

# A keyword-driven service testing framework based on TTCN-3

Wen Yongxin  
Huang Shifu

Testing Technology Research Dept, A&S

[wenyongxin@huawei.com](mailto:wenyongxin@huawei.com)

[www.huawei.com](http://www.huawei.com)

HUAWEI TECHNOLOGIES CO., LTD.

Huawei Confidential



## Contents

- ◆ **Service testing**
- ◆ **TTCN-3 & AW solution**
- ◆ **Typical application**
- ◆ **Benefits**
- ◆ **Future work**





# Service testing – Difficulties

- ◆ Require more programming skill
- ◆ High cost of script-based test case maintenances
- ◆ Without a test API specification, It is easy to bring too many APIs, which is difficult to use .

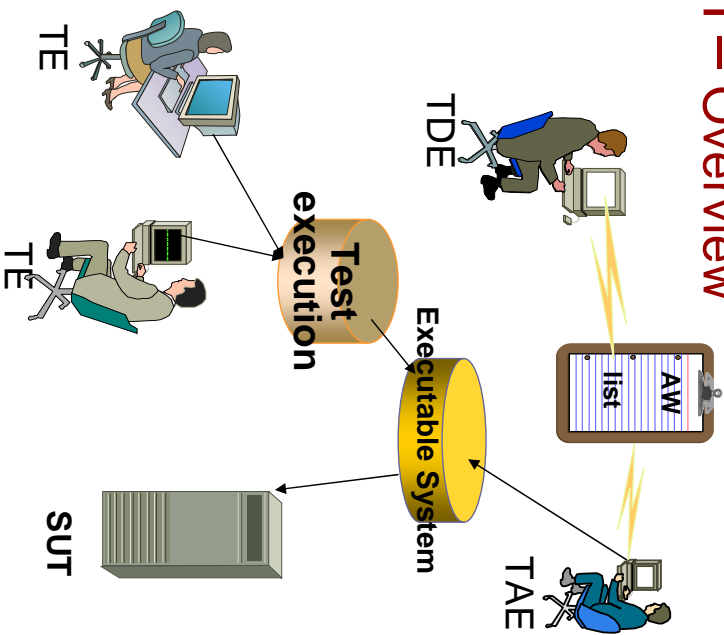
# Contents

- ◆ Service testing
- ◆ **TTCN-3 & AW solution**
  - Overview
  - Framework
  - Presentation layer
  - Script generator
  - Execution layer
  - Application area
- ◆ Typical application
- ◆ Benefits
- ◆ Future work

