MODEL DRIVEN TESTING WITH TIME AUGMENTED MARKOV CHAIN USAGE MODELS

Computations and Test Case Generation Algorithms for Time Augmented Markov Chain Usage Models

Sebastian Siegl, Vladimir Entin, Reinhard German
Department of Computer Science, FAU Erlangen-Nürnberg, Martensstr. 3, 91058, Erlangen, Germany

Gerhard Kiffe
AUDI AG, 85045 Ingolstadt, Germany

Keywords: Software tests, Model based testing, Automated test case generation, Statistical testing, Usage modeling, Testing strategies, Test coverage, Automotive systems, Embedded systems.

Abstract: In statistical testing Markov Chain Usage Models (MCUM) are employed to describe the possible usage of the System-under-test (SUT) and to automatically derive test cases. However, MCUMs, as they are applied today, do not provide a universal concept for the integration of time. The estimated or known delay time of usage states and execution time of actions can only be considered by auxiliary means. But the importance of this aspect grows continously with the integration of increasingly complex and time sensitive systems in the automotive environment, such as hybrid vehicles. To overcome this drawback we present an intuitive extension of MCUMs that allows the consistent integration of time in MCUMs. We discuss how computations for MCUMs can be easily adapted to Time Augmented MCUMs and present algorithms for automatic test case generation from Time Augmented MCUMs together with a case study that has been conducted in the automotive domain.

1 INTRODUCTION

In many application domains, e.g. avionic and automotive systems, software has become the method of choice for innovation. This development is intensified by the customer demand for increasingly safe, comfortable, and environment friendly systems like hybrid or electric vehicles. To satisfy this demand the development of highly complex and distributed systems has become inevitable. These systems pose new challenges to testing. The approach presented in this paper is based on model based testing (Dalal et al., 1999), and can be further categorized into usage model based black box testing (Beyer and Dulz, 2005). Nowadays the focus in usage oriented testing based on Markov chain usage models (MCUM) lies in testing various sampled test cases and in testing that comes close to a known or estimated usage profile (Prowell, 2004). A broad spectrum of different types of algorithms for the generation of test cases exists (Siegl, 2008). Only the variance in actions is currently considered in a consistent manner, however. Time and especially the variance in usage time, i.e. the time between actions and the duration of actions themselves, is not integrated coherently. However, the timing of different actions has an important impact on the behavior of a system and thus on the ability of test cases to discover failures. Timing in this context stands for the duration of the execution of stimuli and the duration of usage states, i.e. the time between the execution of stimuli. In this contribution we present a temporal extension to classical MCUMs that offers the following advantages:

- A realistic, usage time considering, representation of system usage
- Integration of usage timing into computations
- Integration of variance in time of usage into test case generation

Our enhancement is based on the concept of semi-Markov processes as they are used in the performance analysis of systems with non-exponential tim-

Siegl S., Entin V., German R. and Kiffe G. (2009).
MODEL DRIVEN TESTING WITH TIME AUGMENTED MARKOV CHAIN USAGE MODELS - Computations and Test Case Generation Algorithms for Time Augmented Markov Chain Usage Models.
In *Proceedings of the 4th International Conference on Software and Data Technologies*, pages 202-207
DOI: 10.5220/0002255902020207

ings (German, 2000). They are applied in the field of system-modeling and analysis. In the context of usage modeling no application of this concept is known to the authors.

The paper is structured as follows. In the next section model driven testing based on Markov chain usage models is briefly introduced, followed by a presentation of the testing requirements in the automotive domain. Next we present the extension, calculations that are possible with this extension and algorithms for an automatic generation of test cases from our enhanced MCUMs that we call *Time Augmented Markov chain usage models*. Before the conclusion and outlook, we present a case study that has been conducted in the automotive domain.

2 TEST CASE GENERATION FROM MARKOV CHAIN USAGE MODELS

A MCUM is usually described by a directed graph that consists of (Prowell, 2000):

- A set of *states* $S = \{s_1, \ldots, s_n\}$, that represent possible states of usage.
- A set of *arcs* A, representing state transitions. An arc from state i to state j is denoted by a_{ij}, multiple arcs between s_i and s_j are not allowed. A stimulus on the SUT is assigned to each arc.
- The *transition probability* from state i to state j, denoted by $p_{i,j}$ for an existing arc a_{ij}. Otherwise the transition probability $p_{i,j} = 0$. The transition probabilities obey the conditions $0 \leq p_{i,j} \leq 1$ and

$$\sum_{j=1}^{n} p_{i,j} = 1 \quad \forall i = 1, \ldots, n \qquad (1)$$

  that states that the probabilities of all outgoing arcs from a certain state s_i must sum up to one.

