烟设施。
- DSL描述：
 - 如果 建筑 有 房间 并且 房间的 计算标高 > 0 并且 房间 不 有 窗 并且 房间的 面积 > 50 那么 房间 有 排烟风口 或有 排烟设备 或有 排烟系统
 - 如果 建筑 有 房间 并且 房间的 计算标高 < 0 并且 房间的 面积 > 50 那么 房间 有 排烟风口 或有 排烟设备 或有 排烟系统
 - 如果 建筑 的 建筑名称 contains “地下” 并且 建筑 的 总-楼层-面积 > 200 那么 建筑 有 排烟风口 或有 排烟设备 或有 排烟系统
 - 未检查部分: “经常有人停留或可燃物较多”

防火规范：条款8.5.4

- 模型：模型：LTKA-LJ-1-AR-旅检楼-1F-20160505.rvt链接LTKA-LJ-1-JD-旅检楼主体-F1.rvt
- 建筑名称：
- 规范符合性检查

2个 房间: 2个;	房间 办公室 (国检) 1769 8635909	▼	不满足: 如果 建筑 有 房间 并且 房间 的 计算标高 > 0 并且 房间 不 有 窗 并且 房间 的 面积 > 50 那么 房间 有 排烟风口 或有 排烟设备 或有 排烟系统;	经常有人停留或可燃物较多
---------------	-----------------------------	---	--	--------------

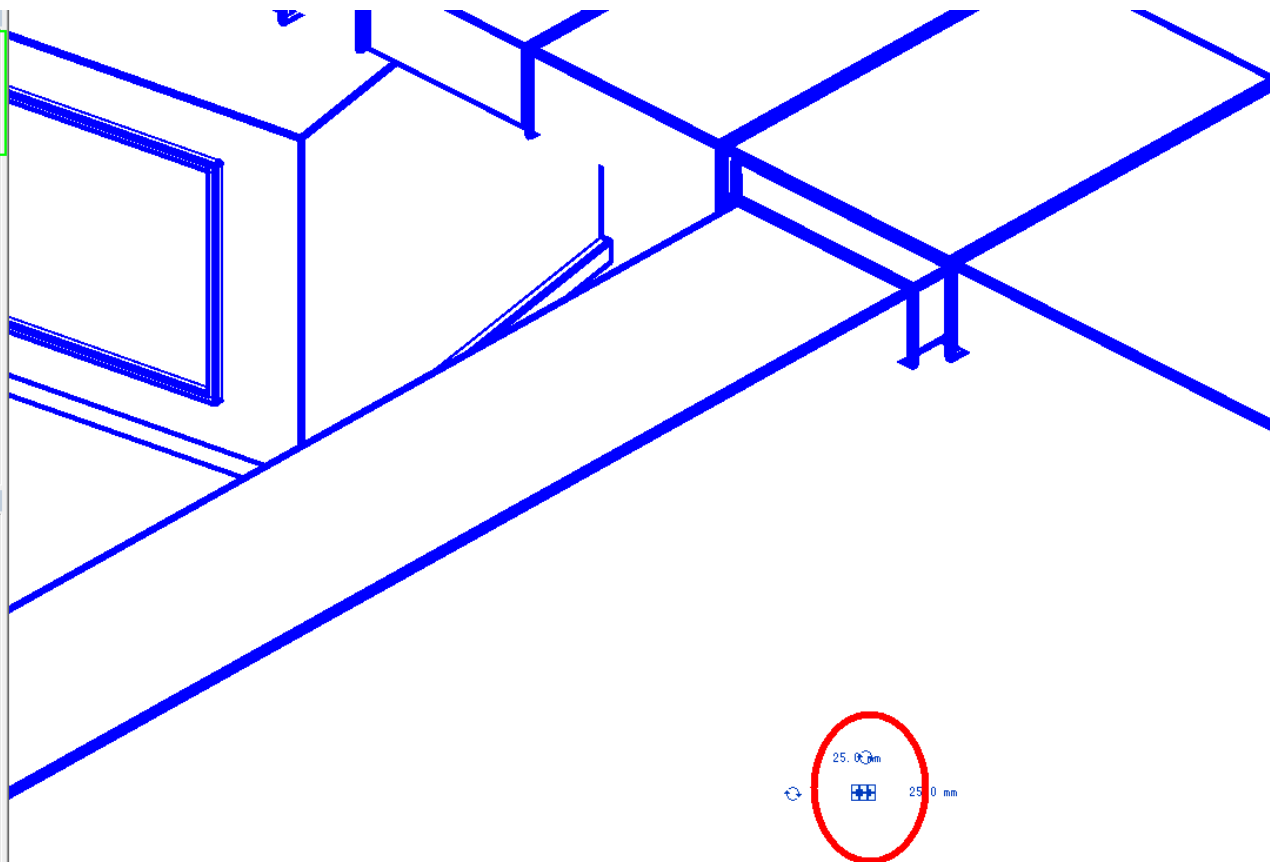
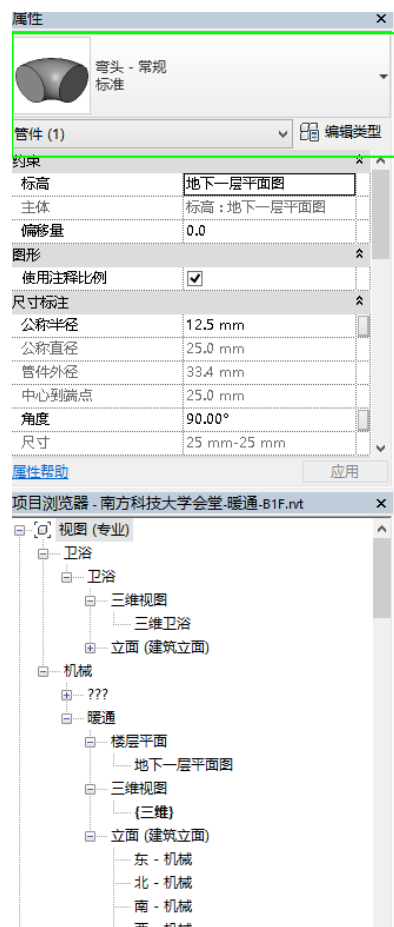
检查结果：国检办公室
(8635909)，没有排烟设置，
不符合规范要求



