2010-6-11

Security Level:

Effective Development and Practice of Automatic Testing based on TTCN3

Wang Shaofeng, Luo Fuliang,

Yan Longguo and Zhang Yanwei

www.huawei.com

HUAWEI

Content

- **Background**
- **Performance**
- **Test Framework**
- **Automation Resources**
- **Conclusion**

Difficulties and Challenges on Automatic Testing

Product requirement
Software becomes more complex

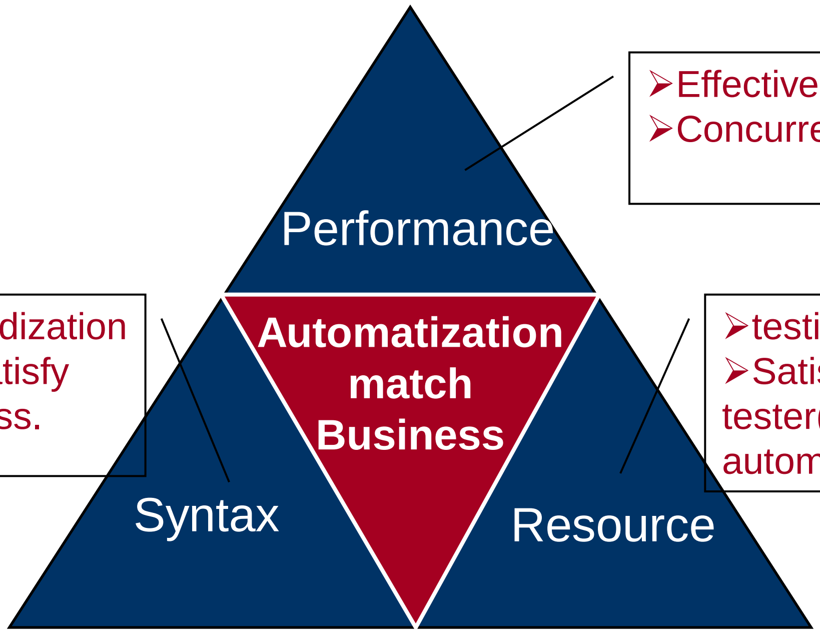
Tester requirement
Testing should be more simple

Agile requirement
Feedback should be quickly

- Hard for TSE&TEE to master Automation skills and join business testing.
- It takes so much time to test a version, but we need feedback quickly.
- TTCN3 can't satisfy the test case linearization.

How do Automation match Business ?

The Concept and Goal of TEP



➢Effective performance.
➢Concurrent performance.

➢Simplification & Standardization
➢Easy to learn & use，satisfy the requirement of business.

➢testing resource detachable
➢Satisfy the requirement for different tester(TE/AE/SE) to manage automation resource.

TEP (Test Execute Platform)
more than a TTCN3 Execution Platform, also offers a workbench to make an effective management for testing resource. Support CI testing、regression testing etc.
Core value of TEP:
Make automation match business easier.

Content

- Background
- **Performance**
- Test Framework
- Automation Resources
- Conclusion

Performance

- **Requirement**
  - **Business becomes more and more complex**
  - **Testsuits become more and more huge (thousands of function/testcase)**
  - **performance & stability always to be a top-level requirement**

How to make an effective management for testing resource?

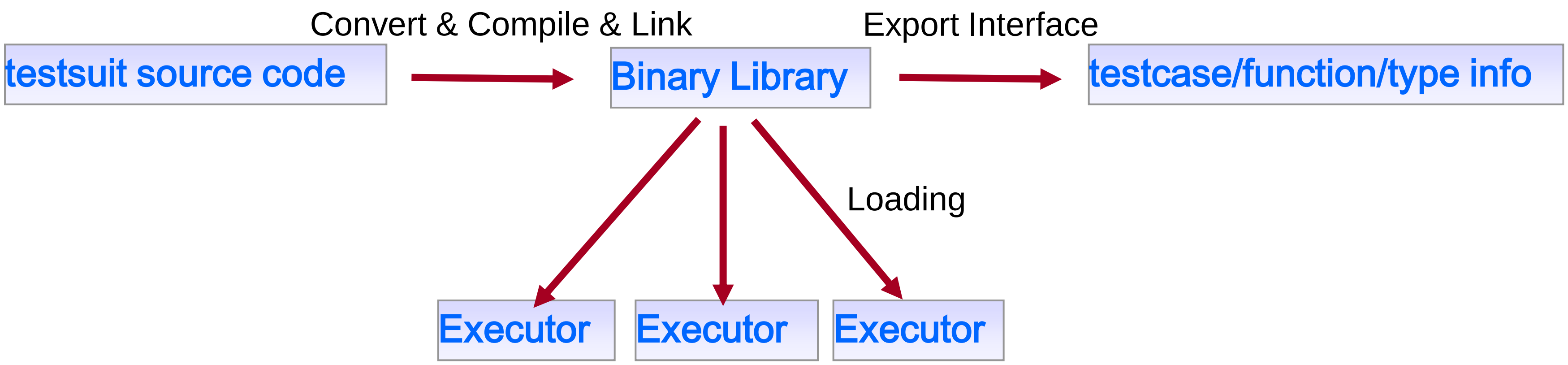
How to upgrade the performance?

