

Statemate[®]

Automatic Test Generation

Release 4.6.1

Release Notes



IBM[®]

License Agreement

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner, BTC Embedded Systems AG.

The information in this publication is subject to change without notice, and BTC Embedded Systems AG assumes no responsibility for any errors which may appear herein. No warranties, either expressed or implied, are made regarding Rhapsody software and its fitness for any particular purpose.

Trademarks

IBM, Rational, and Statemate are registered trademarks of IBM Corporation.

All other product or company names mentioned herein may be trademarks or registered trademarks of their respective owners.

© Copyright 2000-2009 BTC Embedded Systems AG. All rights reserved.

Contacting IBM Rational Software Support

IBM Rational Software Support provides you with technical assistance. The IBM Rational Software Support Home page for Rational products can be found at <http://www.ibm.com/software/rational/support/>.

For contact information and guidelines or reference materials that you need for support, read the [IBM Software Support Handbook](http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html) (<http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html>).

For Rational software product news, events, and other information, visit the [IBM Rational Software Web site](http://www.ibm.com/software/rational) (<http://www.ibm.com/software/rational>).

Voice support is available to all current contract holders by dialing a telephone number in your country (where available). For specific country phone numbers, go to <http://www.ibm.com/planetwide>.

Before you contact IBM Rational Software Support, gather the background information that you will need to describe your problem. When describing a problem to an IBM software support specialist, be as specific as possible and include all relevant background information so that the specialist can help you solve the problem efficiently. To save time, know the answers to these questions:

What software versions were you running when the problem occurred?

Do you have logs, traces, or messages that are related to the problem?

Can you reproduce the problem? If so, what steps do you take to reproduce it?

Is there a workaround for the problem? If so, be prepared to describe the workaround.

1 Package Identification

1.1 Vendor Identification

BTC Embedded Systems AG Buschstrasse 1 26127 Oldenburg +441 96 97 38 0 (voice) +441 96 97 38 64 (fax)

1.2 Tool Identification

IBM Rational Statemate ATG 4.6.1 Commercial Release

1.3 Media Identification

CD-ROM (Solaris): ATG/Documentation/Example Project

CD-ROM (Windows NT/2000/XP): ATG/Documentation/Example Project

1.4 Date of Release

July 2010 for IBM Rational Statemate 4.6.1

1.5 Installation

See the STM_ATG Installation Guide (to be found on the CD).

IBM Rational Statemate 4.6.1 has to be installed previously.

1.6 Platform Issues

Supported platforms are

Sparc Solaris 2.10

Windows 2000, Windows XP

with the same window systems as IBM Rational Statemate.

2 Features

2.1 Statemate ATG

The *Statemate Automatic Test Generation (ATG)* provides a push-bottom generated set of test vectors that cover a Statemate design model. The user can select the degree of exhaustiveness that is desired to cover different criteria of a model. With a set of test vectors for a standard model coverage, ATG reports about covered states, transitions, used transition trigger and static reactions, for activities about covered mini-specs and truth table lines, and about toggled output data items.

2.2 Several Test Formats

The ATG tool is a plugin of Statemate . The integration of ATG directly accesses the contents of the workarea and produces output formats like Simulation Control Programs (SCPs) and Statemate's customizable Test Vectors that are immediately available inside Statemate. Additionally, an ASCII representation of test vectors and an XML format are supported. These test representations describe in particular the changes of inputs as well as outputs at the design interface. Time stamps label the input stimuli and the expected output values.

2.3 Test Vector Applications

The test vectors may be used to validate the design model itself. SCPs and Statemate's test vector format are generated to automate this use case.

Applying the automatically generated test vectors to software implementations and/or hardware (ECU) solutions enable a complete test against the Statemate specification model. Since ATG provides not only the timed sequences of input stimuli that are required for these test applications but also the output values that are expected with respect to the specification model, an automated test evaluation becomes possible.

