



# The Testing and Test Control Notation **TTCN-3**

*Ina Schieferdecker*



## Agenda

- Introduction
- Overview on TTCN-3
- An Example
- Test Execution
- Outlook



## TTCN-3

- A standardized alternative to proprietary test systems
  - Developed by a large group of testing experts
  - Used by a growing community
  - Proven by tools
  - Maintained at ETSI
- A test specification and implementation language
  - Test and Testing Control Notation = TTCN-3
  - Multipart standard covering
    - Testing concepts
    - Semantics
    - Presentation formats
    - Execution interfaces



## TTCN-3

- Applicable for all kinds of **black-box** and **grey-box testing** for reactive and distributed systems, e.g.
  - Fixed and mobile telecommunication (ATM, GSM, GPRS, UMTS)
  - Internet (IPv6, SIP)
  - Middleware platforms (CORBA, CCM, EJB, Web Services)
  - Embedded systems (automotive, avionics)
  - Programming interfaces (Java, XML)
- Applicable to **many kinds of testing**
  - Development, system, integration, interoperability, scalability ...
  - And also conformance testing
- As a test solution in industrial software development and for standardized test suites

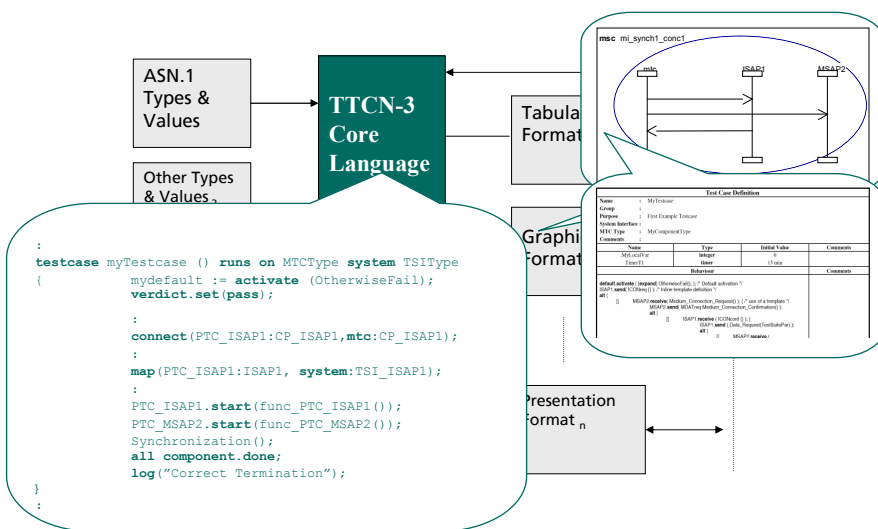


# Main Aspects of TTCN-3

- Triple C
  - **Configuration:** Dynamic concurrent test configurations with test components
  - **Communication:** Various communication mechanisms (synchronous and asynchronous)
  - **Control:** Test case execution and selection mechanisms
- Improved
  - Harmonized with ASN.1
  - Module concept
- Extendibility via attributes, external function, external data
- Well-defined syntax, static and operational semantics
- Different presentation formats

