Combining Time and Concurrency in Model-Based Statistical Testing of Embedded Real-Time Systems

Daniel Homm, Jürgen Eckert, and Reinhard German

Department of Computer Science 7
University Erlangen-Nuremberg
Martensstr. 3, 91058 Erlangen, Germany
{daniel.homm, juergen.eckert, reinhard.german}@fau.de

Abstract. Timed usage models (TUMs) represent a model-based statistical approach for system testing of real-time embedded systems. They enable an automatic test case generation and the calculation of parameters that aid the test process. However, a classical TUM only supports sequential uses of the system under test (SUT). It is not capable of dealing with concurrency, which is required for state of the art real-time embedded systems. Therefore, we introduce TUMs with parallel regions. They also allow automatic test case generation, which is carried out similarly to classical TUMs. But, the semi-Markov process (SMP) that is usually used for analysis is not suitable here. We apply Markov renewal theory and define an SMP with parallel regions, which is used to calculate parameters. We validated our analytical approach by simulations.

Keywords: System testing, timed usage models, concurrency

1 Introduction

Timed usage models (TUMs) are applied in the field of system testing for model-based statistical testing of real-time embedded systems [10,12]. A TUM represents an extension of a Markov chain usage model (MCUM) [13] by time aspects. It specifies possible uses of the system under test (SUT) in a notation similar to state machines. But, it also considers time dependencies related to the use of the SUT. TUMs enable an automatic generation of test cases [12] and the calculation of parameters that aid the test process [11], e.g., they help to decide when to stop testing [9]. Test case generation is achieved by a random walk through the model. Parameters are calculated by mapping the TUM to a semi-Markov process (SMP), which is analyzed subsequently.

Modeling uses in a TUM is restricted to sequences of stimuli (events, interrupts, etc.). Concurrent streams of use are not supported. However, the rising complexity of embedded systems leads to an increase of concurrent aspects. For example, customer requirements in the automotive domain enforce the development of more sophisticated systems [3]. Neglecting concurrent uses during the test of the SUT increases the risk of failures being undetected prior to the release.

In this paper we extend the concept of TUMs by parallel regions. Thus, it is possible to handle both, sequential and concurrent uses of the SUT, with their respective time dependencies and the model stays similar to state machines. Test cases still can be generated automatically from the model by a random walk. But the calculation of parameters becomes a non-trivial task: the SMP used to analyze a TUM assumes a sequential execution of the process in time, which is no longer given due to the introduction of concurrency. Therefore, we apply Markov renewal theory and define an SMP with parallel regions. The calculation of parameters for a TUM with parallel regions is carried out by an analysis of a respective SMP with parallel regions.

2 Related Work

Research in the area of model-based testing already tackled the issue of combining time and concurrency aspects of the SUT [2]. There are also commercial tools available, e.g., TPT[1]. However, the results are not transferable to model-based statistical testing.

Model-based statistical testing aims at testing the expected use of the SUT. Test cases are sampled and executed from the set of possible test cases. Statistical methods are used to draw inferences. This has to be considered by concepts that extend the methodology. To the best knowledge of the authors, only one related work deals with the combination of time and concurrency in the field of model-based statistical testing: usage nets [1]. A usage net is a Discrete Deterministic and Stochastic Petri net with colored transitions. Prior to test case generation or parameter calculation, it is transformed into a respective MCUM. Any time dependencies are thereby discretized, i.e., a single state or transition in a usage net is represented by a set of states and transitions in the underlying usage model. Deploying usage nets in practice may cause increased efforts for training, as the notation for most model-based approaches in the field of system testing is similar to state machines [4]. Additionally, the discretization of time dependencies increases the effort that is required to carry out the analysis, as it provokes a state space explosion in the underlying MCUM.

Our approach is based on a notation similar to state machines. This reduces the effort to deploy it in practice. Test case generation is directly carried out on a TUM with parallel regions. For the analysis, discretization of time dependencies is avoided. Instead, we define and use an SMP with parallel regions: a stochastic process suitable as foundation for the calculation of parameters from a TUM with parallel regions.

3 Semi-Markov Processes with Parallel Regions

We define an SMP with parallel regions as natural extension of an SMP [6] by composite states. It is represented by a state machine with simple and composite

[1] Time Partition Testing: Systematic automated testing of embedded systems, http://www.piketec.com, accessed on May 22, 2015

states $s_i \in S$ at the top level. The stochastic matrix $\mathbf{P} = [p_{i,j}]$ holds the branching probabilities after leaving a state at the top level. The sojourn time of each state s_i is given by the random variable X_i with $F_i(t)$ as distribution, and the steady-state probability is denoted by the respective entry in vector $\boldsymbol{\pi} = [\pi_i]$. Different sojourn times can be configured for each state by attaching a respective distribution. However, a composite state is not annotated with a distribution.

A composite state s_i holds r_i parallel regions $s_{i,j}$. Each region owns a state machine with a final state. A sub-state k of region j of composite state s_i is referred to as $s_{i,j,k}$. Besides final states, each sub-state has a distribution annotated. If a composite state is entered, then each region will enter its initial state. A composite state will be left, if all regions have reached their final state. There are no further synchronizations between the regions. An exemplary SMP with parallel regions is depicted in Figure 1. Note that, we call a final state an absorbing state and refer to it as $s_{i,j,m}$ with $m = |s_{i,j}|$ ($s_{i,j}$ can also be interpreted as set of its sub-states).