2.4 Coverage Reports and Statistics

Detailed coverage reports are generated with the test vectors

For the summery of test vectors that belong to one test goal

For the set of all generated test vectors

These statistic reports tell you which states, transitions, static reactions, mini-specs, etc., and output values are covered. For transition guards, it is marked whether MC/DC coverage is reached independent from the question if this criteria was a specified coverage goal.

A more general Application Report informs about the selected test goals, the ATG application status, and in summery about the achieved coverage.

2.5 ATG Profiles

The ATG profile stores all user input, test goal selections and definitions, in a profile. These profiles can be exchanged via and stored in a databank as all other ingredients of a Statemate design project.

3 Release News

3.1 New in this release of Statemate ATG

This version no longer supports Solaris 2.8 and 2.9. There are no feature updates, but the plugin and its documentation have been adapted to the new Statemate release.

3.2 New in release ATG 4.6

The Automatic Test Generation plugin is now part of the IBM Rational Statemate family.

The version numbering of the previous plugin releases (ending in the previous release number 3.6) has been removed for this and future releases. The plugin releases are simply synchronized and numbered as the Statemate tool itself. Thus we now have Statemate ATG 4.6.

Dependencies from some open source (GNU) tools and libraries have been removed from the release. These free tools and libs can be taken from ftp://stm_plugins@btc-es.de/.

The installation has been adapted to the IBM look&feel. Furthermore, the installer for the Windows platform has been renewed. All documentation got an update.

3.3 New in release ATG 3.6

Statemate ATG 3.6 has been released for Statemate 4.5. There are no feature updates, but the plugin and its documentation have been adapted to the new Statemate release.

3.4 New in release ATG 3.5

The ATG 3.5 release has been adapted to Statemate 4.4. In particular, this includes support for the new Transition Priorities in state charts. The new Complex Constant Value Definition are not supported, only simple constant values may be used for ATG. The Static Check messages of ATG and the documentation have been adapted accordingly.

For Windows users, the ATG user interface has been redesigned to get a better Windows-like look-and-feel. The dependency from an X-server has been removed. The set of packages that are installed for the cygwin runtime environment has been reduced. The user interface has been improved in its graphical presentation and some logging messages have been cleaned up to get better readability.

3.5 New in release ATG 3.4

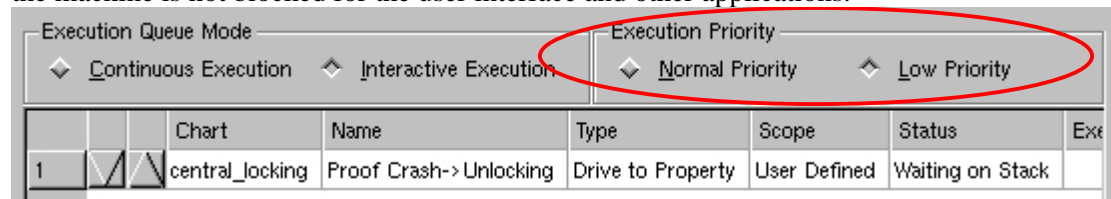
The ATG 3.4 release has been adapted to Statemate 4.3. The key improvement is a replacement of the previously used so-called “Bounded Engine”, now yielding much higher performance. With this release, a derivate of the SAT competition winner, MiniSat¹, has been integrated. Similar to the SAT competition results in particular for industrial applications, MiniSat has turned out to be the best available kernel engine for Bounded Analyses for Statemate models. The benefit is mainly a reduced runtime of Model Coverage tasks (about 35%, depending on the model). Less memory consumption, less time-outs, and better coverage results in Analyses tasks are also measured with the new engine.

Further enhancements are

A new option allowing the user to reduce the priority of the tasks that are executed. To this end, the Execution Queue tab in the user interface contains an option for selection of “Normal” versus “Low” Execution Priority. Low priority for the execution will lead to longer run time, but has the advantage that

¹ <http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/>

the machine is not blocked for the user interface and other applications.



The length of file names and paths has been drastically reduced to avoid the problem that Windows typically supports only a very limited length of file- and path-names. Execution errors due to “file name too long” should be solved now. Model and component names do not influence the (internal) file names any more.

