WHITE PAPER

A Vision for Automated Testing

www.tt-medal.org

Preface

TT-Medal is a European research project on tests and testing methodologies for advanced languages. In TT-Medal key roles are assigned to international standards, the Testing and Test Control Notation (TTCN-3) by ETSI and ITU-T, the Unified Modelling Language (UML2.0) and its testing profile by the OMG.

Reading this white paper provides you a summary on the state of the art of test tools, an introduction to TT-Medal achievements, in particular insight into the industrial case studies that have been performed in the project. Three major directions for research are addressed: a common test tool infrastructure, approaches for the automatic generation of TTCN-3 tests, and the integrated system and test development. Our conclusion picks up these directions and attempts to describe a future picture for automated testing in the next years.

Contents

1. Introduction

The amount of software developed and the dependency of society on this software are still increasing rapidly. Furthermore, software provides the core functionality of most electronics products. These “software-intensive systems” have a large penetration into daily life and this penetration is expected to grow even more in the coming years. In addition to the growth in size and complexity, these software systems need to be of ever-increasing quality and need to be developed in ever-shorter time frames. To tackle these increasing quality requirements, testing is an important step in the development of a software-intensive system, as it checks if requirements to a system have not been met. To remain competitive in the development of these systems, industry needs to increase productivity by means of better testing practices.

TTCN-3 as a Solution

As successor to the widespread TTCN-2 [3] test notation TTCN-3 [2][15] is quickly becoming the most used test notation in the telecommunications and datacom area. Beyond this, TTCN-3 was designed as a general-purpose, technology-independent test language and developed for a wide range of computer applications in mind. A test notation was created, which has been adopted in several sectors and areas, such as functional and conformance testing, regression testing, interoperability, integration, load or stress testing. The development of the third version of TTCN was started at ETSI in 1998 by TC-MTS (Methods for Testing & Specification).

By now the basic concepts of language are both stable and mature and TTCN-3 is well established in the market. Not only due to the fact that it is the only international standardised test language for specification and implementation of complex test systems and solutions, but also because of the potent new features in the third version. These features include support for synchronous and asynchronous communication, data and signature templates with powerful matching mechanisms, a precise execution algorithm, and dynamic concurrent testing configurations. Several vendors are providing compilers for TTCN-3 and a rich set of tools and already standardised test suites is available on the market. As the standard describes a well defined syntax, interchange format and static as well as operational semantics, test suites written in TTCN-3 are portable between toolsets of different vendors. Using TTCN-3 to test an existing system is done by implementing a small set of standardised interfaces only once. Besides the test suites themselves, these adaptors are fully reusable, guaranteeing a high safety of investment in the future.

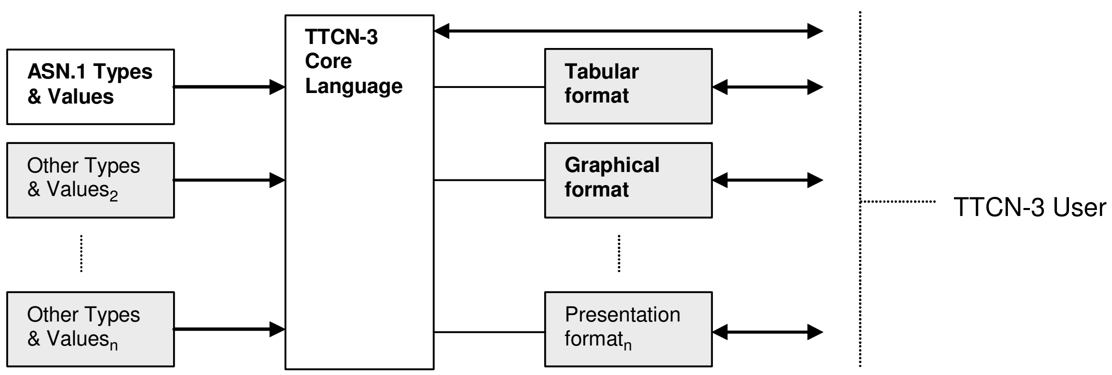

Figure 1: Elements of the TTCN-3 language definition

Reusability is one of the key concepts that drove the evolution of TTCN-3. Because testers in different domains are using a variety of fixed testing data types, TTCN-3 is able to import types and data from several external type systems, e.g. ASN.1, IDL or XML Schemata as indicated in Figure 1. Of course, TTCN-3 can not only be extended to accommodate further type systems, but the test logic itself can be displayed and modified using different presentation formats: Whereas the textual format is similar to a modern programming language, the graphical format displays the test logic like Message Sequence Charts and the tabular format resembles TTCN-2's tabular notation.

2. Test Tool Survey

The quality of test tools has certainly matured during the past number of years. Their scope, diversity and level of application have increased. But how far are we, what tools are used most and what is the level of satisfaction? This paragraph presents the results of a well-founded test tool survey that was carried out in over 400 organisations world-wide ranging from IT-companies with no more than 200 employees to large multi-nationals. The results can be used by the professional tester to understand the current situation, and to identify areas where tools could be used beneficially.

Implementations

Test tools may be classified according to the activities they support. The main support currently offered by test tools is intended for test management and test execution. In Table 1 data is shown regarding the implementation of the various tools - the percentage of companies actually using a certain tool type either off-the-shelf or self-made. No less than 88% of the companies that participated in the survey indicated that they had at least one test tool. The data distinguishes between the areas of technical applications (e.g. industry, embedded software and telecommunications) and information systems (e.g. banking, insurance and government).

Test tool implementation ratio[1]	*Technical Applications*	*Information Systems*	*Overall*
Test Management	38%	31%	35%
Defect management	74%	62%	69%
Configuration mgt.	71%	32%	54%
CM on testware	50%	21%	37%
Static analysis	38%	06%	24%
Test design	11%	20%	15%
Coverage tools	23%	03%	15%
Performance tools	40%	31%	36%
Record & playback	33%	43%	37%

Table 1: Test tools implementation ratio

Table 1 shows that in the area of technical applications substantially more tools are available and applied than in the area of information systems. There seems to be an uptake of test management tools. The implementation ratio of defect management tools is still low compared to the offerings available in the market. It seems one out of three organisations in the area of technical applications has a static analysis tool. The fact that this is much lower for information systems can be explained by the fact that few tools are available for fourth generation languages (4GL). The same reasoning probably also applies to the test coverage tooling. Only a few tools exist for test design but around 15% of the organisations have a supporting tool available. Although often perceived as the most

[1] Implementation ratio is defined as the number of organisations using a certain test tool divided by the total number surveyed

popular test tool, only one out of three organisations seems to have a record & playback tool. This means that most scripted testing is still done manually and a full regression is almost impossible.

