AVAYA

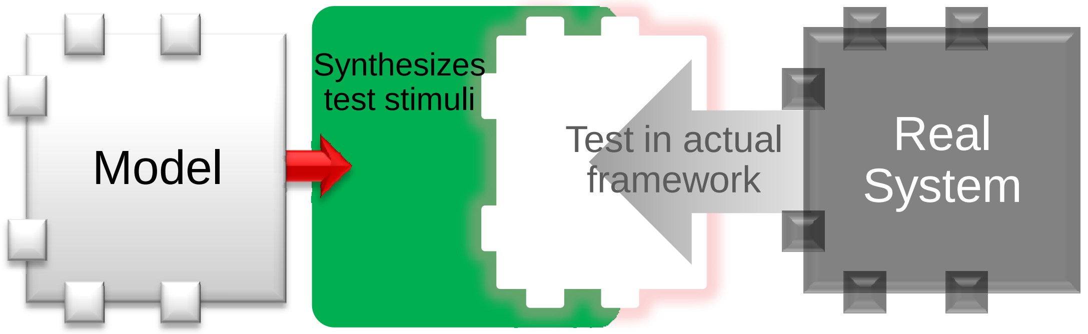
INTELLIGENT COMMUNICATIONS

# Model Based Testing for VoIP Phones

**Dinesh Patil, Avaya**
Jani Koivulainen, Conformiq
Jagadish Vellanki, Ideabytes

September 2010

## What is Model Based Testing?



- The basic idea in Model Based Testing (MBT) is to check the conformance of a system implementation (SUT) to the specification by modeling its expected behavior

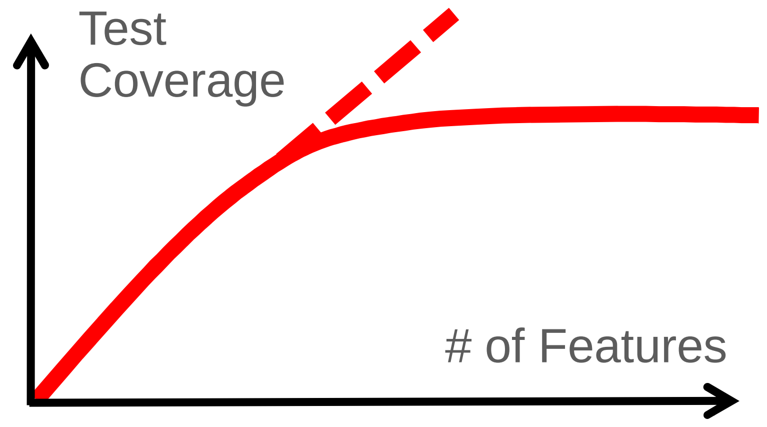
- Directly from this model, the user selects the algorithms to use in the Automated Test Design (ATD) tool

- ATD automatically designs and generates black box feature test scripts and proper test outcomes (test oracle)

## Why Avaya Decided to Use MBT

AVAYA
INTELLIGENT COMMUNICATIONS

- We realized that improving our testing by just bringing in new people was no longer an option, so we had to change our test design methodology



- MBT (Model Based Testing) and ATD (Automated Test Design) seemed like a good way to significantly boost our testing capability