Solaris 2.10 is now supported, too.

The Static Check has been slightly improved for some situations where Correctness Errors in the model (reported by STM’s Check Model tool) avoid an application of ATG.

The installer (setup.exe) for the installation on Windows has been fixed: it could happen that the installer ended up in a deadlock when the user entered a non-existing installation path. Another fix has been implemented concerning the uninstallation of previous STM plugins in automated “Replay” of a recorded installation.

3.6 New in release ATG 3.3.2

ATG 3.3.2 is released for the maintenance release MR-2 of Statemate 4.2. The following changes and enhancements have been implemented:

A performance issue on some large models has been solved. When applying the so-called bounded engine it could happen in previous releases, that starting the engine took several minutes where in the new release only a few seconds are required.

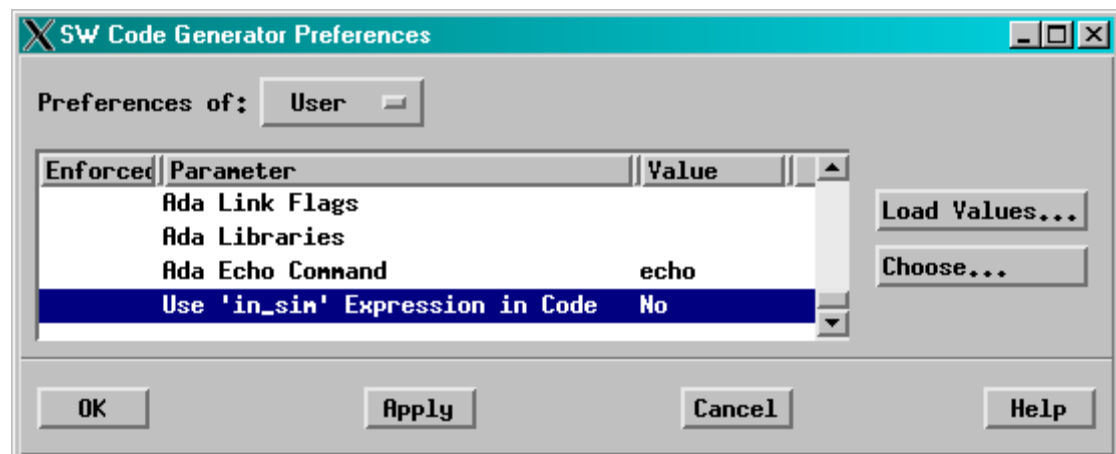
Parsing and checking properties in the drive-to-property dialogs has been improved with respect to the runtime to improve the interaction with the user interface.

The semantics of Statemate’s `in_sim(event_expr)` operator say that the `event_expr` is evaluated and executed by Statemate’s simulator, but is ignored by any other tools in the Statemate tool family. As consequence, the interpretation of the model information by ATG was different from the one by the STM simulator. As further consequence, the SCPs generated by ATG do not run in the STM simulator whenever an `in_sim()` operator is involved.

With this release, the STM plugin takes the information from the Statemate preferences settings into account. There is an option introduced with Statemate’s 4.2 maintenance releases in the

```
Project -> Preferences Management
        -> Statemate Prototype C Code Generator...
        -> Use 'in_sim' Expression in Code
```

that allows the user to enable the interpretation of the `event_expr` in the code generation and in the ATG and ModelChecker and ModelCheckerPlus plugins, too.



To this end, set the value to “Yes” and Re-Compile the design model inside ATG.

A bug has been fixed that occurred when compiling the model into the plugins representation. Whenever the model contains a `length_of ()` operator applied to an component of a structured data item, an unexpected compilation failures has been reported. This is fixed now.

For the XML format of the test vector export, a so-called DTD (*Document Type Definition*) has been added. The file `TestVectorExport.dtd` can be found in `<STM_ROOT>/plugins/atg33/doc` .

