



英睿信息
enways.com

软件测试高级技术

——Pairwise Testing

Better process
Better products
Better service

个人简介

Better process ,Better products, Better services

□ 杨瑞

- 厦门英睿信息科技有限公司联合创始人
- 复旦大学软件工程硕士，同济大学自动控制学士
- 10年软件工程经验，8年过程改进经验，在基于传统和敏捷的开发管理方面具有多年经验，目前是国内敏捷社区内的活跃人士。
- 曾在三五互联、锐捷网络、福州网龙等知名上市公司担任质量经理、研发总监和 EPG Leader等职位。



培训提纲

Better process ,Better products, Better services

- Pairwise Testing理论

- PICT工具介绍

PAIRWISE TESTING

Better process
Better products
Better service

Pairwise Testing理论

- Pairwise Testing称为“成对测试”、“两两测试”或者“结对测试”；
- 和Pairwise Testing类似的还有“Combinatorial Method”、“Orthogonal Arrays”、“All-Pairs Testing”等测试方法；

Pairwise Testing理论

Better process, Better products, Better services

□测试工作的目的

- 最大限度的发现系统中存在的问题，即我们一般所称的“缺陷”（bug）

□测试管理所要解决的问题

- 测试覆盖率
- 测试效率
- 测试工作量

Pairwise Testing理论

- 在不可能对所有测试产品功能点进行覆盖、遍历的情况下，如何用最少的工作量发现最多的缺陷，即“测试性价比”问题就显现出来；

Pairwise Testing理论

- Pairwise Testing能够有效的应对复杂环境下的测试；
- Pairwise Testing能够有效的提高“测试性价比”；
- Pairwise Testing关注与系统的“变量”，同时考虑对“变量”所有可能的“取值”；

Pairwise Testing理论

□ “*Cartesian Products*”（笛卡尔积）

- 给定一组域 $D_1, D_2, \dots, D_n$ ，这些域中可以有相同的；
- 所有域的所有取值的一个组合，不能重复；
- 假设集合 $A=\{a,b\}$ ，集合 $B=\{0,1,2\}$ ，则两个集合的笛卡尔积为 $\{(a,0), (a,1), (a,2), (b,0), (b,1), (b,2)\}$ 。

Pairwise Testing理论

□ 如何测试？

■ GUI 应用程序 V1.0

◆ List1 (0, 1, 2)

◆ List2 (10, 20)

◆ “OK”按钮，点击后计算

List1	List2
0	10
1	20
2	

Pairwise Testing理论

□笛卡尔积

■共有 $2 \times 3 = 6$ 个组合

List1	List2
0	10
0	20
1	10
1	20
2	10
2	20

Pairwise Testing理论

enter process , Better products, Better services

□ 如何测试？

■ GUI 应用程序 V2.0

- ◆ List (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- ◆ Textbox (1~99)
- ◆ Checkbox1 (-X) (相反数)
- ◆ Checkbox2 (X*X) (平方)
- ◆ “OK”按钮，点击后计算

Pairwise Testing理论

笛卡尔积

■ 共有 $10 \times 99 \times 2 \times 2 = 3960$ 个组合

List	Textbox	Checkbox1 (-x)	Checkbox2 (x ²)
0	1	On	On
1	2	Off	Off
2	3		
3	4		
4	...		
5	96		
6	97		
7	98		
8	99		
9			

Pairwise Testing理论

- GUI 应用程序 V1.0测试简单，在有限的时间
内能够完成所有的遍历情况；
- GUI 应用程序 V2.0测试复杂，在有限的时间
内不能够完成所有的遍历情况；
- 如何测试V2.0这样的应用程序？

Pairwise Testing理论

□ 如何测试V2.0这样的应用程序？

■ 通过简单的观察，我们很容易的发现一个“BUG”

:

◆ $0 \times (-1) = ? ? ?$

■ 再深入分析一下，如果做“遍历”组合测试，当List=0时，Checkbox1=On时，以下的取值并不影响测试结果：

◆ Textbox

◆ Checkbox2

Pairwise Testing理论

□如何测试V2.0这样的应用程序？

■那么我们浪费的工作量等于多少（需要设计和执行的用例）？

◆99个

◆如果平均每个用例的设计和执行需要一个工时，则需要浪费的工作量近100个工时，即一个人需要工作2周半

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ “随机”抽检测试

- ◆ $1/3960 \approx 1/4000 = 0.00025$

- ◆ 随机测试的概率基本为 0

■ “遍历”组合测试

- ◆ 3960个测试用例，约23.6个人月，即2个人年

- ◆ 过度的“遍历”组合测试将使测试无法进行

Pairwise Testing理论

□如何测试V2.0这样的应用程序？

- 大部分缺陷是在两个变量取值冲突的测试时发现的
- 不仅仅是在所有的组合情况下才会发现所有的测试缺陷
- 这个是“Pairwise Testing”基本原理，不要测试所有的组合，测试所有的“Pairwise”即可

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ 目标

- ◆ 减少测试用例，去除无效的测试用例
- ◆ 使用“等价类”方法，合并变量取值

Pairwise Testing理论

□ 如何测试V2.0这样的应用程序？

■ 步骤1:

◆ 列出各种“变量”的所有可能“取值”

List	Textbox	Checkbox1 (-x)	Checkbox2 (x ²)
0	Valid int	On	On
Any other	Invalid int	Off	Off
	Alpha		

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V2.0这样的应用程序？

■步骤2:

- ◆将“变量”的所有可能“取值”中，多于“2”的情况放在前面
- ◆将“变量”的所有可能“取值”中，等于“2”的情况放在后面

Textbox (3)	List (2)	Negative (2)	Square (2)

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ 步骤3：

◆ 完成“变量”的所有可能“取值”中，多于“2”的情况遍历组合（每行代表一个测试场景）

Textbox (3)	List (2)	Negative (2)	Square (2)
Valid integer			
Valid integer			
Invalid Integer			
Invalid Integer			
Alpha			
Alpha			

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ 步骤4：

◆ 完成第二列的“两两”情况

◆ 检查是否所有的“两两”都存在？

- (Valid, 0)、(Valid, Other)
- (Invalid, 0)、(Invalid, Other)
- (Alpha, 0)、(Alpha, Other)

Textbox (3)	List (2)	Negative (2)	Square (2)
Valid integer	0		
Valid integer	Other		
Invalid Integer	0		
Invalid Integer	Other		
Alpha	0		
Alpha	Other		

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V2.0这样的应用程序？

■步骤5（续）：

◆完成第三列的“两两”情况

◆“两两”重复没关系，关键是不能有遗漏

Textbox (3)	List (2)	Negative (2)	Square (2)
Valid integer	0	On	
Valid integer	Other	Off	
Invalid Integer	0	On	
Invalid Integer	Other	Off	
Alpha	0	On	
Alpha	Other	Off	

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V2.0这样的应用程序？

■步骤5：

◆检查是否所有的“两两”都存在？

●第一列&第三列

- (Valid , On) 、 (Valid , Off)
- (Invalid , On) 、 (Invalid , Off)
- (Alpha , On) 、 (Alpha , Off)

●第二列&第三列

- (0, On) 、 ~~(0, Off)~~
- ~~(Other, On)~~ 、 (Other, Off)

