

 <p>D-MINT Deployment of Model-Based Technologies to Industrial Testing</p>	Title: Integrated description of all tool chain components
	Version: 1.0 Date : 17/10/09 Pages : 20
	Author: Michal Janiszek (SQS)
	Reviewers: FOKUS, TESTING TECH, DAIMLER
	To: D-MINT Consortium
The D-MINT Consortium consists of: Nokia Siemens Networks, ABB, Åbo Akademi, Conformiq, Daimler, DATAPIXEL, ELIKO, Elvior, Fraunhofer IESE, Fraunhofer FOKUS, IDEKO, Innovalia Association, INSPIRE, iXtronics GmbH, Nethawk Oyj, PikeTec GmbH, SQS, SORALUCE, Tallinn University of Technology, Testing Technologies IST GmbH, TRIMEK, VTT Technical Research Centre of Finland, ETSI, TANDBERG, Simula Research	Printed on:
Status: <input type="checkbox"/> Draft <input type="checkbox"/> To be reviewed <input type="checkbox"/> Proposal <input checked="" type="checkbox"/> Final / Released	Confidentiality: <input checked="" type="checkbox"/> Public - Intended for public use <input type="checkbox"/> Restricted - Intended for D-MINT consortium only <input type="checkbox"/> Confidential - Intended for individual partner only
Deliverable ID:	D.3.4.v1.0
Title:	Integrated description of all tool chain components
Summary / Contents:	
<p>The report provides some statistics of tools and presents a summary of each tool related with D-Mint project. Solution summaries have been extracted from “integrated description of all tool chain components” draft document and the statistics has been calculated based on that information.</p>	



	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 2 of 20
		Version: 1.0
		Date : 24/08/2009
		Status : Final Confid : Public

TABLE OF CONTENTS

CHANGE LOG	3
APPLICABLE DOCUMENT LIST	3
1. Terminology	4
2. INTRODUCTION	4
3. Overview of tools	4
IESE.....	6
3.1.1 Tools by status.....	7
3.1.2 Tool by application	8
4. Tool descriptions	11
4.1 PIKETEC.....	11
4.1.1 TPT	11
4.1.2 TPT CAMEL-View-Plugin	12
4.2 iXtronics	13
4.2.1 CAMEL-View.....	13
4.2.2 CAMEL-View Testing TBX.....	14
4.3 Conformiq.....	14
4.3.1 Conformiq Qtronic.....	14
4.4 AAU.....	14
4.4.1 MATERA (Modelling for Automated Test derivation at AAU)	14
4.5 Nethawk	15
4.5.1 NetHawk EAST	15
4.6 Elvior	15
4.6.1 MessageMagic.....	15
4.6.2 MOTES	16
4.7 VTT	17
4.7.1 Root Cause Analyzer (RCA)	17
4.8 TestingTech.....	17
4.8.1 TTworkbench	17
4.8.2 TTmodeler for TTworkbench	18
4.9 FOKUS.....	19
4.9.1 MDTester	19
4.9.2 ADTgenerator	19
4.9.3 qTTCN-3.....	19
4.10 IESE.....	20
4.10.1 JUMBL.....	20
4.10.2 jSEQ	20


	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 3 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

CHANGE LOG

Vers.	Date	Author	Description
0.1	17.10.2009	Michal Janiszek (SQS)	First version
0.2	27.10.2009	Michal Janiszek (SQS)	Additional tools included
0.3	27.10.2009		Reviewed Version
1.0	06.11.2009	Michal Janiszek (SQS)	Corrections included, three new tools added

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	D-MINT ID

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 4 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

1. TERMINOLOGY

2. INTRODUCTION

Solutions that are summarised in this document have been submitted by the following D-Mint partners:

- PikeTec (2 tools)
- iXitronics (2 tools)
- Conformiq (1 tool)
- AAU (1 tool)
- NetHawk (1 tool)
- Elvior (2 tools)
- VTT (1 tool)
- TestingTech (2 tools)
- FOKUS (3 tools)
- IESE (2 tools)

3. OVERVIEW OF TOOLS

In D-Mint project are involved 10 organisations that offer in summary 17 different tools. Each tool described in “integrated description of all tool chain components” draft document contain following information:

- **Organisation name:** Name of D-Mint partner that has introduced any tool.
- **Overview of the Tool:** A short overview of solution.
- **Tool Status:** Describes the status of the development and tells if the version of the tool is already complete.
- **Tool Application:** Possible values are: Modelling, Test Generation, Test Execution, Planning & Management and Test Analysis.
- **Integrable with:** determines other applications that can integrate with tool.
- **Extensions:** available tools extensions that expand tools functionalities.
- **Contact Person:** Person in each partner organisation responsible for cooperation with D-Mint.