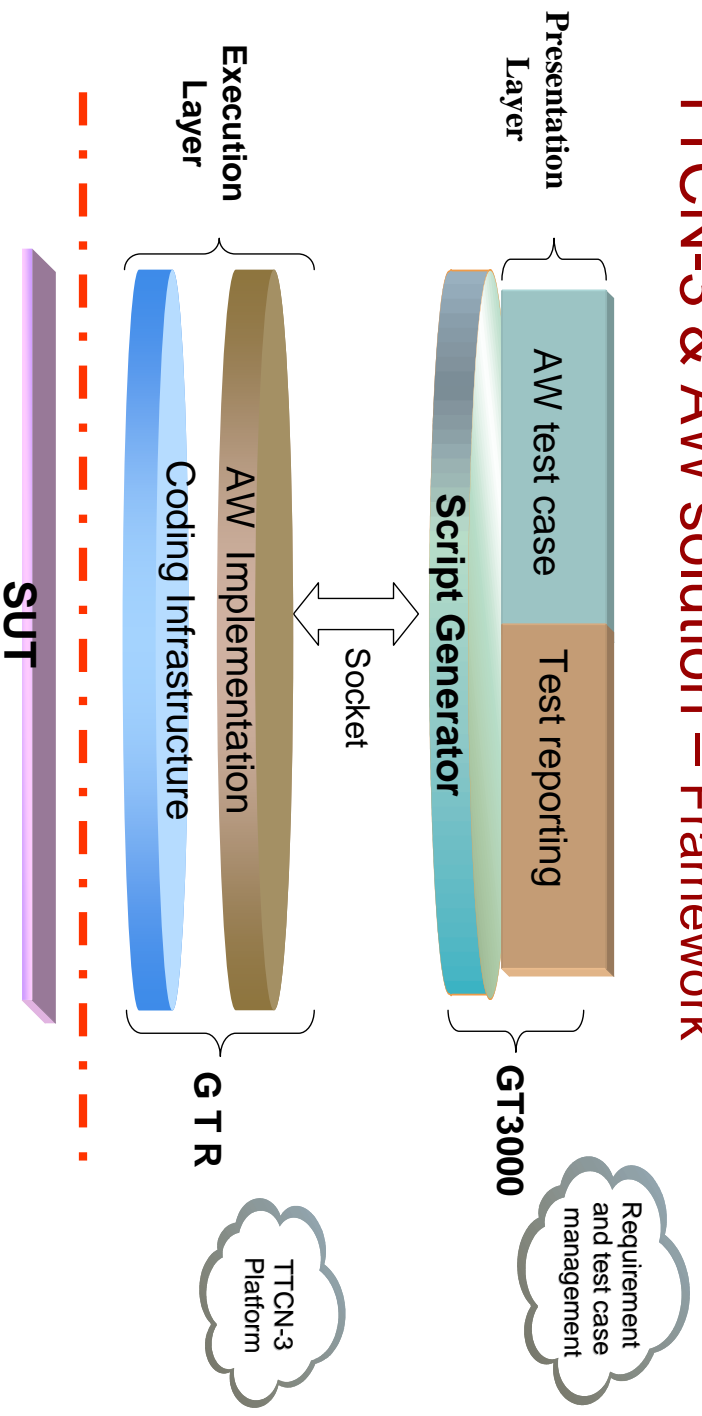


# TTCN-3 & AW solution – Overview

- ◆ Action word (AW ) is a methodology of keyword-driven testing from Huawei
- ◆ A 3rd generation of automatic testing
- ◆ Separates test design from test execution
- ◆ Graphical format of test case.
- ◆ Easy to create and maintain



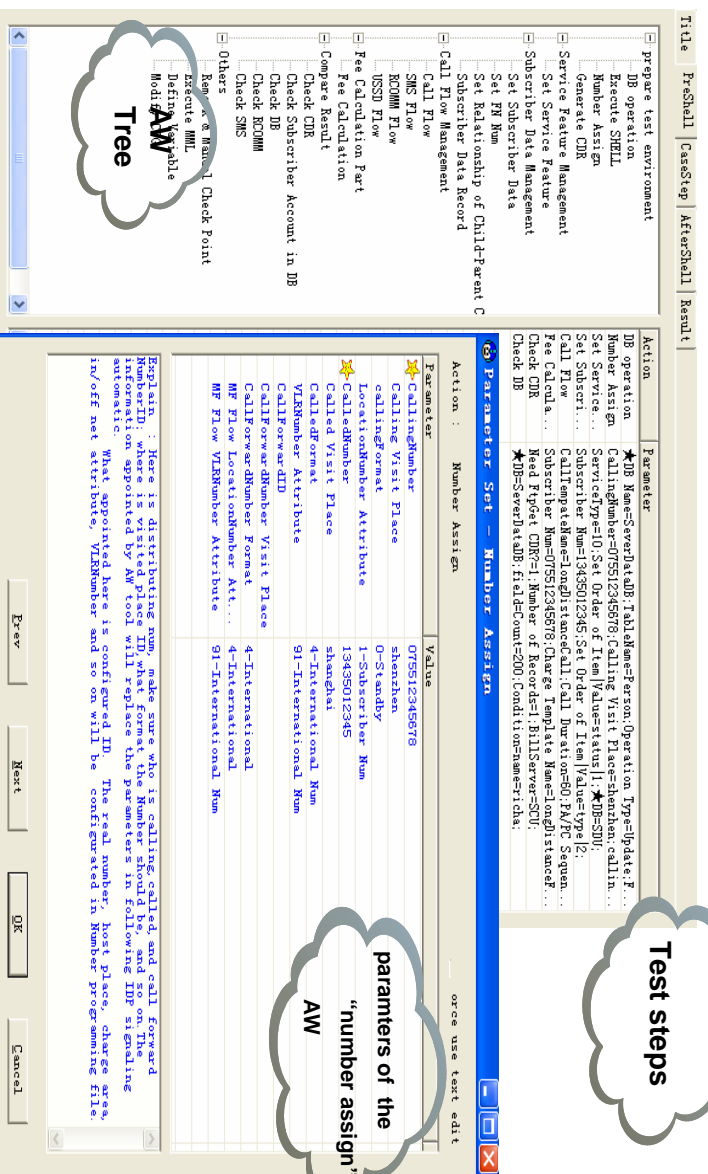
# TTCN-3 & AW solution – Framework



# TTCN-3 & AW solution – Presentation Layer

- ◆ Supports both tabular format and graphical format
- ◆ Different to ETSI standard
- ◆ Ease of complex value assignment
- ◆ Similar to real test environment
- ◆ User friendly
- ◆ Extensible
- ◆ Can be extended by C++, Delphi

