

Basics and application trends for the Testing and Test Control Notation TTCN-3

Axel Rennoch, Ina Schieferdecker

22nd ICSSEA, Paris, 7th December 2010

Contents

Contents

- **TTCN**
  - Introduction
  - History
- **Status**
  - Concepts and tools
  - Applications:
    - Industrial domains, MBT
- **Future**
  - Latest releases, extension packages
  - Outlook: Embedded TTCN-3
- **Conclusion**

How Much Does Testing Cost?

„ ... the national annual cost estimates of an inadequate infrastructure for software testing are estimated to be **$59.5 billion**.
The potential cost reduction from feasible infrastructure improvements is **$22.2 billion**.“

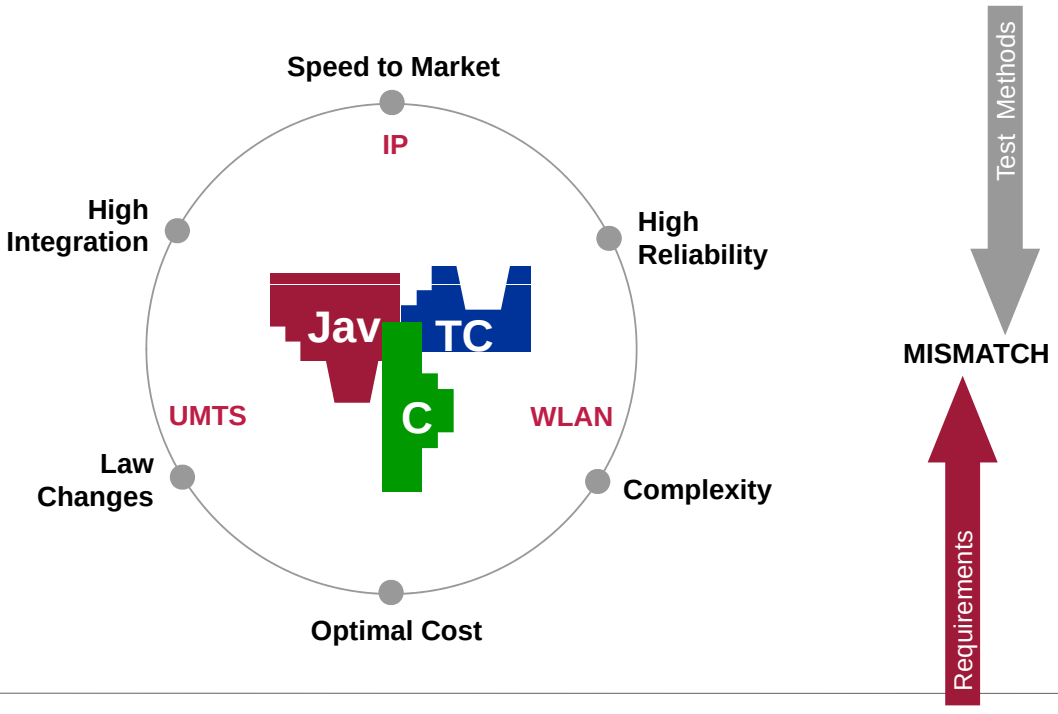
The Economic Impacts of Inadequate Infrastructure for Software Testing

Study by NIST, May 2002

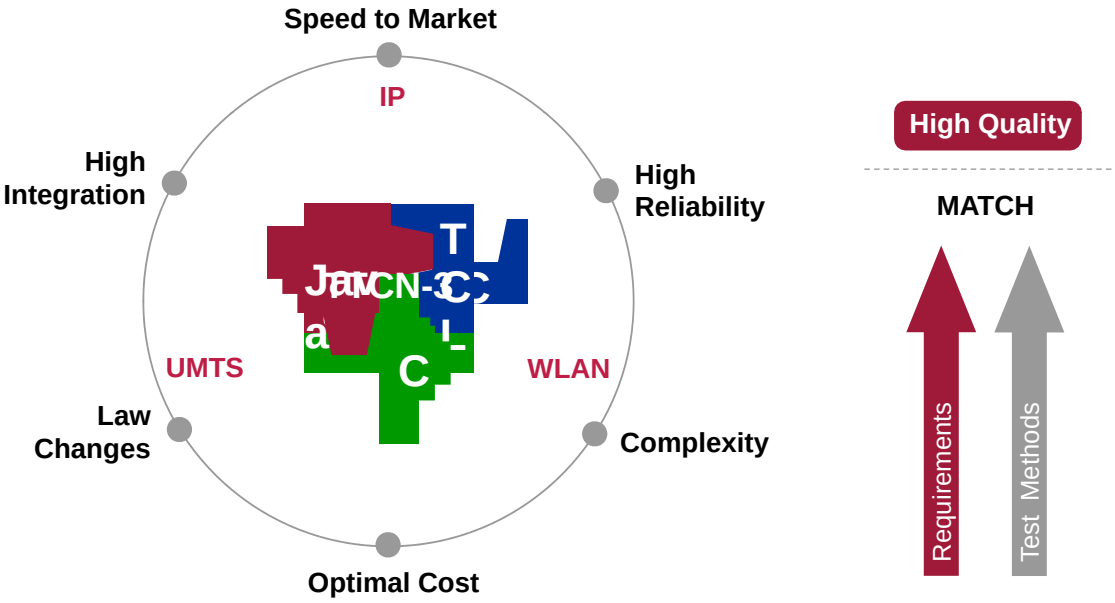
Testing Today

- Is
  - Important
  - Expensive
  - Time critical
- But
  - Only rarely practiced
  - Unsystematic
  - Performed by hand
  - Error-prone
  - Uncool („If you are a bad programmer you might be a tester.“)
  - Destructive

Why Using TTCN-3 (1)



Why Using TTCN-3 (2)

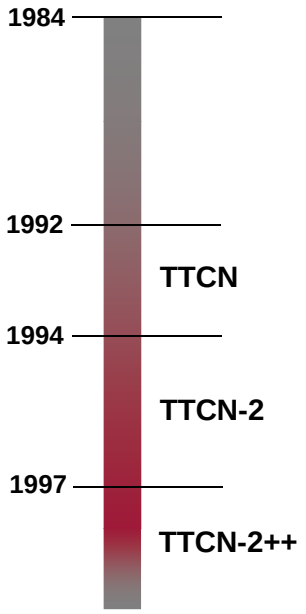


The Testing and Test Control Notation

Idea and design Principles of TTCN-3

- **One test technology for different tests**
  - Distributed, platform-independent testing
  - Integrated graphical test development, -documentation and –analysis
  - Adaptable, open test environment
- **Areas of Testing**
  - Regression Testing
  - Conformance and Functionality Testing
  - Interoperability and Integration Testing
  - Load/ Stress Testing

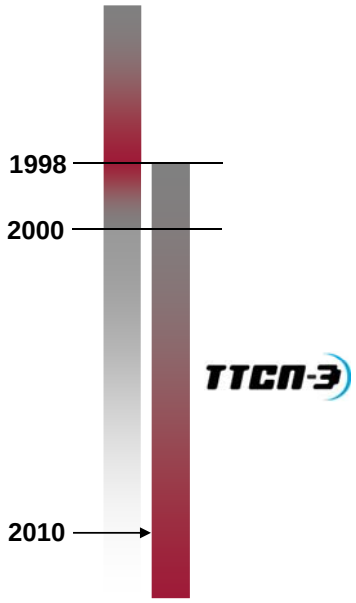
History (1)



- TTCN (1992)
  - Published as an ISO standard
  - Tree and Tabular Combined Notation
  - Used for protocol testing only
    - GSM, N-ISDN, B-ISDN
- TTCN-2/2++ (1997)
  - Concurrent tests
  - Modularization
  - Manipulate external data
  - Rather for conformance testing
  - Developed by ETSI MTS

Competence Center **MOTION** Fraunhofer **FOKUS**

History (2)



- TTCN-3 (2000)
  - Testing and Test Control Notation
  - Developed by ETSI MTS
  - Proper language
    - Well defined syntax and semantics
  - Enhanced communication, configuration and control
  - Standard test specification
    - SIP/IMS, SCTP, HiperLan, HiperAccess, IPv6 ...

TTCN-3 Multipart-Standard

- ETSI ES 201 873-1 TTCN-3 Core Language (CL)
- ETSI ES 201 873-2 TTCN-3 Tabular Presentation Format (TFT)
- ETSI ES 201 873-3 TTCN-3 Graphical Presentation Format (GFT)
- ETSI ES 201 873-4 TTCN-3 Operational Semantics
- ETSI ES 201 873-5 TTCN-3 Runtime Interface (TRI)
- ETSI ES 201 873-6 TTCN-3 Control Interfaces (TCI)
- ETSI ES 201 873-7 Integration of ASN.1
- ETSI ES 201 873-8 Integration of IDL
- ETSI ES 201 873-9 Integration of XML
- ETSI ES 201 873-10 T3Doc

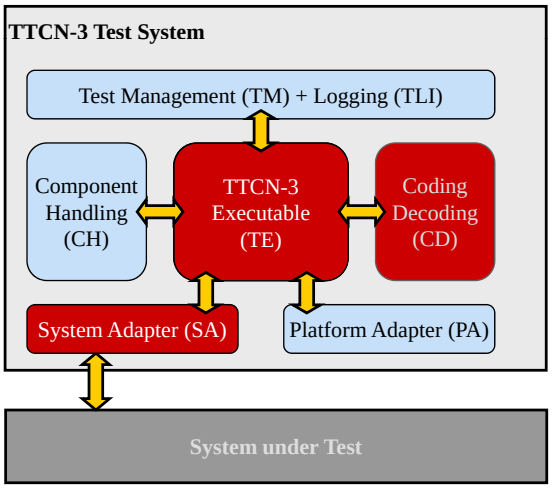
- Standard available for download at http://www.ttcn-3.org
- Also standardized by the ITU-T as ITU-T Z.140

- NEW: TTCN-3 Extension packages ETSI ES 202 78x!