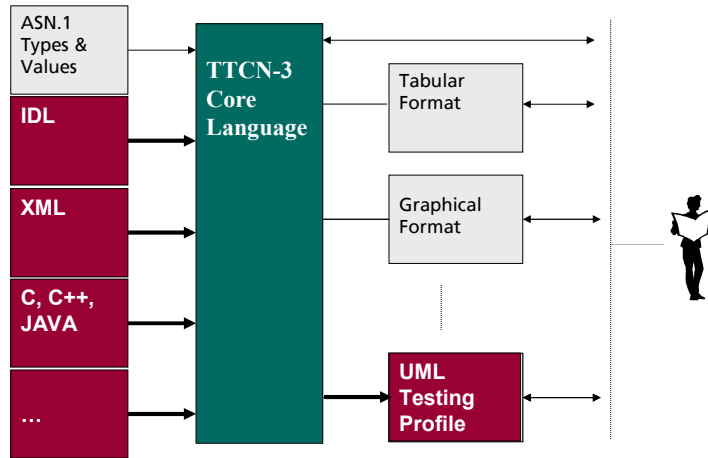


# Overview on TTCN-3

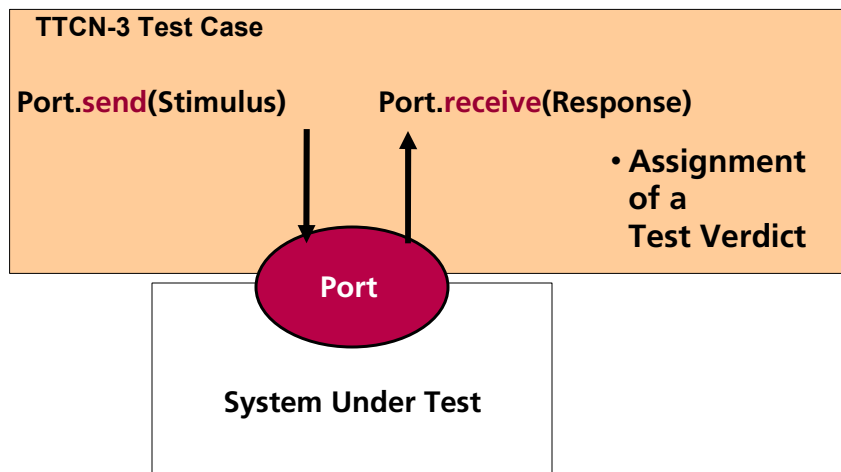




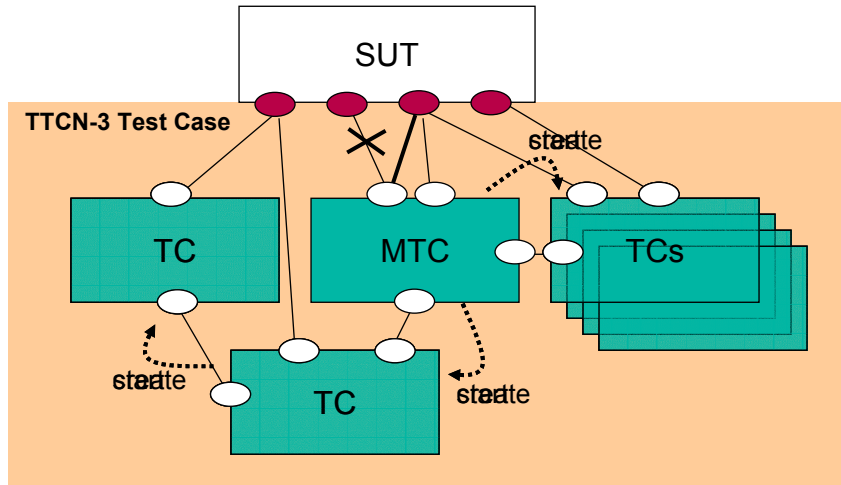
# TTCN-3 and Software Testing



# Based Black-Box Testing



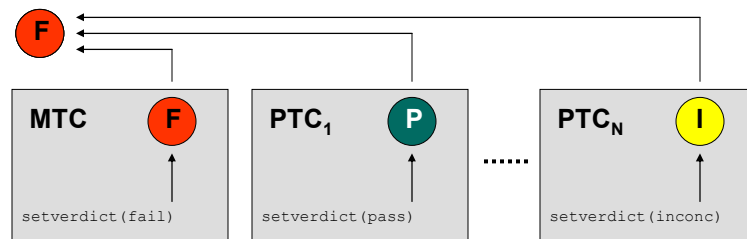
# Component-Based Test System



# TTCN-3 – Test Verdicts

- Test verdicts: **none** < **pass** < **inconc** < **fail** < **error**
- Each test component **has** its own local verdict, which can be set (**setverdict**) and read (**getverdict**).
- A test case returns a global verdict