## 1. Testcase & AW



The screenshot displays the software interface for test case and AW configuration. It is divided into several panes:

- Left Pane (Test Steps):** Lists various test actions such as "Prepare test environment", "DB operation", "Execute SQL", "Number Assign", "Generate CIR", "Service Feature Management", "Subscriber Data Management", "Set PK Num", "Subscriber Data Record", "Call Flow Management", "Call Flow", "USSD Flow", "Fee Calculation Part", "Compare Result", "Check CIR", "Check Subscriber Account in DB", "Check DB", "Check RCDM", "Check SMS", and "Others".
- Top Right Pane (Parameter Set - Number Assign):** A table defining parameters for the "Number Assign" action.
- Bottom Pane (Code Editor):** Contains C++ code for the "Number Assign" action, including comments and function calls.

Annotations in the image:

- A cloud-shaped callout labeled "Test steps" points to the list of actions in the left pane.
- Another cloud-shaped callout labeled "parameters of the 'number assign' AW" points to the "Parameter Set - Number Assign" table.

Action	Parameter	Value
Number Assign	DB Name=ServerIatADB;TaliName=Person;Operation Type=Update;F...	
Number Assign	CallInghNumber=075512345678;Calling Visit Place=Shenzhen;callIn...	075512345678
Set Service...	ServiceType=10;Set Order of Item Value=status 1;*DB=SDU;	Shenzhen
Set Subscri...	Subscriber Num=13495012345;Set Order of Item Value=type 2;	0-Subscriber
Call Flow...	CallTemplateName=longi;stance=call;Call Duration=60;P/M/PC Sequen...	1-Subscriber Num
Fee Calcula...	Subscriber Num=075512345678;Charge Template Name=length;stanceOf...	13495012345
Check CIR	Need PkgSet CIR?=1;Number of Record=1;BillServer=SDU;	longi
Check DB	*DB=ServerIatADB;FieldCount=200;ConditionName=pricea;	91-International Num

```
int main() {
    Here is a structure named num, make sure who is calling callid, and call forward
    Ring ID, check if the Ring ID is correct, if not, return false, if correct, return true.
    Information appointed by AW tool will replace the parameters in following IDP signaling
    automatic. What appointed here is configured ID. The real number here place charge area
    in/off net attribute, VLRNumber and so on will be configured in Number programming file.
}
```

## 2. User interface

Action : UC INFO Scene use text edit

Parameter	Value
Scene	Conference Setup
Role	client
Wait syn	
Set syn	
Local number	`\${_LocalFixPhone}
Access code	`\${_OutGoingAcCCode}
Save	Conference_ID=conferenceId
Request	Members=075512345678&account=\${_AfterAccl}&account ...
Response	Result=success

FIN AW Assistant V1.0 【 UC INFO Scene|Role 】

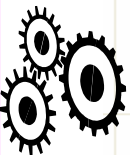
Name	Value
Scene	Conference Setup
Role	client
Wait syn	
Set syn	
Local number	`\${_LocalFixPhone}   075566661001   local telephone
Access code	`\${_OutGoingAcCCode}   900   Out going access code
Save conference ID	conferenceId
Request	Members 075512345678
*Request	Account \${_AfterAccl}   075566661001   PFS Account Number
*Request	Account password 12345678
Request	Subject Discussing work plan
Request	Background music listen.wav
Request	Record Yes
Request	Conference type Multimedia
Request	Conference size 10
Request	Conference password 123456
Response	Result success

## TTCN-3 & AW solution – Script Generator

- ◆ Convert tabular test case into TTCN-3 script
- ◆ Each TTCN-3 AW has only one parameter, containing all the AW parameters in a string, it also supports optional parameters: paramA{valueA} paramB{valueB} paramC{valueC} paramD{valueD}...
- ◆ Generate PTCs to simulate different network elements surrounds the SUT

# 1. Convert Tabular AW to TTCN-3 function

Parameter	Value
Comment	
★DB Name	ServerDB
Compound SQL	Person
TableName	Update
Operation Type	address age
Field Name	shenzhen in China 10
Value	name='richa'
Where Condition	
Input SQL	
SQL Statement	