Contents

TTCN-3: Testing and Test Control Notation

- A standardized alternative to proprietary test systems
  - Developed by a large group of testing experts
  - Used by a growing community
- Enabling a testing middleware,
  - *unifying* methods, tools, test infrastructure, documentation and training
  - domain-specific extensions
- TTCN-3 is more than a notation: **TTCN-3 test system**
- The only international standard for
  - Test specification *and* implementation

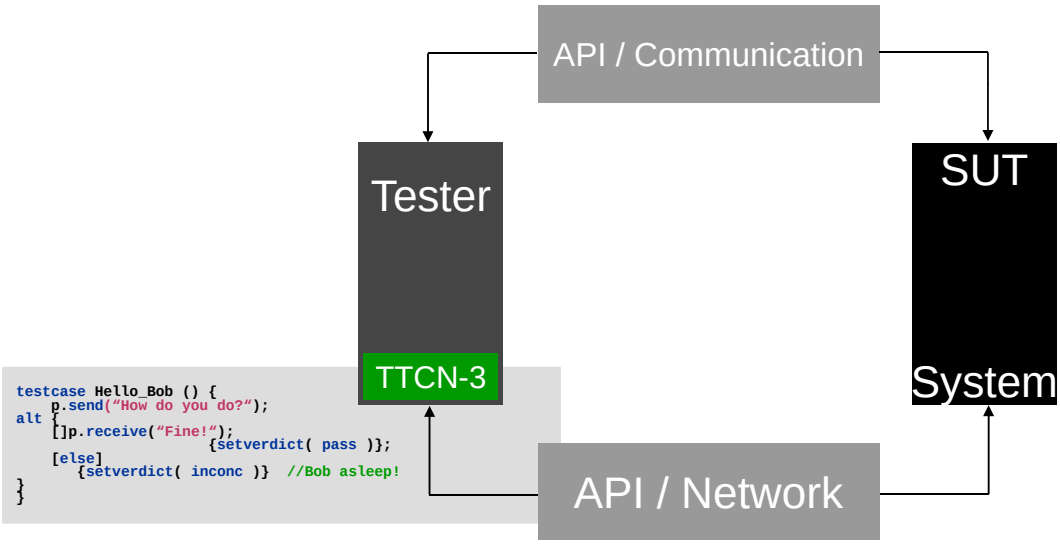


What is TTCN-3?

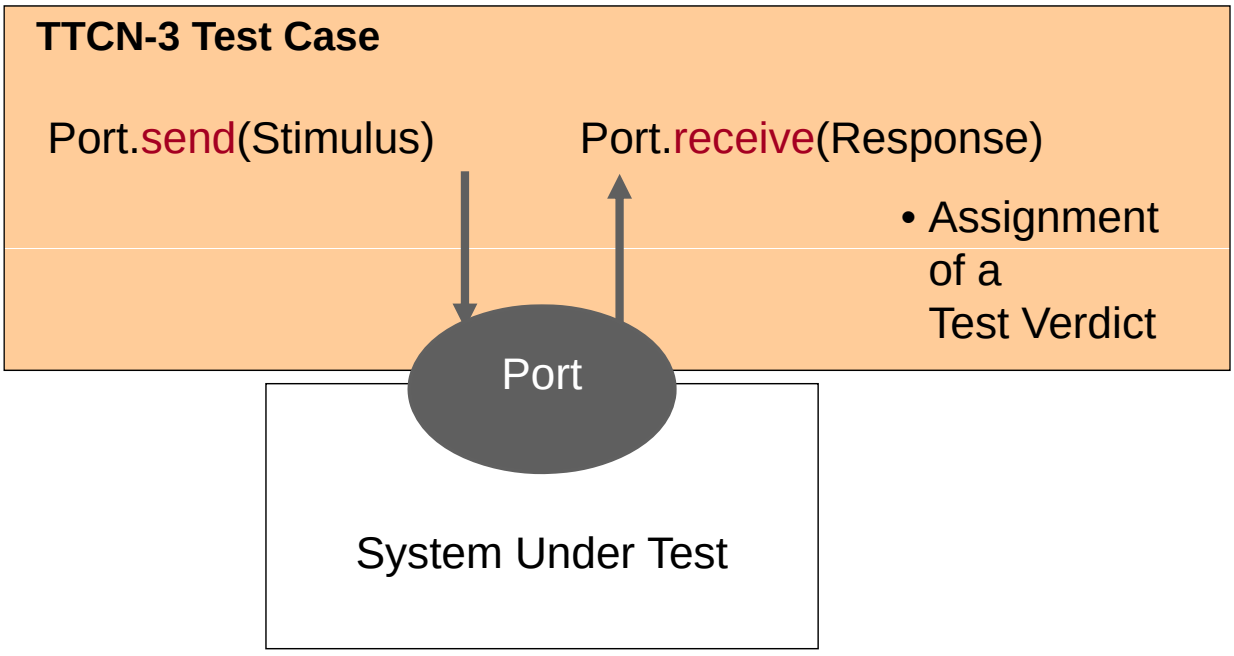
- Testing and Test Control Notation
- Internationally standardized testing language for formally defining test scenarios. Designed purely for testing
- Taking the best bits of TTCN-2 and combining them with a new more powerful textual notation

```
testcase Hello_Bob () {
    p.send("How do you do?");
alt {
    []p.receive("Fine!");
          {setverdict( pass )};
    [else]
          {setverdict( inconc )}  //Bob asleep!
}
}
```

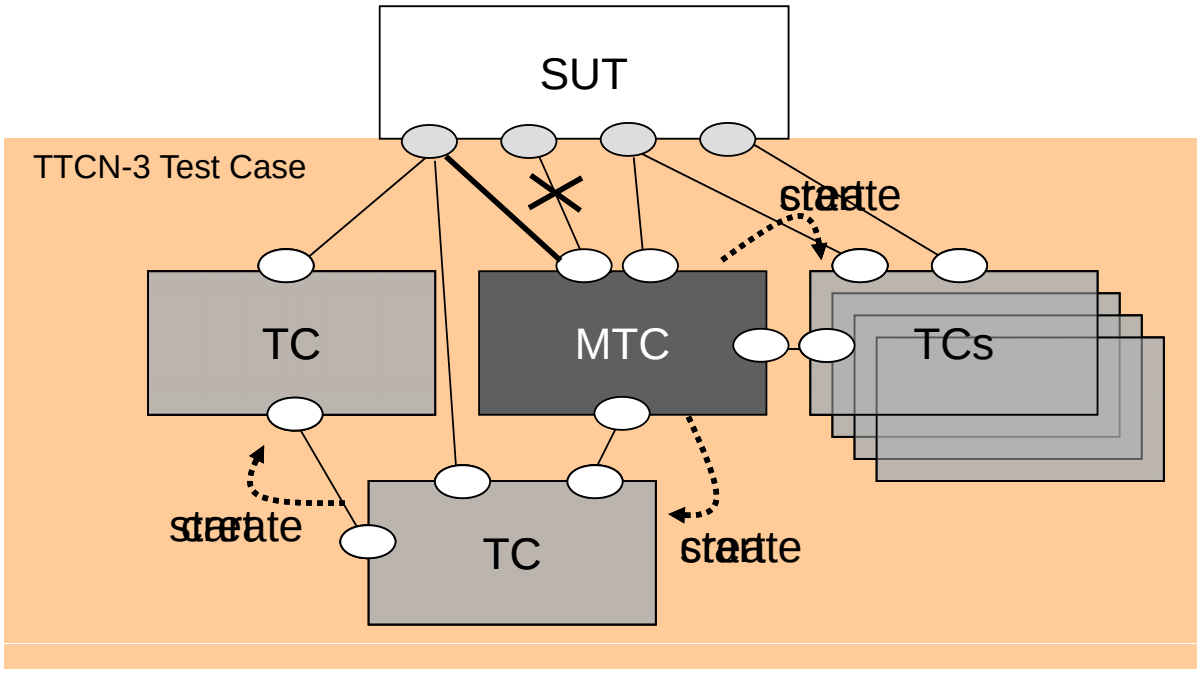
TTCN-3 Execution



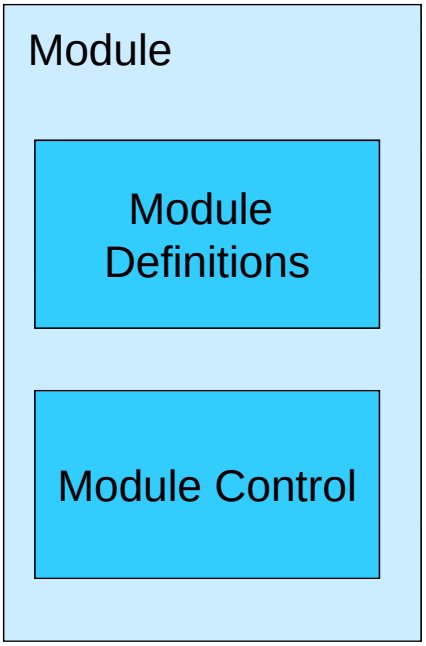
TTCN-3 is designed for Black-Box Testing



TTCN-3 Test Configuration may change dynamically



TTCN-3 Module



- The top level entity of TTCN-3 is **module**.
- A module can **import** definitions from other modules.
- A module contains a definition part and a **control** part.

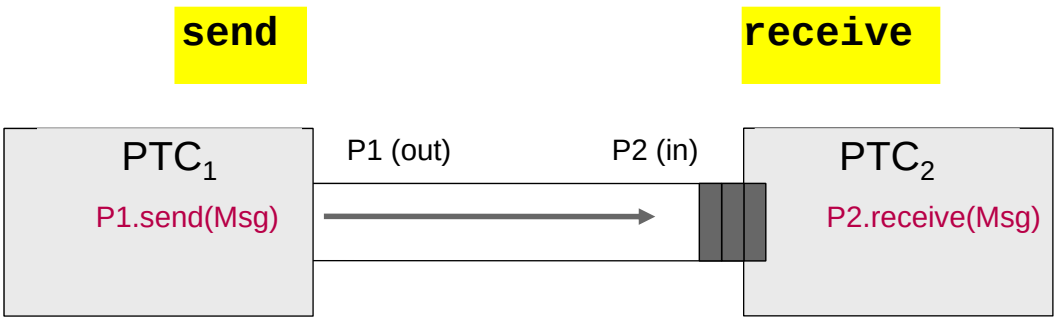
```
module MyModule {
   // definition part

   control {
   // test execution logic
   }
}
```

Major language elements of TTCN-3 notation

module definitions	
Imports	Importing definitions from other modules defined in TTCN-3 or other languages
Data Types	User defined data types (messages, PDUs, information elements, …)
Test Data	Test data transmitted/expected during test execution (templates, values)
Test Configuration	Definition of the test components and communication ports
Test Behavior	Specification of the dynamic test behavior