A Lightweight Executor

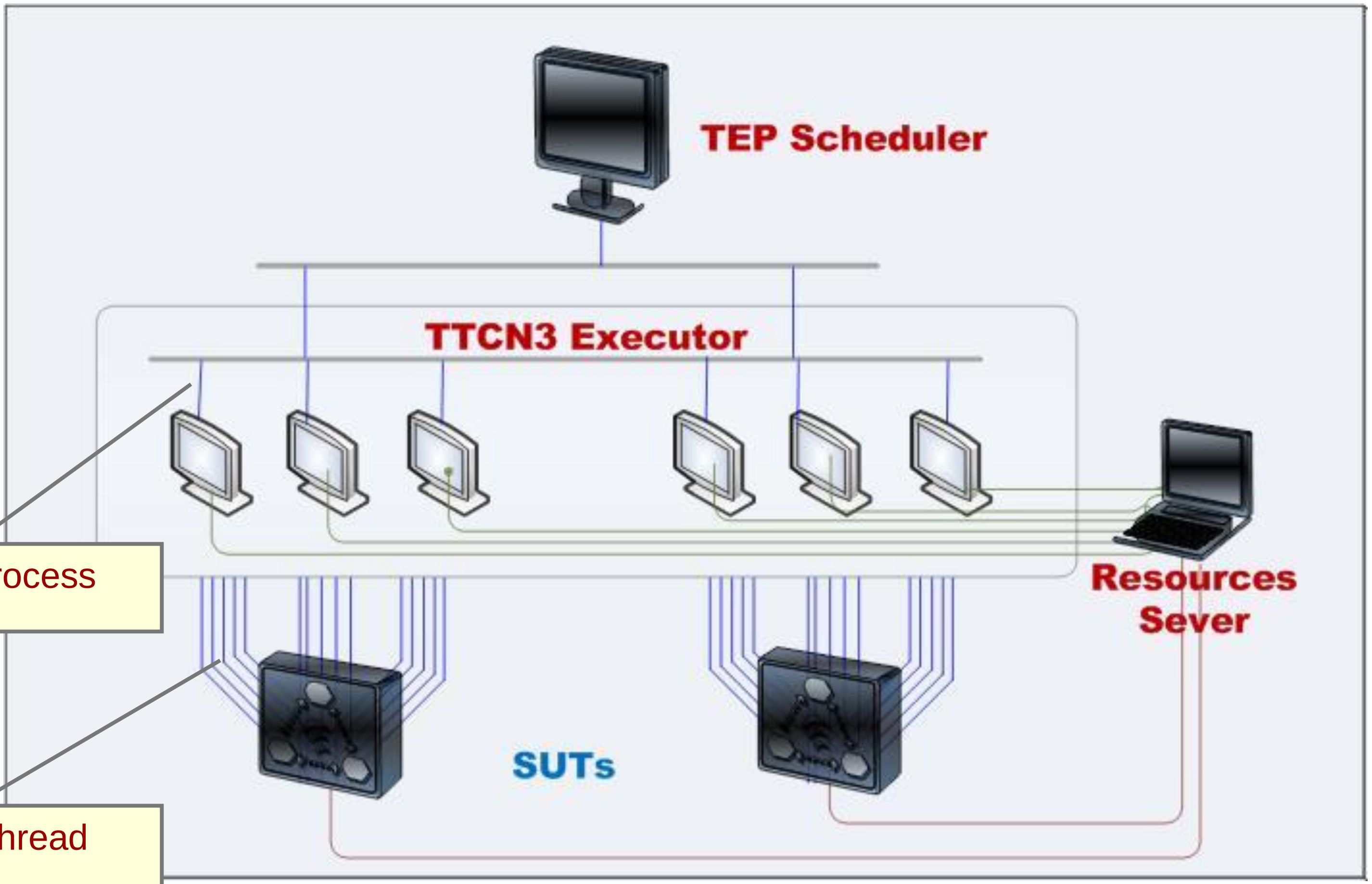
A lightweight executor based on language transform (TTCN3 to C++)

>> Easy to manage testing resource based on binary library

>> Easy to obtain concurrent performance



Concurrent Performance



Multi-Process

Multi-Thread

Content

- Background
- Performance
- **Test Framework**
- Automation Resources
- Conclusion

Test Framework

- **More demand for Automation Engineer**

  Be familiar with a language(TTCN3)？

  knowledge on Application & Business detail？

  Some special testing skills？

  。。。

  Actually，testing should be easy and simple.

  A standard way to generate effective test case?