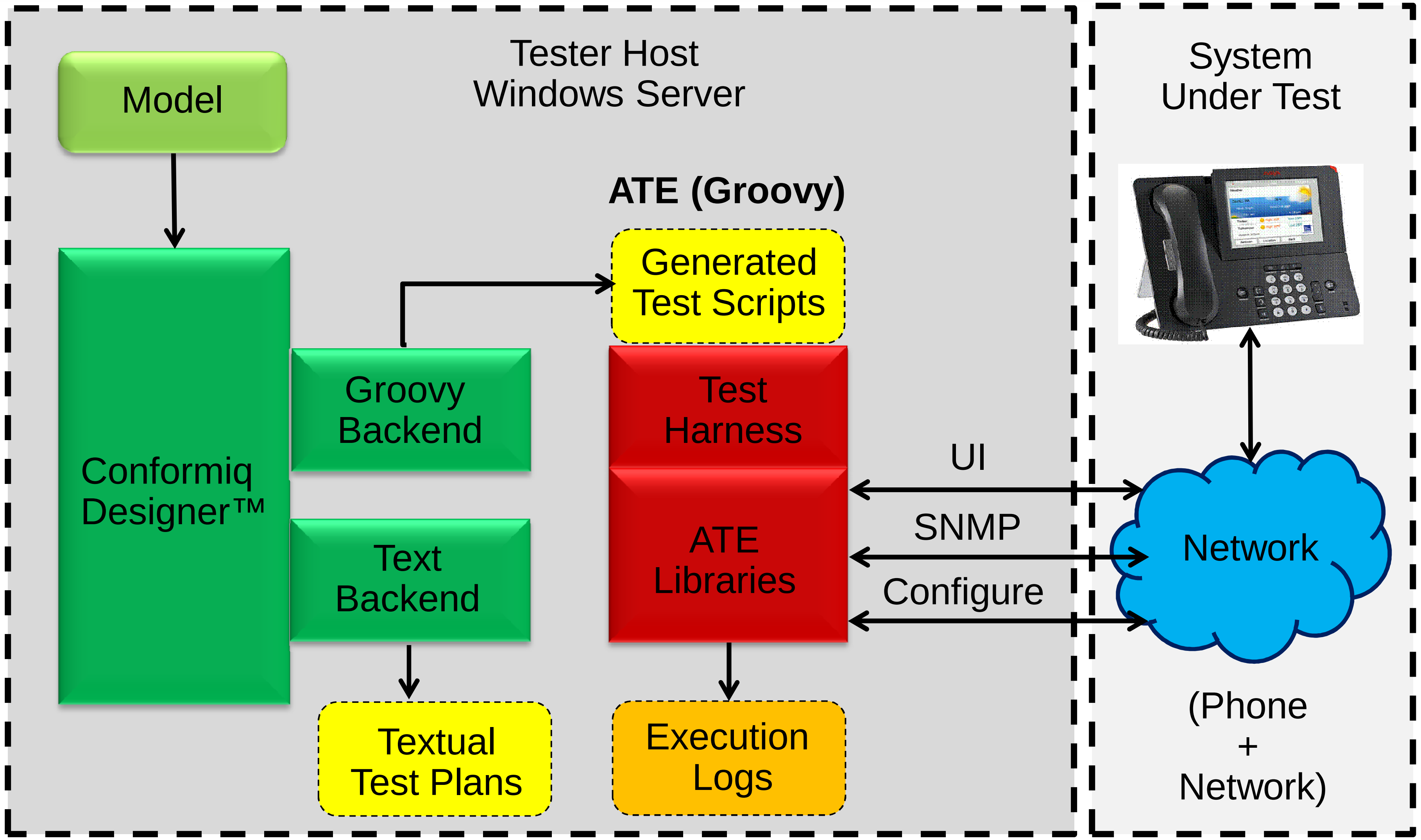
- Conformiq Designer™ software was selected

- Within six months of implementation we established MBT /ATD as an essential part of our testing process