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ 步骤5（续）：

◆ 调整第三列的“两两”情况

Textbox (3)	List (2)	Negative (2)	Square (2)
Valid integer	0	On	
Valid integer	Other	Off	
Invalid Integer	0	Off	
Invalid Integer	Other	On	
Alpha	0	On	
Alpha	Other	Off	

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V2.0这样的应用程序？

■步骤5（续）：

◆检查是否所有的“两两”都存在？

●第一列&第三列

- （Valid，On）、（Valid，Off）
- （Invalid，On）、（Invalid，Off）
- （Alpha，On）、（Alpha，Off）

●第二列&第三列

- （0，On）、（0，Off）
- （Other，On）、（Other，Off）

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ 步骤6：

◆ 完成第四列的“两两”情况

Textbox (3)	List (2)	Negative (2)	Square (2)
Valid integer	0	On	Checked
Valid integer	Other	Off	Unchecked
Invalid Integer	0	Off	Checked
Invalid Integer	Other	On	Unchecked
Alpha	0	On	Unchecked
Alpha	Other	Off	Checked

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V2.0这样的应用程序？

■步骤6（续）：

◆检查是否所有的“两两”都存在？

●第一列&第四列

- （Valid，Checked）、（Valid，Unchecked）
- （Invalid，Checked）、（Invalid，Unchecked）
- （Alpha，Checked）、（Alpha，Unchecked）

●第二列&第四列

- （0，Checked）、（0，Unchecked）
- （Other，Checked）、（Other，Unchecked）

●第三列&第四列

- （On，Checked）、（On，Unchecked）
- （Off，Checked）、（Off，Unchecked）

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V2.0这样的应用程序？

■ 步骤6（续）：

◆ 检查是否所有的“两两”都存在？

● 其余列组合情况已经检查，不再进行检查

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V2.0这样的应用程序？

■通过“两两”组合（同时考虑等价类划分）我们已经极大的减少了测试用例的数量

◆我们只需要6个测试用例就能够进行有效的测试

◆采用“遍历”组合测试，需要的用例数为
 $3 \times 2 \times 2 \times 2 = 24$ 个

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试？

■ GUI 应用程序 V3.0

- ◆ List (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- ◆ Textbox (1~99)
- ◆ Checkbox1 ($-X$) (相反数)
- ◆ Checkbox2 ($X*X$) (平方)
- ◆ Checkbox3 ($! X$) (阶乘)
- ◆ Checkbox4 (Hex (X)) (十六进制)
- ◆ “OK”按钮, 点击后计算

Pairwise Testing理论

Better process, Better products, Better services

□笛卡尔积

■共有 $99 \times 10 \times 2 \times 2 \times 2 \times 2 = 15840$ 个组合

□GUI 应用程序 V3.0测试更复杂，在有限的时间
内不能够完成所有的遍历情况；

□如何测试V3.0这样的应用程序？

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V3.0这样的应用程序？

■ 步骤1~6同V2.0

■ 步骤7:

◆ 完成第五列的“两两”情况

Textbox (3)	List (2)	Negative (2)	Square (2)	Factorial (2)	Hex (2)
Valid integer	0	On	Checked	Yes	
Valid integer	Other	Off	Unchecked	No	
Invalid Integer	0	Off	Checked	No	
Invalid Integer	Other	On	Unchecked	Yes	
Alpha	0	On	Unchecked	No	
Alpha	Other	Off	Checked	Yes	

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V3.0这样的应用程序？

■步骤7（续）：

◆检查是否所有的“两两”都存在？

●第一列&第五列

- （Valid，Yes）、（Valid，No）
- （Invalid，Yes）、（Invalid，No）
- （Alpha，Yes）、（Alpha，No）

●第二列&第五列

- （0，Yes）、（0，No）
- （Other，Yes）、（Other，No）

●第三列&第五列

- （On，Yes）、（On，No）
- （Off，Yes）、（Off，No）

●第四列&第五列

- （Checked，Yes）、（Checked，No）
- （Unchecked，Yes）、（Unchecked，No）

Pairwise Testing理论

□ 如何测试V3.0这样的应用程序？

■ 步骤8：

◆ 完成第五列的“两两”情况

Textbox (3)	List (2)	Negative (2)	Square (2)	Factorial (2)	Hex (2)
Valid integer	0	On	Checked	Yes	Dec
Valid integer	Other	Off	Unchecked	No	Hex
Invalid Integer	0	Off	Checked	No	Hex
Invalid Integer	Other	On	Unchecked	Yes	Dec
Alpha	0	On	Unchecked	No	Dec
Alpha	Other	Off	Checked	Yes	Hex

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V3.0这样的应用程序？

■ 步骤8（续）：

◆ 检查是否所有的“两两”都存在？

● 第一列&第六列

- (Valid, Dec)、(Valid, Hex)
- (Invalid, Dec)、(Invalid, Hex)
- (Alpha, Dec)、(Alpha, Hex)

● 第二列&第六列

- (0, Dec)、(0, Hex)
- (Other, Dec)、(Other, Hex)

● 第三列&第六列

- (On, Dec)、(~~On~~, ~~Hex~~)
- (~~Off~~, ~~Dec~~)、(Off, Hex)

● 第四列&第六列

- (Checked, Dec)、(Checked, Hex)
- (Unchecked, Dec)、(Unchecked, Hex)

● 第五列&第六列

- (Yes, Dec)、(Yes, Hex)
- (No, Dec)、(No, Hex)

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V3.0这样的应用程序？

■ 步骤8（续）：

◆（On， Hex）、（Off， Dec）缺失，怎么办？

◆ 调换位置？

Textbox (3)	List (2)	Negative (2)	Square (2)	Factorial (2)	Hex (2)
Valid integer	0	On	Checked	Yes	Dec
Valid integer	Other	Off	Unchecked	No	Hex
Invalid Integer	0	Off	Checked	No	Hex
Invalid Integer	Other	On	Unchecked	Yes	Dec
Alpha	0	On	Unchecked	No	Dec
Alpha	Other	Off	Checked	Yes	Hex

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V3.0这样的应用程序？

■ 步骤8（续）：

◆ 如果调整Dex和Hex组合顺序，会破坏其他“两两”组合的情况；

Textbox (3)	List (2)	Negative (2)	Square (2)	Factorial (2)	Hex (2)
Valid integer	0	On	Checked	Yes	Dec
Valid integer	Other	Off	Unchecked	No	Hex
Invalid Integer	0	Off	Checked	No	Hex
Invalid Integer	Other	On	Unchecked	Yes	Dec
Alpha	0	On	Unchecked	No	Dec
Alpha	Other	Off	Checked	Yes	Hex

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V3.0这样的应用程序？

■步骤9：

◆增加两个测试用例，分别针对缺失的（On， Hex）和（Off， Dec）

Textbox (3)	List (2)	Negative (2)	Square (2)	Factorial (2)	Hex (2)
Valid integer	0	On	Checked	Yes	Dec
Valid integer	Other	Off	Unchecked	No	Hex
		On			Hex
Invalid Integer	0	Off	Checked	No	Hex
Invalid Integer	Other	On	Unchecked	Yes	Dec
		Off			Dec
Alpha	0	On	Unchecked	No	Dec
Alpha	Other	Off	Checked	Yes	Hex