Overview

Precondition & Postcondition

>most like other TFL

Teststep

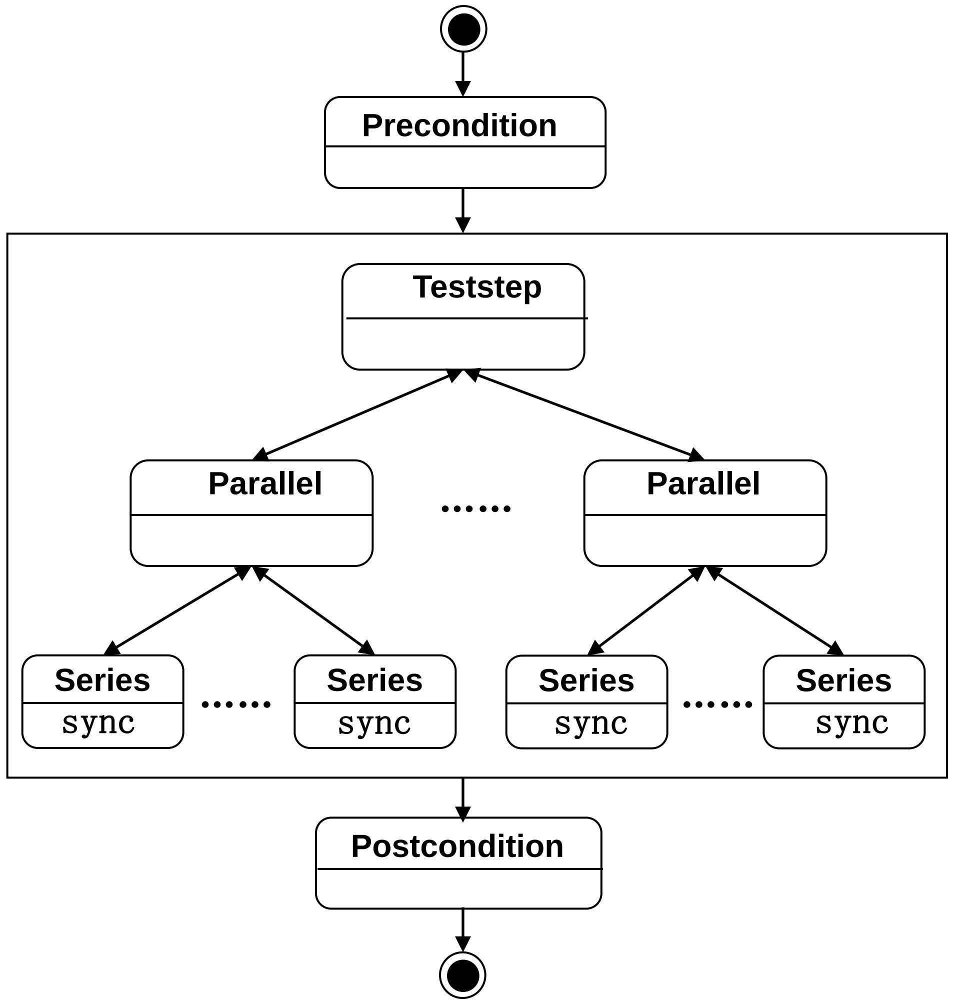
>test framework body

Series & Parallel

>like thread and thread pool

Sync

>block semantic, like barrier.



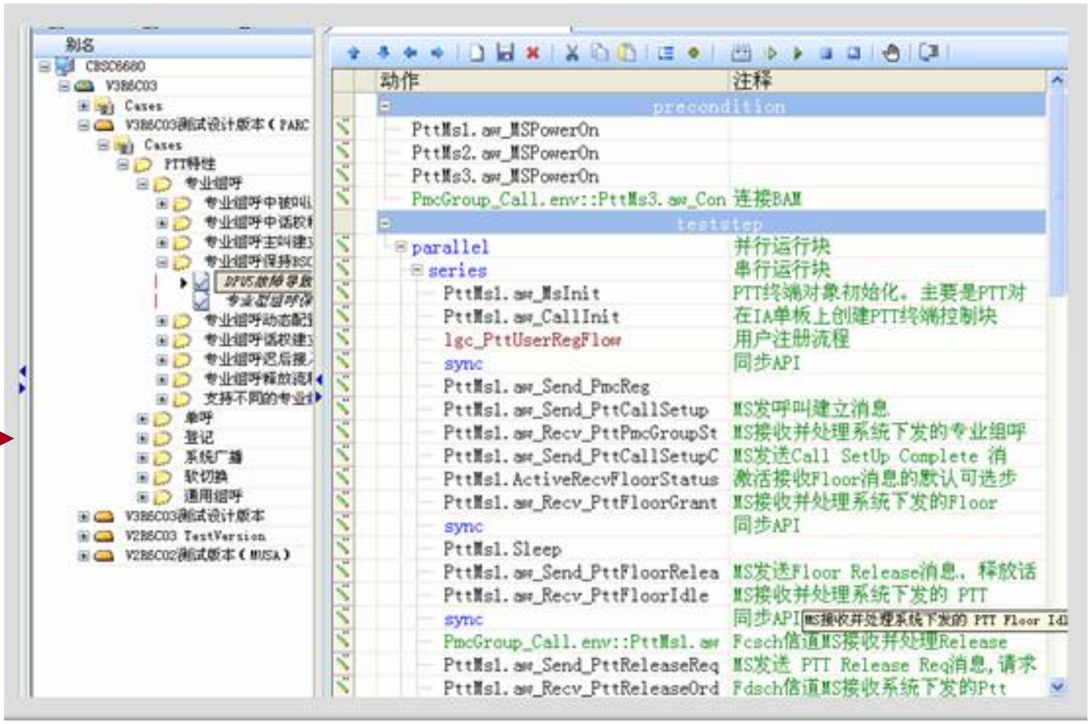
More than a TFL,

also offer a running model like concurrent action

Test Framework

```
testframe TF_001()
{
      var class_MS xPhone
      precondition
      {
      }
      teststep
      {
            parallel
            {
                  series
                  {
                        sync;
                  }
            }
      }
      postcondition
      {
      }
}
```