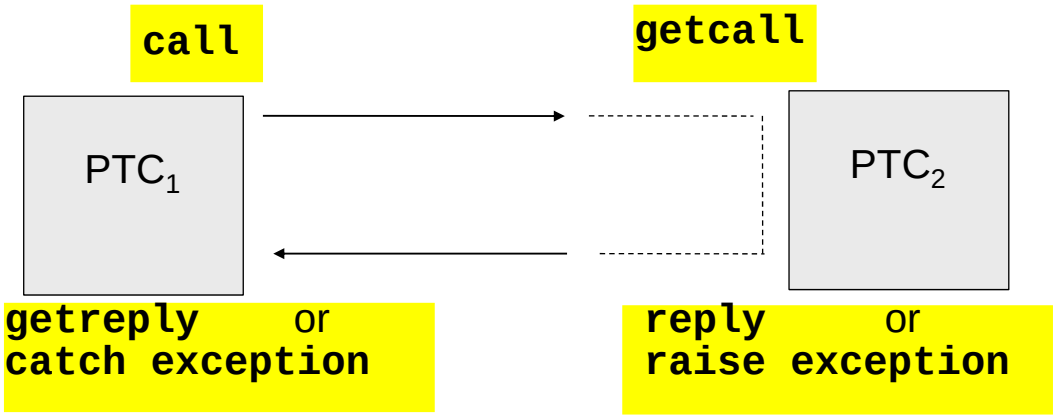
Message-Based Ports

- For sending and receiving *messages* of given type



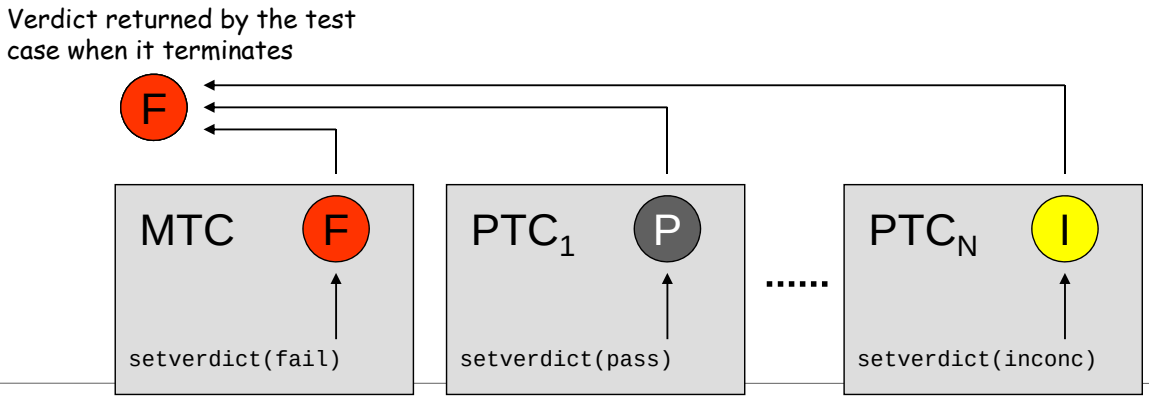
Procedure-Based Ports

- For invoking operations, receiving *operation* calls, replying, raising *exceptions* as well as for receiving replies and catching exceptions



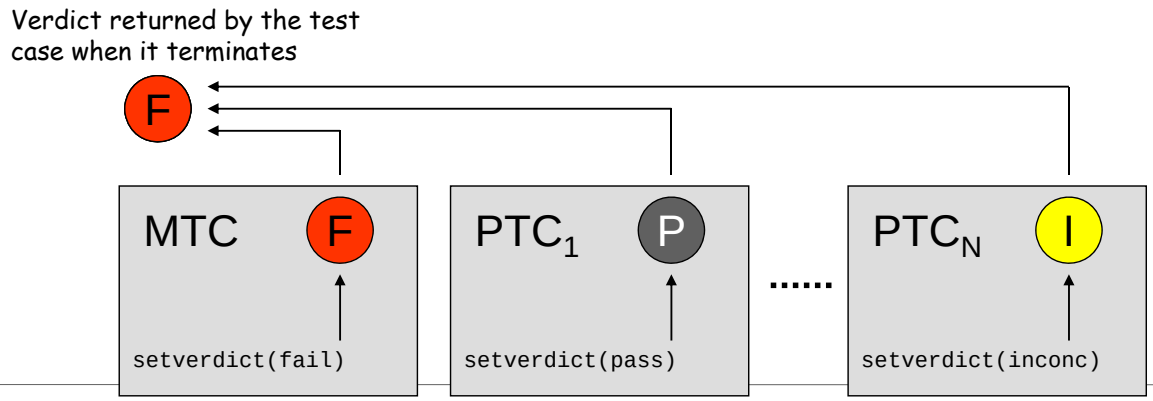
Test Verdicts

- 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

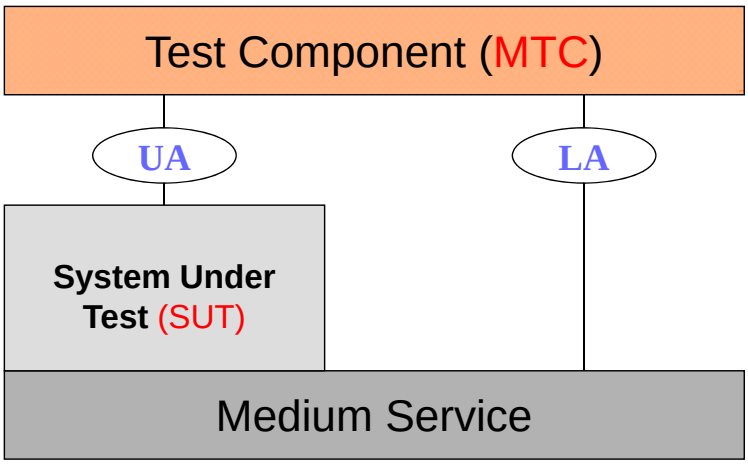


Test Verdicts

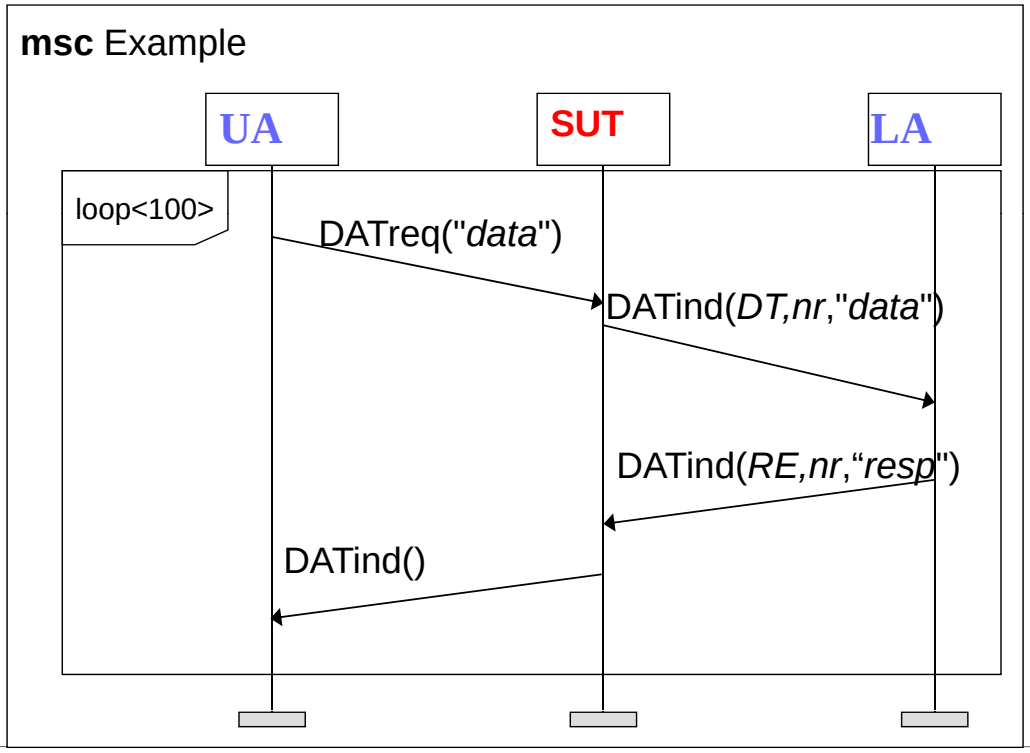
- Test verdicts: none < pass < inconc < fail < error
- Each test component has its own local verdict, which can be set (setverdict) and read (getverdict).
- A test case returns a global verdict



Sample: Test Configuration



Time Sequence Diagram for a Simple Behavour Example



Same Example in TTCN-3 Core Language

```
testcase Example( ) runs on MTC_Type {
        var default mydefault := activate (DefaultDef());
        T1.start;
        for (integer i:=1; i<=100; i:=i+1) {
                UA.send (DATreq:{"data"});
                LA.receive (DATind:{DT, nr, "data"});
                LA.send (DATind:{RE,nr, "resp"});
                UA.receive (DATind :{});
        }
        setverdict(pass);
        T1.stop;
}
```

Typical language elements:

type definitions:

- boolean, integer, float, bitstring, charstring, octectstring, hexstring
- record, set, enumeration, union

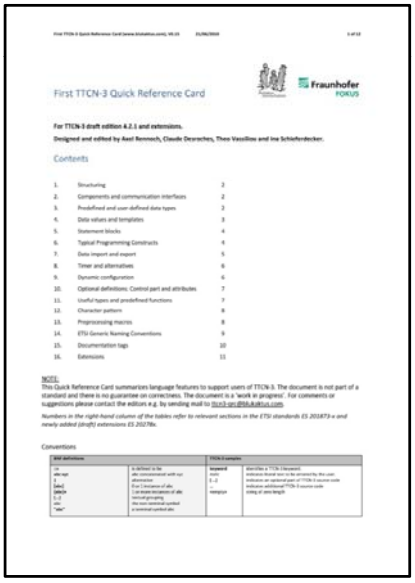
programming constucts:

- *message:* send/receive
- *procedure:* call/getcall, reply/getreply, raise/catch
- if-then-else, loops: for, while, do-while
- functions, alternatives
- component/port/timer control

Predefined functions:

- type conversion, lengthof (string), sizeof (records), …

Overview: e.g. TTCN-3 Quick Reference Card


First TTCN-3 Quick Reference Card

Fraunhofer FOKUS

For TTCN-3 draft edition 4.2.1 and extensions.

Designed and edited by Axel Rennoch, Claude Desroches, Theo Vassiliou and Ina Schieferdecker.