Pairwise Testing理论

Better process, Better products, Better services

□ 如何测试V3.0这样的应用程序？

■ 步骤9：

◆ 其余列则无所谓

Textbox (3)	List (2)	Negative (2)	Square (2)	Factorial (2)	Hex (2)
Valid integer	0	On	Checked	Yes	Dec
Valid integer	Other	Off	Unchecked	No	Hex
<i>Valid integer</i>	<i>Other</i>	On	<i>Unchecked</i>	<i>No</i>	Hex
Invalid Integer	0	Off	Checked	No	Hex
Invalid Integer	Other	On	Unchecked	Yes	Dec
<i>Invalid Integer</i>	<i>Other</i>	Off	<i>Unchecked</i>	<i>Yes</i>	Dec
Alpha	0	On	Unchecked	No	Dec
Alpha	Other	Off	Checked	Yes	Hex

Pairwise Testing理论

Better process, Better products, Better services

□如何测试V3.0这样的应用程序？

- 通过“两两”组合（同时考虑等价类划分）我们已经极大的减少了测试用例的数量

- ◆我们只需要8个测试用例就能够进行有效的测试

- ◆采用“遍历”组合测试，需要的用例数为
 $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$ 个

- 每个测试用例可以覆盖多个“两两”组合

PICT工具介绍

Better process
Better products
Better service

PICT工具介绍

Better process ,Better products, Better services

□ Pairwise Independent Combinatorial Testing tool (PICT)

■ Jacek Czerwotka (jacekcz@microsoft.com) , Test Lead, Microsoft Corporation

■ 下载地址:

◆ <http://download.microsoft.com/download/f/5/5/f55484df-8494-48fa-8dbd-8c6f76cc014b/pict33.msi>

□ PICT不仅仅是“两两”测试工具，其可以生成多种组合测试结果

PICT工具介绍

Better process ,Better products, Better services

- PICT是一个命令行工具，其输入、输出都可以是“纯文本”形式的（输出也可以为MS Excel格式）
- PICT安装后，会在系统路径中增加其路径，因此可以在任意路径下使用

PICT工具介绍

Better process ,Better products, Better services

□ Usage: pict model [options]

■ Options:

- ◆ /o:N - Order of combinations (default: 2)
- ◆ /d:C - Separator for values (default: ,)
- ◆ /a:C - Separator for aliases (default: |)
- ◆ /n:C - Negative value prefix (default: ~)
- ◆ /e:file - File with seeding rows
- ◆ /r[:N] - Randomize generation, N - seed
- ◆ /c - Case-sensitive model evaluation
- ◆ /s - Show model statistics

PICT工具介绍

Better process ,Better products, Better services

□ PICT的初步使用

■ 简单形式

◆ C:\pict ModelFile.txt

■ 使用输出文件（TXT格式）

◆ C:\pict ModelFile.txt > OutputFile.txt

■ 使用输出文件（XLS格式）

◆ C:\pict ModelFile.txt > OutputFile.xls

PICT工具介绍

Better process ,Better products, Better services

□ PICT的输入文件

- 一般称为: Model Files

- 文件格式: 分为三个章节 (Sections)

 - ◆ Parameter definitions [必选]

 - ◆ Sub-model definitions [可选]

 - ◆ Constraint definitions [可选]

- 章节约束

 - ◆ 章节不能调整顺序

 - ◆ 章节不能出现重叠

 - ◆ 章节之间不必使用特殊的分割符号

 - ◆ 空行可以出现在任何位置

 - ◆ 注释行使用“#”标注

PICT工具介绍

Better process ,Better products, Better services

□ PICT的输入文件

■ 简单形式：每个参数占一行

◆ <ParamName> : <Value1>, <Value2>, <Value3>, ...

■ 默认的参数取值分隔符为“，”，但是可以使用“/d: ”进行更改

■ 默认的组合方式为“两两测试”，但是可以使用“/o: ”进行更改

■ 参数名称必须要唯一

■ 取值不必唯一

PICT工具介绍

Better process ,Better products, Better services

□ PICT的输入文件

■ 例子1:

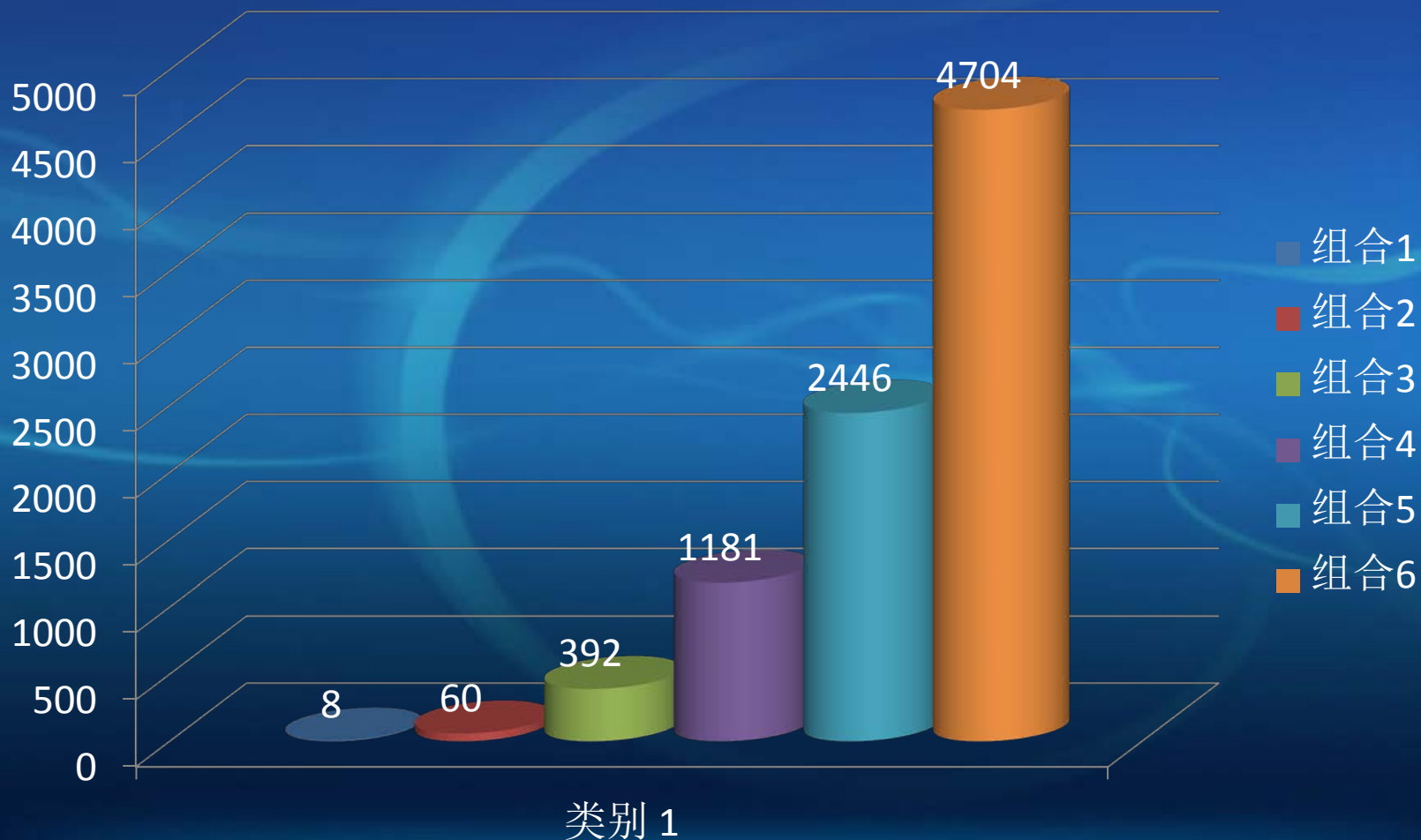
```
#  
# This is a sample model for testing volume create/delete functions  
#  
7  Type:          Primary, Logical, Single, Span, Stripe, Mirror, RAID-5  
7  Size:          10, 100, 500, 1000, 5000, 10000, 40000  
2  Format method: quick, slow  
3  File system:   FAT, FAT32, NTFS  
8  Cluster size: 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536  
2  Compression:  on, off
```