	Integrated description of all tool chain components Deliverable ID: D.3.4.v1.0	Page : 5 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

Table 1. Summary view of all tools

Partner Organization	Name of Tool	Tool Status	Tool Application Area	Integrable with	Extensions
PikeTec	<i>TPT</i>	Tool	Modelling Test Generation	MATLAB, ASCET, Targetlink, MESSINA, TESSY, INCA, PROVEtech:TA, MLBA4	TPT CAMEL-View-Plugin
	<i>TPT CAMEL-View-Plugin</i>	Prototype	Test Execution	CAMEL-View	-
iXtronics	<i>CAMEL-View</i>	Tool	Modelling	TPT from PikeTec TTmodeler from TestingTech	-
	<i>CAMEL-View Testing TBX</i>	Prototype	Modelling Test Execution	TPT from PikeTec TTmodeler from TestingTech	-
Conformiq	<i>Conformiq Qtronic</i>	Tool	Test Generation	IBM/Telelogic Rhapsody IBM/Rational RSARTE Sparx Systems Enterprise Architect	-
AAU	<i>MATERA (Modelling for Automated Test derivation at Aau)</i>	Tool	Modelling	Qtronic, EAST	MagicDraw
NetHawk	<i>NetHawk EAST</i>	Tool	Test Execution	Qtronic (with scripiter plug-in) VTT RCA (through log files)	-
Elvior	<i>MessageMagic</i>	Tool	Test Execution	MOTES Conformiq Qtronic	-
	<i>MOTES</i>	Prototype	Test Generation	UML CASE tools Poseidon, Artisan Studio TTCN-3 test tools MessageMagic, TTworkbench, OpenTTCN, IBM Rational TTCN-3 Suite	-
VTT	<i>Root Cause Analyzer (RCA)</i>	Prototype	Test Analysis	Qtronic (through log and model files) EAST and Wireshark. (through log files)	-

	Integrated description of all tool chain components Deliverable ID: D.3.4.v1.0	Page : 6 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

Partner Organization	Name of Tool	Tool Status	Tool Application Area	Integrable with	Extensions
TestingTech	<i>TTworkbench</i>	Tool	Test Execution Planning & Management Test Analysis	HP Quality Center, IBM Rational Software Architect, Eclipse, Hudson Topcased,	Camel View Testing TBX, TTmodeler, TTplugin
	<i>TTmodeler for TTworkbench</i>	Tool	Modelling Test Generation	IBM Rational Software Architect Enterprise Architect, any other Eclipse based modelling tools	TTworkbench
FOKUS	<i>MDTester</i>	Prototype	Modelling	any Eclipse-based testing or system design tools	-
	<i>ADTgenerator</i>	Prototype	Modelling Test Generation	PREEvision	TTCN-3 or tool specific scripts
	<i>qTTCN-3</i>	Prototype	Test Analysis	Eclipse applications	Graphical result presentation applications
IESE	<i>JUMBL</i>	Tool	Modelling Test Generation Test Analysis	HP Quality Center Matlab / Simulink	Dynamic adaptation of usage profiles in test models Traceability Extensions Test Adapter and Tool Connections
	<i>jSEQ</i>	Prototype	Modelling	JUMBL DOORS (Import of DOORS requirements) Excel (Import of tabular requirements)	New development

3.1.1 Tools by status

The tools presented in “integrated description of all tool chain components” have been under different status of development. Possible values here are Tool and Prototype. The following Table 2 and Figure 1 present how the tools have been distributed based on their current status. Because of that this criterion is has a logic character tool can be assigned only to one category.

Table 2. Number of Tools by status.

