

# Automated testing of automotive telematics systems using TTCN-3

Simon Burton  
DaimlerChrysler AG,  
Research and Technology  
Alt Moabit 96a, 10599 Berlin  
Simon.Burton@DaimlerChrysler.com

André Baresel  
DaimlerChrysler AG,  
Research and Technology  
Alt Moabit 96a, 10599 Berlin  
Andre.Baresel@DaimlerChrysler.com

Ina Schieferdecker  
Fraunhofer FOKUS, TIP  
Kaiserin-Augusta-Allee 31, 10589  
Berlin  
schieferdecker@fokus.fraunhofer.de

## Abstract

Traditionally the Testing and Test Control Notation TTCN has been used and accepted widely in the telecom industry. The new version TTCN-3 is intended to be applicable to a broader scope of communication and software systems. This paper describes the initial results of a research project with the aim of evaluating the use of TTCN-3 within the automotive telematics domain. The paper addresses the use of TTCN-3 to specify the reusable test cases for the telematics domain, the development of a TTCN-3 test platform for telematics systems and future work required before the requirements of the industrial testing processes within the automotive industry can be met.

## 1 Introduction

The telematics features of modern passenger vehicles are becoming increasingly more complex and can only be realised by the correct interaction between a number of devices. In addition, the shortening development cycles and increased cost of system failures require that testing processes be continually improved in order that the required quality can be delivered without exceeding the tight development budgets.

The objective of the TT-Medal project [1] (running under the European Union ITEA Research Programme) is to develop the methodologies, tools and industrial experience to enable European industry to test more effectively and more efficiently. The project will introduce the latest testing technology, developed by ETSI, into European industry. This new technology based on the TTCN-3 testing language [2] will also provide common methods and tools to enable reuse of test artefacts between different phases of a product's life cycle. The methodology and tools developed by TT-Medal are expected to fit to many domains outside of the telecommunications industry, where the technologies originated. The goal of

DaimlerChrysler in the TT-Medal project is to evaluate the applicability of TTCN-3 technologies to the development of automotive telematics systems. The technical objectives of DaimlerChrysler within TT-Medal can be summarised as follows:

- Develop testing processes that contribute to a rigorous evaluation of the product quality – i.e. that can estimate as accurately as possible how many errors of which type remain, what type of errors are definitely not in the system, etc.
- Systematically reuse test specifications across test integration phases, testing types (e.g. reuse of functional test cases in robustness testing) and projects.
- Automate testing to reduce the time to find critical errors in the system.
- Make use of a standard conform and scalable test platform for executing tests that can be re-used across all test integration phases and testing types and development projects. The use of standardised test description languages shall enable test to be exchanged between OEMs and suppliers without mandating the use of particular tools.

This paper describes the initial experiences of DaimlerChrysler in applying TTCN-3 as a means of achieving the above objectives. The paper is structured as follows: Section 2 describes the systems under test. The motivation for applying TTCN-3 to testing these systems is then given in Section 3. Section 4 describes how TTCN-3 can be used to specify the tests for telematics systems. Section 5 summarises the results of the work so far and describes the key technologies currently in use in the TTCN-3-based testing environment. The paper concludes in Section 6 with a review of the future work.

## **2 The systems under test**

The telematics systems under test are 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 telematics 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 (Media Oriented System Transport) [3] and CAN (Control Area Network) [4] are currently the most prominent.

MOST is a multimedia fibre-optic network optimised for automotive applications. Its design allows it to provide a low-overhead and low-cost interface for the

simplest of devices, such as microphones and speakers. At the same time, more intelligent devices can automatically determine the features and functions provided by all other devices on the network and establish sophisticated functionality. The MOST specification provides basic data types, operation types and a number of function block definitions (logical application entities). Function blocks and their associated functions are described in the standardised MOST function catalogue which contains an exact definition of the messages used to communicate between the function blocks.

CAN communication is based on an electrical transmission mechanism – as opposed to MOST which is optical-based. CAN implements a broadcast communication mechanism which is achieved by using a message oriented transmission protocol. CAN devices are typically used to control vehicle functions, such as Climate Control, rear view mirror settings, anti-blocking braking systems and traction control. The CAN protocol is an international standard defined in the ISO 11898.