Contents

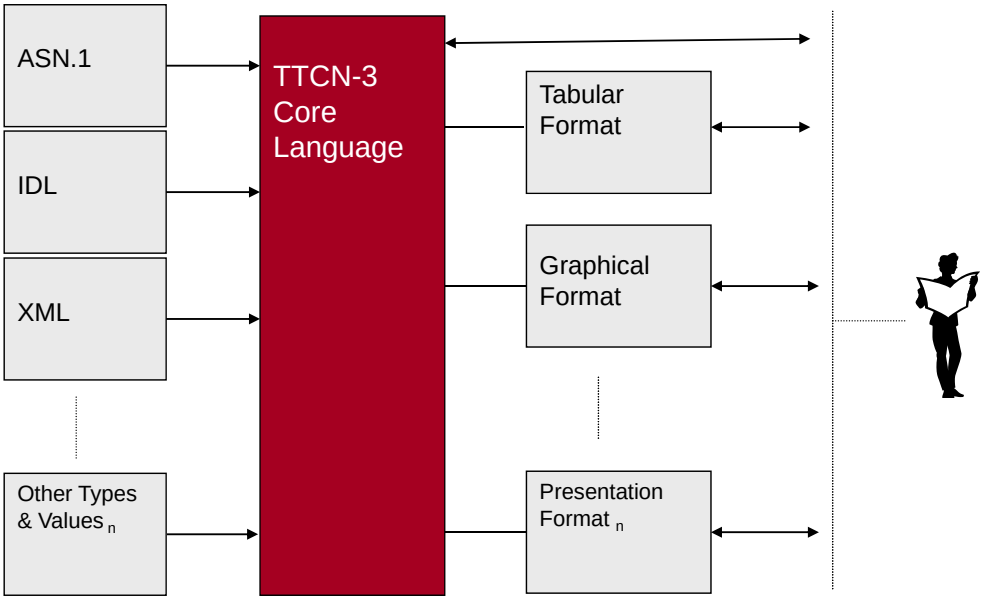
NOTE:
This Quick Reference Card summarises language features to support users of TTCN-3. The document is not part of a standard and there is no guarantee on correctness. The document is a 'work in progress'. For comments or suggestions please contact the editors e.g. by sending mail to ttcn3-sm@fokus.fraunhofer.de.

Numbers in the right-hand column of the tables refer to relevant sections in the ETSI standards ES 201873-x and newly added (draft) extensions ES 202784.

Conventions

BNF definition		TTCN-3 example	
::=	is defined to be	keyword	identifies a TTCN-3 keyword
<abc-xyz>	abc concatenated with xyz	[text]	indicates optional text to be selected by the user
{abc}	zero or more abc	1 .. 2	indicates an enumeration of tokens in TTCN-3 source code
[abc]	0 or 1 occurrence of abc	...	indicates a range of tokens in TTCN-3 source code
{abc}n	1 or more occurrences of abc	repeatpos	indicates a repeatable pattern
(...)	normal grouping		
abc	the non-terminal symbol		
"abc"	a terminal symbol abc		

TTCN-3 is an „open“ standard



Openness for other data types: Example

XML to TTCN-3 Mapping

- Mapping of XML to TTCN-3
  - defined by the ETSI
  - Part 9 of the TTCN-3 multipart-standard
  - restricted to XML data **types** (XSD types) only
- Type import:
  - uses the import mechanism of TTCN-3 for immediate use
  - `import from XSD language "XML1.1" all;`
  - language identifiers:
    - "XML" or "XML1.0" for W3C XML 1.0
    - "XML1.1" for W3C XML 1.1

Technical conversion of XSD to TTCN-3

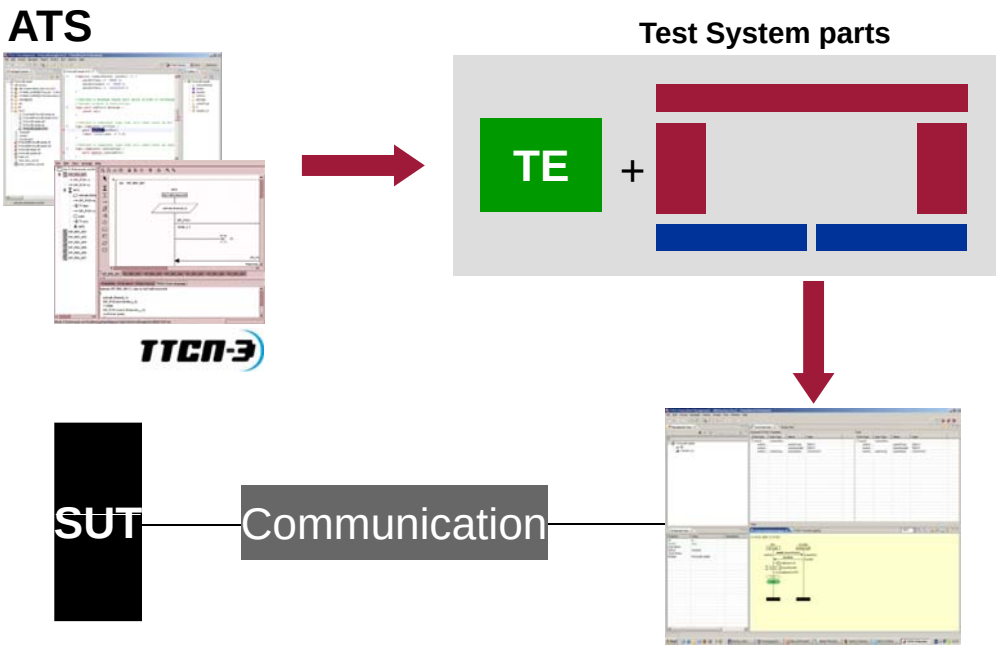
- 16 identifier name conversion rules
- 3 rules to specify the order of mapping
  - **XSD component**

```
<complexType name="e15">
  <sequence minOccurs="5" maxOccurs="10">
    <element name="foo" type="integer"/>
    <element name="bar" type="float"/>
  </sequence>
</complexType>
```

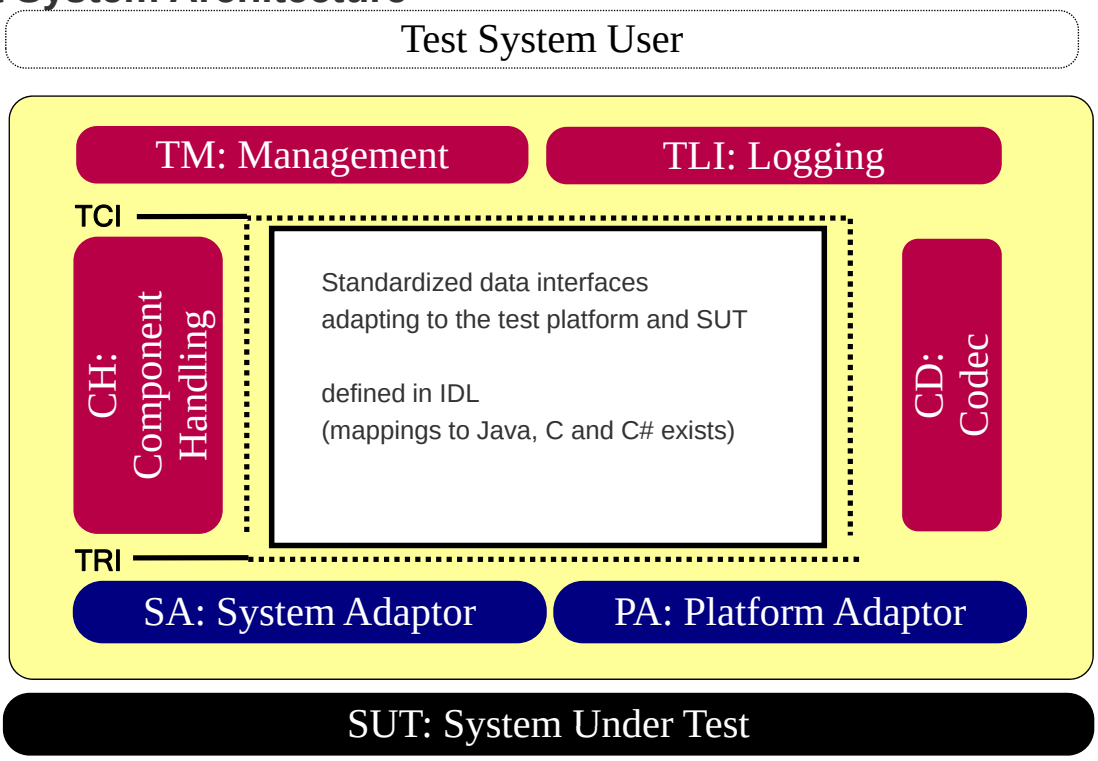
  - **corresponding TTCN-3**

```
type record E15 {
  record length(5 .. 10) of record {
     XSD.Integer foo,
     XSD.Float bar } sequence_list
} with {variant "name as uncapitalized "}
```

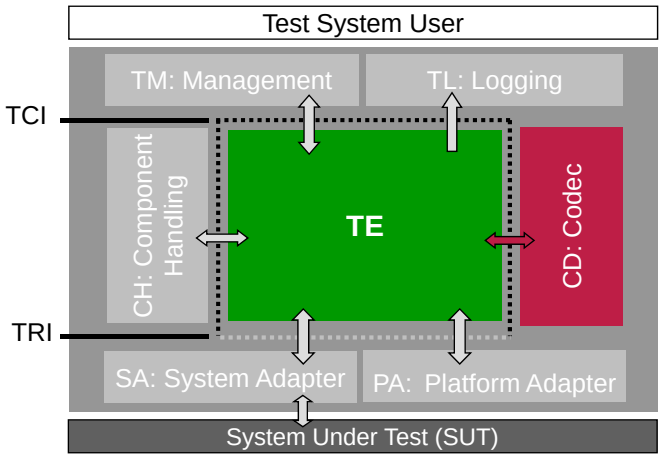
More than a notation: Test system implementation



Test System Architecture



Sample interface: TE - CD



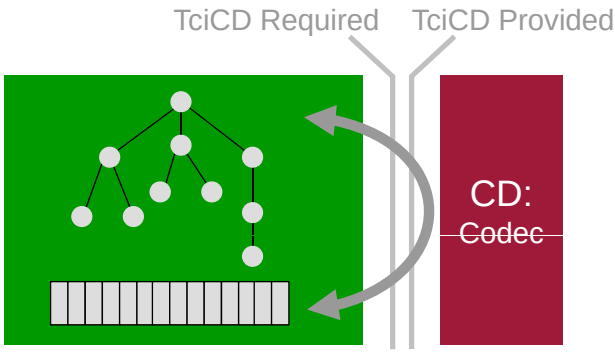
TE – TTCN-3 Executable

CD – Codec

ETSI ES 201 873-5 TTCN-3 Runtime Interface (TRI)