Tool Status	No.
Tool	9
Prototype	8

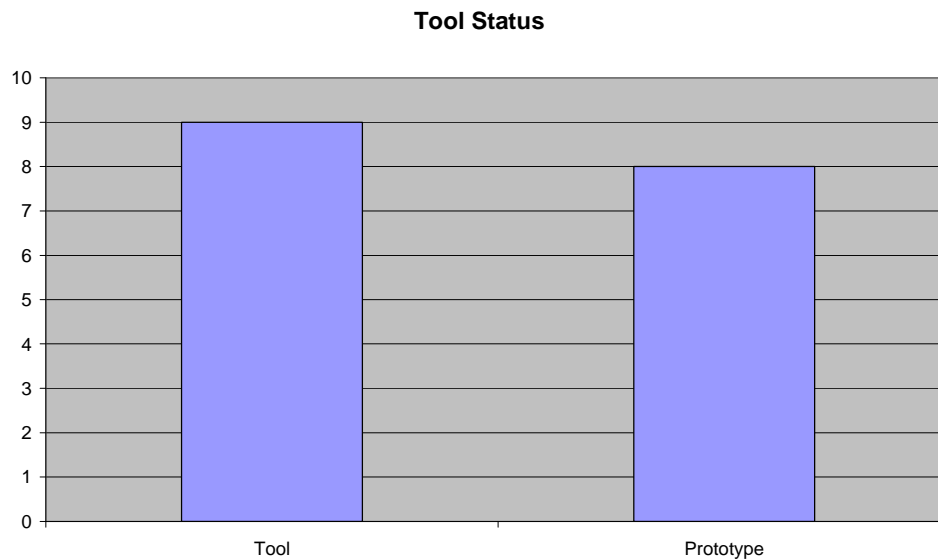


Figure 1. Number of Tools by status

3.1.2 Tool by application

Table 3 and Figure 2 present how the tools have been distributed based on application areas. Possible values here are: Modelling, Test Generation, Test Execution, Planning & Management and Test Analysis. According to tools developers few applications