Two states have special characteristics, that are s_1, which is the sole initial state (also: start state) and s_n, which is the final state (also: end state). The values $p_{i,j}$ can be stored in a *state-transition-matrix P*. Element describes p_{ij} the probability of the transition from state s_i to state s_j.

The representation of a MCUM as directed graph is depicted in Figure 1.

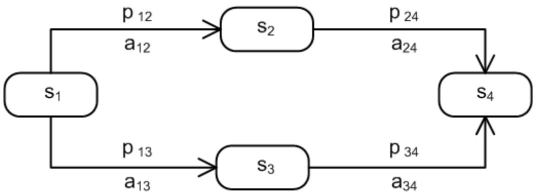

Figure 1: MCUM representation as directed graph.

Each path from the start to the final state is a valid test case. Test cases can be generated straightforward by random walk starting from the initial state, this way obtaining statistical sampled test cases. Different usage profiles, representing e.g. different types of users, can be applied to direct the statistical derivation of test cases. Therefore, distinct user characteristics or testing targets can be considered during test case generation.

For a usage model based on a Markov chain many computations can be performed that provide valuable information for test management (cp. section 4.2).

3 TEST REQUIREMENTS AND CONSTRAINTS IN THE AUTOMOTIVE DOMAIN

Test requirements and constraints in the automotive domain are composed of the

- SUT test requirements
- test automation
- test bench

In the following, the test automation *Extended Automation Method (EXAM)*, applied within the Volkswagen AG, and the test benches employed in the context of this paper are briefly introduced.

EXAM Testing Process. Test automation in the scope of EXAM means the automated generation of platform dependent code and the execution of the derived test suite without human interaction. The majority of test cases is automatically executed on Hardware-in-the-loop (HIL) systems, that are introduced in the next paragraph. However, it is hardly possible to systematically derive test suites that meet the requirements in the automotive domain in an optimal manner.

HIL Testing. Hardware-in-the-loop (HIL) simulation provides a platform for testing embedded systems composed of Electronic Control Units (ECUs) under conditions so that the ECUs *feel* like being in a real car. This is achieved by including the ECUs in a simulated environment.

The main issue is currently how to use the HIL test platform in an efficient manner, since testing time on HIL systems is expensive.

Challenges in Automated Test Case Generation. As test cases until now are created without systematically considerung time and timing constraints the decision which test cases to create and to execute within time constraints remained in the hands of the test designer. Thus, this decision has not been done systematically on the basis of an unambiguous model. One drawback of the existing procedure is that valuable testing time may not be used in an optimal manner.

Strategies have been proposed in (Siegl, 2008) to derive test cases from MCUMs under consideration of time. However, the existing methods for test case generation based on MCUMs lack the systematic integration of usage time in the MCUMs themselves, since they are discrete time Markov chains and each state transition corresponds to one time step on an abstract time base. Until now concepts are known to integrate timing aspects provisionally (Beyer and Dulz, 2005), but not coherently and in a manner that the usage time is integrated by keeping the semantics of the MCUM. Therefore, we present in this paper an extension of MCUMs that keeps the semantic and that supports automated test case generation for embedded systems in test environments as they are established in the automotive industry.

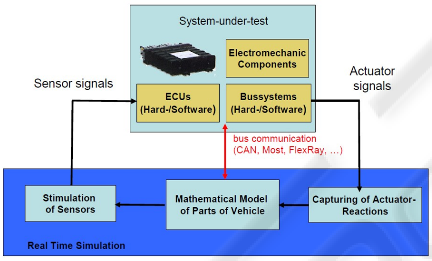

Figure 2: Hardware in the loop simulation.

4 TIME AUGMENTED MARKOV CHAIN USAGE MODELS

This section is structured as follows: First of all the enhancement of MCUMs is described. Next *Markov Computations* are introduced. These can be made before and after test case execution and provide valuable test management information, e.g. the estimated number of test cases required to achieve a coverage level. The enhancement of standard algorithms for test case generation is described in the following paragraph and finally the issue of coverage criteria for Time Augmented MCUMs is presented.

4.1 Enhancement

The enhancement of MCUMs is structured as follows:

- A probability density function (pdf) of the residence time is assigned to each state s_i. We denote it as ts_i.
- A pdf of the stimulus time $ta_{i,j}$ is assigned to each arc a_{ij}. This pdf describes the duration of the execution of a stimulus on the SUT.