User Satisfaction

Having a tool is one thing, but how satisfied are testers regarding their test tools? In the tool survey we inquired regarding the overall level of satisfaction and more specifically per type of tool. Overall 45% responded that they received many benefits, 52% stated that they received some benefits but expected more and only 3% didn't receive any benefits at all. Participants were also asked to rate their satisfaction level for individual tools on a scale of 1 to 10 (see Table 2) but no real high scores can be observed. As a whole this is just a sufficient score for satisfaction and a challenge for the upcoming years.

	Test tool satisfaction level
Test management and control	5.9
Test preparation	6.1
Test execution	6.6

Table 2: Test tool satisfaction level

There are organisations that have successfully chosen and purchased a test tool but many organisations have not achieved any benefit from their investment because their tools have ended up not being used, i.e. on the shelf or "shelfware". In general one can say the test tools that require a significant implementation process are mentioned here. During the survey the respondents were asked whether they had any shelfware, and if so what tools had become shelfware. No less than 22% of the organisations claimed to have some sort of shelfware. Although this is still a high number, the percentage is substantially lower that reported in earlier surveys [16]: 50% (1995), 45% (1997), 40% (1998) and 26% (2001).

Finally it was asked whether people would like more tools. This of course is especially an interesting question for tool suppliers, since it indicates what people are looking for. No less that 73% stated that they would like more tools. The three tools that were mentioned most are record & playback, test management and code coverage. For record & playback and code coverage there seems to be a logical explanation. They both still today have limitations regarding hardware and software platforms and programming language. Many organisations is the area of information systems are looking for coverage tools that support 4GL languages and not for tools that support "just" C, C++ and Java.

3. Achievements

3.1. TT-Medal Scope

Today, testing of software intensive systems is in many cases done in an ad-hoc fashion with languages not designed specifically for testing. The goal of the TT-Medal project is to deliver methodologies, tools and industrial experience to industry allowing the testing process of software intensive systems to be made more effective and efficient. Product quality is improved by exploiting the potential of international standards. The project will introduce the latest standardized testing technology, developed by ETSI, into European industry. This new technology based on the TTCN-3 testing language could provide exactly this competitive advantage to testing. There is also a relation to the upcoming UML 2.0 due to the standard mapping defined from U2TP, the UML 2.0 Testing Profile[2], to

[2] U2TP is an extension of UML 2.0, with features that are specific for test system design. Examples of such features are the notion of test cases, and the enumeration of verdicts.

TTCN-3. As the concepts developed by the TT-Medal project are based on international standards, industry is not required to develop expensive proprietary testing solutions.

To enhance the quality of products and achieve this quality within a given amount of time and resources, the methodologies and tools should address the whole life cycle of testing starting from the initial specifications and ending up with regression testing during the maintenance phase. Methodological aspects in terms of processes, infrastructure and tools need to be addressed for test generation, validation, development, and execution to increase efficiency and other needs of European industry. The methodology and tools developed are expected to fit to many domains not just telecommunications.

Clearly, the tight integration of appropriate tools and methodologies can boost the development on a more efficient level. Especially, a test execution environment (i.e. infrastructure) with open interfaces provides the foundation for automated test campaign with support for distributed test system components (test manager, logger etc.). Open interfaces are a prerequisite for flexible testing solutions that allow the exchange of test components.

The integration of proprietary or legacy test notations and tools needs to be addressed. Currently the effort needed for test development is so high that industry is forced to accept poor test coverage with cost of decreased product quality and the difficulty of defining high quality tests from informal system specifications. This requires automatic test case generation technologies, production of tests from models, and systematic testing methodologies integrated with testing tools. Finally, test reuse provides rapid implementation of test solutions for particular need with predefined tests. Also quality of test is increased by utilizing already developed and executed tests. Test quality can be also ensured by means of test validation methodologies that are integrated into automated tools.

3.2. Innovations

TT-Medal has concentrated in combining methodological research, standardization and new, yet forthcoming application domains by implementing new tool support for the testing methodologies. The work in new domains has lead new tool development and innovative applications in new business domains. Two innovations named ‘Bearer in a box’ and ‘OBSAI Module Tester’ have been taken as examples in the following.

Bearer in a box

Bearer in a box is a test adapter for a SIP laboratory tester trial case, which purpose is to reuse test cases developed originally for protocol conformance testing and functional testing of a network protocol layer. Traditional protocol conformance testing has been done on a workstation such that the protocol entity under test is executed in a workstation and the necessary interfaces are simulated. Laboratory testing will be done in a laboratory environment with real mobile hardware using Bearer in a box, which is combination of GPRS/UMTS radio bearer, simulated radio access network and simulated core network. Thus laboratory testing shall be understood as HW/SW integration testing. The tester will utilise both hand written TTCN-3 and test cases generated from interface models. The SIP Laboratory tester is done together by Nokia, NetHawk & Conformiq, and the Bearer in a box is NetHawk´s contribution to the test environment.

The Bearer in a box points to a new generation of test solutions that can be used to test the entire communications environment. It facilitates the combination of protocol testing and application testing with full featured network simulation.