cover more that one application area, because of that sum of tool numbers presented in development areas is not equal to number of available tools.

Table 3. Number of Tools by application area

Tool Application	No.
Modelling	8
Test Generation	6
Test Execution	5
Planning & Management	1
Test Analysis	4

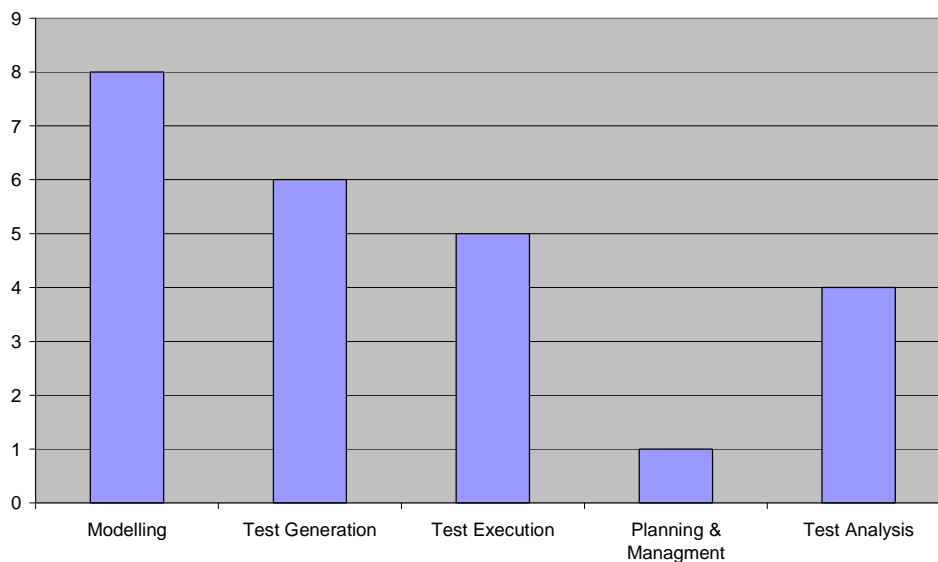
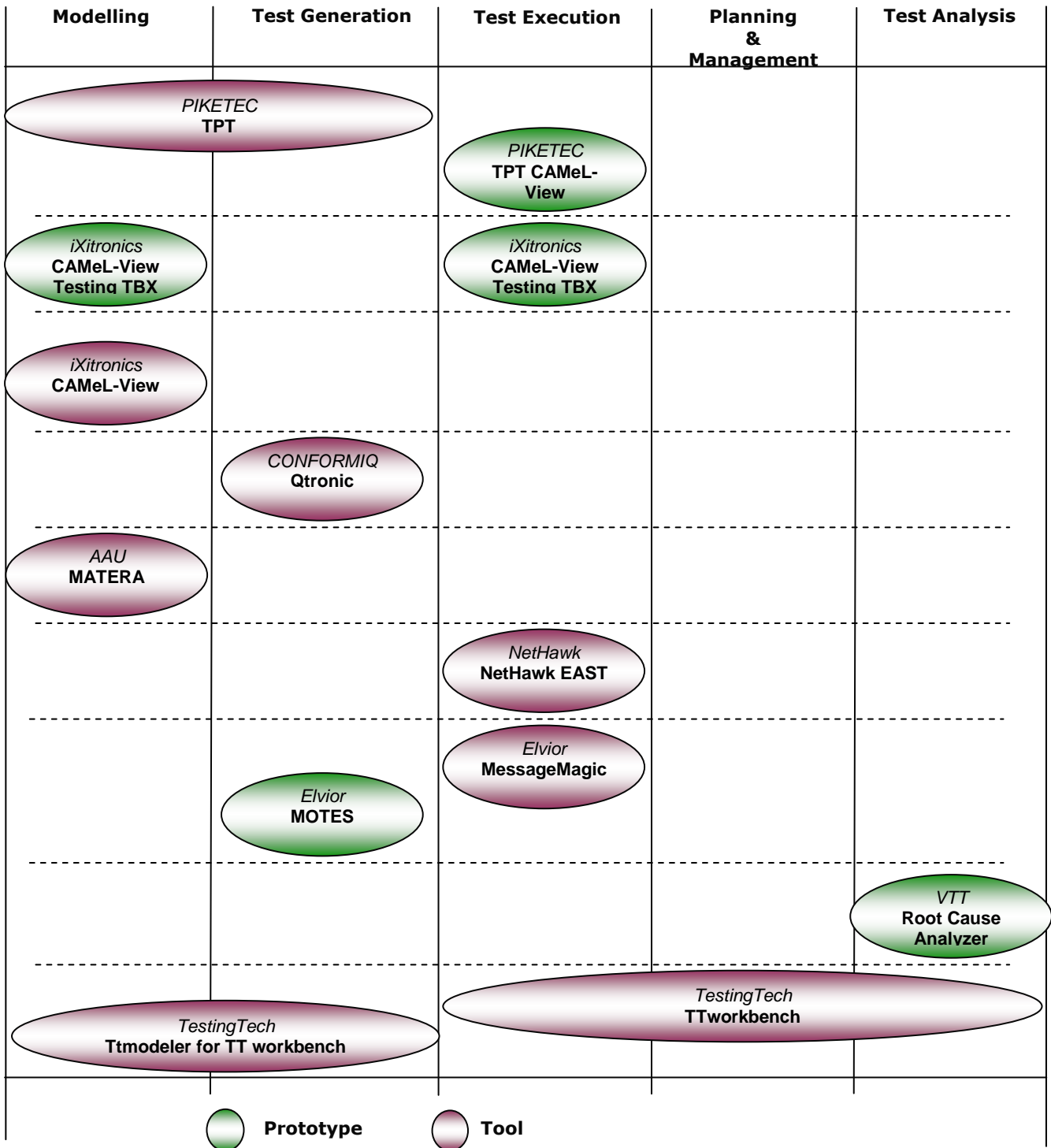