*Verdict returned by the test case when it terminates*





## Basic Elements of TTCN-3

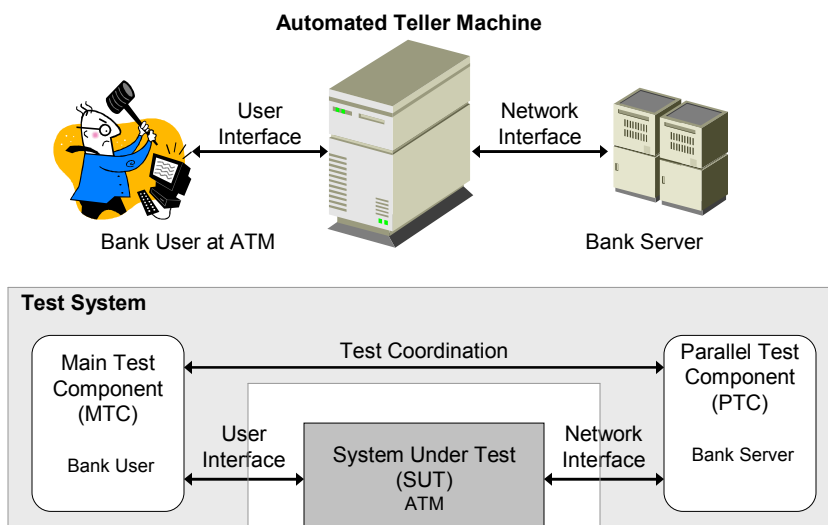
- **Module** covers declarations and control
- **Templates** (test data description) and matching mechanisms (pattern matching)
- **Test configurations**
  - Formally defined interfaces to the SUT
  - Dynamic creation of test component
  - Concurrency to describe distributed test setups
- **Test cases**
  - Small (complete) separate compilable programs
  - Share (type and data) information
- **Test verdicts**

21st TAV, June 2004, Berlin

I. Schieferdecker: TTCN-3



## An Example



21st TAV, June 2004, Berlin

I. Schieferdecker: TTCN-3



## The TTCN-3 Module

```
module ATM_Test {  
  
  modulepar {  
    float      maxDurOfTC_Par;  
    account_Type  validAccount_Par;  
    card_Type    validCard_Par;  
    integer     validPin_Par, validAmount_Par;  
    float      maxDurOfTC_par  
  }  
  
  external const float TestExecutionTime;  
  ...  
}
```

Module Parameter

Constants



## Test Data: Types and Ports

```
type port hardwareInterface_PType message {  
  out authenticationData_Type, operationHWI_Type;  
  in  authenticationComplete_Type, operationComplete_Type,  
      status_Type  
}  
  
type record authenticationData_Type {  
  card_Type card,  
  integer   pin  
}  
  
type record authenticationComplete_Type {  
  boolean success,  
  reason_Type reason  
}  
  
type enumerated reason_Type {  
  noReason,  
  unknown,  
  ...  
}
```

Message Port

Message Types



## Test Data: Templates

Message Type

```
type record operationHWI_Type {  
    HWI_ops operation,  
    integer    argument  
}
```

Message Template

```
template operationHWI_Type  
validWithdrawalOp_Template := {  
    operation := withdrawal,  
    argument  := validAmount_Par  
}
```



## Test Configuration: Components

MTC

```
// Component Type for Hardware Emulator  
type component hardwareEmulator_CType {  
    port hardwareInterface_PType hwiCom;  
    port testCoordination_PType coHWE;  
    timer testCaseGuard := maxDurOfTC_par;  
}
```

PTC

```
// Component Type for Bank Emulator  
type component bankEmulator_CType {  
    port networkInterface_PType niCom;  
    port testCoordination_PType coBE;  
}
```

SUT

```
// Component Type for the Test System Interface  
type component ATM_Interface_CType {  
    port hardwareInterface_PType hwiSUT;  
    port networkInterface_PType niSUT;  
}
```