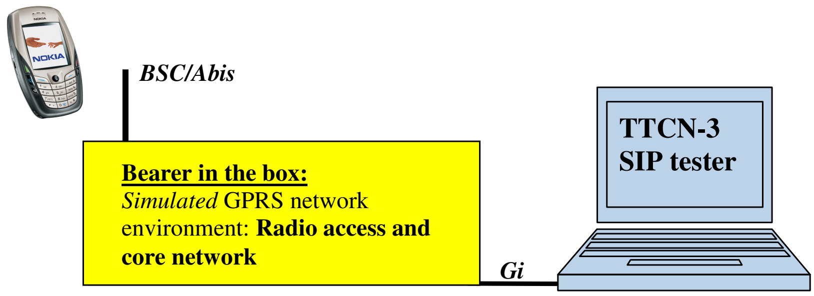

Figure 2: 'Bearer in a box' trial

OBSAI Module Tester

The Open Base Station Architecture Initiative (OBSAI) is an organization formed among leading base station vendors, module and component manufacturers to create a set of open specifications for a base station architecture. By defining a basic modular architecture and the detailed specifications for the internal interfaces between modules, OBSAI aims to create an open market for cellular base stations.

OBSAI has paid a lot of attention to test automation from the very beginning. Test automation requires instrumentation and the support of the instrumentation has been considered in the specifications. Many test tool vendors can provide automated test tools for OBSAI module vendors and for base station integrators to both verify compliance against OBSAI specifications and to validate their applications beyond OBSAI.

Technology developed in TT-Medal project offers a powerful test platform to test tool vendors to develop module testers for OBSAI developers. When fully implemented, test solutions will facilitate automated testing of base station modules to optimize performance, return on investment, and enhancement to business performance. For the test tool vendors this will open up a new, worldwide market.

3.3. Industrial Domains

Successful demonstrations of automated test solutions in various industrial domains are key issues for proving the concepts and convincing users. In the TT-Medal consortium partners from four application domains - telecommunication, railway, finance, and automotive - are involved to validate the technology in their everyday working environments.

Telecommunication case studies

Several case studies have been done in the telecommunications domain. Each of them has been aiming with a different focus. The first of the case studies was done to evaluate the TTCN-3 tools for conformance testing, i.e. the kind of testing which has been done already for more than ten years using the predecessor of TTCN-3. In the second of the case studies it has been investigated whether it is possible to execute the same test suite not only in an environment with an emulated network, but also in an environment where this has been replaced by the real network including the real physical medium. In the third of the case studies it has been investigated whether TTCN-3 can also be applied to new application areas, here the usage of TTCN-3 to test CORBA-based applications has been looked at.

a) Conformance Testing

The behaviour of telecommunication systems needs to be verified for conformance to the relevant standards. In this case study a network element to support location based services has been the SUT. The access to this system has been over TCP/IP connections, and the messages have been defined itself in ASN.1. This kind of message definitions is common within telecommunication systems like GSM. The test system in this case study has taken the role of one or several peer entities of the SUT.

In the case study it has been investigated whether the available TTCN-3 tools are capable of handling the definitions of the messages and test cases at all. Thereafter it has been investigated whether the test cases could be executed. For the execution it has both been necessary to make provide proper access to the transport medium and to properly encode and decode the messages. Finally, it has been investigated, which performance in terms of messages exchanged per second the test system provided.

b) Test Execution Environment

The implementation of a single protocol entity of a telecommunication system is often tested in a workstation environment. Here the implementation is executed on a workstation instead of the target platform, and also the message exchange with the SUT is done by other means than by the target network. I.e, the test system for a SIP implementation was connected by TCP/IP over cable instead of the targeted GPRS or UMTS network.

As already introduced above as 'Bearer in the box', it has been investigated whether an existing test system from a workstation environment can be used also for testing in a real environment without modifying the test cases. To do so, the message exchange over the cable was redirected to a simulation of a GPRS network, including an indoor base station. The test cases have been executed then against the SUT on the target platform.

c) Testing of CORBA Applications

TTCN-3 promises to be applicable for other areas than systems based on message exchange. One such areas are applications which use CORBA as middleware for communication[3]. Usually such applications have interfaces which are defined precisely in IDL. As CORBA is a generic middleware and IDL is also a generic way of defining interfaces, it should be possible to provide a generic means to test such application using TTCN-3. This should be made easier as TTCN-3 provides also procedure-based communication and there is a standardized mapping from IDL to TTCN-3.

In this case study it has been investigated to which extent the IDL to TTCN-3 mapping can be used for real life interface definitions. Once the mapping had matured it has been investigated how a TTCN-3 test system and a CORBA-based application could be connected in a generic way. The tools have been applied to a real life SUT.

Railway case study

The railway case study investigates in testing interlocking subsystem software of railway control systems that consists of three layers: infrastructure, logistic, and interlocking. The infrastructure represents a railway yard that basically consists of a collection of linked railway tracks supplied with such features as signals, points and level crossings. Figure 3 provides a schematic view of the road map complexity of the railway yard at Hoorn-Kersenboogerd in the Netherlands. In this figure, objects denote tracks, signals, points, and a level crossing.

---

[3] CORBA and IDL are systems communication standards by the OMG: www.omg.org/gettingstarted/corbafaq.htm

The interlocking is a layer of railway control systems that guarantees safety, meaning that no accidents can happen. It allows executing commands given by a user only if they are safe; unsafe commands are rejected. The interlocking layer also reacts in dangerous situations that can lead to derailments and collisions. Since the interlocking is the most safety-critical layer of the railway control system, testing this layer is a key issue.

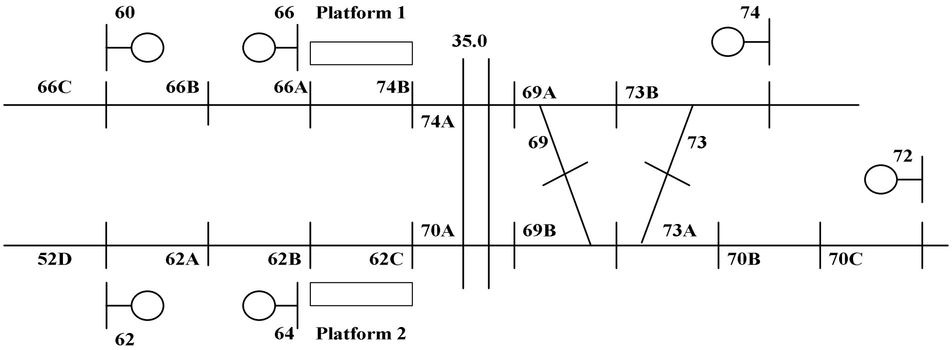

Figure 3: Railway yard at Hoorn-Kersenboogerd

The case study considers interlocking systems based on Vital Processor Interlocking (VPI) that is used nowadays in Australia, some Asian countries, Italy, the Netherlands, Spain and the USA [13]. Using general safety requirements from CENELEC [6][7], in TT-Medal a framework for testing VPI software with TTCN-3 has been developed by CWI. It covers implementing a TTCN-3 test system, developing a test suite, executing the test suite for the interlocking system of the railway station Hoorn-Kersenboogerd and finally interpreting results of testing. This work has been done in cooperation with engineers of ProRail who take care of capacity, reliability and safety on Dutch railways.