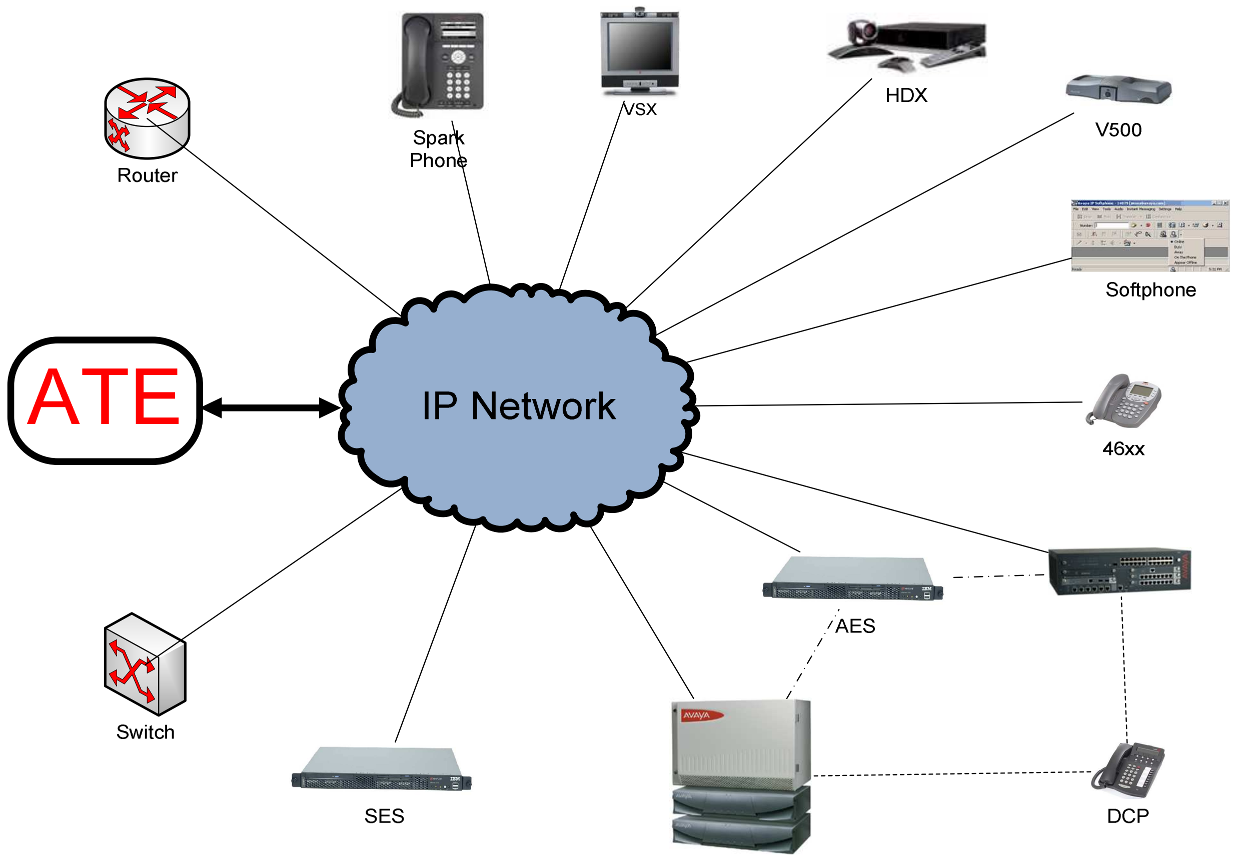
## Productivity Gains Using MBT

- No manual selection and enumeration of test cases
- No need to produce test data
- No debugging of incorrect test cases
- Improved test cases and coverage
- Reports missed requirement tests
- Faster to update model than test scripts
- Maps test cases to requirements
- Consistent test case design
- Automatically matches tests to requirements
- No test automation backlog