◆ 组合遍历测试用例= $7 \times 7 \times 2 \times 3 \times 8 \times 2 = 4704$ 个

PICT工具介绍

Better process ,Better products, Better services

□参数 数目： 6个



PICT工具介绍

Better process ,Better products, Better services

□ Sub-Models

■ {<ParamName1>, <ParamName2>, <ParamName3>, ... } @ <Order>

■ 例子2

```
#
# This is a sample model for testing volume create/delete functions
#

3 PLATFORM: x86, ia64, amd64
3 CPUS      : Single, Dual, Quad
4 RAM       : 128MB, 1GB, 4GB, 64GB
2 HDD       : SCSI, IDE
4 OS        : NT4, Win2K, WinXP, Win2K3
4 IE        : 4.0, 5.0, 5.5, 6.0
3 APP       : SQLServer, Exchange, Office
```

◆ 组合遍历测试用例 = $3 \times 3 \times 4 \times 2 \times 4 \times 4 \times 3 = 3456$ 个

PICT工具介绍

Better process ,Better products, Better services

□ Sub-Models

■ {<ParamName1>, <ParamName2>, <ParamName3>, ... } @ <Order>

■ 例子2

```
#  
# This is a sample model for testing volume create/delete functions  
#  
  
PLATFORM: x86, ia64, amd64  
CPUS      : Single, Dual, Quad  
RAM       : 128MB, 1GB, 4GB, 64GB  
HDD       : SCSI, IDE  
OS        : NT4, Win2K, WinXP, Win2K3  
IE        : 4.0, 5.0, 5.5, 6.0  
APP       : SQLServer, Exchange, Office  
  
{ PLATFORM, CPUS, RAM, HDD } @ 2  
{ OS, IE } @ 2
```

PICT工具介绍

Better process ,Better products, Better services

□ Sub-Models

■ {<ParamName1>, <ParamName2>, <ParamName3>, ... } @ <Order>

■ 例子2

```
#
# This is a sample model for testing volume create/delete functions
#

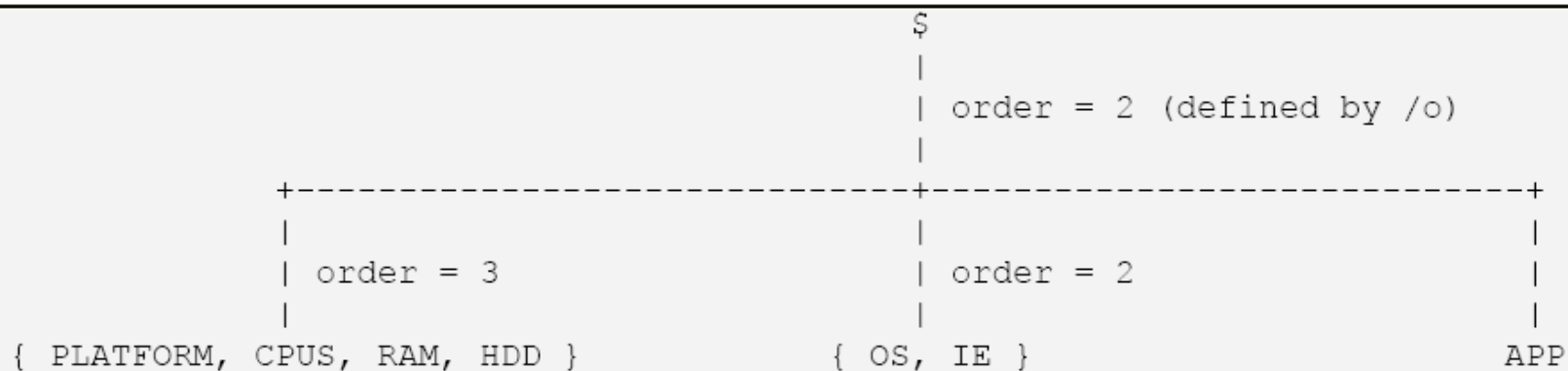
PLATFORM: x86, ia64, amd64
CPUS      : Single, Dual, Quad
RAM       : 128MB, 1GB, 4GB, 64GB
HDD       : SCSI, IDE
OS        : NT4, Win2K, WinXP, Win2K3
IE        : 4.0, 5.0, 5.5, 6.0
APP       : SQLServer, Exchange, Office

{ PLATFORM, CPUS, RAM, HDD } @ 3
{ OS, IE } @ 2
```