## Test Behavior: send/receive

Typically: One send

```
hwiCom.send(validWithdrawalOp_Template);
```

Several receives

```
alt {
  [] hwiCom.receive(operationComplete_Type:
    {withdrawal, true, ?})
    { setverdict(pass); }
  [] hwiCom.receive(operationComplete_Type:?)
    -> value erroneousWithdrawal
    { setverdict(fail); }
  [] ...
}
```



## Test Behaviour: Test Case

MTC, SUT

```
testcase validWithdrawal (inout reason_Type reason)
runs on hardwareEmulator_CType system ATM_Interface_CType {
  ...
  var bankEmulator_CType BE_PTC:=bankEmulator_CType.create;

  map(theSystem:niSUT, BE_PTC:niCom);
  connect(self:coHWE, BE_PTC:coBE);
  map(theSystem:hwiSUT, self:hwiCom);

  BE_PTC.start(BE_validWithdrawal());

  testCaseGuard.start; // start guarding timer

  result :=
  authentication(validCard_Par, validPin_Par, reason);

  ...
}
```

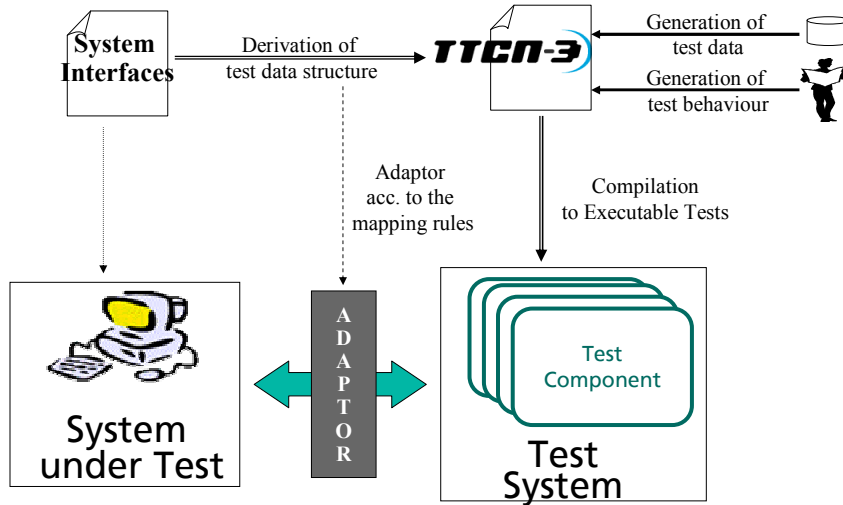
PTC

Connections

PTC behavior



## Overall Picture

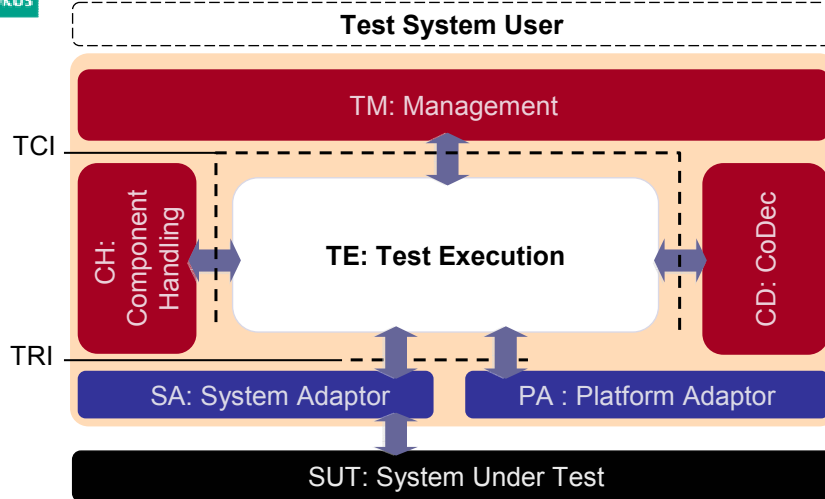


21st TAV, June 2004, Berlin

I. Schieferdecker: TTCN-3



## The Execution Interfaces



TRI – TTCN-3 Runtime Interfaces

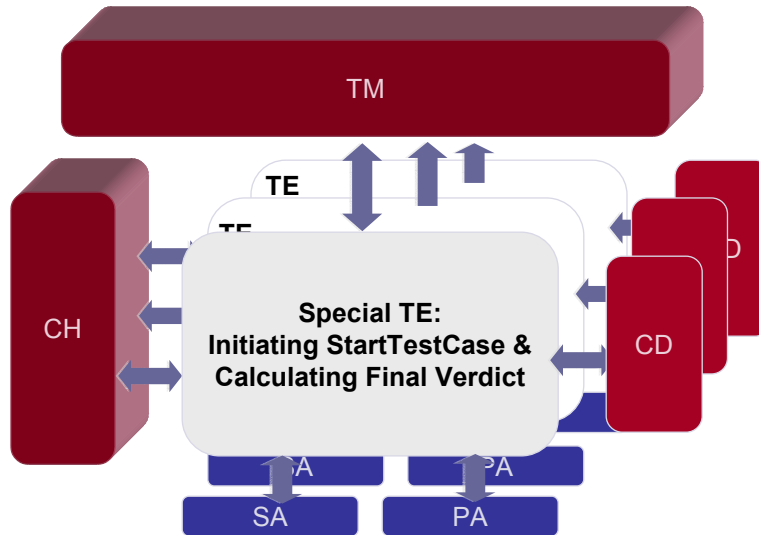