Figure 2. Number of Tools by application area

Tools included in “integrated description of all tool chain components” document focused to Modelling, Test Generation and Test Execution areas. However there are four solutions for Test Analysis area, but two of them are still in prototype phase. There is only one tool (already implemented, not prototype) that covers Planning&Management area.



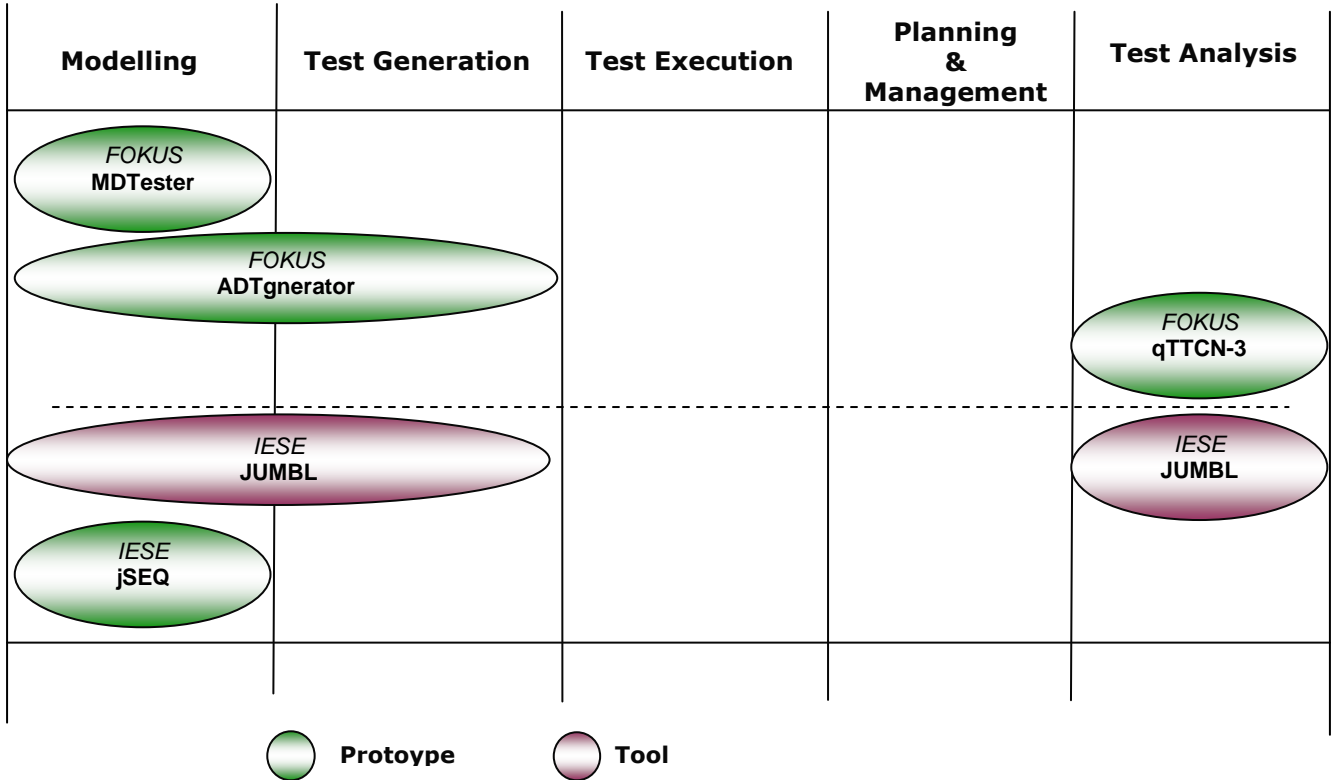



Figure 3. Summary view of all Tools

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 11 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

4. TOOL DESCRIPTIONS

4.1 PIKETEC

4.1.1 TPT

Contact person:

Eckard Bringmann

Tool description:

From document “integrated description of all tool components”:


TPT is a model-based testing tool for testing embedded systems, especially the testing of control systems. TPT supports all important fields of the test process which are in detail the management of tests, test modelling, test execution, test assessment, and test documentation.

From page <http://piketec.com/products/tpt.php>:

TPT is a model-based testing tool for testing embedded systems, especially the testing of control systems. TPT supports all important fields of the test process which are in detail the management of tests, test modelling, test execution, test assessment, and test documentation.

TPT key benefits

- **Cost-efficient consistent testing for all phases of the development:** One test description for all development phases (MiL, SiL, HiL). Early malfunction detection.
- **Intuitive test description:** Model-based test development without extensive programming skills required. The graphical test-modeling is platform-independent.
- **Executable on many different Platforms:** Simple integration and test execution on all platforms (e.g. [MATLAB/Simulink](#), [TargetLink](#), LabVIEW, ASCET, C-Code, CAN, LIN, HiL-Tests)
- **Real-time enabled:** On real-time-platforms the tests can be executed in real-time. The testing of controller systems is possible.
- **Closed-loop testing:** TPT supports reactive tests, that is to say tests, which can react on the system’s behaviour during the test execution. Particularly when testing control systems, this possibility is essential because system mostly show a reactive nature in these fields.
- **Efficient automatic test result evaluation:** After the test run the collected data can be assessed via GUI or assessment script. This means the expected behaviour of the system is analysed systematically. The source of the analysed data can be also a different measure tool like TargetLink or INCA.
- **Efficient automatic test documentation:** The test results are documented as information rich freely configurable HTML reports.
- **Universal test tool:** The test specification (modeling) is identical for all tests (model-testing, unit-testing, Integration-testing, validation-testing, regression-testing).
- **Automatic test process:** The automatic process using TPT includes test execution, test assessment and test documentation. This process is the same using the same tool for all platforms.

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 12 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

- **Highest test-quality** by means of systematic testing.
- Model-based testing of **AUTOSAR** components: In all phases of the development AUTOSAR components can be tested comfortably.
- **ASAM MCD-3 Measurement:** Parallel to the test execution controller internal signals can be measured via [MCD-3](#).
- **Requirements tracing:** [Traceability](#) of requirements and tests.

Support of Tests with Continuous Signals: In the field of control engineering the in- and output variables often are continuous signals. On the market, TPT is the only systematic test tool for reactive tests of embedded systems supporting the easy modelling of test cases with continuous signals.

Automated Tests: Most of the steps of the testing process are being automated by TPT. This considerably shortens the expenditure of time for the test execution, test assessment, test documentation, and the test management. Especially at the development processes for embedded systems which often proceed strongly iterative, the automation allows efficient test repetitions and so an optimal integration of test activities into the development cycles.

Consistency: TPT allows the repetition of tests in different »in- the-Loop«-environments without modifying the test cases for the exchange between Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) and Hardware-in- the-Loop (HiL). This is possible because TPT uses both a platform independent test description language and a platform independent test machine which can be integrated in almost any test environments.