The development and integration of telematics system components is accompanied by a number of test phases. Initially each component is tested alone during a component acceptance test that verifies that the component specification is met (e.g. with respect to the physical properties of the interfaces) and that the basic functionality is available. However, much of the functionality can only be tested when the device is integrated with other devices in the network. This type of testing is performed during the integration test whose goal it is to verify applications (e.g. telephony, navigation etc.) that are distributed across a number of devices. During the integration test, the communication between the various devices is checked for conformance to the system specification. The focus of system tests is the interoperability of the various features within the telematics system and the interaction between the telematics system and its environment.

### **3 The benefits of TTCN-3 as a standardised test notation**

One of DaimlerChrysler's goals within the TT-Medal project is to develop a test platform, based on standardised technologies, which can be used for all test phases and test types mentioned above. A number of requirements have been identified for the test platform and can be summarised as follows:

- It shall be possible to graphically design/represent the behavioural aspects of test cases.
- The test cases should be written in a standardised test specification language to ensure that they can be exchanged between various stakeholders independent of the testing tools used.
- The test platform shall support integration with other CASE (Computer Aided Software Engineering) tools. For example, to support requirements

tracking using the DOORS [6] tool or defect tracking with the TestDirector [7] tool.

- It shall be possible to simulate devices that are not yet integrated into the system in order that distributed functionality can be tested in as early an integrations phase as possible.
- The test platform shall allow for test cases to be executed in a fully automated manner, i.e. without user intervention during the execution of test suite.

The Testing and Test Control Notation TTCN is a standardised testing technology, which has been identified as having the potential to fulfil the above requirements. The newest version of TTCN, TTCN-3 is based on a flexible, abstract and expressive test specification notation, that also includes a graphical representation format, GFT (graphical format for TTCN-3) that is heavily based on the Message Sequence Chart (MCS-2000) [5] standard. This allows tests to be specified and documented at a similar level of abstract as the system specifications.

Besides the syntax of the test notation itself, TTCN-3 also specifies the operational semantics of the languages which enables a consistent behaviour of the test scripts independent of the tools platforms in which they are executed. This will enable OEMs such as DaimlerChrysler to exchange tests with their suppliers without the need for mandating the use of a particular tool. Furthermore, TTCN-3 standardises the interfaces to the system under test via the test runtime interface (TRI) [7] and to test management tools via the test control interface (TCI) [8]. This makes it possible for various tool vendors to offer TTCN-3 solutions independently of the technical details of the system under test. Only the interface to the system under test (the "system adapter") needs to be developed in order to apply TTCN-3 to a new domain. As a result, TTCN-3 has the potential to establish itself as a testing method for a number of types of systems and domains. TTCN-3 therefore has the potential to dramatically reduce the cost of testing in industrial contexts due to the reduced reliance on a plethora of proprietary tools and methods.

## **4 Specifying telematics tests using TTCN-3**

By specifying tests at the same level of abstraction as the system specifications, similar terminology can be used (e.g. the message names in the test can correspond directly to those in the specification) and the logic of the tests becomes more transparent. This facilitates the design of test generation algorithms as well as making the test descriptions more accessible to the system experts who need to review their correctness with respect to the original specification. An abstract specification of the tests also separates the specification of the test from their execution at the system interface. Changes in the system interface therefore do not require a change in the test specifications, but only the test interface. This section explores how the use of TTCN-3 can lead to an abstract specification the

tests for automotive telematics systems communicating via a MOST bus and how TTCN-3 can additionally facilitate re-use of test specifications.

#### 4.1 Specifying message structures and contents

Besides the definition of test configuration (elements) and behaviour, a significant part of TTCN-3 specifications address a declaration of communication operations and/or message structures and values. Since TTCN-3 is an abstract specification technique it allows the definition of application dependent (complex) format structures using basic data types (e.g. Boolean, Integer, String etc.) similar to modern programming languages. So-called "Template" definitions correspond to type definitions and are used to describe stimuli to be sent as well as valid or invalid data expectations. Templates are very important for the interpretation of test observations and verdict settings.