mapping



```
别名
⊟ CBSC6680
  ⊟ V3R6C03
    ⊞ Cases
    ⊟ V3R6C03测试设计版本（PARC
      ⊟ Cases
        ⊟ PTT特性
          ⊟ 专业组呼
            ⊞ 专业组呼中被叫
            ⊞ 专业组呼中话权
            ⊞ 专业组呼主叫建
            ⊟ 专业组呼保持BSC
                DPU5故障导致
                专业组组呼保
            ⊞ 专业组呼动态配
            ⊞ 专业组呼话权建
            ⊞ 专业组呼迟后接
            ⊞ 专业组呼释放流
            ⊞ 支持不同的专业
          ⊞ 单呼
          ⊞ 登记
          ⊞ 系统广播
          ⊞ 软切换
          ⊞ 通用组呼
    ⊞ V3R6C03测试设计版本
    ⊞ V2R6C03 TestVersion
    ⊞ V2R6C02测试版本（MUSA）
```

动作	注释
precondition	
PttMs1.aw_MSPowerOn	
PttMs2.aw_MSPowerOn	
PttMs3.aw_MSPowerOn	
PmcGroup_Call.env::PttMs3.aw_Con	连接BAM
teststep	
parallel	并行运行块
series	串行运行块
PttMs1.aw_MsInit	PTT终端对象初始化。主要是PTT对
PttMs1.aw_CallInit	在IA单板上创建PTT终端控制块
lgc_PttUserRegFlow	用户注册流程
sync	同步API
PttMs1.aw_Send_PmcReg	
PttMs1.aw_Send_PttCallSetup	MS发呼叫建立消息
PttMs1.aw_Recv_PttPmcGroupSt	MS接收并处理系统下发的专业组呼
PttMs1.aw_Send_PttCallSetupC	MS发送Call SetUp Complete 消
PttMs1.ActiveRecvFloorStatus	激活接收Floor消息的默认可选步
PttMs1.aw_Recv_PttFloorGrant	MS接收并处理系统下发的Floor
sync	同步API
PttMs1.Sleep	
PttMs1.aw_Send_PttFloorRelea	MS发送Floor Release消息，释放话
PttMs1.aw_Recv_PttFloorIdle	MS接收并处理系统下发的 PTT
sync	同步API MS接收并处理系统下发的 PTT Floor Idl
PmcGroup_Call.env::PttMs1.aw	Fcsch信道MS接收并处理Release
PttMs1.aw_Send_PttReleaseReq	MS发送 PTT Release Req消息，请求
PttMs1.aw_Recv_PttReleaseOrd	Fdsch信道MS接收系统下发的Ptt

Test framework base on class variable.

What’s class?

Why need a new concept : class

How to reuse a runs on function in other components?

Function func1() runs on Comp1 …

```
//Choice 1: using component compatibility
type component CompA
{       Now CompA is compatibility with Com1
    some Comp1's content;
}
```

```
//Choice 2: using component extends
type component CompB extends Comp1
{       Now CompB is extends from Comp1
    ……..
}
```

What an image When component relationship is very complex?

```
type component CompC extends Comp1,Comp2,Comp3,Comp4,Comp5 ……
{       Do you face any situation like CompC?
}
```

Implementation Detail

A new type more like OO: class (vs component)

Old style:

```
type compoennt CompC extends Comp1,Comp2,Comp3,Comp4,Comp5 …… {  … }
```

New style:

```
type class CompC
{
    Function Reference + Attribute;
}
```

```
//Function Reference different component
Function func1() runs on Comp1 return integer;
Function func5() runs on Comp5 return integer;
//Member Attribute
var charstring strVersion := "V1.0.0" ;
var charstring strIP := "10.78.75.69";
```

Usage: ***More like a component composite style***

```
var CompC myCompC;
myCompC.strIP := "10.78.75.100";
myCompC.func1();
```

Content

- Background
- Performance
- Test Framework
- **Automation Resources**
- Conclusion