3.7 New in release ATG 3.3.1

ATG 3.3.1 is released for the maintenance release MR-1 of Statemate 4.2. The following changes and enhancements have been implemented:

The interface computation has been fixed for the top level of models with asynchronous simulation semantics. Data_items which are read from the environment as inputs and modified in the model as well are now treated as so-called “inouts” in ATG. The gap between the ATG interpretation of the model and the STM simulator behavior is closed at this point now.

The operator `in_sim` has been added to the list of operators in the User Guides and Limitations document. Note that, as defined by Statemate “`in_sim(event_expr)`” is replaced by “`empty_event`” by all Statemate tools but the simulator. Thus, ATG does not take the `in_sim(event_expr)` into account.

The analysis with the bounded engine has been improved. In drive-to-transition and drive-to-property tasks you may get the result “unreachable” where previous releases reported “Not reached” with “Engine depth <n>” steps. In model coverage tasks you may get more “unreachable” coverage goals where previously “uncovered” was the result status.

3.8 New in release ATG 3.3

ATG 3.3 is released for Statemate 4.2. For this Statemate version the following improvements and enhancements are implemented in ATG:

Support for Statemate’s new `reset_element` and `reset_all_elements` operators.

A new option in ATG’s test vector export allows the user to generate test vector specific coverage reports for each test vector.

A new version (1.5.21) of the cygwin runtime environment for Windows is used. The new environment avoids conflicts that occurred when the user has cygwin for other purposes too. Furthermore, ATG benefits from memory allocation improvements that come with the new cygwin.

Inside the ATG technology building blocks, several improvements have been implemented. Overall performance improvements:

Reduced amount of allocated memory during runtime

Shorter runtime – in particular for model coverage tasks and applications with the bounded analysis engine

Higher coverage on models where memory and runtime were the limiting factors in previous releases.

Applications of ATG to small models will probably not take advantage of the improvements. Performance profits on larger models depends on the length of the (expected, generated) test vectors and on the principle reachability of the coverage goals inside the model. Runtime reductions of up to 35% compared to the previous release have been measured on customer models.

A complexity problem that yielded an internal parse error and blocked the generation of test vectors has been solved.

A bug fix for a compilation failure in the context of assignments of don't care values to outputs in truth tables has been made.

The support for Solaris 2.6 and 2.7 platforms and for Windows NT have been skipped, since these are not supported by Statemate any more.

3.9 New in release ATG 3.2.2

The ATG 3.2.2 plugins is released for Statemate 4.1 MR-2. There are only slight modifications compared to the previous release.

A bug has been fixed that concerns the treatment of history connectors. There semantics in general and resetting history in particular were faulty in previous releases.

A second bug fix is related to user defined functions of type string. When they are used in assignments to context variables then the previous release crashed during the static analysis of the model.

The analysis of Boolean expressions transition trigger, mini-specs, static reactions etc. was not complete with respect to the observation of MCDC coverage with generated test vectors. As result, the reported coverage was better than the really reached MCDC coverage. This has been fixed. Due to the corrected computation the reported MCDC coverage is now in some applications below the previously reported (wrong) statistic.

In those cases where Extended Trigger coverage or Full Set is selected as coverage criteria the whole application run-time may be higher than in previous ATG releases, but will probably also generate more test vectors to get a better MCDC coverage.

A limitation of previous releases was that the test vector resolution (TVR) setting in the ATG profile was related to the Model Time Unit (MTU). In particular, it is now possible to specify models with respect to a unit in milliseconds (e.g. 10 msec) and to specify as TVR a value of, for example, 50msec. The vectors generated by ATG stimulate the system then only at every 50 milliseconds. Note that the user has to take care that implementations of such specification models are fast enough to get stable in time.

An advantage of models which use milliseconds as time basis is that there the `tm()` and `delay()` operators usually talk about integer values while models that use seconds (the default in the Statemate simulator) often specify REAL values: `dly(12)` interpreted in msec instead of `dly(0.012)` in sec. The advantage for ATG is that there are no precision and fixed-point representation problems with integer values, while REALs are inside ATG represented on a fixed point basis what sometimes lead to concretization errors during the test vector generation.