PICT工具介绍

Better process ,Better products, Better services

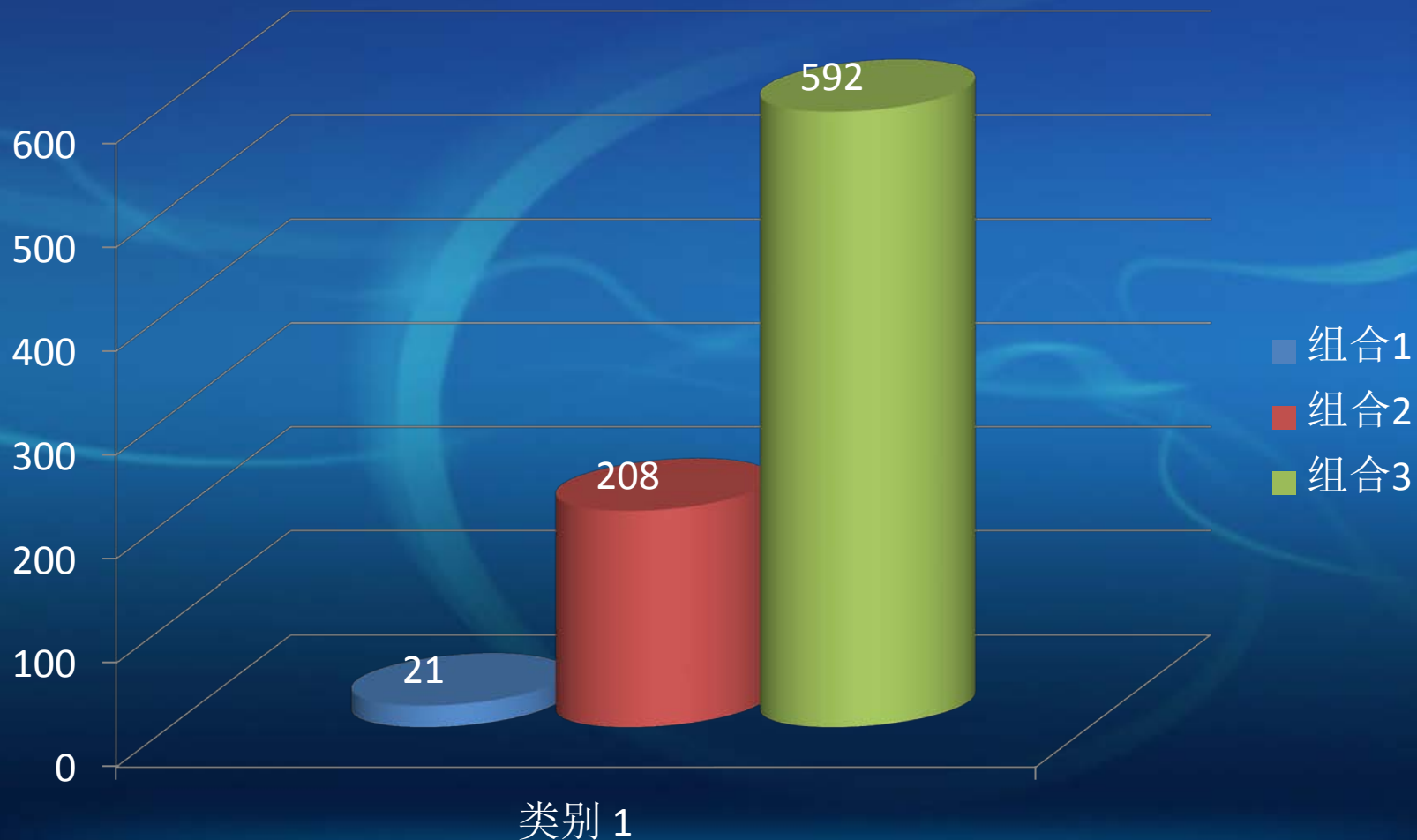
□ Sub-Models



PICT工具介绍

Better process ,Better products, Better services

□ Sub-Models



□ Sub-Models约束

- 子模型只能进行一级，不能深层嵌套
- 子模型分组可以为任意多个
- 参数可以存在于多个子模型中
- 每个子模型的组合数最多不超过其参数的最大数目
- 如果不明示子模型组合数，将使用默认的组合数

PICT工具介绍

Better process ,Better products, Better services

□约束条件表达

- =, <>, >, >=, <, <=,

- LIKE (通配符)

 - ◆ “*” (任意)

 - ◆ “?” (单个)

- IN

- IF, THEN, and ELSE

- NOT, AND, and OR

- ()

□约束表达

■例子3

```
#  
# Model 3  
# This is a sample model for specifying constraints  
#  
  
Type: Primary, Logical, Single, Span, Stripe, Mirror, RAID-5  
Size: 10, 100, 500, 1000, 5000, 10000, 40000  
Format method: quick, slow  
File system: FAT, FAT32, NTFS  
Cluster size: 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536  
Compression: on, off
```

PICT工具介绍

Better process ,Better products, Better services

□约束

■例子3

◆FAT16: 4G

◆FAT32: 32G

```
#  
# Model 3  
# This is a sample model for specifying constraints  
#  
  
Type: Primary, Logical, Single, Span, Stripe, Mirror, RAID-5  
Size: 10, 100, 500, 1000, 5000, 10000, 40000  
Format method: quick, slow  
File system: FAT, FAT32, NTFS  
Cluster size: 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536  
Compression: on, off  
  
IF [File system] = "FAT" THEN [Size] <= 4096;  
IF [File system] = "FAT32" THEN [Size] <= 32000;
```

PICT工具介绍

Better process ,Better products, Better services

□例子4

```
#  
# Example 4  
# This is a sample model for specifying constraints  
#  
  
Type: Primary, Logical, Single, Span, Stripe, Mirror, RAID-5  
Size: 10, 100, 500, 1000, 5000, 10000, 40000  
Format method: quick, slow  
File system: FAT, FAT32, NTFS  
Cluster size: 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536  
Compression: on, off  
  
IF [Cluster size] in {512, 1024, 2048} THEN [Compression] = "Off";  
IF [File system] in {"FAT", "FAT32"} THEN [Compression] = "Off";
```

PICT工具介绍

Better process ,Better products, Better services

□例子5

```
#  
# Example 5  
# This is a sample model for specifying constraints  
#
```

Type: Primary, Logical, Single, Span, Stripe, Mirror, RAID-5

Size: 10, 100, 500, 1000, 5000, 10000, 40000

Format method: quick, slow

File system: FAT, FAT32, NTFS

Cluster size: 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536

Compression: on, off

```
IF [File system] <> "NTFS" OR ( [File system] = "NTFS" AND [Cluster size] > 4096 )  
THEN [Compression] = "Off";
```

```
IF NOT ( [File system] = "NTFS" OR ( [File system] = "NTFS" AND NOT [Cluster size] <= 4096 ))  
THEN [Compression] = "Off";
```

PICT工具介绍

Better process ,Better products, Better services

□例子6

```
#  
# Example 6  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG_1: EN, DE
```

```
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server|AdvServer, Datacenter  
LANG_2: EN, DE
```

```
IF [LANG_1] = [LANG_2] THEN [OS_1] <> [OS_2] AND [SKU_1] <> [SKU_2]
```

PICT工具介绍

Better process ,Better products, Better services

□例子6

■无约束

	OS_1	SKU_1	LANG_1	OS_2	SKU_2	LANG_2+
1.	Win2000	Datacenter	EN	WinXP	Professional	EN+
2.	WinXP	Datacenter	DE	Win2000	Server	DE+
3.	WinXP	WinPowered	EN	Win2000	AdvServer	EN+
4.	Win2000	Professional	DE	WinXP	Server	DE+
5.	WinXP	Server	EN	WinXP	Datacenter	DE+
6.	Win2000	AdvServer	DE	Win2000	Professional	EN+
7.	Win2000	WinPowered	DE	Win2000	Datacenter	EN+
8.	WinXP	WinPowered	EN	WinXP	Professional	DE+
9.	Win2000	Server	DE	WinXP	AdvServer	EN+
10.	WinXP	Professional	EN	Win2000	Professional	EN+
11.	WinXP	Datacenter	EN	WinXP	Datacenter	EN+
12.	WinXP	Professional	EN	Win2000	Datacenter	DE+