Requirement of Automation Resource

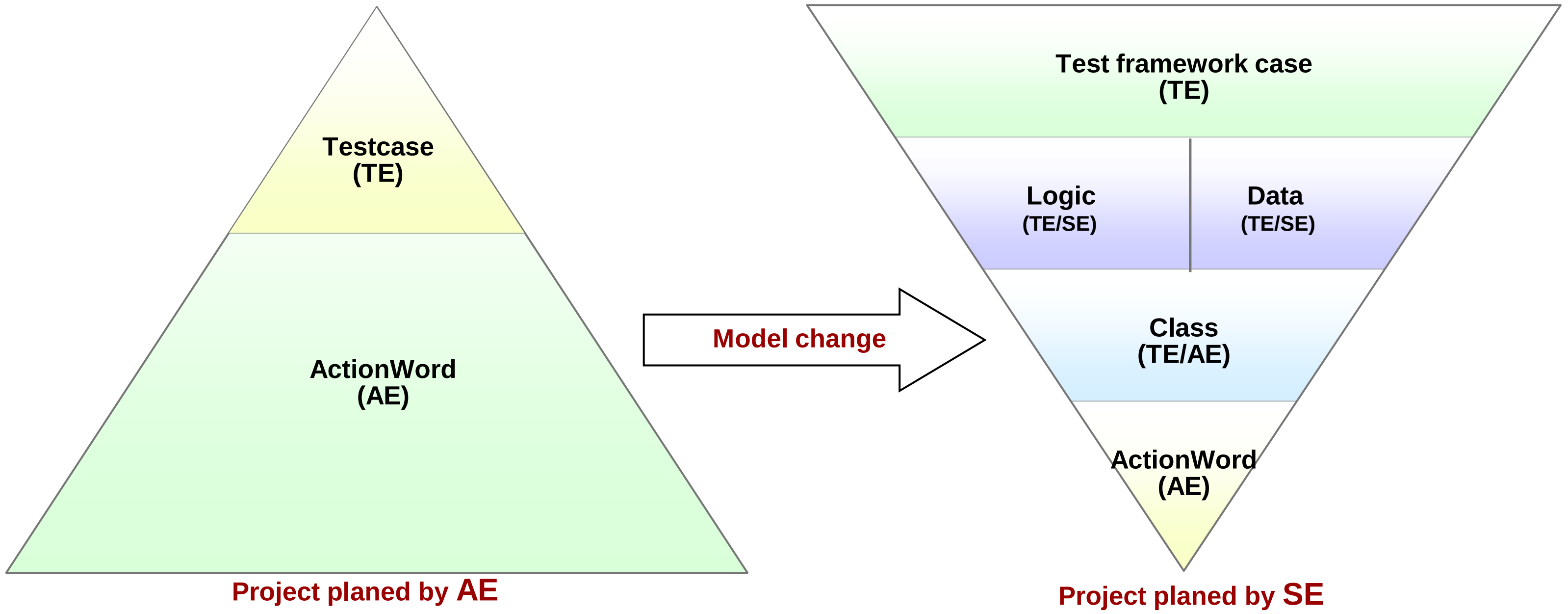
- **Concept**
  - **Automation Resource**

    Testcase、ActionWord、Data(environment data)、logic、class etc.

- **Defect**
  - **Automation Resources is too complex to design.**
  - **Not a easy work for TSE/TEE/TAE, testing and business not always to be the same.**

- **Requirement**
  - **Automation Project need to be divided into Automation Resource independently.**
    - **TSE take charge of planning the Automation Resources(Class).**
    - **TEE take charge of developing Logics & Testcases.**
    - **TAE take charge of developing ActionWords.**