Due to adaptations to the latest API of Statemate this MC release does not work with previous Statemate releases. STM 4.1 MR-2 is mandatory.

3.10 New in release ATG 3.2.1

ATG 3.2.1 is a maintenance version of the previous release. It is tested with and released for the Statemate maintenance release STM 4.1 MR-1. Fixed problems in ATG 3.2.1 relate

Truth-Table support: default rows in truth-tables were erroneously executed always at the end of the truth-tables execution.. Thus, the generated test vectors became useless for models with default rows in truth-tables.

Truth-Table support: implicit false “fs()” and true “tr()” events in table columns yielded a consistency error in the representation of the model's behavior in ATG. As consequence, the execution of ATG failed during test vector generation with an “internal error” message.

Models where ATG signals with an orange status lamp that some constructs are supported by abstraction only, got ATG-generated SCPs that could not be compiled and executed. This is fixed.

In some rare situations ATG generated for asynchronous models sets of test vectors, where the first vectors tried to change input values in non-stable states, i.e. within an asynchronous step. Afterwards, the expected behavior in the vectors differs from the simulated behavior. This was not intended and has been changed now.

In applications, where the so-called model time unit was set to a value that is different from the default “1 sec”, the test vectors exported by ATG in STM’s test vector format got time stamps in a wrong unit. The STM simulator generates time stamps in test vectors with an non-interpreted time unit as in `tm()` and `delay()` operations. Previous ATG releases used time information from the profile to generated real-time time stamps. Now, ATG works confirm to the STM simulator.

Array limits in user-defined types raised a crash in the (Re-)Compile step. This bug is fixed.

The so-called *bounded engine* reported always one step to much for generated test vectors in drive-to-transition and drive-to-property tasks. This regression problem is solved.

The *Installation Guide* has been updated.

A few optimizations have been implemented that will in particular speedup the reaction of the Graphical User Interface (GUI) when showing or opening (for edit) existing task definitions.

3.11 New in release ATG 3.2

This ATG release is adapted to the new API of Statemate 4.1. New features and fixed bugs are

- Support for Windows XP’s Service Pack 2.
- A solution for the so-called “rebase” problem: in previous releases it frequently happened on Windows systems that after the ATG installation a rebase procedure had to be executed. This problem is solved.
- A problem with the display of interfaces in the user interface for models with case sensitive data items is fixed.
- The coverage of truth tables has been improved, and the report about the achieved truth table line coverage has been added as new information to the coverage statistics report. Note that truth tables with empty cells prevent ATG to connect to the workarea.
- For model coverage tasks, for unreached test goals available information about the reason for the missing coverage (definitely unreachable or unreached due to time-out, out-of-memory, concretize problem indicating overflows, races or insufficient fixed point resolutions) has been added to the tasks coverage statistics. Note that this information is not available for all potential coverage goals.

3.12 New in release ATG 3.1.2

There are no changes compared to the previous release, except that the installer allows the installation for the MR-2 of Statemate 4.0.

3.13 New in release ATG 3.1.1

The Statemate ATG plugin Release 3.1.1 is nearly identical to the 3.1 in its functionality, but adapted to Statemate’s maintenance release 4.0 MR-1.

When you install this maintenance release of Statemate it is necessary to install this ATG 3.1.1 update too.

New is a bug fix that is related to synchronous time models only, where test vectors exported in Statemate's TV format and in XML were not correct.

Slightly modified is the Coverage Statistics report. Here the reported total number of covered transition, mini-spec, and static reaction guards is extended by the number of those of these objects that are even covered in compliance to the MC/DC coverage definition.

3.14 New in release ATG 3.1

ATG 3.1 is released for Statemate 4.0.

New is that major limitations in the support of expressions in timing operators has been removed. Timing expressions that contain Reals and user-defined functions in Action Language do not raise red lamps any more. The related part in the restrictions section in the User Guide document is adapted.