Besides, the transition probabilities $p_{i,j}$ as well as the values ts_i and $ta_{i,j}$ can be stored and exchanged by means of a matrix P. The single values $\forall p_{ij}$ are replaced by the two elements $p_{i,j}$ and $ta_{i,j}$. An additional row or column that extends the matrix by elements p_{in+1} or p_{n+1j} is used to store the pdf ts_i assigned to the usage state s_i. This information could also be stored in a one-dimensional vector related to a matrix P. The extensions introduced by our approach are presented in Figure 3.

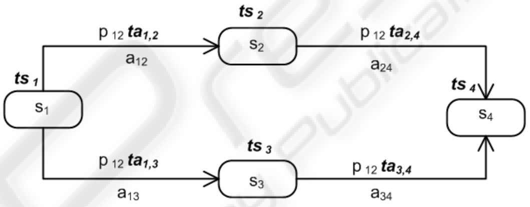

Figure 3: MCUM extended by usage time pdfs.

The pdfs of time we used for this approach are:

- Deterministic time as a step function.
- Uniform distribution (discrete uniform distribution)
- Gaussian distribution (binomial distribution)
- Exponential distribution (geometric distribution)

In brackets the discrete approximation of the continuous functions is given.

By assigning these distributions it is possible to describe usage that

- is always exactly predictable in time.
- is characterizable that it is equally likely within time boundaries.
- happens typically after a certain time. The standard deviation is known.
- whose occurrence probability does not change over time.

Therefore not only the variance of stimuli themselves, but also the variance of the timing, i.e. the execution time of stimuli and the time between them can be described by a MCUM. There is one issue that has to be considered when creating such a model. Multiple arcs a_ij, that have their origin in the same state s_i, may be assigned the same stimulus. In this case the range of values of the pdfs that describe the duration of the arc traversal must be disjoint.

4.2 Computations

Markov Computations for MCUMs are (Prowell, 2000):

- Expected number of occurrences of a state in a test case
- Long-run state probabilities
- Long-run arc probabilities
- State occurrence probability
- Arc occurrence probability
- Expected length of a test case

Up to now these computations have produced abstract and relative numbers with respect to the other elements of the MCUM, i.e. describing characteristics of the model. An example is the long-run state probability, that is computed for each state in the following way: The expected number of occurrences of the considered state is divided by the number of occurrences of all states. This is quite an abstract measure that does not represent the testing time spent in each state. With our extension this computation becomes more valuable, because we can take the expectation value of each ts_i and determine in this manner the fraction of time spent in each state during testing. This result might differ considerably from the relative occurrence rates and can thus be a better indicator for the decision making which test cases should be executed. Also, the other computations gain importance, e.g. the number of occurrences of arc and state weighted by the pdfs of time give an important indicator for test planning. To complete the list, the following computations are now possible with Time Augmented MCUMs:

- mean time consumed in transitions
- mean time consumed by execution of each stimulus
- Expected state residence time in a test case
- Long-run state mean residence time
- Long-run arc mean execution time

4.3 Algorithms

The existing algorithms for the generation of test cases can still be used, however, they do not make use of the timing information (Prowell, 2004). For this reason we propose new algorithms that incorporate the knowledge of time for test generation strategies. The variance in time is considered during test generation and can this way also be found in the test cases. Furthermore, the estimated execution time for each generated test case is automatically available, which is an important information for the test management.

Random Walk. Random walk is the standard sampling technique to derive test cases from MCUMs. In the following algorithms it is assumed that a certain usage profile is given from which the values t_{sc}, $p_{c,j}$, and $ta_{c,j}$ are obtained. These values describe the duration of a state t_{sc} or of the execution of a stimulus t_{cj} in the concrete test case that is generated. The extended random walk for Time Augmented MCUMs is described in Figure 4.

```
Data:
Time Augmented MCUM;
$s_c$ state under consideration;
Initialization:
$s_c \leftarrow s_1$;
while $s_n$ not reached do
    Get $t_{sc}$ by sampling $ts_c$;
    Get $a_{cj}$ randomly from all $p_{c,j}$;
    Get $t_{cj}$ by sampling $ta_{c,j}$;
    $s_c \leftarrow s_j$;
end
```

Figure 4: Random walk for Time Augmented MCUMs.