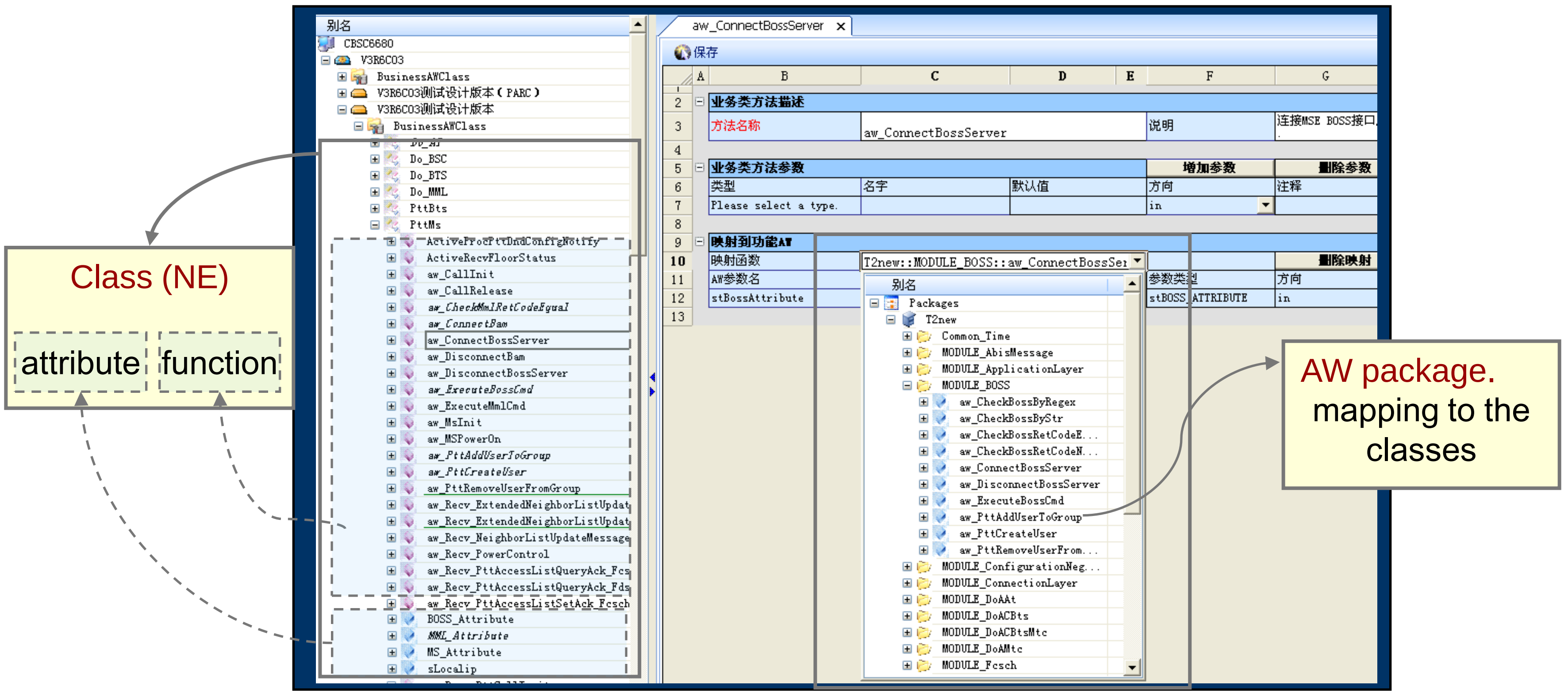
What Automation Resources to be



Automation Project to be divided into resources independently。
Less change when the version update.

SE&TE who understand business are suffice for Automation。

Testing Resource Management



```
别名
CBSC6680
⊟ V3R6C03
  ⊞ BusinessAWClass
  ⊞ V3R6C03测试设计版本（PARC）
  ⊟ V3R6C03测试设计版本
    ⊟ BusinessAWClass
      ⊞ Do_AT
      ⊞ Do_BSC
      ⊞ Do_BTS
      ⊞ Do_MML
      ⊞ PttBts
      ⊟ PttMs
        ⊞ ActiveProcPttDndConfigNotify
        ⊞ ActiveRecvFloorStatus
        ⊞ aw_CallInit
        ⊞ aw_CallRelease
        ⊞ aw_CheckMmlRetCodeEqual
        ⊞ aw_ConnectBam
        ⊞ aw_ConnectBossServer
        ⊞ aw_DisconnectBam
        ⊞ aw_DisconnectBossServer
        ⊞ aw_ExecuteBossCmd
        ⊞ aw_ExecuteMmlCmd
        ⊞ aw_MsInit
        ⊞ aw_MSPowerOn
        ⊞ aw_PttAddUserToGroup
        ⊞ aw_PttCreateUser
        ⊞ aw_PttRemoveUserFromGroup
        ⊞ aw_Recv_ExtendedNeighborListUpdat
        ⊞ aw_Recv_ExtendedNeighborListUpdat
        ⊞ aw_Recv_NeighborListUpdateMessage
        ⊞ aw_Recv_PowerControl
        ⊞ aw_Recv_PttAccessListQueryAck_Fcs
        ⊞ aw_Recv_PttAccessListQueryAck_Fds
        ⊞ aw_Recv_PttAccessListSetAck_Fcsch
        ⊞ BOSS_Attribute
        ⊞ MML_Attribute
        ⊞ MS_Attribute
        ⊞ sLocalip
```

aw_ConnectBossServer ✕

保存

	A	B	C	D	E	F	G
2	⊟	业务类方法描述					
3		方法名称	aw_ConnectBossServer			说明	连接MSE BOSS接口.
4							
5	⊟	业务类方法参数				增加参数	删除参数
6		类型	名字	默认值		方向	注释
7		Please select a type.				in	
8							
9	⊟	映射到功能AW					
10		映射函数	T2new::MODULE_BOSS::aw_ConnectBossSer				删除映射
11		AW参数名				参数类型	方向
12		stBossAttribute				stBOSS_ATTRIBUTE	in
13							

```
别名
⊟ Packages
  ⊟ T2new
    ⊞ Common_Time
    ⊞ MODULE_AbisMessage
    ⊞ MODULE_ApplicationLayer
    ⊟ MODULE_BOSS
      ⊞ aw_CheckBossByRegex
      ⊞ aw_CheckBossByStr
      ⊞ aw_CheckBossRetCodeE...
      ⊞ aw_CheckBossRetCodeN...
      ⊞ aw_ConnectBossServer
      ⊞ aw_DisconnectBossServer
      ⊞ aw_ExecuteBossCmd
      ⊞ aw_PttAddUserToGroup
      ⊞ aw_PttCreateUser
      ⊞ aw_PttRemoveUserFrom...
    ⊞ MODULE_ConfigurationNeg...
    ⊞ MODULE_ConnectionLayer
    ⊞ MODULE_DoAAt
    ⊞ MODULE_DoACBts
    ⊞ MODULE_DoACBtsMtc
    ⊞ MODULE_DoAMtc
    ⊞ MODULE_Fcsch
```