Bug fixes are provided concerning

the usage of constant arrays on string constants, and

the availability of generated abstraction functions in the interface when the model is marked Orange for abstracted model contents.

the generation of implicit written and changed events in test vectors, when a user selected startup duration time has been used. The first step after the startup phase, when ATG actively generated input stimuli for the system, could in some rare cases yield test vectors that made a wrong assumption on the generation of implicit input events.

the install script and documentation for the installation on Solaris systems has been made more robust.

3.15 New in release ATG 3.0

ATG 3.0 is released for Statemate 4.0. Based on feedback from professional users, new requirements, and improvements in technical challenges this release of ATG implements a many news. The changes compared to previous releases include an easier usage achieved by a complete redesign of the user interface and a better performance due to internal technical improvements. Details are

User Interface

- New layout of the user interface with different views to settings, test definitions, and results.
- New tool buttons "Statistic Report" and "Test Vector Export" and new icons for old functions.
- Global settings directly under the General Tab, no clicking through menus any more for this.
- Application Report "online" as view in top level interface.
- The mode (interactive/continuous) of the Execution Queue can be selected directly at the execution queue view.
- Some menu items renamed to become more intuitive.
- Easier entering and correction/editing of fixed-point resolution and boundary values.
- New result status display.
- New syntax info text for Property and Simple Assumption definitions.
- Detailed error messages and highlighting for inconsistent or invalid Properties and Assumptions.
- New hints about problems that arise when Freezing lists or Simple Assumptions are adapted/adjusting to a new model in a Re-Compile step.
- List in the user interface allow to hide columns (header in context menu).

- Special symbols and national characters are allowed to be used in names and comments.
- “Are you sure?” popup before overwriting file with new application report.
- Field for comments in Application Report.
- Improved control via keyboard.

Property Language like StateMate Action Language

- The language used for entering properties and assumptions (and Freezing values) has been adapted to the StateMate action language used in trigger and guards.
- Syntax as well as some type checking is performed on user input. In case of errors, the erroneous part of the property is highlighted and a detailed message about the problem is shown.
- In cases where old definitions slipped through the syntax check in the past test definitions may become “not ready” for execution. Example are tr() and fs() applied to basic states or conditions compared with Reals. In trivial cases, the definitions are automatically updated, for example the old notation BASIC_STATE^^ENTERED is changed to en(BASIC_STATE).

Installation

- Slightly modified procedure.
- New cygwin version (1.5.10) as run-time environment on PCs.
- Check about required packages when using existing cygwin installations.
- Updated Installation Guide. Hints to deal a sporadic “rebase” problem.

Documentation

- Complete update of all documents with respect to new user interface and tool usage
- Extended description of known limitations and workaround in User Guide

3.16 New in previous release ATG 2.5.3

The STM ATG 2.5.3 is released for the maintenance version StateMate 3.3.1-MR2.

It is technically equal to ATG 2.5.2 (see below), but comes with a completely redesigned and improved install setup for Windows platforms. In particular, recording (logging) installations and replaying recorded installations are now easy-to-use functionalities. On windows platforms, the installation of the cygwin runtime environment is now one click only.

3.17 New in previous release ATG 2.5.2

The STM ATG 2.5.2 is released for StateMate 3.3.1-MR1. Compared to the previous ATG release, it contains no new features, but the following bug fixes:

Profile Update Progress window always in background: fixed

InstallGuide “adjusting cygwin memory limit” (Call 53767)

The adjustment is not necessary any more. This is no mentioned in the Install Guide.

When using the Startup Duration Assumption in test definitions with startup values that are not a factor of the test vector resolution value, then the generated test vectors in ASCII representation contained some additional superfluous lines that may made trouble in post processing the ASCII vectors. This is fixed.

Using the Startup Duration Assumption together with Freezing and Simple Assumptions failed sometimes, when the frozen values were not the STM default values or the simple assumption was not fulfilled in the default initial state. This is fixed.

Statemate models with library components raised failures of the ATG compile step. Fixed.