# Avaya Automated Testing Environment



# Avaya VoIP Test System Architecture

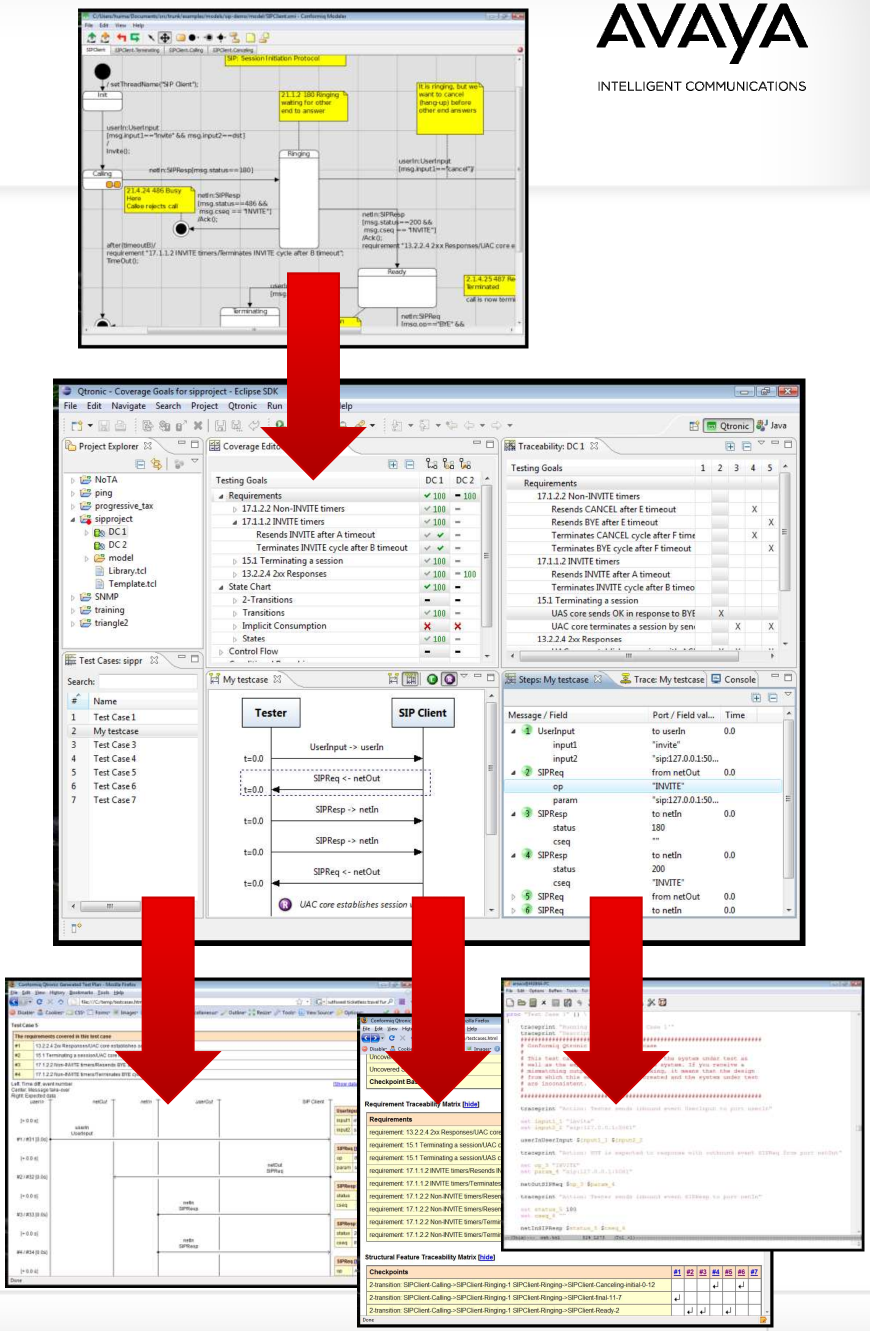


Our Groovy-based Automated Test Environment (ATE) is very complex and includes an extensive set of functionalities for interacting programmatically with our VoIP phones