ETSI ES 201 873-6 TTCN-3 Control Interfaces (TCI)

The Codec and Value Interface

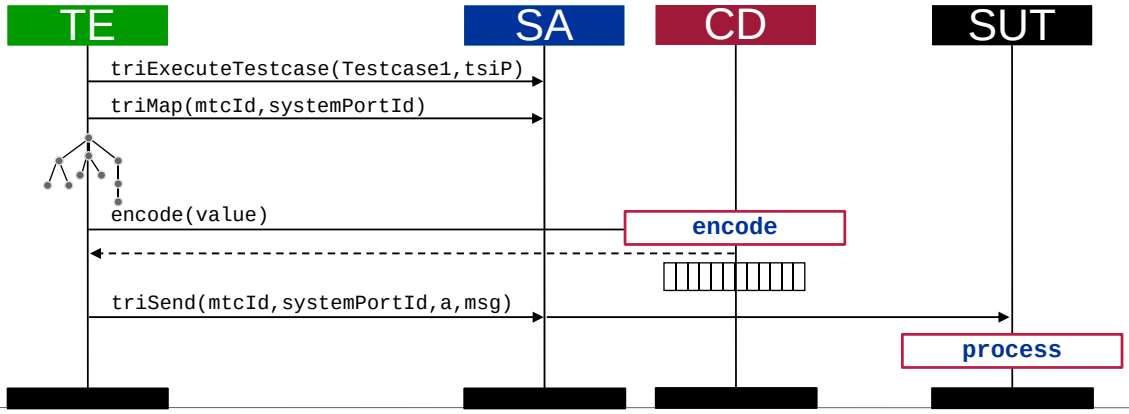


- Codec translates between abstract and concrete presentation

- Complete set of provided (**`TciCDProvided`**) operations
  - `TriMessageType` ***`encode`*** `(in Value value)`
  - `Value` ***`decode`*** `(in TriMessageType message, in Type hyp)`

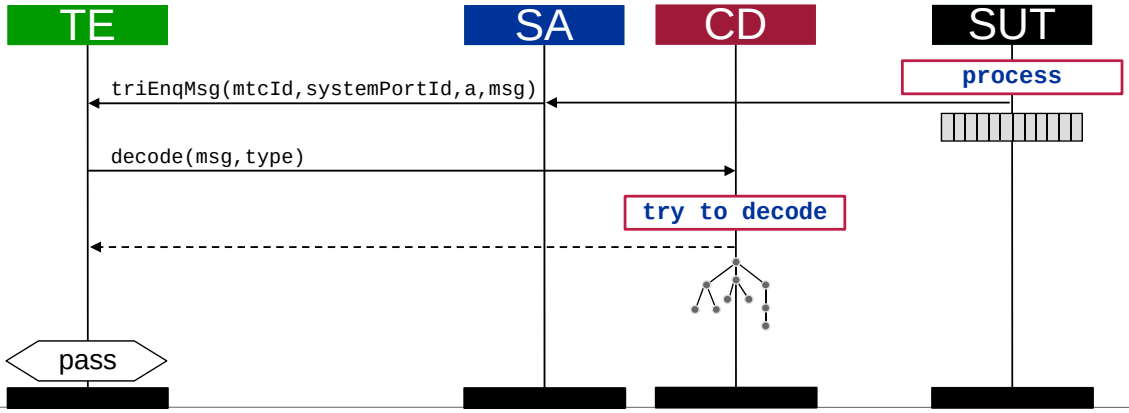
Dynamics of the Codec (Sending)

```
testcase Testcase1() runs on DNSTester system TSI {
        map(mtc:P, system:P);
        P.send(query);
        P.receive(answer);
        setverdict(pass);
}
```



Dynamics of the Codec (Receiving)

```
testcase Testcase1() runs on DNSTester system TSI{
        map(mtc:P, system:P);
        P.send(query);
        P.receive(answer);
        setverdict(pass);
}
```



Commercial TTCN-3 tools (source: www.ttcn-3.org)

TTCN-3 Compilers and Interpreters

- **Exhaustif/TTCN**: compiler (C++) produced by Métodos y Tecnología (MTP), Spain.
- **OpenTTCN:** interpreter (C, Java, C# interfaces) produced by OpenTTCN Ltd, Finland.
- **MessageMagic:** compiler (C/C++, Java, C#) produced by ELVIOR, Estonia.
- **Real Time Developer Studio**: modelling tool including TTCN-3 compiler by PragmaDev, France.
- **TAU Tester:** compiler (C) by IBM.
- **TTCN-3 toolbox**: compiler (C) by Danet Group, Germany.
- **TTCN-3 Express**: compiler (C#) by Fraunhofer FIRST and Metarga GmbH, Germany.
- **TTworkbench**: compiler (C, Java) by Testing Technologies, Germany.

TTCN-3 Generators

- **Qtronic** by Conformiq OY, Finland. generate complete TTCN-3 test suites from e.g., UML, Java, or C# models.
- **MaTeLo** by All4Tec, France (TTCN-3 test suites from usage models specified using Markov chains).
- **MOTES** by ELVIOR, Estonia (from the state model of the SUT)

Open source TTCN-3 tools (source: www.ttcn-3.org)

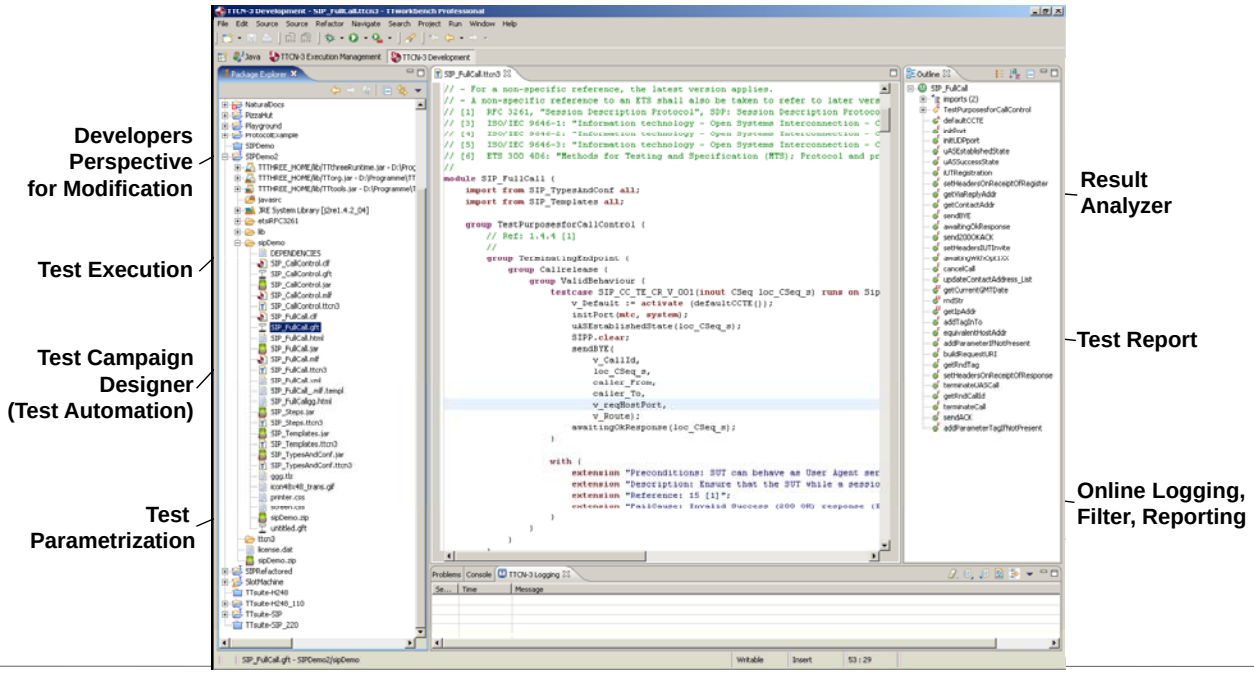
- **LoongTesting** testing platform including TTCN-3 **compiler** and **integrated development environment** by Information Processing Center of USTC , China.
- **BBT** TTCN-3 **Compiler**, by BroadBit, Hungary.
- **TRex:** by University of Göttingen to provide **IDE** functionality for TTCN-3 core notation, and to support assessment and automatic restructuring of TTCN-3 test suites. (open-source Eclipse plug-in).
- **T3doc** by Federico Engler and further developed by ETSI. for generating HTML **documentation** via tagged TTCN-3 comments.
- **Codec generator** by IRISA as part of T3DevKit. It automatically **generates a codec** based on TTCN-3 type module(s), C++ codec functions.
- **T3DevLib** by IRISA as part of T3DevKit. It allows the development or **integration of Codec, SUT and Platform Adapter** implementations written in C++.

... and more academic prototype/research tools
(guideline checking, quality analysis, ...)