The messages that can be sent on a MOST bus have been standardised and are defined by the MOST-Database which uses XML as an exchange format. All MOST messages have the same basic structure, therefore it is possible to use a single TTCN-3 type to represent all MOST messages, and vary the contents using templates. Alternatively, a separate TTCN-3 type could be used for each MOST message type. As an example the following type definitions could be used to define the structure of a particular MOST message (for changing the current radio waveband via the AMFMTuner function block).

```
type record AMFMTuner_SetWaveband_Type {
  integer      FunctionBlock,
  integer      InstanceID,
  integer      Function,
  integer      OpType,
  Waveband_Type Waveband
}
```

A MOST specific codec would be required to encode these types into the correct sequence of bytes to be sent as a packet over the most bus. The system adapter also needs to add information to the data packet based on which function blocks are deployed to which devices and other details of the system architecture, e.g. the hardware addresses of the various devices. A template based on this type could be created where the waveband to be changed to is defined as a parameter to the template:

```
template AMFMTuner_SetWaveband_Type
  AMFMTuner_SetWaveband (Waveband_Type Waveband) := {
    FunctionBlock := 64,
    InstanceID := 1,
    Function := 512,
    OpType := 0,
    Waveband := Waveband
  }
```

This template could then be used to define a message to be sent within a test case as follows:

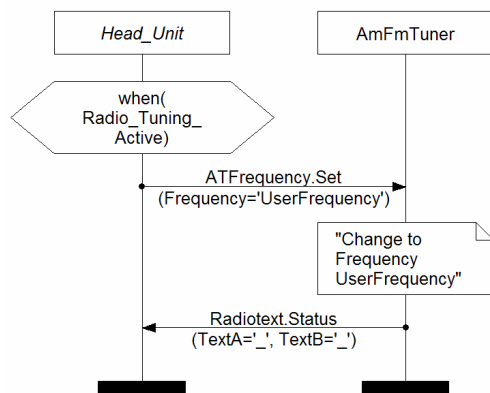
```
AmFmTuner.send(AMFMTuner_SetWaveband (FM));
```

The same message definitions based on the MOST/CAN catalogues will need to be used for all tests within a particular project and possibly across a number of different projects. Therefore it is desirable to reuse the definitions of the messages across a number of test suites. The modular structure of the TTCN-3 allows messages to be declared in a separate module to the test behaviour. Separate modules have been created that contains the message definitions for each MOST function block. When specifying the test behaviour, only those modules corresponding to the function blocks referenced by the test need then be imported.

Due to the nature of test descriptions and the broad scope of TTCN-3, also data types defined in other languages such as ASN.1, IDL and XML can be used in TTCN-3 test definitions in an easy way. We distinguish an explicit mapping to TTCN-3 and insertion of definitions into a test suite in contrast to an import and implicit usage of non-TTCN-3 definitions. In the later case it is obvious that the same language mapping rules as in the explicit variant are applied, but test developers do not wish to see the translated definitions within the test suite. Mappings for several types systems have already been defined by ETSI but we have to remark that XML-based structures (as used in MOST) need some pre-processing prior to a mapping as proposed in [14] due to the high level of flexibility that XML DTD/Schema definitions allow. In particular, the MOST function catalogue contains many elements not required when specifying tests.

## **4.2 Specifying test behaviour**

The dynamic behaviour of tests for automotive telematics systems consists of a sequence of either messages sent by the tester to a particular function block or messages between function blocks that are observed by the tester by monitoring the bus. The exact sequencing of the messages can depend on conditions that are decided at run-time or on the observed behaviour under test. A test case can therefore contain a number of paths that can lead to a pass verdict. In addition it may also be necessary to test behaviour that runs in parallel in the system under test. Due to its background in the testing of telecommunication systems, TTCN-3 appears to be well suited for specifying this type of behaviour and a preliminary examination of the TTCN-3 language indicates that all control structures are included that are necessary for specifying the logic of telematics tests. As an example, the following specification Message Sequence Chart could be modelled as the following TTCN-3 test case, where the tester simulates the head unit behaviour.



```

tescase ManualFrequency(float UserFrequency)
runs on MOSTTester {
  timer MyTimer:= 2000; // 2000 ms

  if( Radio_Tuning_Active){
    AmFmTuner.send( AMFmTuner_SetFrequency( UserFrequency));

    MyTimer.start;

    alt {
      [] AmFmTuner.receive (AMFmTuner_Radiotext_Status (?,?))
        setverdict(pass);
        stop;
      [] MyTimer.timeout
        setverdict(fail);
        stop;
    }
  }
}