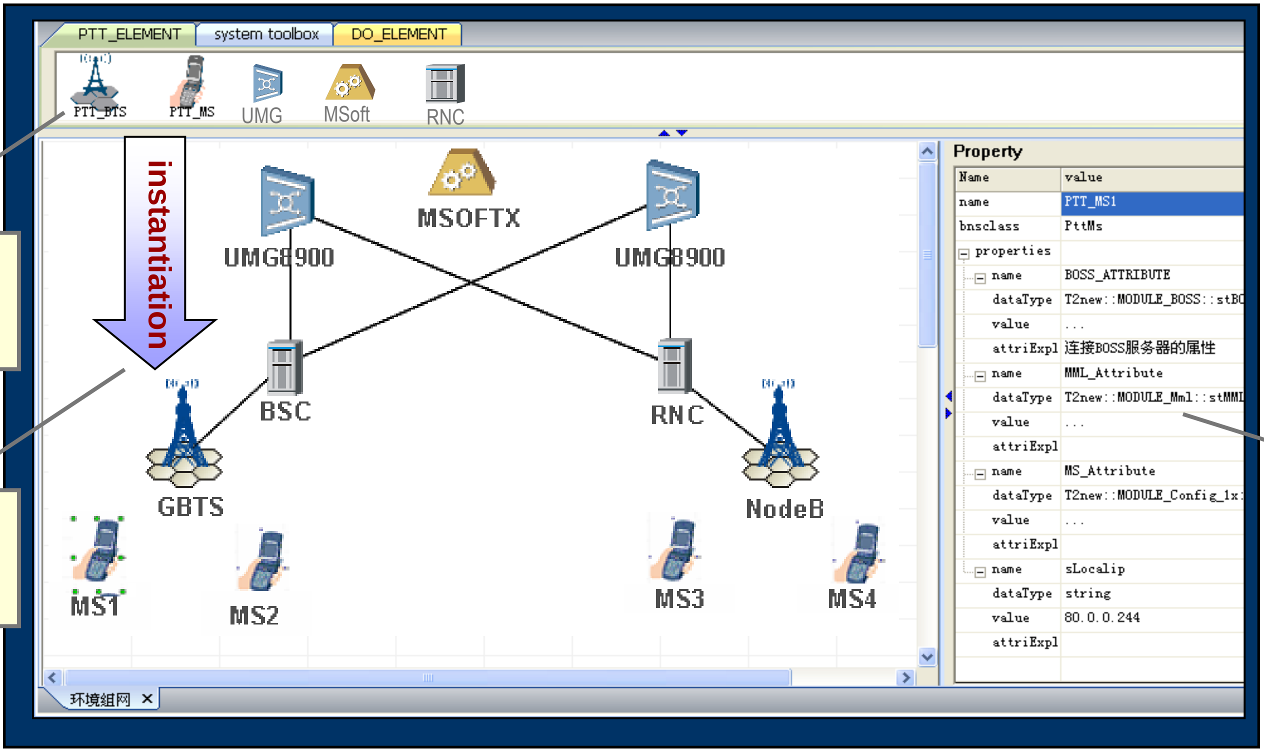
Class mapping NE model。The functions & attributes mapping NE’s AWs & parameters. New Class can be Parsed from AW-package automatically.

Easy Configuration。The TTCN3 templates instead of NE’s para templates which created by parsing asn.1 syntax

Graphics Configuration of Class

NE Classes
Saved at server

NE Objects
Saved at local.