Many Platforms: Simple integration in all test-platforms (e.g. [MATLAB/Simulink](#), [TargetLink](#) ⇄ LabVIEW, C-Code)

Systematic Testing: A systematic and structured choice and modelling of the test cases is of particular importance, because testing always is an empiric operation that can not prove the system quality entirely. In TPT test cases are not being looked at isolated of each other. Instead, all test cases are being created in a common model with the objective of working out common and different characteristics in order to get an overview about which aspects have been tested and which might have been neglected.


4.1.2 TPT CAMeL-View-Plugin

Contact person:

Eckard Bringmann

Tool description:

The TPT CAMeL-View-Plugin is an extension of TPT that allows TPT models to be executed on CAMeL-View based HiL test rigs.

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 13 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

4.2 IXTRONICS

4.2.1 CAMEL-View

Contact person:
Ulrich Meier-Noe

Tool description:

From document “integrated description of all tool components”:


CAMEL-View is an environment for the development of mechatronic systems and offers a variety of functions to fully support the development cycle of innovative systems. This includes the modelling of the physical system based on its topology as well as the implementation of the more functional control concepts.

From page http://www.ixtronics.com/ix_hist/English/CAMELViewProg.htm :

CAMEL-View provides a comfortable interactive way to build up models of complex mechatronic systems which include different system domains (e.g. multibody, hydraulic, control-engineering, and discrete systems). Moreover, the developer is supported by an extensible database comprising predefined components. Furthermore, CAMEL-View allows the integration of models taken from other modelling environments, such as Matlab/Simulink. The domain-specific model of multibody systems includes also the 3-D graphical description which can be imported from numerous CAD systems (OpenInventor, VRML, DXF and IGES) and automatically reduced for animation purposes. For analysis, the domain-specific model can automatically be transferred to a mathematical representation and furthermore to optimized C code. What is more, CAMEL-View allows to export the model to Matlab/Simulink as an mdl-file including s-functions. This Simulink model maintains the model hierarchy and allows access to the model parameters.

Features:

- domain-specific modelling
- interactive modelling
- extensible database of predefined components
- automatic model documentation (MS-Word, HTML, MIF)
- CAD import of the 3-D geometry representation
- support of the type concept
- object-oriented modelling
- automatic derivation of the mathematical representation
- export of optimized C code especially for real-time
- simulation and Hardware-in-the-Loop applications

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 14 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

4.2.2 CAMEL-View Testing TBX

Contact person:

Ulrich Meier-Noe

Tool description:

A tool-box for the software tool CAMEL-View which supports the integration of D-MINT technologies into CAMEL-View as well as the interaction with other tools for the model based testing.

4.3 CONFORMIQ

4.3.1 Conformiq Qtronic

Contact person:

Kimmo

Tool description:

Conformiq Qtronic is our tool for automatically designing and creating test cases. The tool uses a high-level system model as its input. It calculates a high-quality set of test cases mathematically from the system model and exports the test cases in user-definable formats, for example TCL, TTCN-3, HTML, XML, or Python.

Based on the acclaimed Eclipse environment, Conformiq Qtronic is available cross-platform for Windows as well as for Linux and other UNIX variants. Available both as a stand-alone application and as an Eclipse plug-in, the tool can be easily installed and deployed without prior knowledge about Eclipse, but it also fits seamlessly into an existing Eclipse-based workflow.

Feature highlights:

- Generates test plans and executable test scripts automatically using state-of-the-art test generation algorithms
- Produces automatically traceability matrices, test case dependency information, human-readable test plans and computer-executable test scripts
- Fully interactive and visual test generation and review workbench based on the Eclipse environment, including customizable views, graphical representation of tests as message sequence charts, and a graphical mapping between the input models and the generated test cases
- Completely user-definable output formats for test case output
- Support for multiple, co-existing test generation settings with fine-grained control over requirements and model features to be used as basis for test generation
- Automatic support for state-of-the-art black-box design techniques such as boundary value analysis, atomic condition coverage, and orthogonal array testing
- Makes full use of multi-core PCs for optimal processing performance during test generation

4.4 AAU


4.4.1 MATERA (Modelling for Automated Test derivation at AAU)

Contact person:

Dragos

Tool description:

MATERA is a toolset that integrates UML modelling with several tools used across a custom MBT process. The GUI of MATERA is implemented as a plug-in in the NoMagic's MagicDraw modelling tool, combining existing capabilities of MagicDraw with custom ones. MATERA supports graphical specification of the

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 15 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

requirements using SysML and tracing of these requirements to the UML models specifying the SUT. Model validation is performed in MagicDraw using both predefined as well as custom validation rules. The resulting models are automatically transformed into input for the Conformiq Qtronic tool, used for automated test generation. Upon executing the test scripts generated by Qtronic in the Nethawk East execution environment, statistic analysis of the test run is presented in the GUI. The back-traceability of the covered requirements from test to models is also provided in the GUI in order to facilitate the identification of the source of possible errors in the models.