```

**Figure 1:** Example MSC and TTCN-3 test case

In many cases, the exact values of some message parameters are omitted from the telematics specifications (see message "Radio\_Text.Status" above). When designing test cases, it is therefore often difficult to predict in advance which concrete values should be expected for certain received messages. TTCN-3 allows received messages to be under-specified, by using the wildcard ("?") symbol to denote parameters that may take any value. The use of the wildcard symbol will help to avoid over-specialisation of the test cases and therefore facilitate their re-use in different contexts. Where the messages to be *sent* by the tester contain undefined parameters in the specification, these must be given concrete values at test run-time. In order to facilitate re-use, these parameters are typically left undefined in the test case specification and set to various values at run-time. This can be achieved via the test case parameter or module parameter interfaces in TTCN-3. In the example above, the variable "UserFrequency" used to define the

parameter value in the message "ATFrequency.Set" is defined as a parameter to the test case.). The Classification Tree Method [9] is currently being evaluated as a means of systematically identifying suitable message parameters to be used in the tests.

## 5 Initial experiences with TTCN-3

An initial investigation of the use of TTCN-3 for testing telematics system has shown that it is well suited for specifying a broad range of tests. In addition, the standardised TTCN-3 architecture has allowed a telematics test platform to be developed which reuses a number of existing tool components. The use of a standard test notation not only increases the potential for exchanging test cases between various stakeholders in the test process, it also offers more options when deciding which test methodology to use for particular test problems. For example, several tools already exist for generating TTCN-3 test cases based on a number of specification types and test goals [10, 11, 12]. Each of these tools can be combined with any TTCN-3 compatible test execution environments.

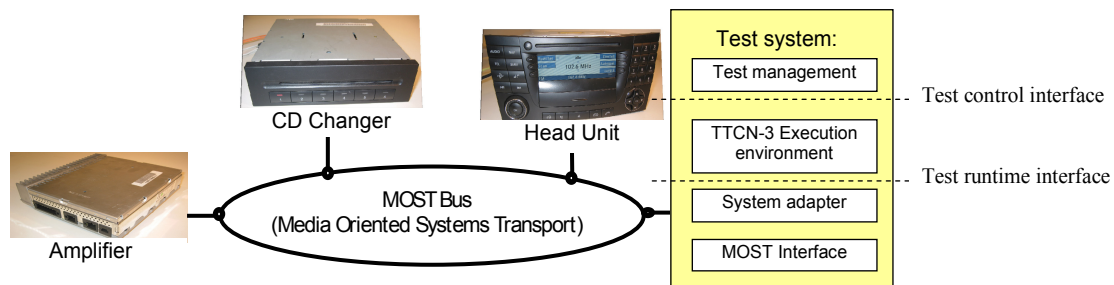
The application of TTCN-3 to the automotive telematics domain involved two main tasks. Firstly, a suitable TTCN-3 representation of the messages types and contents was chosen. This representation is used by the tester to develop the test suites as well as the system adapter as the basis for creating the concrete messages to be sent out over the bus (see Section 4.1). The decision as to which representation to use was based on a trade-off between the need for specifying the tests at the same (high) level of abstraction as the specification and the complexity required in the system adapter to encode and decode the messages. The more abstract the TTCN-3 representation is (i.e. message destinations are referenced by function block and not by hardware address) the more intelligent (and therefore less efficient and less generally applicable) the system adapter needs to be. The representation chosen allowed for an efficient encoding of the messages by making heavy use of templates to hide the details of the message contents, whilst abstracting from the concrete deployment of function blocks to devices in the system by ensuring that the system adapter fills in the appropriate source and destination hardware addresses.