# MBT Tool Operation

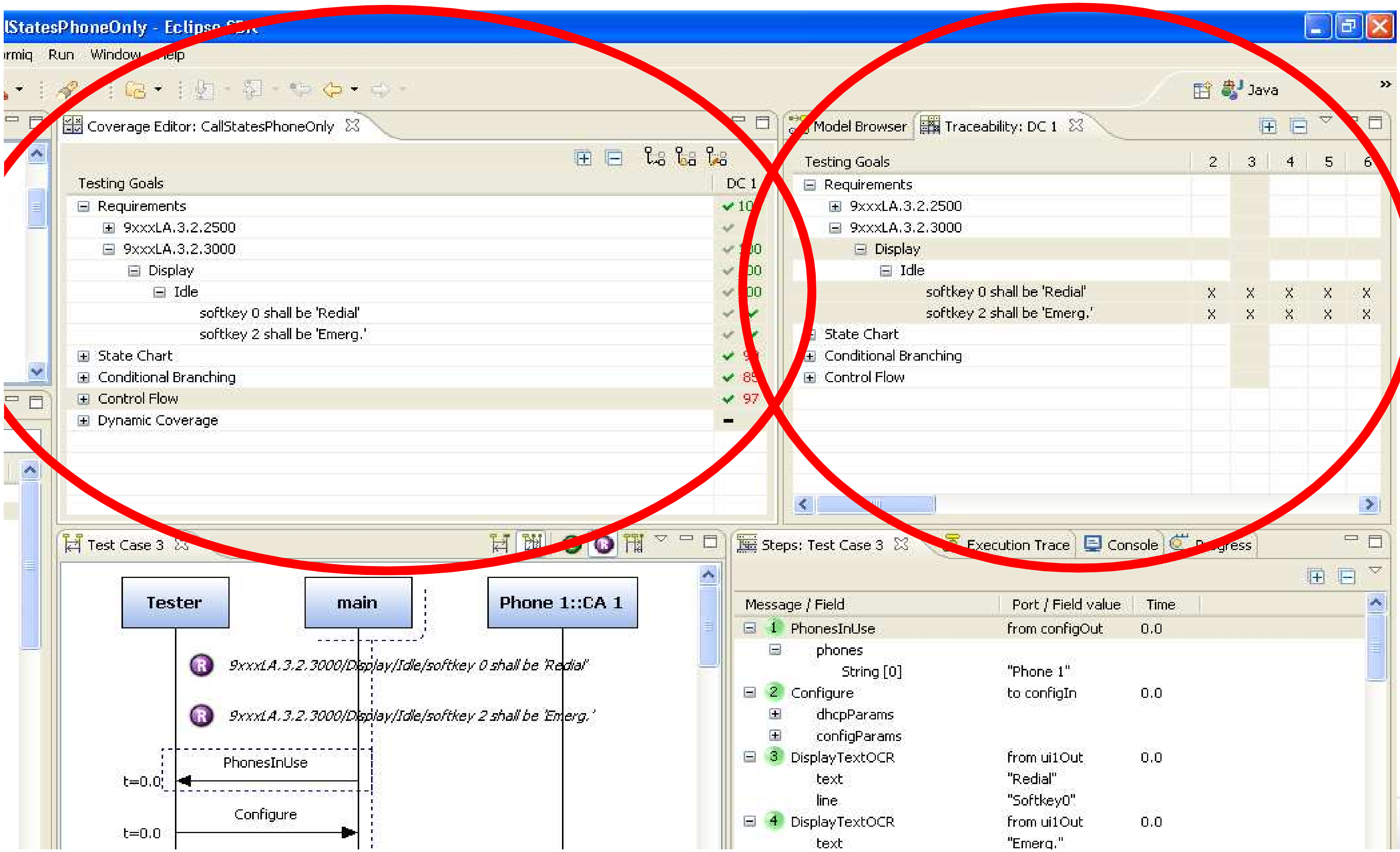
AVAYA
INTELLIGENT COMMUNICATIONS

- Manually create system model
- Automatically designs test input and expected output with data and timer handling
- Generates test reports
- Renders test scripts in Groovy



# Automated Test Design Generation

AVAYA
INTELLIGENT COMMUNICATIONS



| Testing Goals | DC 1 |
|---|---|
| Requirements | ✓ 10 |
| 9xxxLA.3.2.2500 | ✓ |
| 9xxxLA.3.2.3000 | ✓ 00 |
| Display | ✓ 00 |
| Idle | ✓ 00 |
| softkey 0 shall be 'Redial' | ✓ ✓ |
| softkey 2 shall be 'Emerg.' | ✓ ✓ |
| State Chart | ✓ 9 |
| Conditional Branching | ✓ 85 |
| Control Flow | ✓ 97 |
| Dynamic Coverage | — |

| Testing Goals | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| Requirements | | | | | |
| 9xxxLA.3.2.2500 | | | | | |
| 9xxxLA.3.2.3000 | | | | | |
| Display | | | | | |
| Idle | | | | | |
| softkey 0 shall be 'Redial' | X | X | X | X | X |
| softkey 2 shall be 'Emerg.' | X | X | X | X | X |
| State Chart | | | | | |
| Conditional Branching | | | | | |
| Control Flow | | | | | |