Selected test devices with TTCN-3 support – www.testingtech.com

- DCT2000© (Catapult Communications Corporation)
- E6651A Mobile WiMAX Test Set (Agilent Technologies)
- G35 Protocol Analyzer (Tektronix)
- MiNT T2230/1 AIME - 802.16e (Aeroflex)
- Nexus8610 Traffic Simulation System (Nexus Telecom)
- MobileRobot (ServiceForce.Com GmbH)
- Testerlyzer (Ruetz Systems Solutions GmbH)
- …

The seemless chain of TTCN-3 Tools

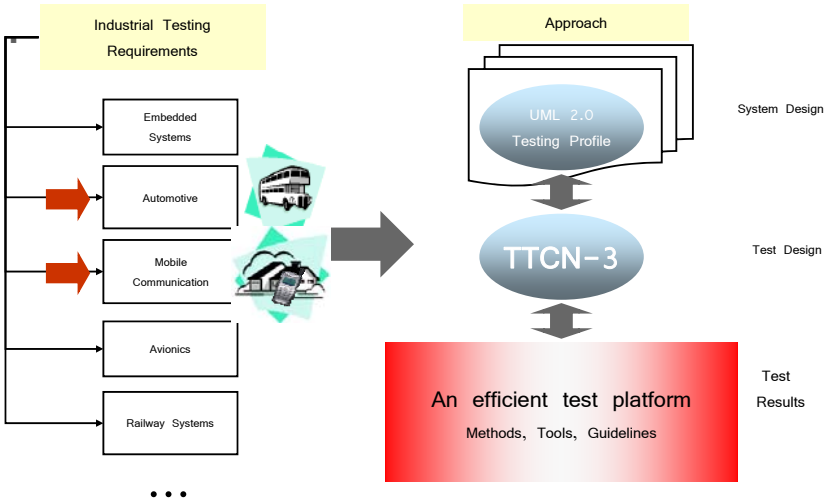


Contents

- **TTCN**
  - Introduction
  - History
- **Status**
  - Concepts and tools
  - Applications:
    - Industrial domains, MBT
- **Future**
  - latest releases, Extension packages
  - Outlook: Embedded TTCN-3
- **Conclusion**

TTCN-3 applications: various industrial domains

Objective:
- To develop an efficient **test platform** fulfilling **industrial testing requirements**
- To **execute high-level test models**, e.g. UML2 test profile



TTCN-3 domains: Telecom

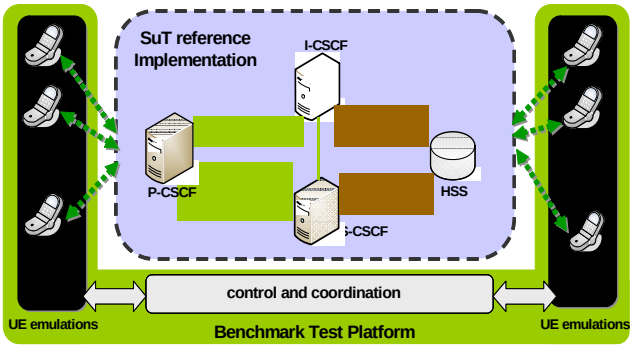
- Industrial use
  - Big companies with hundreds of TTCN-3 engineers: Ericson, Nokia, Siemens, Motorola
  - large distribution among SME
- Standardization bodies
  - standardized test suites:
    ETSI / 3GPP / OMA / TETRA and its members
  - IMS performance benchmark project:
    Intel, HP, BT, FOKUS and others
- Test tool manufacturer:
  - Commercial Tektronix, Catapult, Nexus, R&S, …
  - Open source community projects
- Certification program based on TTCN-3: WiMax forum

Telecom example: Performance Benchmarking

Sample SUT: IMS components
Test IMS components separately and in different configurations

Definitions using TTCN-3:
- standard call scenarios
  (e.g. voice call, conference call, application call, push-to-talk call)
- traffic set as an
  aggregation of call scenarios

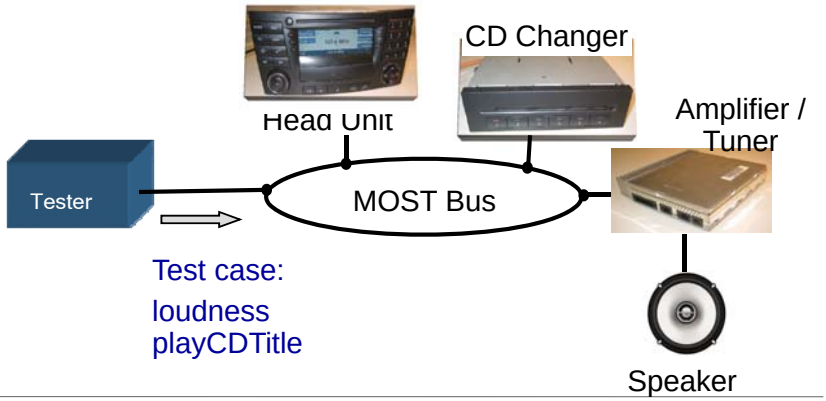
Performance:
- 5.000 - 10.000 IMS subscribers
  (per server)
- Up to 250 requests per second
  (per server)



TTCN-3 domains: Automotive

- Car communication systems
  - Daimler, Volkswagen, SiemensVDO
  - edutainment bus system (test suite)
- Standardization groups:
  - AUTOSAR consortium
  - MOST cooperation
- Car-to-car communication

Telematics Applications in the Cockpit
- Audio (CD / Radio), Video
- Telephone, SMS
- Navigation
- Speech recognition
- User interface for body electronic



TTCN-3 domains: other domains

- Medicine
  - SiemensMED (image processing)
  - HL7 eHealth protocols (Interoperability)

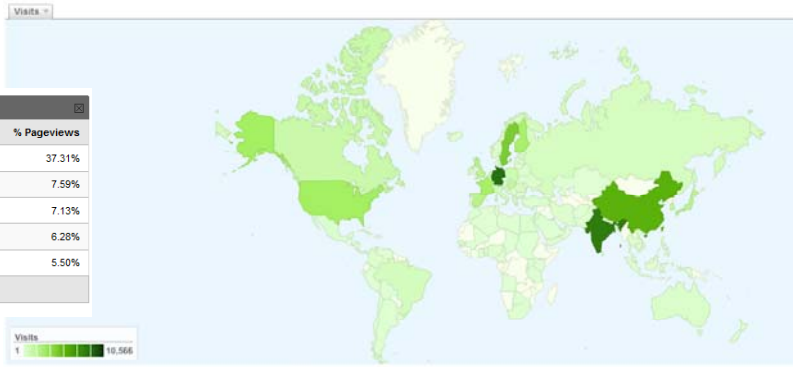
- Power transmission and distribution:
  - SiemensPTD (safe and reliable energy system)
- Financial data warehouse:
  - International bank (functional / regression testing)
- Avionics
  - European Space Agency
- Railways
  - Dutch railways (www.tt-medal.org)

Google analytics for www.ttcn-3.org

- Statistics since June 2008

Content Overview

Pages	Pageviews	% Pageviews
/	67,547	37.31%
/CoursesAndTutorials.htm	13,741	7.59%
/StandardSuite.htm	12,915	7.13%
/WhatisT3.htm	11,365	6.28%
/OpenSourceTools.htm	9,957	5.50%
view report		



65,194 visits came from 121 countries/territories

65,194 Visits
181,050 Pageviews
2.78 Pages/Visit
36.31% Bounce Rate
00:02:42 Avg. Time on Site
56.46% % New Visits

- Search Engines 32,669.00 (50.11%)
- Referring Sites 17,819.00 (27.33%)
- Direct Traffic 14,706.00 (22.56%)

A Selection of TTCN-3 Users

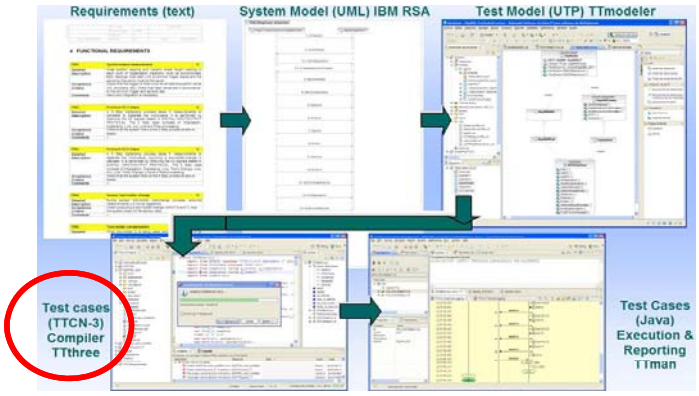
- **Telecommunication Vendors**
  - ERICSSON, MOTOROLA, SIEMENS, NOKIA, TELES GROUP, Telefonica, ciena, hotsip, Sonus Networks, ISKRATEL
- **Service Provider**
  - O2, vodafone, TELEKOM AUSTRIA, france telecom, Deutsche Telekom, VeriSign, sunrise
- **Test Device Vendors**
  - Agilent Technologies, Tektronix, NAVTEL, Alcatel·Lucent, AEROFLEX
- **Automotive and Avionics**
  - DaimlerChrysler, carmeq, esa, SIEMENS VDO
- **Hardware Vendors**
  - Texas Instruments, FUJITSU, Seagate, intel, SAMSUNG, CONEXANT, wistron
- **System Integrators**
  - NTT Data, HITACHI, ORACLE, utimaco
- **Standardisation**
  - AUTOSAR, oma, ETSI, 3GPP, WiMAX Forum

TTCN-3 in MBT

- TTCN-3 is used in several domains as binding link between modelling and execution
- Commercial tools like Qtronic (Conformiq) and TTmodeler (TestingTech) do generate TTCN-3 code for test execution
- lots of academic prototype tools
- selection of industrial case studies: see e.g. European D-MINT project www.d-mint.org

MBT example 1: Trimek/Datapixel production engineering case study

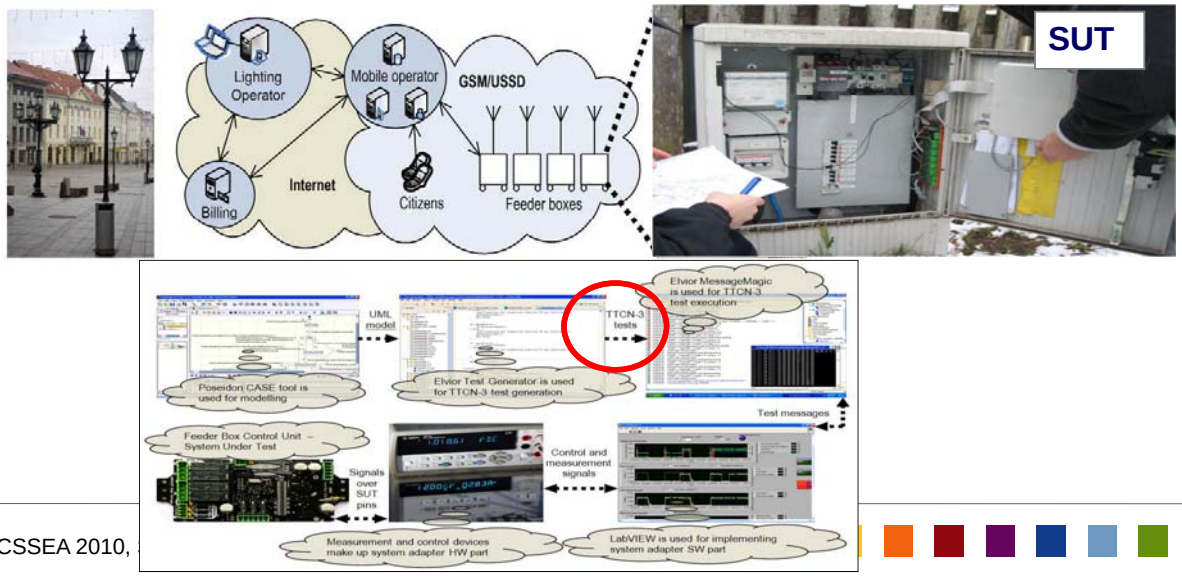
- **SUT: Coordinates Measuring Machines (CMM) control software (CDMS) for controlling a measuring system**
  - ➢ Focus: test case derivation from UML models
- Models in use: *UML class, sequence, state diagrams*