The second main task was to develop a system adapter for telematics systems that conforms to the standardised TTCN-3 interfaces. This task turned out to be relatively simple. Existing hardware drivers used to monitor and produce traffic on the bus were packaged into a form that was compatible with the TTCN-3 TRI



interface. Messages to and from each function block are directed into separate TTCN-3 ports within the TTCN-3 execution environment, which allows the test behaviour to be described in terms of the communication between function blocks rather than the devices the function blocks are deployed to in the system. This further ensures reuse of the tests between different system configurations.

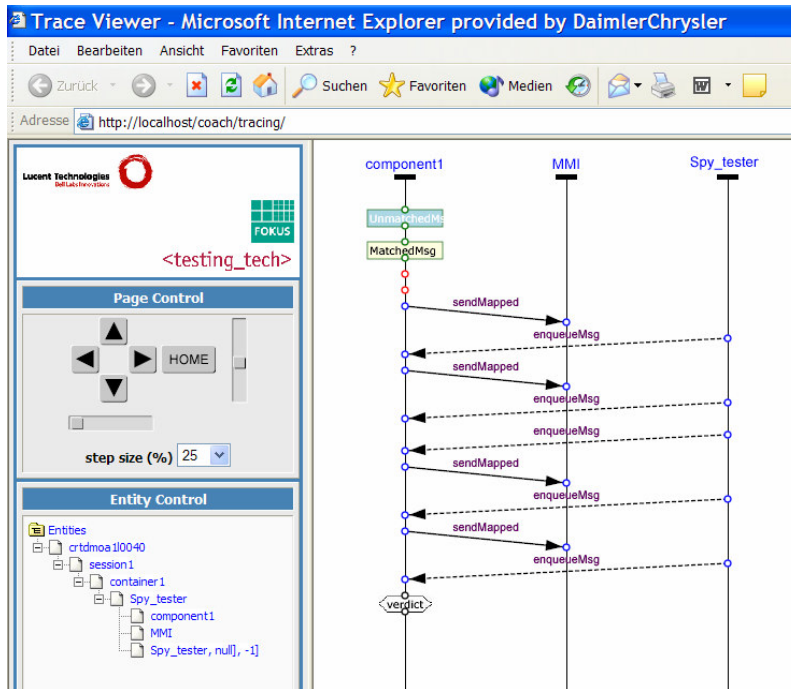
Figure 2 shows an example TTCN-3-based test environment for testing the integration of several telematics components which communicate via the MOST-Ring.



**Figure 2:** TTCN-3 based test architecture for telematics systems

The TTspec [9] tool from TestingTechnologies was used to graphically develop the test cases. The generated TTCN-3 code was compiled using the TTthree [8] Compiler and executed using the TTman tool also from TestingTechnologies [8]. The results of the test could be viewed as a graphical trace using the TraceViewer [13] tool (see Figure 3). As part of future work, the TTworkbench tool will be evaluated as a means of providing an integrated development environment in which tests can be developed, executed and evaluated.

At present, the graphical traces produced by the tool chain represent the communication between the test execution environment and the system under test (SUT). However, in order that the traces can be better compared with the specification, it is necessary to represent the monitored communication as message exchanges between function blocks within the system. In order to achieve this, the graphical tracing functionality needs to be extended to retrieve the source and destination function blocks from the observed messages (currently modelled as receives on the tester side). This information could then be used to describe the observed behaviour in terms of the interaction between the function blocks in the system. Obviously this goes beyond a simple representation of interactions between the SUT and the test system. Currently a distinction of SUT messages according to parameter values is possible via an introduction of PTC (parallel test components).



**Figure 3:** TTCN-3 based test architecture for telematics systems

## 6 Future work

A number of areas of future have been identified and will be addressed during the remainder of the TT-Medal project.

The effort required in manually designing and implementing tests based on system specifications of growing complexity and size is becoming increasingly impractical. More efficient approaches to test design will therefore be developed. Methods of generating test based on graphical specifications will be investigated, e.g using the Conformiq test generator [10]. In addition a more systematic and tool supported approach to defining the data aspects of the test behaviour will be developed based on an integration of the classification tree method [9] in to the TTCN-3 integrated development environment.