3. 独立构件问题

- 规则分析
 - 工具中通过距离计算，进行检查。若管件、风管道等与其附近的管道、风管距离过远，则认为其存在独立构件问题。
- 工具中，具体化为6条规则描述
 - 所有 机械设备的 机械设备-风管道最小距离 <50
 - 所有 管道附件的 管道附件-管道最小距离 <50
 - 所有 管件的 管件-管道最小距离 <50
 - 所有 风管附件的 风管附件-风管件最小距离 <50
 - 所有 风道末端的 风道末端-风管最小距离 <50
 - 所有 风管道管件的 风管道管件-风管附件风管最小距离 <50

• 出问题的构件管件（714955）及截图



4. 硬碰撞问题

- 规则分析
 - 工具中通过距离计算，进行检查。即，若管件、风管等两两之间的距离 <0 ，那么则认为其碰撞问题。
- 规则描述：使用工具中封装的“不穿越”关键词表示为6条规则
 - 所有管道不穿越管道
 - 所有管道不穿越风管
 - 所有风管不穿越风管
 - 所有管道不穿越风管管件
 - 所有管件不穿越风管
 - 所有管件不穿越风管管件

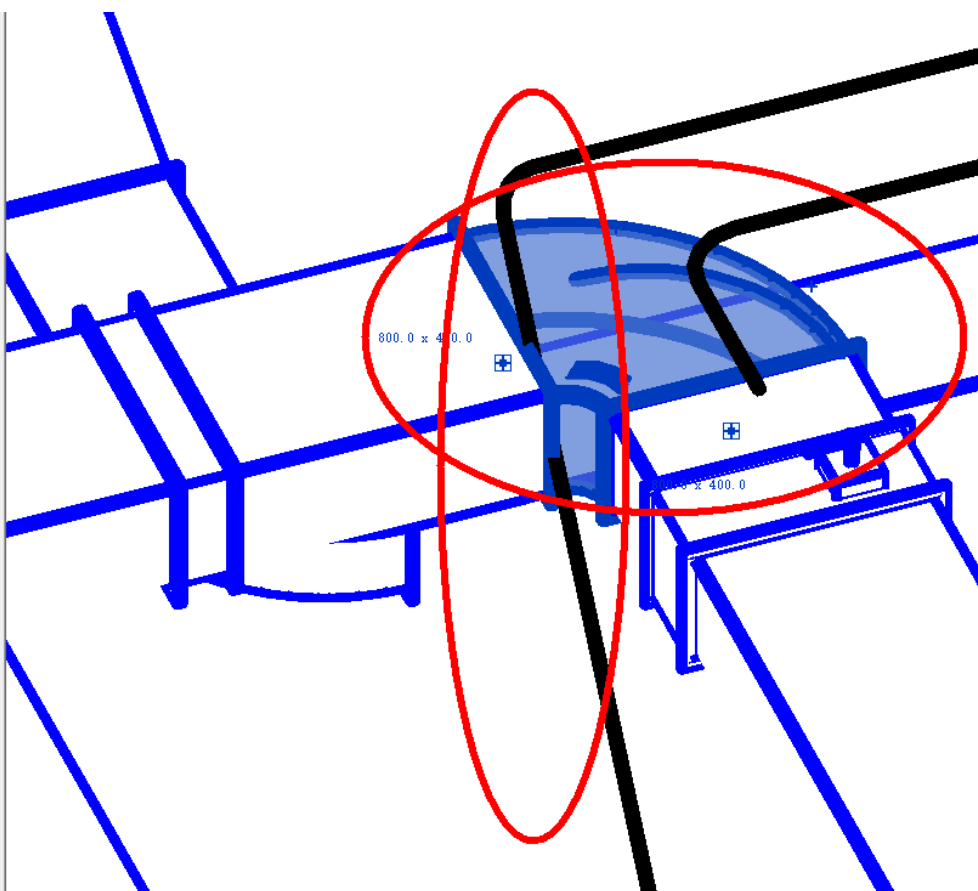
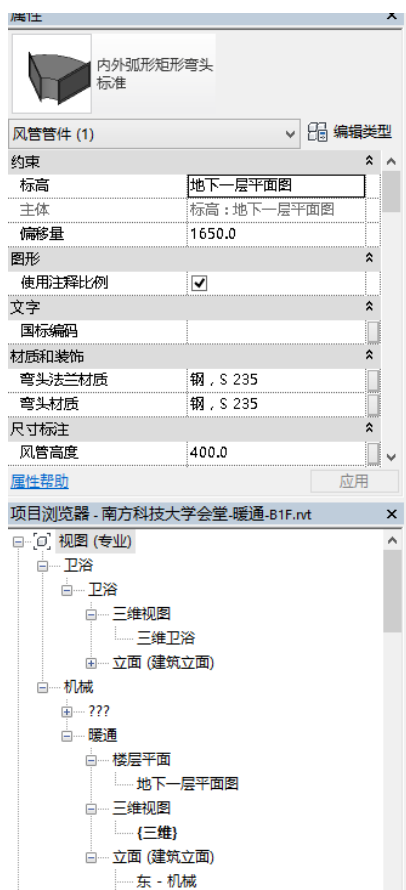
• BimChecker的检查结果

- 风管管件与管道碰撞（1处：确认1处）
- 管道与管道碰撞（2处：确认1处）
- 管道与风管碰撞（4处：确认3处）
- 管件与风管碰撞（2处：确认2处）
- 风管与风管碰撞（13处：确认13处）

条款编号	条款内容	检查结果	错误类型	错误原因	错误个数
1.1.1	构件冲突检查	未通过	4硬性错误	所有 管道 不穿越 风管管件	风管管件: 1个;管道: 1个;
		未通过	4硬性错误	所有 风管 不穿越 风管	风管: 23个;
		未通过	4硬性错误	所有 管道 不穿越 管道	管道: 3个;
		未通过	4硬性错误	所有 管道 不穿越 风管	风管: 3个;管道: 3个;
		未通过	4硬性错误	所有 管件 不穿越 风管	风管: 1个;管件: 2个;