PICT工具介绍

Better process ,Better products, Better services

□例子6

■有约束

	OS_1	SKU_1	LANG_1	OS_2	SKU_2	LANG_2↵
1.	Win2000	Datacenter	EN	WinXP	Professional	EN↵
2.	WinXP	Professional	EN	Win2000	Professional	DE↵
3.	Win2000	Server	DE	Win2000	Datacenter	EN↵
4.	Win2000	WinPowered	DE	WinXP	Professional	DE↵
5.	WinXP	WinPowered	EN	Win2000	Server	EN↵
6.	Win2000	Professional	DE	WinXP	AdvServer	DE↵
7.	WinXP	WinPowered	EN	WinXP	Datacenter	DE↵
8.	WinXP	Datacenter	DE	Win2000	Server	DE↵
9.	Win2000	Datacenter	EN	Win2000	Datacenter	DE↵
10.	WinXP	AdvServer	EN	WinXP	AdvServer	DE↵
11.	WinXP	Professional	EN	Win2000	Datacenter	EN↵
12.	WinXP	Server	DE	Win2000	Professional	DE↵

PICT工具介绍

Better process ,Better products, Better services

□例子7

```
#  
#Example 7  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG_1: EN, DE  
  
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server|AdvServer, Datacenter  
LANG_2: EN, DE  
  
# At least one parameter must be different to be a meaningful test case (use the OR operator)  
[OS_1] <> [OS_2] or [SKU_1] <> [SKU_2] or [LANG_1] <> [LANG_2];
```

PICT工具介绍

Better process ,Better products, Better services

□例子7

	OS_1	SKU_1	LANG_1	OS_2	SKU_2	LANG_2+
1.	Win2000	Datacenter	<u>EN</u>	WinXP	Professional	<u>EN</u> +
2.	WinXP	Datacenter	<u>DE</u>	Win2000	Server	<u>DE</u> +
3.	WinXP	WinPowered	<u>EN</u>	Win2000	AdvServer	<u>EN</u> +
4.	Win2000	Professional	<u>DE</u>	WinXP	Server	<u>DE</u> +
5.	<u>WinXP</u>	Server	EN	<u>WinXP</u>	Datacenter	DE+
6.	<u>Win2000</u>	AdvServer	DE	<u>Win2000</u>	Professional	EN+
7.	<u>Win2000</u>	WinPowered	DE	<u>Win2000</u>	Datacenter	EN+
8.	<u>WinXP</u>	WinPowered	EN	<u>WinXP</u>	Professional	DE+
9.	Win2000	Server	DE	WinXP	AdvServer	EN+
10.	WinXP	<u>Professional</u>	<u>EN</u>	Win2000	<u>Professional</u>	<u>EN</u> +
11.	<u>Win2000</u>	<u>Datacenter</u>	EN	<u>Win2000</u>	<u>Datacenter</u>	DE+
12.	Win2000	Professional	DE	WinXP	Datacenter	EN+

PICT工具介绍

Better process ,Better products, Better services

□例子7

```
#  
#Example 7  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG_1: EN, DE  
  
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server|AdvServer, Datacenter  
LANG_2: EN, DE  
  
# All parameters must different (use the AND operator)  
[OS_1] <> [OS_2] and [SKU_1] <> [SKU_2] and [LANG_1] <> [LANG_2];
```

PICT工具介绍

Better process ,Better products, Better services

□例子7

	OS_1	SKU_1	LANG_1	OS_2	SKU_2	LANG_2↵
1.	Win2000	WinPowered	DE	WinXP	Professional	EN↵
2.	WinXP	Professional	EN	Win2000	Datacenter	DE↵
3.	WinXP	Server	EN	Win2000	Professional	DE↵
4.	WinXP	Professional	DE	Win2000	Server	EN↵
5.	WinXP	Datacenter	EN	Win2000	AdvServer	DE↵
6.	Win2000	Professional	EN	WinXP	Datacenter	DE↵
7.	WinXP	WinPowered	EN	Win2000	Server	DE↵
8.	Win2000	Datacenter	DE	WinXP	Professional	EN↵
9.	Win2000	AdvServer	DE	WinXP	Datacenter	EN↵
10.	Win2000	Datacenter	EN	WinXP	AdvServer	DE↵
11.	Win2000	WinPowered	DE	WinXP	Datacenter	EN↵

□ 参数类型

■ 两种类型:

◆ 字符型

◆ 数字型

■ 仅当参数为数字型的才可以和数字进行比较

■ 仅当参数为字符型的才可以和字符进行比较

■ 如果取值有多个名字时，仅计算一次

■ 字符型的参数大小写敏感，可以使用“/C:”进行改变

□别名

- 使用别名可以为一个取值取多个名称
- “|”（管道符）,使用“/a:”可以更改
- 多个名称并不能改变模型的组合复杂度，一个值不论有多少个名字，都仅作为一个“实体”来看待
- 不同的只是在输出文件中显示的不同，名称将采用“轮流”的形式出现
- 当采用参数和取值“比较”时，只使用第一个值
- 当决定参数类型时，只使用第一个值

PICT工具介绍

Better process ,Better products, Better services

□ 别名

■ 例子8

```
#  
# Example 8: This is a sample model for Aliasing (No Aliasing)  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server, Datacenter, WinPowered  
LANG_1: EN, DE  
  
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server, Datecenter  
LANG_2: EN, DE  
  
# WinXP is always Professional in our case  
if [OS_1] = "WinXP" then [SKU_1] = "Professional";  
if [OS_2] = "WinXP" then [SKU_2] = "Professional";  
  
# No German WinPowered  
if [SKU_1] = "WinPowered" then [LANG_1] = "EN";  
  
# Let' s not test the same OS on both sides  
[OS_1] <> [OS_2];
```

PICT工具介绍

Better process ,Better products, Better services

□ 别名

■ 例子8

```
#  
# Example 8: This is a sample model for Aliasing (Aliasing)  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG_1: EN, DE  
  
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server|AdvServer, Datacenter  
LANG_2: EN, DE  
  
# WinXP is always Professional in our case  
if [OS_1] = "WinXP" then [SKU_1] = "Professional";  
if [OS_2] = "WinXP" then [SKU_2] = "Professional";  
  
# No German WinPowered  
if [SKU_1] = "WinPowered" then [LANG_1] = "EN";  
  
# Let' s not test the same OS on both sides  
[OS_1] <> [OS_2];
```

PICT工具介绍

Better process ,Better products, Better services

□别名

■例子8

- ◆不使用别名：13个“两两”组合
- ◆ 使用别名：13个“两两”组合

PICT工具介绍

Better process ,Better products, Better services

□ 别名

■ 例子8

```
#  
# Example 8: This is a sample model for Aliasing(Names rotated)  
#
```