An additional means of increasing the efficiency of test design is to apply reuse techniques. These reuse techniques can be at the level of applying design patterns for TTCN-3 code or by applying reuse of entire test cases for different test goals. A methodology for applying test reuse will be developed and applied to the industrial case studies. In order to reuse test cases at various levels of system integration it will be necessary to be simulate those devices in the system not yet developed. This poses a challenge for the system adapter which must be able to mimic the network level behaviour of a number of devices simultaneously. In addition, techniques for

describing the behaviour need to developed that allow the same test behaviour to be used to simulate devices at early stages of system integration as that used to monitor the actual devices once they have been developed and integrated into the system.

The technologies described in this paper have so far been used to test the integration of a small number of telematics devices. More experimentation is needed to investigate how the performance of the test platform scales to tests the integration of a larger number of components or when performing the test of a single component where a large amount of bus traffic must be simulated by the tester in order to provide a realistic context for the test.

Based on our initial experiences of TTCN-3 in the telematics domain, we anticipate that neither generic nor specific extensions to the core language will be required to adequately describe the tests for the telematics case studies. However, for the test of real-time reactive systems, such as engine management control devices or intelligent braking systems some extensions to the language may be desirable. The test of real-time reactive systems is a significantly different paradigm to that which leads to the development of TTCN. Therefore it is not surprising that the requirements on a standardised test language should be different. These extensions could be realised as extensions to the language itself or alternatively, existing tools and approaches could be integrated via the TTCN-3 external function mechanism. The extensions that are required include, amongst others, the ability to describe test stimuli as continuous signal curves and the ability to define a threshold values on the output of the system under test and use this to trigger behaviour within the test case. E.g. wait for the engine revolutions value to reach 4500 before sending the signal to change gear.

## REFERENCES

- [1] TT-Medal: Tests and Testing Methodologies with Advanced Languages. <http://www.tt-medal.org/>
- [2] ETSI ES 201873, The Testing and Test Control Notation TTCN-3, Part 1 (Core Language), Part 2 (Tabular Format), Part 3 (Graphical Format), Part 4 (Operational Semantics), Part 5 (TTCN-3 Runtime Interface), Part 6 (TTCN-3 Control Interfaces), Sophia-Antipolis, France, October 2003.
- [3] MOST cooperation: <http://www.mostnet.de/downloads/Specifications/MOST>
- [4] CAN (Control Area Network): <http://www.can-cia.org/can/>

- [5] ITU-T Z.120: Message Sequence Charts, MSC-2000, Geneva, Switzerland, October 1999.
- [6] S. Schulz and T. Vassiliou-Gioles, Implementation of TTCN-3 Test Systems using the TRI, IFIP 14th International Conference on Testing of Communicating Systems - TestCom 2002, March 2002.
- [7] I. Schieferdecker, T. Vassilou-Gioles, Realizing Distributed TTCN-3 Test Systems with TCI, IFIP TC 6 / WG 6.1 15th International Conference on Testing of Communicating Systems - TestCom 2003, May 2003.
- [8] TestingTechnologies IST GmbH : TTCN-3 Toolset.  
<http://www.testingtech.de/products/>
- [9] DaimlerChrysler: Classification Tree Method, [http://www.systematic-testing.com/ctm\\_cte.htm](http://www.systematic-testing.com/ctm_cte.htm)
- [10] Conformiq: UML based test generation, <http://www.conformiq.com/>
- [11] TTspec: Graphical TTCN-3 Test Development Tool,  
<http://www.testingtech.de/products/ttspec.php>
- [12] MaTeLo: European IST project 2001-32402 on Markov Test Logic,  
<http://www.alitec.net/matelo/>
- [13] H. Batteram, W. Hellenthal, W. Romijn, A. Hoffmann, A. Rennoch, A. Vouffo: Implementation of an Open Source Toolset for CCM Components and Systems Testing Testcom 2004, Oxford (UK), Springer-Verlag, March 17-19, 2004.
- [14] D-M. Jeaca, G. Din, A. Rennoch: Importing XMLSchema datatypes into TTCN-3, Submitted to SV04 Workshop, Paris, December 2004.