*9xxxLA.3.2.3000/Display/Idle/softkey 0 shall be 'Redial'*

*9xxxLA.3.2.3000/Display/Idle/softkey 2 shall be 'Emerg.'*

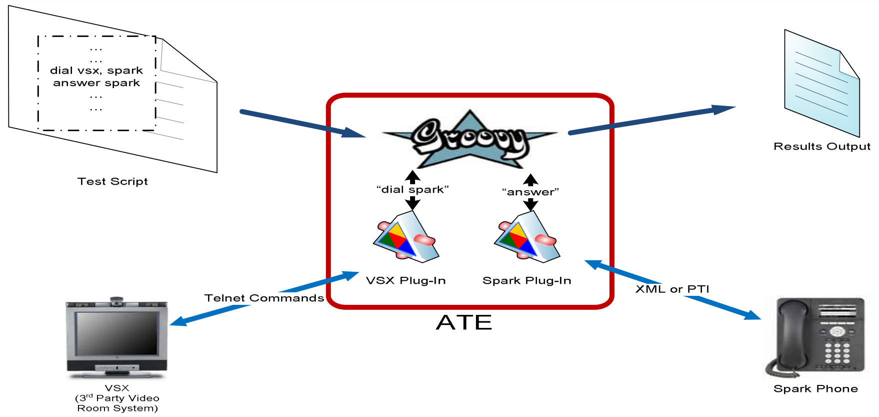
t=0.0 PhonesInUse

t=0.0 Configure

| Message / Field | Port / Field value | Time |
|---|---|---|
| 1 PhonesInUse | from configOut | 0.0 |
| phones | | |
| String [0] | "Phone 1" | |
| 2 Configure | to configIn | 0.0 |
| dhcpParams | | |
| configParams | | |
| 3 DisplayTextOCR | from ui1Out | 0.0 |
| text | "Redial" | |
| line | "Softkey0" | |
| 4 DisplayTextOCR | from ui1Out | 0.0 |
| text | "Emerg." | |

# Generated Test Scripts

- In order to generate directly executable test cases for ATE, Conformiq Designer™ uses a Groovy scripting backend



- This Groovy-fragment shows what a typical test script looks like:

```
// Define phone object i.e. the "end point"
endpoint1 = define(type:'spice',alias:'endpoint1',
                   ip:endpoint1_ip,backend:"pti-only",
                   spiceversion:"2.0")
register(endpoint:endpoint1,extension:endpoint1_extension,
       password:'1234',gatekeeper:mygk1)

// Navigate to contacts application using the phone object
button(endpoint1,"AddressBook")

// press "New" soft key on phone
button(endpoint1,"SoftKey0")
```

## Testing Efficiency Results

| Test Area | No. of Manual Test Cases | No. of Conformiq Test Cases | No. of Manual Test Steps | No. of Conformiq Test Steps | Test Case Coverage Gain |
|---|---|---|---|---|---|
| **Network Config** | 400 | 1440 | 5445 | 93000 | **3.60** |
| **Phone Apps** | 544 | 1360 | 6859 | 85000 | **2.50** |

| Test Area | Manual Effort in Hours | Conformiq Effort in Hrs | Productivity Gain |
|---|---|---|---|
| **Network Configuration** | 608 | 120 | **5.07** |
| **Phone Applications** | 827 | 428 | **1.93** |

## Automated Test Design Benefits

AVAYA
INTELLIGENT COMMUNICATIONS

- Higher Productivity in Test Design
- Improved Test Coverage
- Simplified Test Maintenance
- Model Reuse
- Simplified Test Harness
- Specification Issues Are Found Earlier
- Test Automation Backlog Is Eliminated

## Thank You

AVAYA
INTELLIGENT COMMUNICATIONS

# Questions