MBT example 2: City street lights case study

- **SUT: Eliko *street lighting control system* feeder box control unit (FBCU)**
- Models for the SUT: *UML state charts*, produced with tool Poseidon

➢ Elvior test generator derives *TTCN-3 test cases* from state charts



Status: further activities

- Establishment of a **TTCN-3 syllabus and certificate**
  - Cooperation of ETSI, GTB and iSQI
  - accredited TTCN-3 training providers
  - lots of certificates issued
- Improvement of **TTCN-3 tool interoperability**
  - Development of a TTCN-3 reference test suite
  - 1st TTCN-3 **Plugtest** event with commercial tools at ETSI (2009)
- TTCN-3 **user conferences**
  - Europe: since 2004 (F, D, E, S), next: 2011 (SLO)
  - Asia: 2007 (China), 2009 (India), 2010 (China)

Contents

- **TTCN**
  - Introduction
  - History
- **Status**
  - Concepts and tools
  - Applications:
    - Industrial domains, MBT
- **Future**
  - latest releases, Extension packages
  - Outlook: Embedded TTCN-3
- **Conclusion**

Maintenance of TTCN-3

- Standard is constantly maintained
  - Through Change Requests (CRs)
  - Extension proposals
  - Active contributions in the TTCN-3 community
    - TTCN-3 mailing list, TTCN-3 users conference
- ETSI STFs (Specialist Task Force)
- Change requests result in new editions of the standard
  - 2000: Edition 1
  - 2003: Edition 2
  - 2005: Edition 3
  - 2009: Edition 4.1
  - 2010: Edition 4.2

Status of TTCN-3 extension packages

- Published language extensions
  - ES 202 784 v1.1.1 (2009-07) Advanced Parameterization
  - ES 202 785 v1.1.1 (2010-01) Behaviour Types
- New language extensions
  - ES 202 781 v1.1.1 (2010-08) Configuration and Deployment Support
  - ES 202 782 v1.1.1 (2010-07) Real-Time and Performance Testing Support
- Reference Test Suite
  - TS 102 995 v1.1.1 Proforma for TTCN-3 Test Suite

    Completed in October. 2010, to be published soon

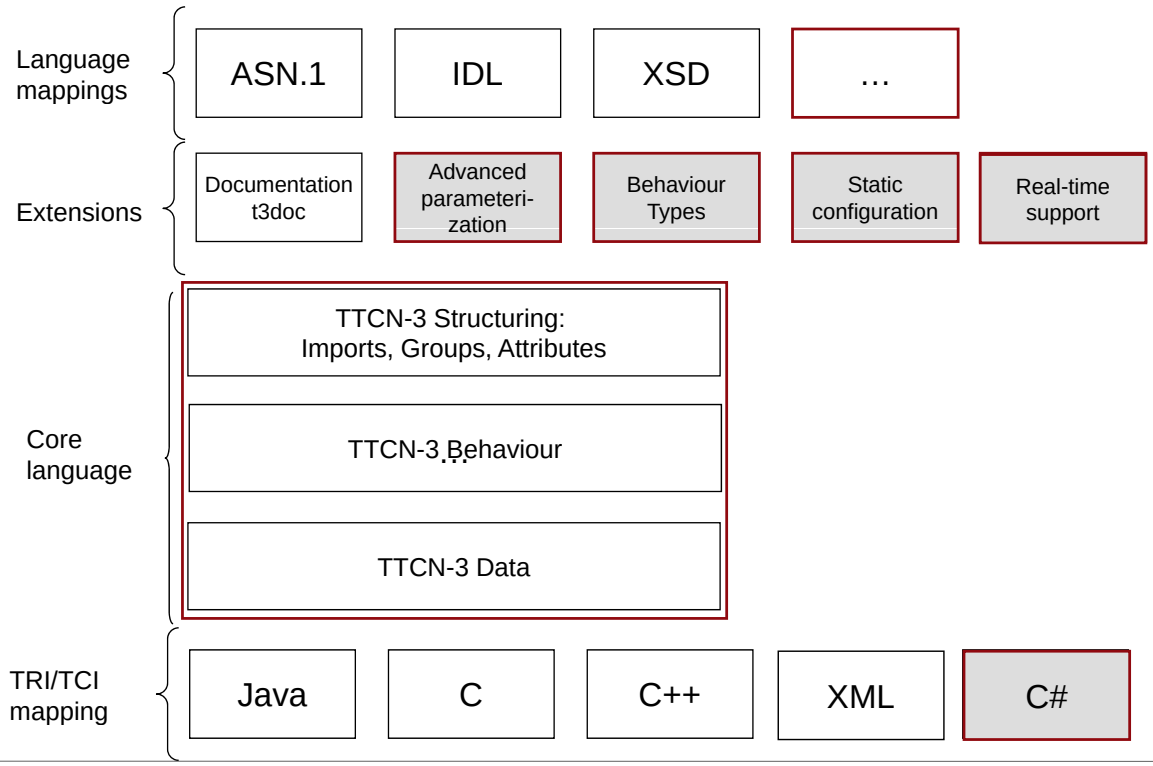
New in ES 201 873, v4.2.1

- TTCN-3 language
  - **Import of `import` statements**
  - **Special real values** (infinity, -infinity, not_a_number)
  - **Exclusive bounds for range subtyping**
  - **`ispresent()`, `str2hex()`, `testcase.stop`**
  - **Several smaller additions** (template arguments for superset, allows templates of all types, __BFILE__ macro)
- TRI/TCI
  - **C# mapping**
- Language mappings
  - **Unambiguous mapping of real/float types from XSD and ASN.1**
  - **Full support of mapping ASN.1 (2008)**

New in ES 201 873 (contd)

- Language mappings (contd)
  - **Parts are re-structured to be "package-style"**
    (e.g. TRI support for ASN.1->TTCN-3 is moved to Part-7)
- Source code documentation
  - **New tags added** (@priority, @requirement and @reference)
  - **Documentation block for local definitions of component types**
- Several clarifications, amendments and editorial corrections

TTCN-3 multipart standard overview: 2010 extensions



Extension packages #1: Advanced Parameterization

- **ETSI ES 202 784 V1.1.1** 2009-07
- Formal parameters of kind “type” for: type, template, and behaviour defintions
- Effects ES 201 873 parts: -1 (core), -4 (semantics), -6 (TCI), -7 (ASN1), -10 (T3DOC)
- Examples:

```
type record Data <in type p_PayloadType> {
        Header hdr,
        p_PayloadType payload}

function f_myfunction <in type p_myType >
                (in MyList<p_myType> p_list, in p_myType p_elem)  return p_myType
        {return p_list[0] + p_elem)}

f_myfunction <integer> ({1,2,3,4}, 5)
```


Extension packages #1: Advanced Parameterization

- Allows
  - Static value parameterization of types
  - Static type parameterization of types, templates, functions, altsteps and testcases
  - Default type is allowed for type parameters, like default values/templates for value parameters
- **Example use cases**
  - ***Message*** types, where e.g. the ***payload (field)*** can be a structured data as well as an octetstring/charstring etc.
  - Message templates, supporting both positive and negative testing in one parameterized definition
  - Port types with parameterized in/out/inout lists
  - ***Component*** types, where the ***type(s*)** ***of port(s)*** are parameterized
  - Functions that executes some kind of user-defined actions on data of different types (e.g. inserting extra fields into encoded messages for negative testing)
  - Functions with parameterized runs on clauses

Extension packages #2: Behaviour types

- **ETSI ES 202 785 V1.1.1** 2010-01
- More flexibility: variables may carry "behaviour" (functions/altsteps etc.)
- Effects ES 201 873 parts: -1 (core), -4 (semantics), -6 (TCI), -10 (T3DOC)
- Example:

```
type function MyFuncType ( in integer p1 );      // formal type definition (w/o body)
function f_myFunc1 ( in integer p1 ) {…};          // concrete behaviour
…
var MyFuncType v_func;
v_func := f_myFunc1;
apply (v_func(0));
```


Extension packages #2: Behaviour types

- Allows
  - Defining types of function, altstep and testcase (FAT) "prototypes"
  - Defining module parameters, constants, variables and templates of FAT types and storing references to "real" FATs in them
  - References to FATs can be stored, passed as parameters and sent to another component
  - Calling FATs via their references (including starting PTC behaviours)
  - "runs on self": specific runs on compatibility checking rules and cannot be sent to another component
- **Example use cases**
  - Efficient handling of state machines; the actual state is an integer number that – in case of an event - allows calling the function executing the action from an array of function references, instead of using e.g. select case to determine the actual state; also allows easy dynamic changing of the FSM
  - Using ***dynamically*** registrable callback ***functions in*** TTCN-3 SW ***libraries***
  - Flexible test configuration by e.g. passing PTC behaviour function references to MTC configuration functions

Extension packages #3: Configuration and Deployment Support

- **ETSI ES 202781 V1.1.1** 2010-08
- Introduces predefined static test components/configuration independent of a testcase (save time for testcase setup)
- Effects ES 201 873 parts: -1 (core), -4 (semantics), -5 (TRI), -6 (TCI)

- Example:

```
configuration f_StaticConfig() runs on MyMtcType system MySystemType {
        myComponent := MyPTCType.create static;
        map (myComponent :PCO, system:PCO1) static;...
}

testcase TC_test1 () execute on f_StaticConfig {...}
control {
        var configuration myStaticConfig;
        myStaticConfig := f_StaticConfig();        // configuration setup
        execute(TC_test1);
        myStaticConfig.kill;...
}
```


Extension packages #3: Configuration and Deployment Support

- Allows
  - Defining special “static” test configurations, creating and destroying them from the control part
  - Testcases (identified by `execute on`) may be called on a created static test configuration sequentially
  - Testcases may add “dynamic” PTCs and connections to the test configuration but shall not destroy any part of the static test configuration

- **Example use cases**
  - Semi-automatic (interactive) execution of test cases, where the SUT state may change due to loosing connections and/or missing keep-alive signals for a long time
  - ***Keep configuration setup***: Automatic test execution, where *loosing connections may raise unwanted alarm* signals and/or blockings/resets