Most Probable Test Case. Algorithms to derive the most probable test case usually choose the most probable arc at each state. For the timing informations assigned to states and arcs we assume that the expectation value (in the following also: *mean*) of the pdf for the time, denoted as $E[pdf]$ represents the most probable value for the time.

```
Data:
Time Augmented MCUM;
$s_c$ state under consideration;
Initialization:
$s_c \leftarrow s_1$;
while $s_n$ not reached do
    $t_{sc} \leftarrow E[ts_c]$;
    $a_{cj} \leftarrow a_{cj}$ with max $p_{c,j}$;
    $t_{cj} \leftarrow E[ta_{c,j}]$;
    $s_c \leftarrow s_j$;
end
```

Figure 5: Most probable test case from Time Augmented MCUMs.

Note that the determination of the next arc is left untouched, i.e. the arc with the highest probability. The time that is determined for each arc and state is the expectation value for the pdf that is assigned to the arc and state respectively.

Minimal Arc Coverage. Minimal arc coverage algorithms can be used without modification. However, the timing information has to be applied depending on the test strategy that one follows. If testing time is short, one could take the minimal limiting value of the pdfs.

Boundary Value Testing. Boundary value testing is supported by the time information incorporated in the Time Augmented MCUM. The following four cases have to be considered:

Deterministic Distribution. In this case there is no choice of time.

Uniform Distribution. Also in this case the determination of boundary values is straightforward, as they are given by the limits of the interval.

Gaussian Distribution. In these cases the range of values is potentially infinite. A reasonable value for boundary value testing has to be determined. If one prefers to have a unique definition of the boundary values the values at $\sigma_r = \mu[pdf] \pm 3\sigma[pdf]$ can be taken. These values encompass 99.7% of the whole set.

Exponential Distribution. In the case of the exponential distribution, a minimal value is given by the point of origin. As a maximal boundary value we suggest to use $E[pdf] = \mu[pdf]$.

4.4 Coverage Criteria

Graph based coverage criteria rely on the evaluation of how the test paths belonging to the test cases cover the graph abstraction (Hierons et al., 2008, p. 119). This concept is hardly transferable to an infinite domain like time, so a coverage criterion considering timing aspects has to be defined. These definitions are based on the assumptions described in section 4.3. These definitions, of course, can only give a general idea for orientation, since the significance depends also on the special characteristics of the system under test (SUT). That is to say that for testing a low level non functional property such definitions can be easier applied than for testing a high level non functional property. To give an example *hybrid vehicle operating strategies* shall be stated, where beside the *operating strategy* (Wallentowitz and Reif, 2006) itself the potential timing in usage of various participants has to be considered. A general definition of coverage concerning the timing aspects is hard to give and coverage criteria dependent on characteristics of the tested functionality (e.g. its safety criticality) is part of our future work. Some basic definitions shall be given:

Deterministic Distribution. In this case there is no range so only the specified value has to be considered for coverage under a usage point of view.

Uniform Distribution. The minimal, mean, and maximal value could be good choices for minimal coverage testing.

Gaussian Distribution. Mathematical indicators, such as the included fraction of possible realizations of the value domain, or requirements, can be a sound basis.

Exponential Distribution. The minimal or expectation value $E[pdf] = \mu[pdf]$ can be sufficient, strongly dependent on the functionality.

5 CASE STUDY

For the evaluation of our approach, a test issue in the automotive domain has been chosen. It concerns an aspect of the operation of the Electronic Stability Program (ESP) switch in modern cars. Different functionalities are invoked depending on the state of the car and the pressing duration of the switch (AUDI AG, 2005). Some of these functionalities are directly desired by the user and some are assumed to represent the reaction desired by the user.

In some cars, the ESP switch was placed in the center console, where car passengers often deposit their hand bag. This led to frequent misactivations of this switch and as a consequence to undesired deactivation and activation operations of the ESP functionality. Due to this problem the *handbag switching mechanism* is now integrated in the cars. This mechanism activates ESP irrevocable for the whole journey if the ESP Switch is pressed longer than a certain time, thus preventing the undesired deactivation of ESP if an object is placed on top of the ESP switch.

In Figure 6 the model is presented that describes the different activation durations of the ESP switch.

A subset of the computations in the classical way is shown in table 1.

The probability is the same, that is 0.03844, to stay during a test case in the start state (*S9998*), final state (*S9999*), and in the state in which the ignition has been turned on (*S1*). The probability, however, that the test case is in the state (*S234*) where the car drives at 15kmph and the ESP switch has been pressed for less than 3 seconds is only 0.01443.

Table 1: Computations for Case Study ESP Switch.