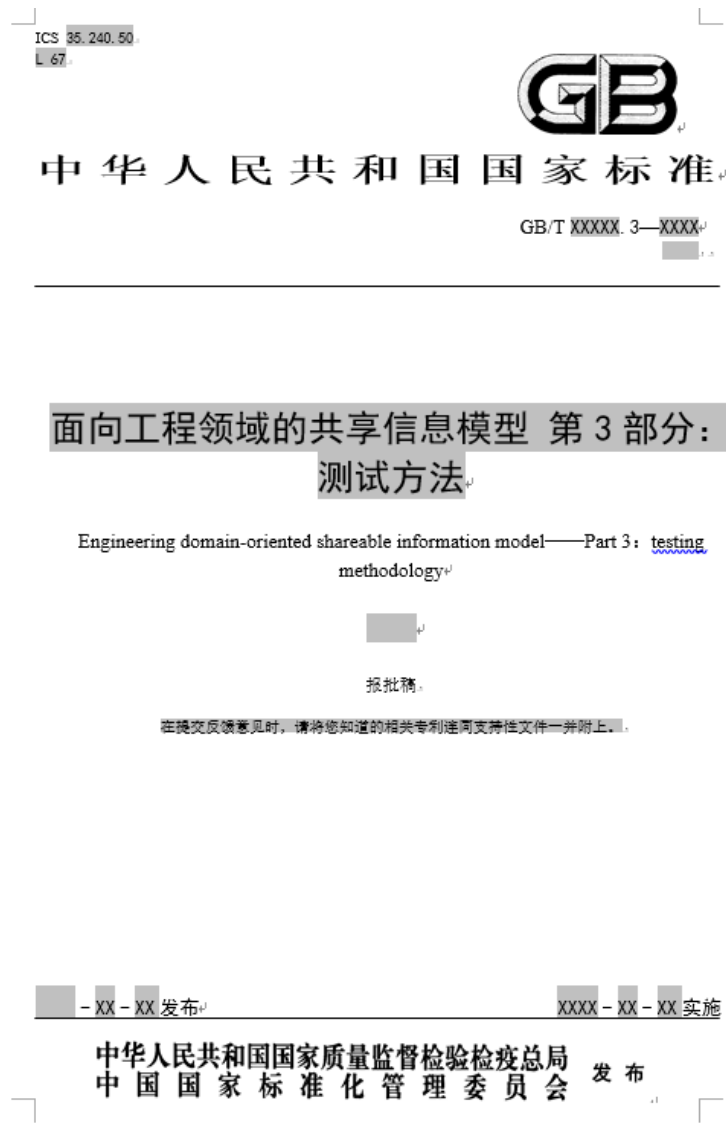
风管管件与管道碰撞（1处）

ID	类	名称
(792489, 808808)	风管管件 管道	标准



应用和推广： 国家标准

- 《面向工程领域的共享信息模型》
 - 第1部分：领域信息模型框架
 - 第2部分：服务接口
 - 第3部分：测试方法



目次	
前言	
面向工程领域的共享信息模型 第 3 部分：测试方	
1 范围	
2 规范性引用文件	
3 术语和定义	
4 概述	
5 一般要求	
6 测试过程概述	
7 判定方法	
8 报告生成	
9 形式化描述语言	
10 测试	
附 录 A （资料性附录） 形式化描述语言使用	

附 录 A (资料性附录) 形式化描述语言使用实例

A.1 简单句

所有 建筑 的 高度 > 60
存在一个 管道 A 的 直径 > 55
存在一个 建筑 B 的 高度 ≤ 60 并且 ≥ 45

总结：简单句的语法结构可以详细参考第一部分语法内容；需要注意的地方就是逻辑关键词的使用，逻辑关键词会在下面详细举例介绍。（特别注意的是，在使用“存在一个”关键词时，构件后面必须有代词连接，如上例所示）

A.2 复合句

不存在 建筑 的 高度 ≤ 60 或者 > 80
并且 所有 建筑 的 耐火等级 匹配“一级”
并且 存在一个 建筑 A 的 层数 ≥ 8

存在一个 楼板 A 的 耐火极限 ≥ 2
并且 所有 楼板 的 耐火等级 匹配“二级”

总结：复合句是由多个简单句组成的，中间使用逻辑连接词“并且”连接，可以连接任意个简单句，根据用户的实际需要进行撰写，关于“并且”这个关键词的介绍会在后面具体示例。

A.3 条件句

如果 管道 的 名称 包含“中水”
并且 管道 的 直径 ≤ 80
那么 管道 的 管道材质 包含“不锈钢”
并且 管道 的 连接方式 包含“卡压”

如果 阀门 的 名称 包含“给水” 或者 包含“排水” 并且 包含“泄压阀” 并且
阀门 的 直径 ≤ 50
那么 阀门 的 阀体材质 匹配“青铜” 并且
阀门 的 连接方式 匹配“螺纹连接”

DSL介绍的内容

- 为什么？（Why）
- 是什么？（What）
- 怎么做？（How）
- 怎么用？（Where）

谢谢！