To test VPI software more efficiently, a concept of testing VPI software in simulated time has been proposed [8]. It shows that not only real time and scaled time testing is possible with TTCN-3, but also testing with simulated time where the system clock is a discrete logical clock.

Already with a small test suite, it was possible to discover a failure in VPI software. The failure had also been found by the specialists from ProRail and corrected in the currently running VPI software. The failure was caused by a misplaced use of 'OR' operation in the VPI code. The test suite can be reused in case VPI software on Hoorn-Kersenboogerd station is updated and should be tested again.

Finance case study

For the finance domain case study by LogicaCMG, administration and decision support tool functionalities are under test. The SUT that was chosen is GLOBUS, a widely used banking application that is representative for this domain. It offers administration of core banking processes, decision support in core banking processes, a user-friendly (GUI) interface for user-machine interaction, and it operates on the standard platform for end-user client workstations (MS Windows).

Showing that GLOBUS can be tested using TTCN-3 is an achievement. It has been proven that it is feasible that TTCN-3 can be executed on a Windows GUI; that a TTCN-3 test tool [18] for the telecommunications domain can be adapted to fit the financial domain; that separation of test control and SUT control can be achieved; and that existing SUT adapter code from test solutions other than TTCN-3 (here: TestFrame) can be fitted into the TTCN-3 architecture.

Showing that TFL, the test specification notation of TestFrame [10], can be mapped to TTCN-3 is an achievement. This mapping allows for a higher level of abstraction for test analysis and design, for enlargement of the user domain of TTCN-3 (accessible for non-technical users), and for the introduction of TTCN-3 in acceptance testing of administrative systems.

Case study findings are that logging results need to be more accessible (search, zooming in/out for level of detail/overview); that logging processing time appears to be long compared to test execution time (and that there may be a risk of memory overflow); that splitting SUT control from test control requires three layers: the 'classic' TTCN-3 SUT adapter that is the interface to the TTCN-3 environment, the SUT handling code that provides the communication with the SUT, and a TTCN-3 SUT facade that is the interface between the other two.

Automotive case study

The automotive case study by DaimlerCrysler addresses integrated in-car entertainment and infotainment components such as the Head Unit (central user interface for the driver), Navigation System, Radio, Audio amplifier, CD Changer etc. Most of the system features are realised by the simultaneous interplay of these components (e.g. on receiving a telephone call, the CD player is paused and the audio amplifier allocates audio channels to allow for hands free speech using microphones and speakers built into the dashboard).

The most cost effective way to continue innovating in all these areas is to allow the devices to be developed independently by different suppliers and then be networked together using standard hardware and software interfaces. Challenges for testing arise due to the limited testing interfaces available at the integration testing phase. The behaviour of the delivered components can only be observed via their interfaces to the rest of the system. The various components of the system are connected using several networking standards of which MOST[4] and CAN[5] are currently the most prominent.

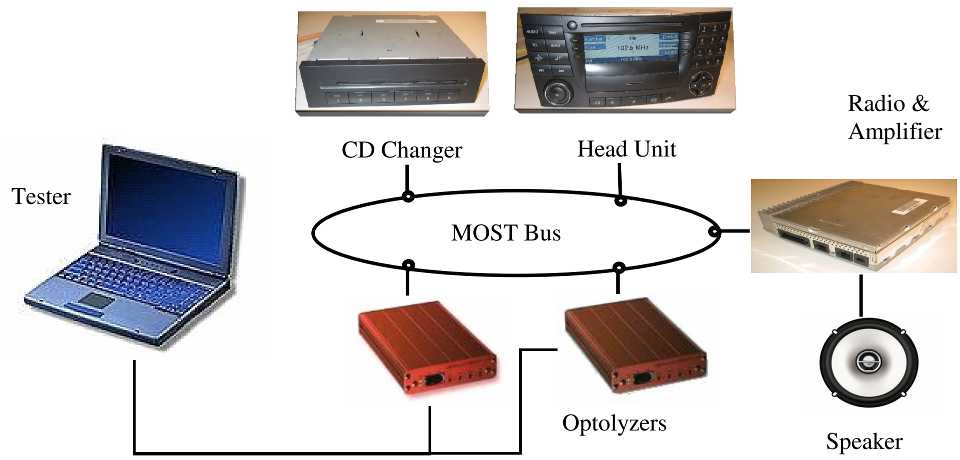

Figure 4: Automotive case study

[4] MOST Cooperation, Media Oriented System Transport, www.mostnet.org
[5] CAN, Controller Area Network, www.can-cia.org/can

Figure 4 shows the architecture of the automotive case study. A test system connected to the MOST ring executes test cases, i.e. it sends messages and observes the behaviour of the components. For accessing and observing the MOST ring two special devices, called Optolyzers[6], are used.

Test cases, specified in TTCN-3 and executed on this configuration, are simulating user actions on the Head Unit as for example the switching between the CD player and the radio or between several CDs (or several titles on one CD). First results in applying TTCN-3 for testing automotive applications based on MOST are very promising [9]. Furthermore it has been investigated that data type definitions for the test system could be derived in a generic approach automatically from the MOST functional catalogue that is based on XML schemata.

4. Directions

4.1. Test Infrastructure

The industrial application of the TTCN-3 technology requires a tool chain that supports customers during all phases of the testing process. In TT-Medal the common tool infrastructure and platform is of importance for future tool extensions and developments.