Extension packages #4: Real time and performance testing support

- **ETSI ES 202782 V1.1.1** 2010-07

- Introduces **system time progress** (since start of a testcase), **delays and timestamps**
- Effects ES 201 873 parts: -1 (core), -4 (semantics), -5 (TRI)

- Examples:

```
module MyModule {...
    type port myPort message [realtime] {...};
    var float v_specified_send_time, v_sendTimePoint, v_myTime;
    ...
    myPort .receive(t)-> timestamp v_myTime;
    …
    wait (v_specified_send_time);
    myPort.send(m_out);
    v_sendTimePoint:= now;
} with {stepsize "0.001"}; // precision of a milisecond
```


Extension packages #4: Real time and performance testing support

- Allows
  - Identifying the required precision of the system time
  - `now`: getting the actual system time (from starting the test case)
  - `wait`: suspends the execution of a component until a given point in time (from starting the test case)
  - Identifying ports with `real-time` requirements; timestamp of entering messages in the TSI shall be stored and sent to the TE
  - Retrieving the timestamp of the receiving in the TTCN-3 code

- **Example use cases**
  - Set the required system preciseness by the writer of a TTCN-3 module (e.g. by a SW library module)
  - More precise time measurements, e.g. round-trip delays
  - More precise and ***convenient handling of timing-related actions***

Proforma for TTCN-3 Test Suite

- A framework for a TTCN-3 Reference Test Suite
- Standardized chapters includes
  - ATS structure
    - grouping of positive/negative tests
    - test case identifiers
  - ATS library proforma
  - General requirements
  - TTCN-3 naming conventions
  - Comments tags
  - Annex for
    - ICS proforma
    - TSS and TP
    - IXIT proforma
    - ATS in TTCN-3 core (text) format
- draft version TS 102 995 (2010-10)

Ongoing TTCN-3 Maintenance in 2011

- Planned TTCN-3 release(s,) planned: beginning of 2011
  - TTCN-3 v4.3.1
  - Extension packages and proforma for TTCN-3 test suite v1.2.1
  - Proforma for TTCN-3 Test Suite v1.2.1
- CRs can be submitted and followed in http://t-ort.etsi.org/view_all_bug_page.php or via http://www.ttcn3.org/ChangeRequest.htm
- CR priorities (for processing order)
  - high: bug fixes and clarifications
  - normal: minor additions
  - low: major additions

Outlook: TTCN-3 embedded

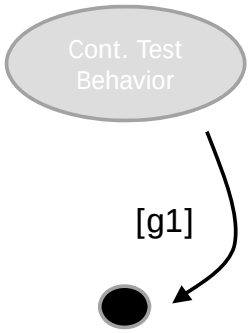
- Extensions for testing
  - real time systems (RT-TTCN-3)
  - continuous systems (Continuous TTCN-3)

- RT TTCN-3 Concepts
  - **clock** as a common basis for time measurement.
  - **timestamp** redirection for exact time measurement of message interaction.

- Continuous TTCN-3 Concepts
  - **sampled clock** as a common basis for discretization and stream definitions.
  - sampled **streams** that provide a data structure to define, access and manipulate discretized signal values and their history in time.
  - **hybrid automata** that provides a control flow structure to enable and control the simultaneous stimulation and evaluation of **stream ports**.

TTCN-3 embedded: Hybrid Behavior Specification

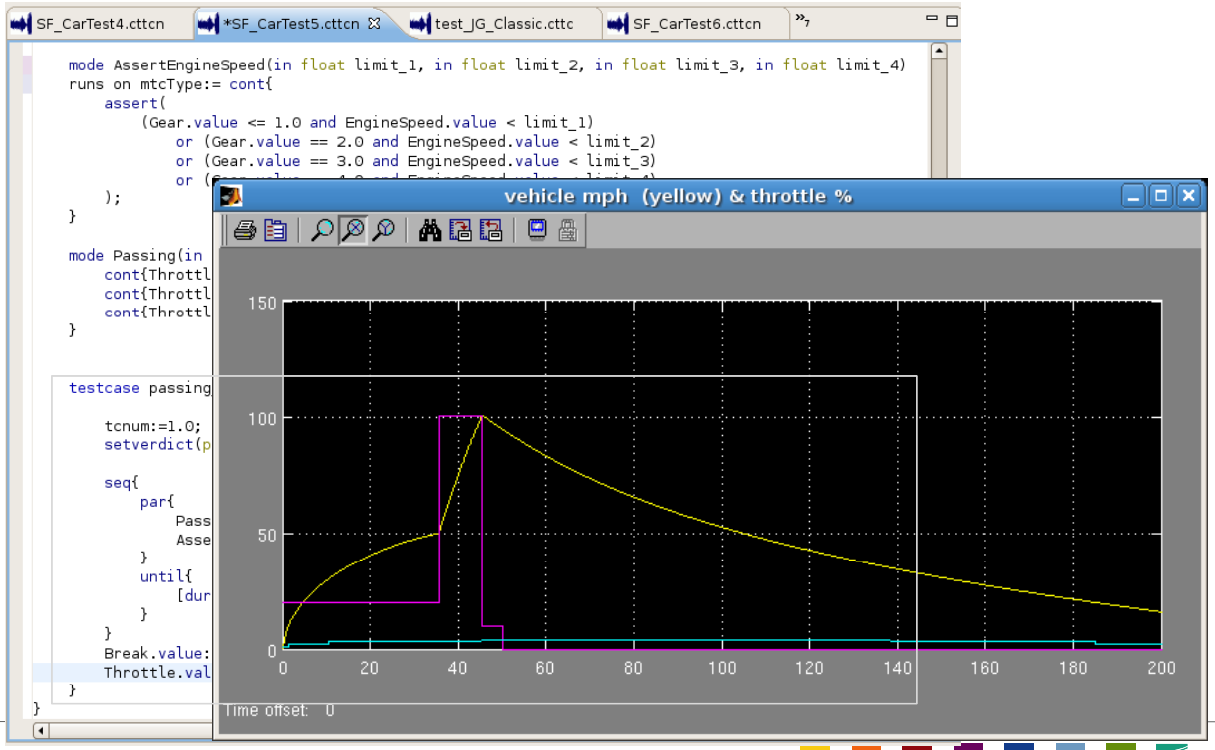
```
cont{
  var integer x :=0;
  onentry{Brake.value := 0.0}
  inv{ Brake.value <= 0.1 }
  Throttle.value := duration * 2.0;
  assert(EngineSpeed.value < 4000.0);
  onexit{Throttle.value := 0.0}
} until (duration >= 10.0)
```



- Modes represent the macro state of the test system
- **Repetitive execution** of the mode's content (micro states)
  - `onentry` and `onexit` are executed **once**
  - `inv`, `until`, `assertions` and `assignments` are executed **continuously**

- Parallel and sequential composition (using *par* and *seq*)

TTCN-3 embedded sample: vehicle passing



Contents

- **TTCN**
  - Introduction
  - History
- **Status**
  - Concepts and tools
  - Applications:
    - Industrial domains, MBT
- **Future**
  - latest releases, extension packages
  - Outlook: Embedded TTCN-3
- Conclusions

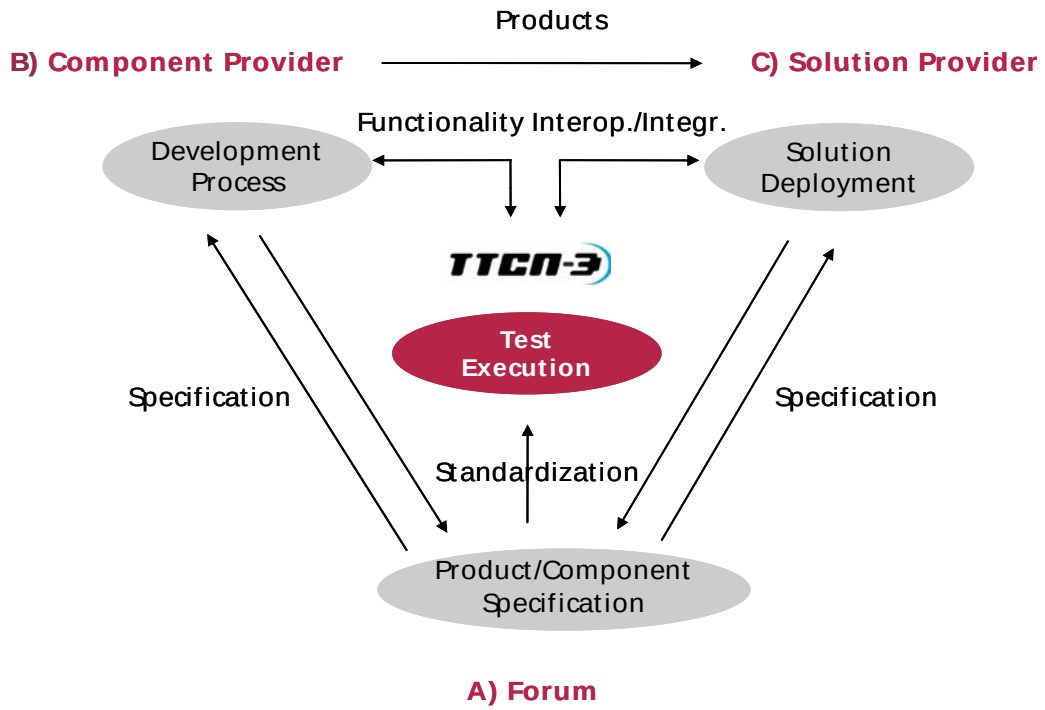
Major technical innovations / benefits

- TTCN-3 as standardized test language ***and*** implementation
- Easy (human readable) description of test scenarios
  - free from programming issues, (transparent framework for end-customers)
  - platform and tool vendor independence
  - different presentation formats
- Clear separation of testing issues:
  - test configuration, data and behaviour
  - SUT specific adaptation and codec

Conclusions

- 1980s – today, long history and value
- Wide range of applicability
  - different communication paradigms, testing types
  - domains: Telecom, Automotive, Medicine, Finance, Railways, Avionics…
  - used in research and industry
- Creates International community, expertise, commercial and open source tools
- TTCN-3 is established and in progress for the future

TTCN-3 as a testing middleware



TTCN-3 Sources

- Online information
  → www.ttcn-3.org
- TTCN-3 User Conference
  → 2011 in Bled, Slovenia
- TTCN-3 Standards, Papers, Book
  → http://www.ttcn.de/
- Quick Reference
  → http://www.blukaktus.com/
- Exercises and Tooling
  → research licenses

Thank you
for your attention!