Data items with type Condition Array or Event Array raised internal simulation errors with the effect that desired test vectors and SCPs could not be generated.

3.18 New in previous release ATG 2.5

The STM ATG 2.5 plugin is released for STM 3.3.1. STM 3.3.1 extends the action language by new else/default trigger. These new constructs are fully supported by ATG 2.5 .

New in the list of action language elements that are supported by ATG 2.5 are the predefined functions trunc() and round(). Models that use these functions do not yield a Red lamp any more, only the usage of trunc and round in properties and assumptions in test definitions is still not possible.

The Statemate simulator and code generator options for the evaluations of truth tables are fully supported now. Like in Statemate, the setting of an “Upon Change” or “Every Step” evaluation of truth tables is selected in the ATG profile.

On the technical side, some improvements have been made. In particular when input freezings are used, the required run-time and memory is reduced in ATG applications. States and transitions that become directly unreachable as consequence of frozen inputs are detected as unreachable by a fast static analysis, i.e. without (expensive) formal verification engines. The generated reports about coverage statistics separate uncovered from unreachable transitions.

The application of generated test vectors in HIL/SIL environments is addressed by a new feature: a new possibility to specify a **Startup Duration Time** has the effect that all generated test vectors start the system under test but then wait with its stimulation (setting inputs) until the systems startup phase is passed. Details on this are in the User Guide (see section 1.3, Testbench Assumption: Startup Duration Time).

3.19 New in previous release ATG 2.4

This release implements several improvements:

Performance and usability: compared to ATG 2.3 the execution run-time is usually up to 40% shorter. This applies in particular for ModelCoverage goals and the usage of freezings for inputs. The progress is visualized in terms of ‘examined coverage goals’ of ‘total coverage goals.’ ModelCoverage tasks can be stopped during execution and continued later. Stopped tests provide full access to all so-far reached results, i.e. test vectors in all representations, coverage information and statistics. Test vectors exports in all representation formats allow now to add a time offset to the absolute starting point of the test. This makes it possible to give the design under test a startup phase before the test vectors are applied.

Coverage Goals/ Coverage Criteria: In addition to the previous release, transition trigger and mini-spec trigger are covered more exhaustive when the new criteria “Extended Trigger Coverage” is selected. This requires more execution time for ATG, but gives you reports that tell you whether the generated test vectors fulfill the MCDC criteria according to DO 178b.

A second new criteria applies the Full Set of the previous test goal plus better sequential analysis of trigger combinations and automatic internal expansion of Boolean truth table Don’t Care starts ‘*’.

Statemate constructs: Data items of type REAL in time-out operators are now supported, when the ATG profile settings specify a minimal time unit. Furthermore, the new possibility in Statemate 3.3 to specify bounds for REALs and default values for REALs and Integer is taken into account in fixed point mappings

in ATG 2.4. If REAL data items are used in timing expressions, the performance of ATG is much better when the lower and upper bounds are already specified in the model.

ATG 2.4 supports the Windows XP platform. Statemate 3.3 is a prerequisite.

3.20 The Previous Release ATG 2.3

ATG 2.3 supports 32 bit integer ranges and fixed point resolutions with 32 bit. This allows a higher precision in the treatment of computations on REALs. Data items of type REAL are now presented in groups, where each group contains data items that are directly related to each other inside the model. Thus, a more fine granular resolution per group allows to use high precision where necessary and a lower resolution to decrease complexity whenever it is possible.

The user interface (GUI) is improved: the data item list in the drive-to-property window can now be filtered like in the Statemate data dictionary browser 'Name pattern:' search. The whole window is resizable now, thus there is more room for the property definition. The available operators that may be used in the property definition can be selected in a new window that can be put aside and used with double clicking.

Of high value is the possibility to copy/past test definitions, when existing test definitions contain large and detailed freezing lists and simple assumptions. Entering them for each test again is not necessary any more.