The test platform illustrated in Figure 5 offers components dedicated to the synthesis, validation and analysis of tests. Tests are developed along different types (such as functional, interoperability, performance, or load tests) and purposes (i.e. objectives) for which they are designed. The TT-Medal platform basically supports tests that are specified in TTCN-3 or U2TP. The results of the test development are abstract test suites in TTCN-3, which are compiled into executable code and executed by the test infrastructure.

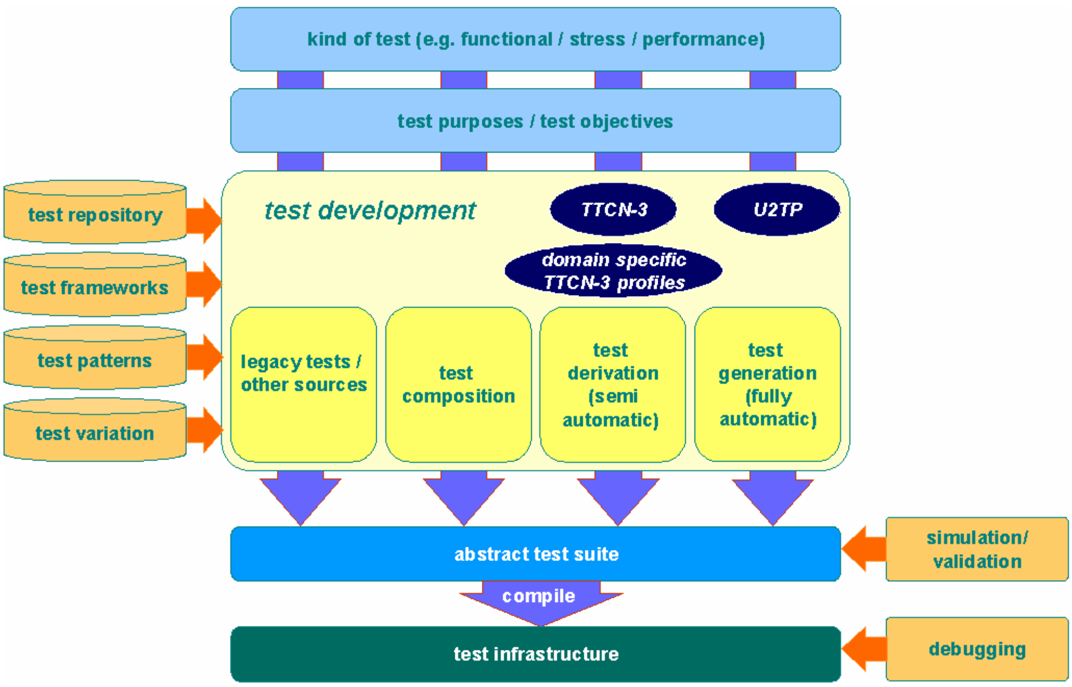

Figure 5: TT-Medal test platform

[6] Optolyzer for MOST: www.oasis.de/eng

The test infrastructure[7] provides means to execute TTCN-3 tests including test setup and test result reporting. Abstract test suites (taken from test development) are compiled with a TTCN-3 compiler (e.g. TTthree [18]). During test set-up all components needed for the test execution are (locally or remotely) deployed. These components include the executable test code, the test engine and adapters (often taken from adapter repositories).

The results of test execution via the test manager are test logs, which are the basis for determining the final test results. The test logs can be visualized using various presentation formats. They may be stored in a test result repository to be considered during the development of new test scenarios.

Mapping from proprietary languages

As already introduced in the previous chapter in TT-Medal particular interest has been given also to a mapping to TTCN-3 from TFL [10]. This mapping has been implemented in translator software that generates TTCN-3 core notation.

This version of the mapping takes only TFL for stand-alone set-ups, and its direction is one-way from TFL to TTCN-3. Future mapping versions can be expanded to include master/slave set-ups, as well as data typing. The direction to TFL can be investigated further to see how e.g. parallel test components and default behaviour can be expressed best in TFL, or if TFL should be expanded. The direction to TFL would make TFL a complete representation format in the TTCN-3 language architecture as illustrated in the introduction.

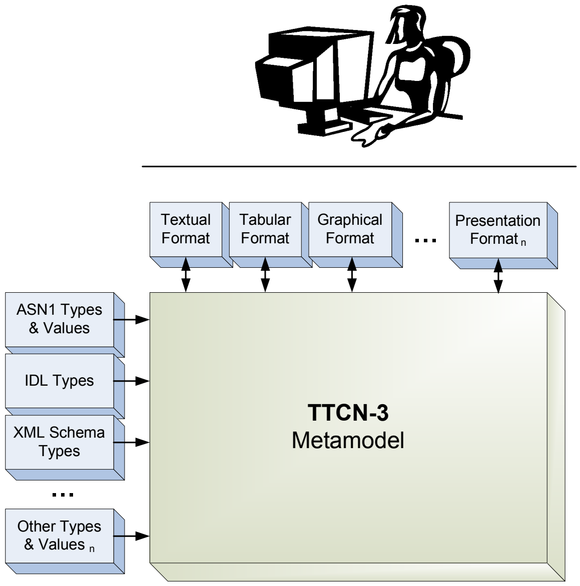

Figure 6: TTCN-3 metamodel

As TTCN-3 test suites can be internally represented in a Metamodel structure[8], new presentation formats can be developed with ease. This is planned for U2TP and TFL.

All in all, the results up to now give positive expectations for the feasibility of mapping other proprietary or legacy languages to TTCN-3, such as the scripting languages of several commercially available tool suites.

[7] In TT-Medal the terms "Test Platform" and "Test Infrastructure" have been used to avoid conflicts with the term "Test System" defined by ETSI. "Test Infrastructure", nevertheless, encapsulates the standardized concepts of TRI and TCI.
[8] Note: The TTCN-3 metamodel is under discussion. Currently the exchange between different presentation formats is defined via the TTCN-3 core notation.

4.2. Test Generation

It is well known in the telecom and related industries that developing test cases is costly and error-prone. According to a rule of thumb, one fifth of all test cases developed are incorrect before execution. In the general software industry there has been a steady trend towards higher and higher levels of abstraction. There is a need to raise the abstraction level also in the testing domain. TTCN-3 brings in the general programming language constructs, making it easier to implement, albeit in a restricted form, dynamic variation and parameterization.