Size: 1, 2, 3, 4, 5|6|7|8|9

Value: a, b, c, d

High: o, p, q, r, s, t

1.	4	d	p↵	16.	4	c	r↵	32.	1	d	r↵	46.	9	c	q↵	65.	1	a	s↵
2.	5	a	t↵	17.	2	a	o↵	33.	4	a	r↵	47.	5	a	o↵	66.	1	a	t↵
3.	4	d	t↵	18.	2	a	p↵	34.	3	d	r↵	48.	1	b	p↵	67.	2	b	s↵
4.	4	d	s↵	19.	2	b	q↵	35.	4	b	r↵	49.	4	c	p↵	68.	2	a	r↵
5.	1	a	q↵	20.	2	a	q↵	36.	4	a	t↵	50.	6	b	q↵	69.	4	d	r↵
6.	6	b	r↵	21.	2	c	s↵	37.	6	b	o↵	51.	7	d	t↵	70.	3	c	s↵
7.	1	b	q↵	22.	5	c	o↵	38.	3	a	o↵	52.	4	d	o↵	71.	6	c	t↵
8.	7	a	s↵	23.	2	d	t↵	39.	7	b	t↵	53.	2	b	r↵	72.	3	c	r↵
9.	8	c	p↵	24.	4	b	o↵	40.	1	b	s↵	54.	1	b	r↵	73.	2	d	q↵
10.	4	c	s↵	25.	1	c	p↵	41.	8	d	o↵	55.	1	a	r↵	74.	3	a	q↵
11.	4	a	p↵	26.	2	c	t↵	42.	3	b	o↵	56.	8	b	p↵	75.	7	c	s↵
12.	1	d	s↵	27.	4	b	p↵	43.	2	a	t↵	57.	9	d	r↵	76.	2	b	p↵
13.	3	b	r↵	28.	1	c	o↵	44.	3	d	p↵	58.	1	c	t↵	77.	4	b	q↵
14.	9	c	r↵	29.	4	a	s↵	45.	1	c	s↵	59.	2	c	r↵	78.	8	a	r↵
15.	4	b	s↵	30.	3	a	s↵					60.	5	d	s↵	79.	9	a	q↵
																80.	3	c	o↵

PICT工具介绍

Better process ,Better products, Better services

□ 别名

■ 例子8

```
#  
# Example 8: This is a sample model for Aliasing (Parameter Types-No Error)  
#
```

```
Size: 1, 2|8|x|y|z, 3, 4, 5  
Value: a, b, c, d
```

```
IF [Size] > 3 THEN [Value] > "b";
```

```
#  
# Example 8: This is a sample model for Aliasing (Parameter Types-No Error)  
#
```

```
Size: 1, x|y|z|2|8, 3, 4, 5  
Value: a, b, c, d
```

```
IF [Size] > "3" THEN [Value] > "b";
```

```
#  
# Example 8: This is a sample model for Aliasing (Parameter Types-Error)
```

```
# Input Error: Parameter/value type mismatch: IF [Size] > 3 THEN [Value] > "b";  
#
```

```
Size: 1, x|y|z|2|8, 3, 4, 5  
Value: a, b, c, d
```

```
IF [Size] > 3 THEN [Value] > "b";
```

PICT工具介绍

Better process ,Better products, Better services

□ 别名

■ 例子8

```
#  
# Example 8: This is a sample model for Aliasing(Evaluating constraints-No Error)  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG_1: EN, DE  
  
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server|AdvServer, Datecenter  
LANG_2: EN, DE  
  
# WinXP is always Professional in our case  
if [OS_1] = "WinXP" then [SKU_1] = "Professional";  
if [OS_2] = "Win2000" then [SKU_2] = "Server";
```

PICT工具介绍

Better process ,Better products, Better services

□ 别名

■ 例子8

```
#  
# Example 8: This is a sample model for Aliasing(Evaluating constraints-Error)  
# Constraints Warning: All or no values satisfy relation [SKU_2] = "AdvServer"  
# Constraints Warning: Restrictive constraints. Output will not contain following values: OS_2: Win2000  
#  
  
# Machine 1  
OS_1: Win2000, WinXP  
SKU_1: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG_1: EN, DE  
  
# Machine 2  
OS_2: Win2000, WinXP  
SKU_2: Professional, Server|AdvServer, Datecenter  
LANG_2: EN, DE  
  
# WinXP is always Professional in our case  
if [OS_1] = "WinXP" then [SKU_1] = "Professional";  
if [OS_2] = "Win2000" then [SKU_2] = "AdvServer";
```

□ 逆向测试

- 逆向测试是在构造仅有一个输入条件在取值“非法”时使系统产生缺陷的测试
- 测试中往往一个无效的输入会阻止另外的无效输入被测试
- 逆向测试要求一个测试用例中不要出现多个无效输入

□ 逆向测试

- PICT使用“~”来标记逆向测试中的无效输入，可以使用“/n:”来改变标识符
- PICT通过逆向测试标记可以做到
 - ◆ 所有的有效输入组合被覆盖
 - ◆ 所有的无效输入和有效输入至少组合一次
- 逆向测试标记仅仅是一个标记，并不是取值的一部分
- 在约束条件，直接使用无效输入，而不需要使用逆向测试标记，例如 `if [A] = -1 then [B] = 0`
- 如果取值存在别名，则只使用第一个值

PICT工具介绍

Better process ,Better products, Better services

□ 逆向测试

■ 例子9

```
float SumSquareRoots( float a, float b )  
{  
    if ( a < 0 ) throw error; // [1]  
    if ( b < 0 ) throw error; // [2]  
  
    return ( sqrt( a ) + sqrt( b ) );  
};
```

PICT工具介绍

Better process ,Better products, Better services

□ 逆向测试

■ 例子9

```
#  
# Example 9: This is a sample model for Negative Testing  
#  
  
A: -1, 0, 1, 2  
B: -1, 0, 1, 2
```

```
#  
# Example 9: This is a sample model for Negative Testing (Use Negative)  
#  
  
A: ~-1, 0, 1, 2  
B: ~-1, 0, 1, 2
```

PICT工具介绍

Better process ,Better products, Better services

□ 逆向测试

■ 例子9

16个“两两”
组合

15个“两两”
组合

A B
1 0
1 1
-1 -1
-1 1
2 2
2 0
0 1
2 -1
-1 0
1 -1
-1 2
0 -1
0 2
1 2
2 1
0 0

{-1, -1}	{0, -1}	{1, -1}	{2, -1}
{-1, 0}	{0, 0}	{1, 0}	{2, 0}
{-1, 1}	{0, 1}	{1, 1}	{2, 1}
{-1, 2}	{0, 2}	{1, 2}	{2, 2}

{-1, -1}	{0, -1}	{1, -1}	{2, -1}
{-1, 0}	{0, 0}	{1, 0}	{2, 0}
{-1, 1}	{0, 1}	{1, 1}	{2, 1}
{-1, 2}	{0, 2}	{1, 2}	{2, 2}

A B
2 0
2 2
0 0
0 1
1 0
2 1
0 2
1 2
1 1
~-1 2
1 ~-1
0 ~-1
~-1 1
2 ~-1
~-1 0

□权重

- 权重必须是正整数，默认的权重为1
- 权重可以使PICT更喜欢某些取值，但这个并不是绝对的，要由PICT依靠如下两个规则确定
 - ① 确保以最小的用例数覆盖所有的组合
 - ② 尽量满足权重要求
- ◆其中①的优先级高于②的
- 权重仅仅能够“尝试”让PICT尽量“偏向”于某些取值，决定权在于PICT
- 权重可以和其他因素共同使用

PICT工具介绍

Better process ,Better products, Better services

□权重

■例子10

```
#  
# Example 10: This is a sample model for Weighting(No Weighting)  
#  
  
# Let' s focus on primary partitions formatted with NTFS  
Type: Primary , Logical, Single, Span, Stripe, Mirror, RAID-5  
Format method: quick, slow  
File system: FAT, FAT32, NTFS
```

```
#  
# Example 10: This is a sample model for Weighting(Weighting)  
#  
  
# Let' s focus on primary partitions formatted with NTFS  
Type: Primary (10), Logical, Single, Span, Stripe, Mirror, RAID-5  
Format method: quick, slow  
File system: FAT, FAT32, NTFS (10)
```