Property

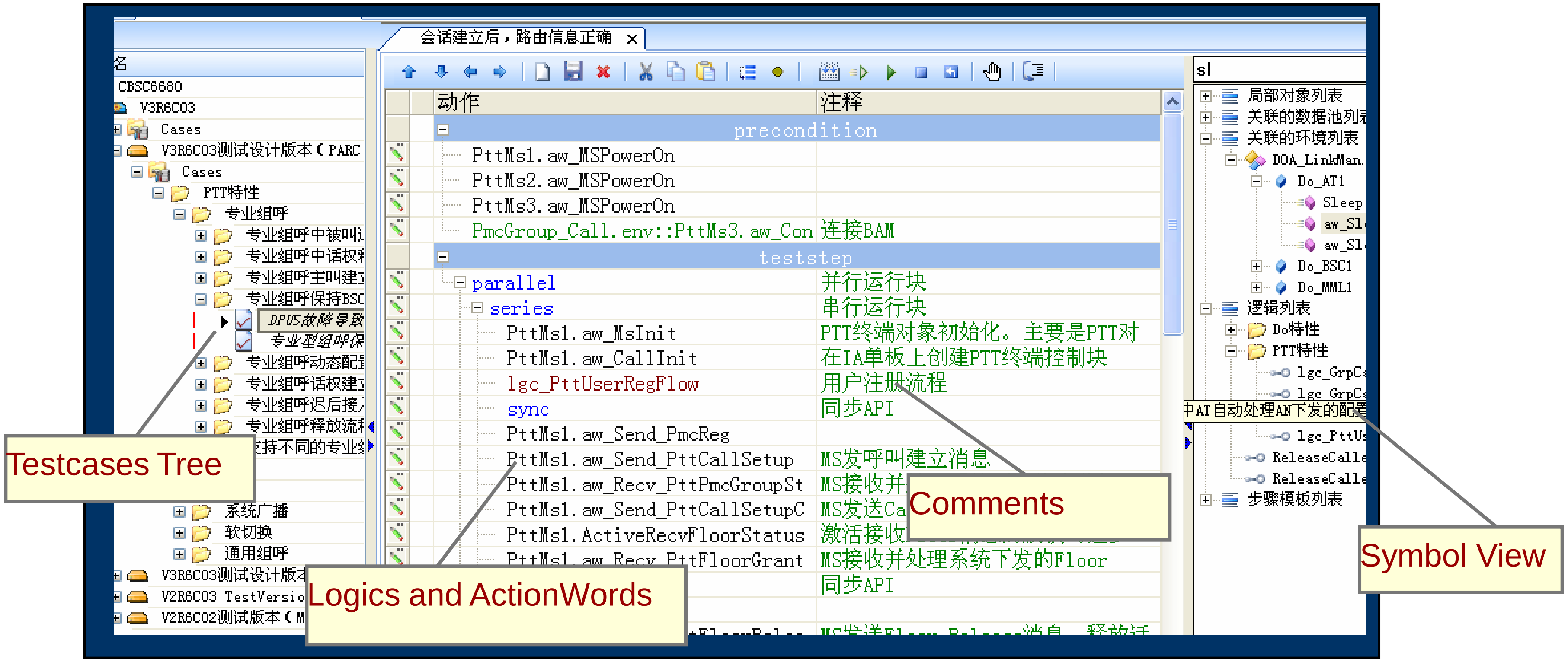
Name	value
name	PTT_MS1
bnsclass	PttMs
properties	
name	BOSS_ATTRIBUTE
dataType	T2new::MODULE_BOSS::stBO
value	...
attriExpl	连接BOSS服务器的属性
name	MML_Attribute
dataType	T2new::MODULE_Mml::stMML
value	...
attriExpl	
name	MS_Attribute
dataType	T2new::MODULE_Config_1x:
value	...
attriExpl	
name	sLocalip
dataType	string
value	80.0.0.244
attriExpl	

NE para templates
Saved at local.

The topological graphics mapping the hardware network。

Better reusable and shareable template。

The Main Workbench



```
名
CBSC6680
V3R6C03
 + Cases
 - V3R6C03测试设计版本（PARC
   - Cases
     - PTT特性
       - 专业组呼
         + 专业组呼中被叫i
         + 专业组呼中话权
         + 专业组呼主叫建
         - 专业组呼保持BSC
             DPU5故障导致
             专业型组呼保
         + 专业组呼动态配
         + 专业组呼话权建
         + 专业组呼迟后接
         + 专业组呼释放流
           支持不同的专业
       + 系统广播
       + 软切换
       + 通用组呼
 + V3R6C03测试设计版本
 + V2R6C03 TestVersio
 + V2R6C02测试版本（M
```

动作	注释
precondition	
PttMs1.aw_MSPowerOn	
PttMs2.aw_MSPowerOn	
PttMs3.aw_MSPowerOn	
PmcGroup_Call.env::PttMs3.aw_Con	连接BAM
teststep	
parallel	并行运行块
series	串行运行块
PttMs1.aw_MsInit	PTT终端对象初始化。主要是PTT对
PttMs1.aw_CallInit	在IA单板上创建PTT终端控制块
lgc_PttUserRegFlow	用户注册流程
sync	同步API
PttMs1.aw_Send_PmcReg	
PttMs1.aw_Send_PttCallSetup	MS发呼叫建立消息
PttMs1.aw_Recv_PttPmcGroupSt	MS接收并
PttMs1.aw_Send_PttCallSetupC	MS发送Ca
PttMs1.ActiveRecvFloorStatus	激活接收
PttMs1.aw_Recv_PttFloorGrant	MS接收并处理系统下发的Floor
	同步API

```
sl
+ 局部对象列表
+ 关联的数据池列表
- 关联的环境列表
  - DOA_LinkMan.
    - Do_AT1
        Sleep
        aw_Sl
        aw_Sl
    + Do_BSC1
    + Do_MML1
- 逻辑列表
  + Do特性
  - PTT特性
      lgc_GrpC
      lgc_GrpC
      lgc_PttU
    ReleaseCalle
    ReleaseCalle
+ 步骤模板列表
```

Easy to learn and use The Objects, which planed by SE at first, relative to the new Testcase will be created automatically.

Powerful running model
precondition / postcondition / parallel / series / sync etc.

Content

- Background
- Performance
- Test Framework
- Automation Resource
- **Conclusion**

The Benefits

Automation match Business

High quality
Actionword/Class based on binary library;
Testcase based on framework;
Less change while the version update.

Less demand for customer
Based on framework workbench
Effective & Concurrent performance

Effective management for Automation.
Resource detachable & shareable.
Based on binary library (not source code)

Thank You

www.huawei.com