The concept of test generation is analogous to compilation. In the same way as a compiler transforms (high-level language) source programs to (lower-level language) compiled code (e.g. C to assembler, Java to JVM bytecode, EMF metamodels into Java), a test generator creates concrete, lower-level test cases from more abstract test specifications or test models.

TTCN-3 sequence generation

TTCN-3 programs, created by human engineers, do a priori suffer from the same quality problems as computer programs do in general. TTCN-3 programs can be also quite large if there are many features to be tested. These facts imply that developing TTCN-3 test suites can be quite costly. Hence, there is room for improvements: a method that reduces test suite development costs could be successfully deployed to boost TTCN-3 testware development productivity, bringing benefits to the European industry.

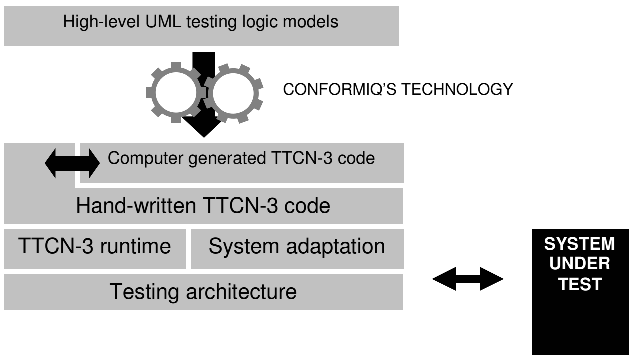

Figure 7: TTCN-3 generation by Conformiq's technology

A test generator for TTCN-3 can be extremely effective in generating test cases of a certain type, yet almost useless for generating test cases of a different nature. It seems therefore that test generation and manual testware development complement each other naturally in the context of TTCN-3. In this way, test generation can bring significant benefits into a TTCN-3 development cycle without destroying any of the original value proposals of the TTCN-3 language itself. The TTCN-3 generator developed by Conformiq Software within the TT-Medal project is a manifestation of this vision and blueprint for value creation.

CTG is a commercial tool for dynamically testing systems based on test models described with extended UML statechart diagrams. One contribution from Conformiq to the TT-Medal project has been an extension to this tool that generates TTCN-3 test cases from the same models that the tool can use for direct on-the-fly (i.e. online) testing, in which test derivation from a model and test execution are combined.

Based on the experience obtained in the TT-Medal project, it seems that the idea of merging computer-generated TTCN-3 with manually written code is the most fruitful way to benefit from test generation in the TTCN-3 context. With this approach, it is possible to introduce e.g. model-driven test generation into a testing process afterwards when commitment to the TTCN-3 platform has been already done. Also, the possibility to make a choice between manually written and automatically generated code reduces technology lock-in risks and adds project planning options.

Test data generation

Test data specification plays an important role during test execution in order to explore all aspects of the system which is to be tested. Although TTCN-3 supports good means for test data specification, it provides no methodology to generate test data automatically. This will be improved by using the classification tree method which allows the automatic data generation by categorizing test data in equivalence classes. The TT-Medal partner FOKUS has developed an Eclipse plug-in which enables the integration between TTCN-3 and the Classification Tree Editor (CTE) [16].

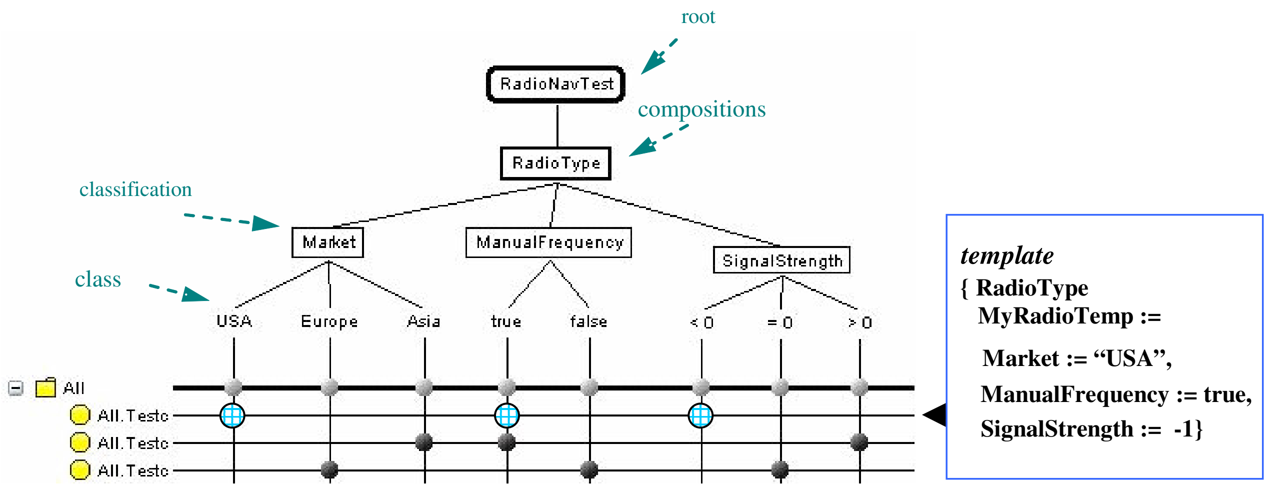

Figure 8: TTCN-3 test data generation using CTE

The idea of the CTE integration with TTCN-3 is to specify TTCN-3 test data as data partitions and visualizing these in a classification tree. Having the classification tree, the user may define test data manually or may let the CTE generate the test data automatically. Afterwards, the re-conversion from CTE test data into TTCN-3 test data can be performed, so that the newly generated CTE test data are integrated into the TTCN-3 test suite. The steps of conversions and re-conversion are realized for TTCN-3 data types, templates and testcases. Figure 6 shows a classification tree and its appropriate TTCN-3 templates.

CoDec generation

According to the standardized TTCN-3 test system architecture it is obvious that test execution requires system adapter and codec modules. Since system adapters can be addressed often in a generic implementation approach there is also a technical solution needed for the CoDec generation. Thus the TT-Medal partner TestingTech introduced a syntax about essential decoding information for TTCN-3 data types and developed an automatic codec generation tool for test engineers to get rid of boring implementation work.