4.5 NETHAWK

4.5.1 NetHawk EAST

Contact person:

Tapio

Tool description:

From document “integrated description of all tool components”:

NetHawk EAST is a multi-technology testing platform for NetHawk Simulators and testing services.

From page https://www.nethawk.fi/products/nethawk_simulators/ :

NetHawk EAST is a multi-technology testing platform for NetHawk Simulators and testing services. NetHawk EAST covers technologies including wireless, VoIP and legacy networks. A single multi-technology test case enables true end-to-end testing. NetHawk testers are used in regression, functional, load and conformance testing. Even the most complicated testing needs can be satisfied by combining the testing platform with NetHawk’s professional testing services.

The flexible test case editing and execution tools enable customers to modify standard tests cases and develop new sophisticated testing scenarios limited only by the user’s imagination and skills. NetHawk EAST is the most flexible testing tool on the market with virtually unlimited high density scaling to meet the most rigorous heavy load test requirements.

4.6 ELVIOR

4.6.1 MessageMagic

Contact person:

Andres Kull


Tool description:

From document “integrated description of all tool components”:

MessageMagic is a TTCN-3 test development and execution environment.

From page <http://www.elvior.ee/messagemagic/>:

MessageMagic is a TTCN-3 test development and execution platform. It can be used for testing software or hardware components in a wide range of industry sectors. Candidate systems for testing with MessageMagic

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 16 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

can be found from the industry sectors where TTCN-3 is widely used as an accepted generic test language - communication, transport and automotive, military and defence.

MessageMagic is ideal for incremental project development. It can be used for testing of an individual task or process. One can continue with the integration tests of a subsystem consisting of processes and later to integrate those subsystems and test them together as a whole software subsystem. It is even possible to continue to perform SW/HW integration tests where MessageMagic can be used to test SW running in target HW.

MessageMagic simplifies and accelerates product development, with all the benefits that derive from shorter development cycles. MessageMagic relies on TTCN-3 architecture and supports TTCN-3 core language. MessageMagic works on Windows.

4.6.2 MOTES

Contact person:

Andres Kull

Tool description:

From document “integrated description of all tool components”:

MOTES is a tool for generating TTCN-3 tests from the UML model of the system under test.

From page <http://www.elvior.ee/motes>:

Test cases development is efficient


- tests are generating from model by pressing a button
- good enough amount of test cases is generated
- high test coverage

Test cases maintenance is under control

- model is built once
- changes in implementation cause change needs in model
- all test cases are regenerated from model by pressing a button

Platforms

MOTES is Eclipse plug-in. It runs in every platform that runs Eclipse. MOTES currently uses UPPAAL verification engine for test generation, which is available for Windows and Linux on x86 architectures only.

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 17 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

4.7 VTT

4.7.1 Root Cause Analyzer (RCA)

Contact person:

Tomi

Tool description:

The purpose of the RCA system is to deduct causes of failures observed under model-based testing. Faults are detected and analyzed based on the state machine model of the system under test (SUT), as well as logs from the test executing and communicating software (EAST, Wireshark and Qtronic).

4.8 TESTINGTECH

4.8.1 TTworkbench

Contact person:

Stephan

Tool description:


TTworkbench is a graphical test development and execution environment based on the international standardized testing language TTCN-3. Apart from text-based TTCN-3 test specification, TTworkbench also offers the option of graphic definition of test cases.

Once tests have been defined in TTCN-3, TTworkbench compiles them in an executable Test Suite. The comprehensive test management and test execution environment feature allows users to administer, execute and analyze their tests. It supports test automation methods which significantly reduce costs and ensure optimal quality throughout the whole test cycle.

The Eclipse based TTworkbench can be deployed for testing products and services in a wide range of different industry sectors. Candidate systems for testing can be both hardware- and software-specific, as testing is based on TTCN-3. This generic test language is THE medium to evolve technology-independent test suites.

Features:

- Test management, execution and analysis
- TTCN-3 Core Language Editor including T3Doc
- TTCN-3 Compiler
- Graphical TTCN-3 Editor
- TTCN-3 source code level debugger
- Runtime plug-in development environment
- Distributed Test Execution Management System
- Built-in adaptation extensions for common protocols

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 18 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

4.8.2 TTmodeler for TTworkbench

Contact person:

Stephan

Tool description:

From document “integrated description of all tool components”:

TTmodeler integrates the model-based system and test design and creates a direct, automated connection to the powerful test development and execution environment TTworkbench. The UML Testing Profile (UTP) defines additional model elements for UML, which allow to define specifications of test objectives, test procedures and test valuations for systems and system components. UTP is a vital component for the alignment of system and test development methods. It contributes to the concept of model-centric systems design: System models and test models can be developed and aligned in all system development phases.