State ID	Stationary distrib.	Visiting probab.	Expected number of visits
S9998	0.03844	1.00000	1.00000
S1	0.03844	1.00000	1.00000
S234	0.01443	0.22232	0.37538
S4	0.07687	1.00000	2.00000
S9999	0.03844	1.00000	1.00000

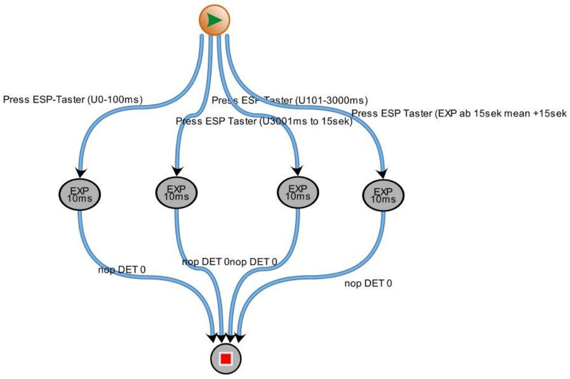

Figure 6: Sub-Usage scenario for ESP push button.

Table 2: Computations for Case Study ESP Switch with Time Augmented MCUM.

State ID	Mean residence time	Expected number of visits	Expected residence time
S9998	0.00000	1.00000	0.00000
S1	9.00000	1.00000	9.00000
S234	10.0000	0.37538	3.75380
S4	0.01000	2.00000	0.02000
S9999	0.00000	1.00000	0.00000

Now have a look at the computations for a Time Augmented MCUM, depicted in Table 2. All measures for the time are given in seconds. Now we can see that for the given usage profile we would expect to be in state *S1* 9 seconds in a typical test case and 3.75 seconds in state *S234*. Next we see that states *S9998*, *S9999* and *S4* have less impact on the testing time, though they are visited once or twice in each test case. This information is valuable for the person engaged to decide when to test and what to test and to assess, if the usage profile is suitable to derive test cases that meet the testing goals within time limits.

6 FUTURE WORK

The future work will focus on computations for the test planing, integrating coverage aspects, and test case generation strategies. The elaboration of timing coverage criteria for functional requirements is necessary. So test case generation strategies are developed that should optimize the usage of the test benches, considerung functional-based timing requirements.

7 CONCLUSIONS

In this paper an extension has been presented and evaluated that integrates usage time in MCUMs. The case study has been conducted with an example from the automotive domain. Substantial advantages arised from the application of our extension, since the integrated time information can be used either for the test planing or for the test case generation. In the first case indicators such as test case length are not any more abstract numbers, but provides the test designer with information about the expected duration of a test case. Secondly this information can be used by the test case generation algorithms for one thing to generate test cases that are also variant in time and not only in operations. For another thing test case generation algorithms can make use of this information to generate test suites that meet test requirements within time constraints in an optimal manner. In this way the whole testing process can be supported by the presented approach.

REFERENCES

AUDI AG (2005). *Spezifikation ESP Anzeige- und Warnkonzept*. 85057 Ingolstadt, Germany.

Beyer, M. and Dulz, W. (2005). Scenario-based statistical testing of quality of service requirements. In *Springer LNCS 3466, International Dagstuhl Workshop on "Scenarios: Model, Transformations and Tools"*, pages 152–173.

Dalal, S. R., Jain, A., Karunanithi, N., Leaton, J. M., Lott, C. M., Patton, G. C., and Horowitz, B. M. (1999). Model-based testing in practice. In *ICSE '99: Proceedings of the 21st international conference on Software engineering*, pages 285–294, New York, NY, USA. ACM.

German, R. (2000). *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. John Wiley & Sons, Inc., New York, NY, USA.

Hierons, R. M., Bowen, J. P., Harman, M., Hierons, R. M., Bowen, J. P., and Harman, M. (2008). *Formal Methods and Testing: An Outcome of the FORTEST Network. Revised Selected Papers (Lecture Notes in Computer Science)*. Springer Publishing Company, Incorporated.

Prowell, S. J. (2000). Computations for markov chain usage models. Technical report, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, USA. UT-CS-03-505.

Prowell, S. J. (2004). State of the art of model-based testing with markov chain usage models. Dagstuhl Seminar 04731, Perspectives on Model Based Testing.

Siegl, S. (2008). Evaluation of Model-Driven Test Case Generation based on Markov-Chain Usage Models in the Automotive Domain. Master's thesis, Department of Computer Science, University of Erlangen-Nuremberg, Germany.

Wallentowitz, H. and Reif, K., editors (2006). *Handbuch Kraftfahrzeugelektronik: Grundlagen, Komponenten, Systeme, Anwendungen*. Vieweg, Wiesbaden.