One parameter contains all the tabular parameter value

`SQM_AW("DBName{ServerDB},TableName{Person},Operation{Update},Fields{address|age} Values{shenzhen in China|10},Condition{name='richa'}");`

# 2. PTCs simulate different network elements

**two PTCs**

- prepare test environment
- DB operation
- Execute SQL
- Number Assgn
- Generate CDR
- Service Feature Management
- Set Service Feature
- Subscriber Data Management
- Set Subscriber Data
- Set FN Num
- Subscriber Data Record
- Call Flow Management
- Call Flow
- SMS Flow
- RCDMM Flow
- USSD Flow
- Fee Calculation Part
- Fee Calculation

Title	PreShell	CaseStep	AfterShell	Result
No	Memo			
10	Caller	CALL Flow	CALLTempate{Name=1, onGd; stanceCall; Call ...	
20	Callee	RCDMM Flow	RCDMM Template Name=RecomMsg; Value Lis...	

```

//Caller
AWDemoComp awdemoComp1 := AWDemoComp.create;
//Callee
var AWDemoComp awdemoComp2 := AWDemoComp.create;
awdemoComp1.start(AWdemoComp1.Func());
awdemoComp2.start(AWdemoComp2.Func());
awdemoComp1.done;
awdemoComp2.done;

function AWdemoComp1_Func() runs on AWdemoComp
{
  MakeCallFlow("signalFlowName{longDistanceCall},callTime{60},releaseRole{Calling hang up}
  IDPNNumType{0-Full Number},CONNNECTDest{1-user}");
  SendRCDMM("tmpRCDMMName{RecomMsg},RCDMMValue{10,20}");
}

function AWdemoComp2_Func() runs on AWdemoComp
{
  MakeCallFlow("signalFlowName{longDistanceCall},callTime{60},releaseRole{Calling hang up}
  IDPNNumType{1-Calling Short Number},CONNNECTDest{1-user}");
}

```

### 3. Convert tabular test case to TTCN-3 test case

```

module Test_20070910154641_1 {
  import from ServerFunc all;
  testcase Test_20070910154641_1_TESTCASE() runs on AWDemoComp system AWDemoComp <
    //Preshell
    SM_AW("<DBName>ServerDB")tableName{Person}Operation{Update}Fields{address|age}
      values{shenzhen in china|10}condition{name="richa"};
    PrePhoneNum{callingId{075512345678}callingDis{shenzhen}callingId{13435012345}
      farwardDirFormat{91-International Num}};
    SetServFlag{"servType{1}&servFlag{args{status}1}&DBName{SDU}}";
    SetUserData{"phoneNum{13435012345}userData{args{type}2}}";
    //Caller
    var AWDemoComp awdemocomp1 := AWDemoComp.create;
    //callee
    var AWDemoComp awdemocomp2 := AWDemoComp.create;
    awdemocomp1.start(AWDemoComp1_Func());
    awdemocomp2.start(AWDemoComp2_Func());
    awdemocomp1.done;
    awdemocomp2.done;
  }
  //AfterShell
  callFee{"phoneNum{075512345678}chargeModel{1}omgDistance{chargeRate{calling}default{count}{normalCount}}";
  JudgePShell{"billTemplate{billCallTemplate}billType{calling_CDR}isGet{1}&billNum{1}&billServer{SDU}}";
  CompareDatabase{"DBName{ServerDataDB}table_Field_Bit{count}compareValue{count=200}condition{name="richa"}
}
}
function AWDemoComp1_Func() runs on AWDemoComp
  MakeCallFlow{"signalFlowName{1}omgDistance{call}callTime{60}releaseRole{calling hang up}
  IDPNumType{0-Full Number}CONNECTDest{1-user}}";
  sendRCOM{"tempRCOMName{RecomMsg}RCOMValue{10,20}}";
}
function AWDemoComp2_Func() runs on AWDemoComp
  MakeCallFlow{"signalFlowName{1}omgDistance{call}callTime{60}releaseRole{calling hang up}
  IDPNumType{1-Calling Short Number}CONNECTDest{1-user}}";
}
}