4.3. Integrated System & Test Development

The widespread modelling standard UML 2.0 provides a uniform way to model systems. UML 2.0 consists of a number of different types of diagrams that are well-suited to describe various aspects of a system. The availability of models in a unified notation provides new opportunities. In particular, in TT-Medal it is investigated how UML models can be used in the testing phase of system development.

In the approach of Model based system development the models of a system are built, and gradually refined to an implementation. This means that the system under development is represented by several models, varying in the aspects described (e.g. structure, data, behaviour, time) and in the abstraction level, to be used in the different phases of system development (e.g. requirements, analysis, design).

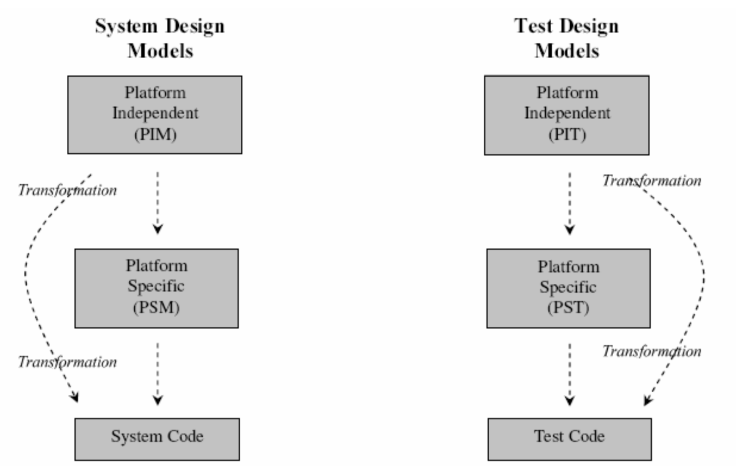

Figure 9: (a) System Design Models, **(b)** Test Design Models

As shown in Figure 9a, platform independent system design models (PIM) can be transformed into platform specific system design models (PSM). While PIMs focus on describing the pure functioning of a system independently from potential platforms that may be used to realize and execute the system, the relating PSMs contain a lot of information on the underlying platform. In another transformation step, system code may be derived from the PSM. Certainly, the completeness of the code depends on the completeness of the system design model.

Similarly in a Model-driven testing[9] approach (Figure 9b), the platform independent test design model (PIT) can be transformed either directly to test code or to a platform specific test design model (PST). The test design model can be transformed into executable test code from either PST or PIT. In this context in TT-Medal U2TP is considered with an implementation that transforms U2TP specifications into TTCN-3.

[9] Note: The terms “Model-based testing” and “Model-driven testing” have been distinguished in the sense that “Model based testing” starts the test derivation from the SUT model while “Model driven testing” implies the use of a test model.

Model-based testing: Methods and Techniques

In order to derive test cases (manually or automatically) in a systematic way, several well-established testing techniques (e.g. boundary value analysis) can be applied. As the TTCN-philosophy is based on black-box testing, these techniques should not be applied to the actual code, but rather to the UML models of the SUT. Depending on the kind of models used, this allows generation of tests from the requirements or the design of a system in a systematic way.

It has been investigated in TT-Medal which kinds of diagrams are used in industrial practice. Furthermore, the most important testing techniques were identified. Finally, these testing techniques are mapped to UML diagrams.

The conclusion is that existing UML models in industrial practice are often too abstract to derive test cases. In particular, the behavioural model of a system is useful for many testing techniques, such as transition coverage. The most detailed diagrams for modelling behaviour, the State Diagrams, are absent in most industrial software projects. However, some widely used diagrams can be useful as well, such as class diagrams and sequence diagrams. Of course, also using Object constraint language (OCL) annotations provide very helpful information for test case generation, in particular when using equivalence partitioning.

A precise mapping of testing techniques to specific UML diagrams has been done. This will probably change industrial practice. It has been made explicit what information is needed in order to derive tests. So depending on the test techniques to be applied, the system modellers could provide models of the required quality.

Algorithms and Tools

Various diagrams of UML 2.0 originate from mathematical logic and theoretical computer science. As an example, state diagrams are highly related to extensions of finite automata. In this area, a whole bunch of algorithms have been developed in order to automatically analyse such models (e.g. automatic test generation from state diagrams). In TT-Medal it was investigated how to apply such algorithms to UML models.

First, it is needed to map UML diagrams to the underlying mathematical models. This has been studied in previous European IST projects[10]. Second, more importantly, several limitations of the traditional algorithms had to be overcome. In particular, the presence of data and timing information had to be dealt with. This requires so-called symbolic test generation methods. Several solutions, based on data abstraction and constraint solving have been proposed, implemented, and tested in a number of case studies.

Here the solution is found in first generating test case templates by means of algorithms on finite automata. Next, the data parts must be filled in, which can be done by using constraint solvers or by other methods like e.g. classification trees. Another approach is to annotate UML 2.0 state diagrams with coverage requirements, constraints and preconditions.

5. Conclusions

For the future of automated testing as applied in TT-Medal we see an evolution that can be described by a series of significant technical achievements. During the project lifetime in TT-Medal the planned industrial case studies were demonstrated due to the development

[10] Such as Omega (aimed at verification; www-omegal.imag.fr) and Agedis (aimed at testing; www.adedis.de).

and application of new TTCN-3 methods and tools. However, the continuous use of the TTCN-3 technology will speed up its maturity and propagation. If the number of TTCN-3 test suites and corresponding generic system adapters and codec modules increases this brings customers and suppliers into a situation to benefit from reusing programming resources.

Methodological strategies will support migrations e.g. from functional to load and benchmark test applications that invites even more interested forums and standardization bodies to investigate. Since e.g. benchmark applications already require performance optimizations this work will enable real-time and statistical testing with TTCN-3. Other directions and challenges for the future are the elaboration of the TTCN-3 metamodel and the openness of the TTCN-3 notation towards transformations with other test notations (integration and/or representation).

Moreover, the work on Model-based testing is a key subject and needs to be continued. Ideally for reuse and test derivation purposes, test relevant information (requirements, purposes etc.) is included into the SUT model. This could be done by the SUT engineers due to specific system knowledge. They must be enabled to create and perform appropriate tests without involving external test experts. Test tools need to be part of the Integrated Development Environments that are already in use by the system developers.