PICT工具介绍

Better process ,Better products, Better services

□编排（“种子选手”）

- 有特别重要的组合需要在所有的测试生成中产生对应的用例。PICT会使用这些重要的组合，并按照要求的方式覆盖其余的组合。
- 你可以“编排”原有的测试用例集，将改动模型所带来的测试用例集改变最小化。
- “编排”必须定义在单独的文件中
- 命令参数
 - ◆pict model.txt /e:seedrows.txt

PICT工具介绍

Better process ,Better products, Better services

□编排（“种子选手”）

■“编排”文件格式

- ◆第一行为参数行，每个参数由TAB进行分隔
- ◆“编排”行为指定的组合形式，由TAB进行分隔
- ◆“编排”行的每个参数可以指定特定的数值，也可以不进行指定

OS	SKU	Lang+ ¹
Win2000	Professional	EN+ ¹
Win2000		DE+ ¹
WinXP	Professional	EN+ ¹

PICT工具介绍

Better process ,Better products, Better services

□编排（“种子选手”）

■“编排”文件约束

- ◆如果“编排”文件中存在对应的“模型”文件中所没有的参数，则此参数对应的取值都会被忽略
- ◆如果“编排”文件中存在对应的“模型”文件中所没有的取值，则此取值会被忽略。PICT仅使用取值范围内的值
- ◆如果“编排”文件中的“编排”行中取值违法现有的约定，则PICT会忽略此“编排”行
(以上的 1 和 2 PICT会产生警告信息)

PICT工具介绍

Better process ,Better products, Better services

□编排（“种子选手”）

■“编排”文件约束

◆以下情况下PICT会产生歧义

- 如果参数和取值为空
- 如果参数和取值中包含TAB键
- 如果取值和它的别名在参数中不唯一

PICT工具介绍

Better process ,Better products, Better services

□编排

■例子11

◆“模型”文件

```
#  
# Example 11: This is a sample model for Seeding  
#  
  
OS: Win2000, WinXP  
SKU: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG: EN, DE
```

◆“编排”文件

```
OS  SKU  Lang  
Win2000 Professional  EN  
Win2000    DE  
WinXP Professional  EN
```

PICT工具介绍

Better process ,Better products, Better services

□编排

■例子11

No Seeding[↵]

	OS	SKU	LANG [↵]
1.	WinXP	WinPowered	DE [↵]
2.	Win2000	Professional	EN [↵]
3.	Win2000	Datacenter	DE [↵]
4.	WinXP	Server	EN [↵]
5.	WinXP	Datacenter	EN [↵]
6.	Win2000	AdvServer	DE [↵]
7.	WinXP	Professional	DE [↵]
8.	Win2000	WinPowered	EN [↵]

[↵]

Seeding[↵]

	OS	SKU	LANG [↵]
1.	Win2000	Professional	EN [↵]
2.	Win2000	<u>WinPowered</u>	DE [↵]
3.	WinXP	Professional	EN [↵]
4.	WinXP	Server	DE [↵]
5.	WinXP	WinPowered	EN [↵]
6.	Win2000	Datacenter	EN [↵]
7.	WinXP	Datacenter	DE [↵]
8.	Win2000	AdvServer	EN [↵]
9.	WinXP	Professional	DE [↵]

[↵]

PICT工具介绍

Better process ,Better products, Better services

□ 随机输出

- 如果“模型”和相关参数不变， PICT总是产生同样的测试用例集
- 如果“模型”和相关参数不变，使用“/r”参数， PICT就会产生随机的测试用例集
- 如果“模型”和相关参数不变，使用“/r:seed”参数， PICT就会产生随机的测试用例集（注意： seed的取值范围是0~65535，超过65535后，会取其 与65536的MOD数）

□随机输出

- 测试用例集的生产取决于模型的初始条件
- 某些测试用例集的生成既要满足组合需求，又要受限于模型的初始条件，会使其生成的测试用例集效率低于其他的测试用例集
- 尝试最小化测试用例数使其更有效率时，可以尝试多次使用seed随机输出不同的结果来达到目的

PICT工具介绍

Better process ,Better products, Better services

□ 随机输出

■ 例子12

```
#  
# Example 12: This is a sample model for Output Randomization  
#  
  
OS: Win2000, WinXP  
SKU: Professional, Server|AdvServer, Datacenter, WinPowered  
LANG: EN, DE
```

```
C:\Example12> pict modelfile.txt > OutFile1.txt  
  
C:\Example12> pict modelfile.txt > OutFile2.txt /r  
Used seed: 50618  
  
C:\Example12> pict modelfile.txt > OutFile3.txt /r  
Used seed: 50630
```

PICT工具介绍

Better process ,Better products, Better services

□ 随机输出

■ 例子12

No Randomization↵				Randomization↵				Randomization↵			
OS	SKU	LANG↵		OS	SKU	LANG↵		OS	SKU	LANG↵	
1.	WinXP	WinPowered	DE↵	1.	Win2000	Professional	DE↵	1.	Win2000	Datacenter	DE↵
2.	Win2000	Professional	EN↵	2.	Win2000	WinPowered	EN↵	2.	Win2000	Professional	EN↵
3.	Win2000	Datacenter	DE↵	3.	WinXP	WinPowered	DE↵	3.	Win2000	Server	DE↵
4.	WinXP	Server	EN↵	4.	WinXP	Server	DE↵	4.	WinXP	Datacenter	EN↵
5.	WinXP	Datacenter	EN↵	5.	WinXP	Datacenter	EN↵	5.	Win2000	WinPowered	EN↵
6.	Win2000	AdvServer	DE↵	6.	Win2000	AdvServer	EN↵	6.	WinXP	Professional	DE↵
7.	WinXP	Professional	DE↵	7.	WinXP	Professional	EN↵	7.	WinXP	AdvServer	EN↵
8.	Win2000	WinPowered	EN↵	8.	Win2000	Datacenter	DE↵	8.	WinXP	WinPowered	DE↵

PICT工具介绍

Better process ,Better products, Better services

□ 随机输出

■ 例子12

```
C:\Example12>pict modelfile.txt>OutFile3.txt /r:2008
```

```
Used seed: 2008
```

```
C:\Example12>pict modelfile.txt>OutFile3.txt /r:20080808
```

```
Used seed: 26792
```

```
C:\Example12>pict modelfile.txt>OutFile3.txt /r:65535
```

```
Used seed: 65535
```

```
C:\Example12>pict modelfile.txt>OutFile3.txt /r:65536
```

```
Used seed: 0
```

```
C:\Example12>pict modelfile.txt>OutFile3.txt /r:65537
```

```
Used seed: 1
```

PICT工具介绍

Better process ,Better products, Better services

□大小写敏感

- PICT默认情况下是“大小写不敏感”的

 - ◆ 参数必须唯一

 - ◆ 取值不必唯一

- PICT使用“/c”来改变“大小写敏感”