TCI – TTCN-3 Control Interfaces

21st TAV, June 2004, Berlin

I. Schieferdecker: TTCN-3



# Local and distributed test setups



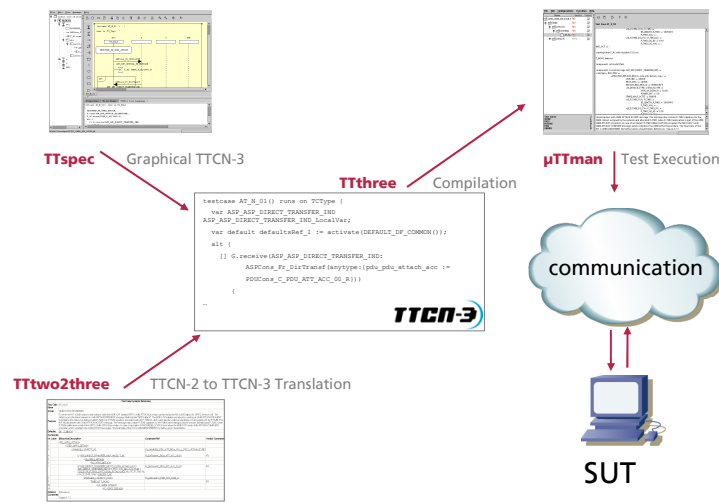
21st TAV, June 2004, Berlin

I. Schieferdecker: TTCN-3



# The Tool Chain

TTthree free for education  
[www.testingtech.de](http://www.testingtech.de)



21st TAV, June 2004, Berlin

I. Schieferdecker: TTCN-3



## Upcoming TTCN-3

- MockUp end of 2004: mainly error correction
- New version v3.1.1 in 2005
  - Better structuring: packages, core language extensions and profile
  - Possibly support for
    - Real-time and performance testing
    - Logging
    - Object-oriented data
    - External applications
    - Etc.



## Summary

- TTCN-3 as the new standardized test specification and implementation language
- TTCN-3 has applications e.g. for
  - Telecommunication (e.g. mobile communication and Internet)
  - Software (e.g. object and component based systems, middleware platforms, Web services)
  - Control systems in automotive, transportation and avionics
- TTCN-3 has many tool vendors and users
- TTCN-3 is a continuously maintained and enhanced test technology



Thank You !

*Any Questions ?*