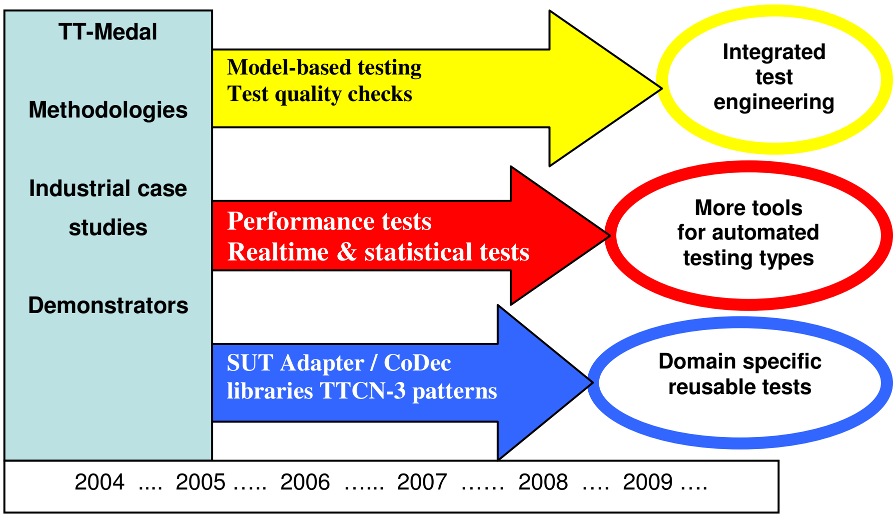

Figure 10: Roadmap for automated testing

A time projection for the future evolution of TTCN-3 in automated testing is very difficult. Figure 10 summarise and illustrates research directions and plans for the next years that have been mentioned in this white paper. It is obvious that the industrial case studies in TT-Medal are most important and constitute some fundamentals for the success of the technology. The continuous application in various domains is of course the only way to archive economic and comfortable conditions for future test automation.

Due to standardization work at ETSI today TTCN-3 is mature for industrial use. The language will increase publicity since the principles on automated testing are covered and could evolve according to future experiences.

6. Contact

This white paper has been written with the support of all TT-Medal partners: Conformiq Software, CWI, DaimlerChrysler, Fraunhofer FOKUS, Improve Quality Services, LogicaCMG, Nethawk, Nokia, ProRail, Testing Technologies, and VTT Electronics. In particular we like to thank the following colleagues for contributions and comments: A. Ahtiainen, S. Burton, Z.R. Dai, T. Deiss, J. Herrmann, E. Höfig, A. Huima, N. Ioustinova, M. Kärki, B.-J. d. Leuw, J. v.d. Pol, A. Rennoch, H. Rytilä, I. Schieferdecker, E. v. Veenendaal, L. Wolters, and J. Zander-Nowicka.

For questions, suggestions or if you want more information about the TT-Medal project you can contact:

Dr. Colin Willcock
Nokia
Itämerenkatu 11-13, FIN-00100 Helsinki
Phone: +491732984166
Email: colin.willcock@nokia.com

7. Glossary

ASN.1	Abstract Syntax Notation One
CAN	Controller Area Network
CENELEC	European Committee for Electrotechnical Standardization
CORBA	Common Object Request Broker Architecture
CTE	Classification Tree Editor
CTG	Conformiq Test Generator
EMF	Eclipse Modeling Framework
ETSI	European Telecommunications Standards Institute
GPRS	General Packet Radio Service
IDL	Interface Definition Language
ITEA	Information Technology for European Advancement
MOST	Media Oriented Systems Transport
OBSAI	Open Base Station Architecture Initiative
OMG	Object Management Group
SIP	Session Initiation Protocol
SUT	System under Test
TFL	TestFrame Language
TTCN-3	Testing and Test Control Language
TT-Medal	Test and Testing Methodologies for Advanced Languages
U2TP	UML 2.0 Testing Profile
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunication System
VPI	Vital Processor Interlocking
XML	Extensible Markup Language

8. Bibliography

[1] TT-Medal website: http://www.tt-medal.org

[2] ETSI ES 201 873-X TTCN-3 standards: http://www.etsi.org/ptcc/ptccttcn3.htm, also published as ITU-T Recommendation Z.140, Z.141, Z.142.

[3] ISO/IEC 9646-3:1998: Information technology - OSI - Conformance testing methodology and framework -- Part 3: The Tree and Tabular Combined Notation (TTCN), also published as ITU-T Recommendation X.292.

[4] OMG: UML 2.0, http://www.uml.org/

[5] OMG: UML 2.0 testing profile, http://www.fokus.fraunhofer.de/u2tp/

[6] CENELEC EN 50126 (1999): Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS).

[7] CENELEC EN 50128 (2001), EN 50129 (final draft 2002): Railway applications – Communications, signalling and processing systems.

[8] S. Blom et al.: Simulated Time for Testing Railway Interlockings with TTCN-3. 5th International Workshop on Formal Approaches to Testing Software, Springer 2005.

[9] S. Burton et al.: Automated testing of automotive telematics systems using TTCN-3. 3rd Workshop on Systems Testing and Validation, IRB Verlag 2004.

[10] H. Buwalda et al.: Integrated Test Design and Automation. Addison-Wesley 2002.

[11] M. Fewster, D. Graham: Automating Software Testing: Effective use of test execution tools. Addison-Wesley 1999.

[12] W. Fokkink et al.: Three Countries' Offensive towards Testing with Advanced Languages. ERCIM News No. 58, July 2004.

[13] U. Marscheck: Elektronische Stellwerke – Int. Überblick. SIGNAL+DRAHT, 89, 1997.

[14] E. v. Veenendaal: The Testing Practitioner, UTN Publishing 2002.

[15] C. Willcock et al.: An Introduction to TTCN-3. Wiley 2005.

[16] Classification Tree Editor: http://www.systematic-testing.com

[17] Conformiq Test Generator: http://www.conformiq.com

[18] Testing Technologies TTCN-3 test tools: http://www.testingtech.de