```

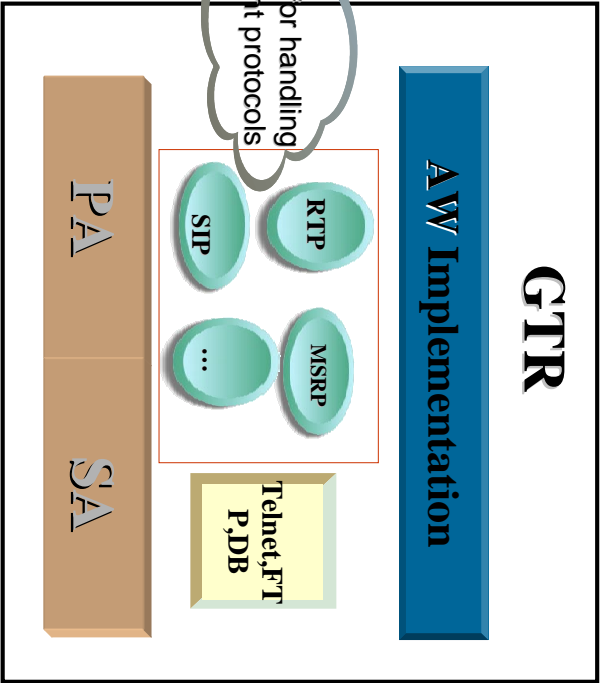
PostShell

CaseStep

Preshell

### TTCN-3 & AW solution – Execution Layer

- ◆ **Action Word Implementation** implements AW function with TTCN-3
- ◆ **Protocol library** implements protocol stacks like SIP stack, with TTCN-3
- ◆ **Common library** implements common operations on database, file, telnet, ftp, etc, with TTCN-3 / C++ (PA) / TCL





# TTCN-3 & AW solution – Application Area

- ◆ Adapt to service testing
- ◆ Not recommended for protocol testing/ API testing
- ◆ Good to test service that is :  
driven by many protocols  
stable, less than 10% changing would be perfect  
(so that the AW test cases can be inherited )

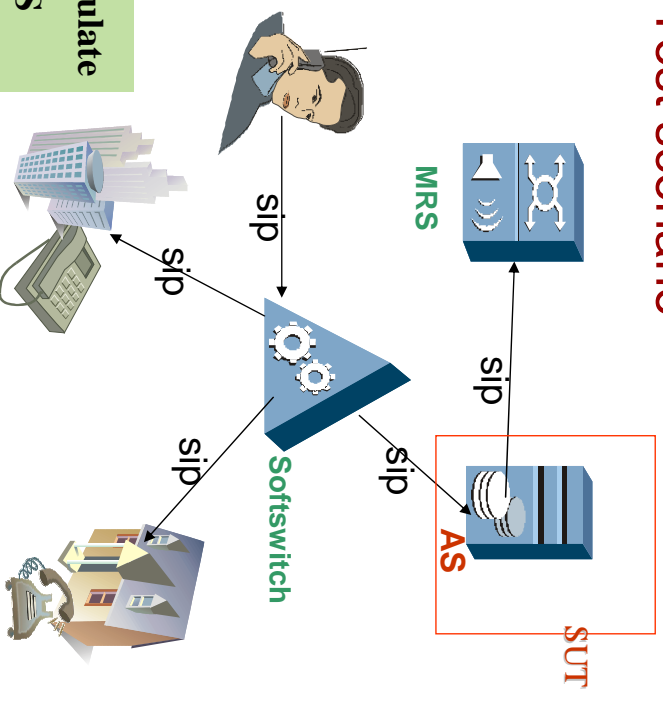
## Contents

- ◆ Service Testing
- ◆ TTCN-3 & AW solution
- ◆ **Typical application**
- ◆ Benefits
- ◆ Future work



# Typical application – Test scenario

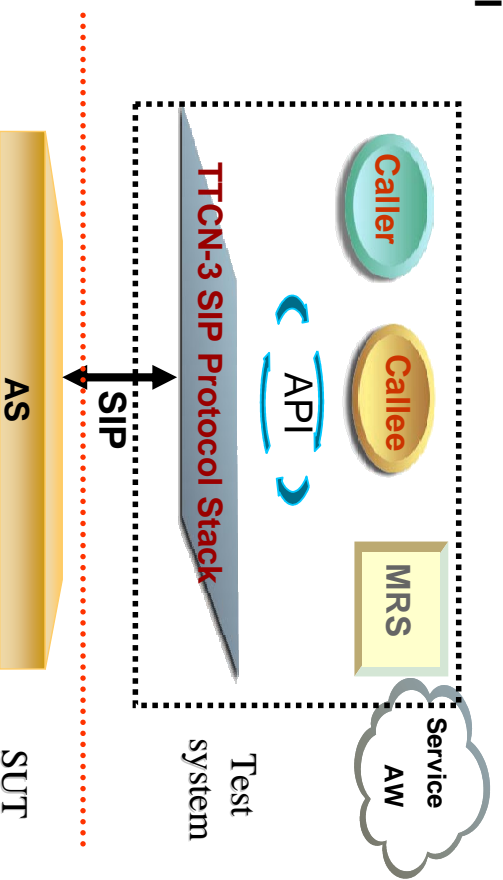
- ◆ A User dialed a virtual number
- ◆ The AS called the office phone and the home telephone at the same time
- ◆ The home telephone is picked up