From page http://www.testingtech.com/products/ttpugins_modeler.php :


TTmodeler integrates the model-based system and test design and creates a direct, automated connection to the powerful test development and execution environment TTworkbench. The UML Testing Profile (UTP) defines additional model elements for UML, which allow to define specifications of test objectives, test procedures and test valuations for systems and system components. UTP is a vital component for the alignment of system and test development methods. It contributes to the concept of model-centric systems design: System models and test models can be developed and aligned in all system development phases.

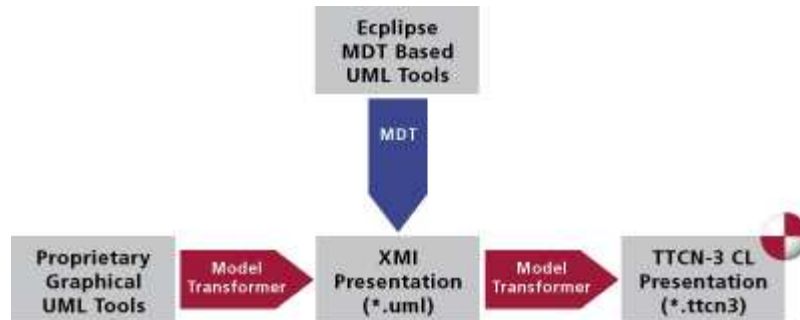
Highlights/Features:

- TTCN-3 mapping of UML Testing Profile
 - Extension of test development life cycle with model-based concepts
 - Generation of TTCN-3 code from UTP test models
- Embed in TTworkbench
 - Immediate use of TTCN-3 based test implementation and execution tools
 - TTCN-3 test management available for UML2 and UTP based systems
- Widely used graphical UML tools integrated
 - Eclipse-based e.g. IBM Rational Software Architect (RSA), TOPCASED
 - Non eclipse-based e.g. Sparx Systems Enterprise Architect (EA)
- In-line with Eclipse concepts and open to Eclipse modelling tools (e.g. MDT- Model Development Tools)

Benefits

- Alignment of system design and test design
- Model-driven system development approach
- Direct automated execution of test models
- Standards-based testing with TTCN-3 and UML testing profile

	Integrated description of all tool chain components Deliverable ID: D.3.4.v1.0	Page : 19 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public



4.9 FOKUS

4.9.1 MDTester

Contact person:

Axel

Tool description:

MDTester is an application to support pattern-oriented model-driven design of tests. Using the graphical UTML notation, tests can be designed at a high level of abstraction and then transformed automatically into test scripts for the target environment. MDTester supports a unified approach to test design by covering other aspects of testing such as test planning and facilitating integration with other aspects of the software development process.

4.9.2 ADTgenerator

Contact Person:

Axel

Tool description:

Based on an architectural model defined within the PREEvision modelling tool (<http://www.aquintos.info/>), textual test descriptions can be derived. It starts with semi-formalized description of requirements and is able to take into account further information kept in the architecture model. It intensively uses links between the model artefacts in different architectural viewpoints. The ADTgenerator demonstrates the ideas of architecture based test derivation using model queries, metrics calculations and an early version of the report generation of the modelling tool.


4.9.3 qTTCN-3

Contact Person:

Axel

Tool description:

The Test Data Quality Analyser (qTTCN-3) is an implementation to generate statistical and dynamic metrics in the context of TTCN-3 test quality improvement. qTTCN-3 allows guidelines checks (e.g. user-defined naming conventions) and calculation of test data variance for test cases using methods e.g. for clustering TTCN-3 template definitions and coverage analysis.

	<p>Integrated description of all tool chain components</p> <p>Deliverable ID: D.3.4.v1.0</p>	Page : 20 of 20
		Version: 1.0 Date : 24/08/2009
		Status : Final Confid : Public

4.10 IESE

4.10.1 JUMBL

Contact Person:

Thomas Bauer

Tool description:

JUMBL is a command line application for model-based statistical testing, programmed in JAVA. JUMBL provides the functionality for specifying and analyzing test models, generating test cases from models, and analyzing test results, and estimating the reliability of the test object.

4.10.2 jSEQ

Contact Person:

Thomas Bauer

Tool description:

jSEQ covers the core functionality for test model construction from the original textual requirements until the test model structure. The test model construction approach is called sequence-based specification (SBS). In the SBS a set of tables is developed that describe the system interface and the functional system requirements. The information can be exported to several output formats including JUMBL-test models.