On a technical level, the 32 bit support has been added (previously there was a 16 bit limitation) without recognizable performance lost. The test search routines got a new heuristic that performs much better when all test goals are reachable, but may use more time if coverage goals can not be reached. An improvement of an internal simulation step brought a speed up of single drive-to-transition or -property tasks of 10%.

4 Known Limitations

The Automatic Test Generation plugin supports the different time models of the Statemate Simulator (i.e., synchronous and asynchronous). However, no support is given to the code-generator "Real Time" time model nor to C or ADA code implementations.

4.1 Unsupported Chart and Action Language Features

Not all definitions that are possible in a Statemate design model are suitable for ATG. For example, combinational assignments are not supported. A detailed list of unsupported features can be found at the end of the User Guide.

When applying Automatic Test Generation to a design (for the first time), a Static Check will be performed as internal preprocess. The check result marks the design model as Red, Orange, Yellow, or Green. Red models contain features that are not supported. The source of the problem is explained to the user in the User Interface. But: like a compiler terminates compilation after it finds an error, it may happen that the static check fails to show *all* sources for Red markings at once.

4.2 Execution Error

An Execution Error will appear in drive-to transitions and property tests, when ATG finds a core test vector on the model in fixed point representation, but the vector can not be lifted to a valid vector on the concrete model with REAL data items.

Run-time overflows may occur in a Statemate simulation. All ATG applications are performed under the assumption that the design model is free of dynamic (range or array boundary) overflows. If they occur in a design model, this may prevent the ATG plugin from generating SCPs.

4.3 SCP do not drive the Statemate simulation as expected

This may happen if a chart is marked by the Static Check as Orange. Due to automatic abstraction from some function or subroutine the analysis or proof may assume effects that are not produced during the simulation. Data items that are affected by abstracted functions or subroutines become input objects that can be restricted with assumptions inside ATG. Thus, user provided assumptions can avoid these failing SCPs. Other solutions may be to change the design model or to provide implementations for subroutines in Action Language instead of user code.

If the model contains write-write races, it can happen that the simulation does not follow the path expected by the generated SCP.

4.4 Executing ATG on a changed Design/Workarea

ATG performs user selected/defined test generations on its own representation of the Statemate design. To this end, there is a compile step from the Statemate design data to a representation suitable for automatic test generation. If a design is modified, or if an ATG profile is checked out from the databank, then a (re-) compilation is necessary. If doing this while keeping old test definitions it may happen that these definitions become invalid. For example, a drive-to-transition becomes invalid if the previously selected source or target state is actually removed from or renamed in the design. Now, when keeping definitions in a re-compile the ATG plugin adapts the definitions – if necessary and possible – automatically. Modified definitions are marked by “* *” which prefix the definition name. The stars “* *” disappear when the user edits and executes the definition again.

Mapping tables, that map data items of type real or (unbounded) integer to finite fixed point representations, are automatically extended if there are more such data items in the re-compiled design than in the old, kept definitions. The newly added data items get a default range from zero to zero. The user has to edit such definitions afterwards to enter appropriate values.

4.5 Simulated Time difference during SCP execution

ATG generates test vectors and in particular SCPs with respect to the time unit and resolution settings in the ATG profile. In cases, where the timing expressions in the model would require a higher resolution than specified in the profile, the SCPs may not run as expected. This is shown by a “time difference ...” message in the simulator in Statemate.

4.6 Bugs

There is a known bug in the compilation step of the Statemate design data into an internal representation: the timing of scheduled actions is not always accurate. There is an erroneous additional delay of one step, that may become visible when defining properties which relate a scheduled to an unscheduled actions.

Simulation control programs generated by ATG do not run, if the selected scope for the ATG run was a basic activity that is implemented by a truth table or mini spec.

5 Testing Performed

Interactive as well as regression tests with main focus on the correctness of results are performed on all platforms.

6 Documentation

There are

- These Release Notes
- Installation Guide

- ATG User Guide
- ATG Tutorial

The documents are accessible via the Help menu of the ATG plugin.

7 Training

Training on applying Automatic Test Generation is available.