◆ Here, we use TTCN-3 & AW to simulate Softswitch and MRS to test the AS

# Typical application – Test system details

- ◆ TTCN-3 SIP Protocol Stack handles sip message and SIP transaction, dialog, etc.
- ◆ Service AW are TTCN-3 functions that simulate the Softswitch (caller, callee) and MRS actions



# Typical application – Demo Testcase

The screenshot shows the 'AutoTestIM View - newTest' interface with two tabs: 'newTest' and 'newTest1'. The 'newTest' tab displays a table of test steps:

No	Memo	Action	Parameter
10	Caller	Wait SIP syn	When-before:SIMSType=7;response:SIPMsg...
20	Caller	Callse Flow	Element:identity=LocalCaller;Number=0...
30	Mrs		

The 'newTest1' tab shows a detailed configuration for the 'Callse' step:

Action	Parameter
Initial environment	Import DB type=SDU/SCDU;
DB operation	DB operation=Ipdate;DB type=SDU/SCDU; tab...
DB operation	operation=Ipdate;DB type=SDU/SCDU; tab...
Refresh memory data	Memory tables=ALL;

Below the screenshot is a 'Call flow' diagram showing the sequence of messages between the Caller, AS, MRS, and Callee:

```

sequenceDiagram
    participant Caller
    participant AS
    participant MRS
    participant Callee

    Caller->>AS: invite
    AS->>MRS: invite
    MRS->>Callee: invite
    Callee-->>MRS: 200
    MRS-->>AS: 200
    AS-->>Caller: 200
    Caller-->>AS: ack
    AS->>MRS: bye
    MRS->>Callee: 200
    Callee-->>MRS: ack
  
```

## Contents

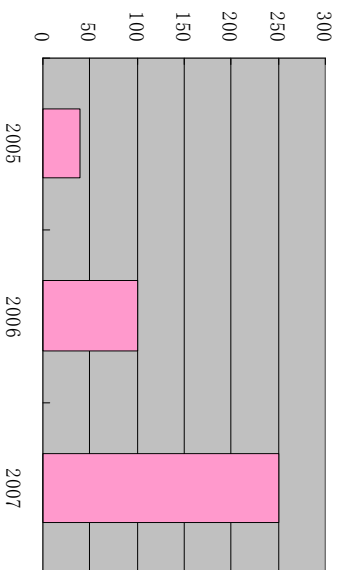
- ◆ Service Testing
- ◆ TTCN-3 & AW solution
- ◆ Typical application
- ◆ **Benefits**
- ◆ Future work



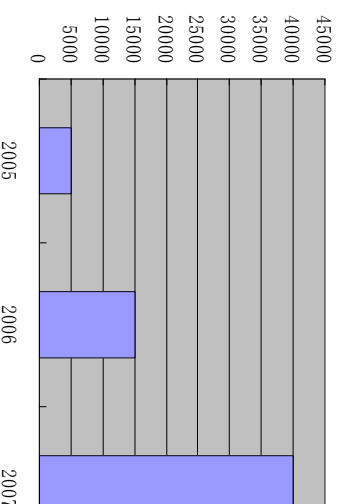
# Benefits



Total users



Total test cases



# Contents

- ◆ Service Testing
- ◆ TTCN-3 & AW solution
- ◆ Typical application
- ◆ Benefits
- ◆ **Future work**



# Future work

- ◆ **AW design with layers**  
High level AW can be implemented by lower level  
AWs
- ◆ **Object oriented AW design**  
With object oriented design, it is similar to real  
entity
- ◆ **Integrate AW implemented in C++/ TCL/ java**  
Share AW pool largely



# Thank you!