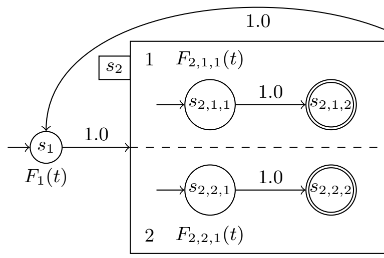

Fig. 1: Exemplary SMP with parallel regions.

3.1 Analysis

The analysis focuses on the calculation of the steady-state probabilities. They represent the fraction of time spent in each state in the long run and are used to calculate further parameters (see Section 3.2).

Prior to analysis it is necessary to define regeneration points, i.e., time instants when the process is memoryless [6]. We set regeneration points upon leaving a state at top level. This yields an embedded Markov chain (EMC) that consist of top level states only. A solution to the steady-state probabilities in the EMC can be obtained by solving the linear system of equations $\mathbf{u} = \mathbf{uP}$ with the normalization condition $\mathbf{ue} = 1$ [6]. For the final solution at top level it is necessary to consider the mean sojourn times for each EMC state. Therefore, we define matrix $\mathbf{C} = [c_{i,j}]$.

$$c_{i,j} = \begin{cases} E[X_i] & \text{, if } i = j \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

For simple states, the mean sojourn time is derived by $E[X_i] = \int\limits_0^\infty \overline{F}_i(t)dt$. For composite states, it is derived by a transient analysis of the regions, which is stated below. Finally, the solution to the steady-state probabilities with regard to the individual mean sojourn times is obtained by solving

$$\boldsymbol{\pi} = \frac{\mathbf{uC}}{\mathbf{uCe}}. \tag{2}$$

The mean sojourn time in a composite state depends on the time spent in each region in isolation until absorption. Its calculation requires the definition of some parameters:

– the mean sojourn time in a region $s_{i,j}$ is given by the random variable $X_{i,j}$ with distribution $\sigma_{i,j}(t)$
– the time spent in a region $s_{i,j}$ depends on the sojourn times of its sub-states $s_{i,j,k}$, which is given by the random variable $X_{i,j,k}$ with distribution $F_{i,j,k}(t)$
– the vector $\mathbf{F}_{i,j}(t)$ holds the distributions of the sojourn time in each state in region $s_{i,j}$
– the vector $\mathbf{f}_{i,j}(t)$ holds the respective density function for each state in $s_{i,j}$
– the branching probabilities after leaving state k and entering state l in region $s_{i,j}$ are given by the stochastic matrix $\mathbf{\Delta}_{i,j} = [\delta_{i,j,k,l}]$
– the steady-state probabilities for sub-states are given by vector $\boldsymbol{\pi}_{i,j} = [\pi_{i,j,k}]$
– the transient probabilities of states in region $s_{i,j}$ in isolation are given by the vector $\mathbf{v}_{i,j}(t) = [v_{i,j,k}(t)]$, starting at $t = 0$
– the matrix $\mathbf{V}_{i,j}(t) = [v_{i,j,k,l}(t)]$ holds the conditional transient probabilities in isolation, i.e., the probability that region $s_{i,j}$ is in state l at time t given it was in state k at time 0
– the mean sojourn times in states of region $s_{i,j}$ are given by vector $\boldsymbol{\sigma}_{i,j} = [\sigma_{i,j,k}]$
– the matrix $\mathbf{\Sigma}_{i,j} = [\sigma_{i,j,k,l}]$ holds the conditional mean sojourn times in $s_{i,j}$

Note that, we denote vectors and matrices that are restricted to non-absorbing states with a bar $\overline{\circ}$, e.g., the vector $\overline{\sigma}_{i,j}$ denotes the mean sojourn times for all states in region $s_{i,j}$ except for its final state. This does not affect distributions, i.e., the complement of a distribution is still given as $\overline{F}(t) = 1 - F(t)$.

For the calculation of the mean sojourn time spent in a composite state we consider three stochastic processes: Continuous Time Markov Chains (CTMCs) with exponentially distributed sojourn times, Discrete Time Markov Chains (DTMCs) with geometrically distributed sojourn times with time step τ, and SMPs with sojourn times described by general distributions.

First, the mean sojourn time is calculated for all non-absorbing states. In case composite state s_i contains only one region, i.e., $r_i = 1$, the mean sojourn times in non-absorbing states is determined by solving only a linear system of equations. For a CTMC, it is given by solving $-\overline{\mathbf{v}}_{i,j}(0) = \overline{\boldsymbol{\sigma}}_{i,j}\overline{\mathbf{Q}}_{i,j}$, with $\overline{\mathbf{Q}}_{i,j}$ as generator matrix. For a DTMC, the equation

$$-\overline{\mathbf{v}}_{i,j}(0) = \overline{\boldsymbol{\sigma}}_{i,j}(\overline{\mathbf{P}}_{i,j} - \mathbf{I}) \tag{3}$$

has to be solved, with $\overline{\mathbf{P}}_{i,j}$ as transition matrix. In case of an SMP, it is necessary to derive the conditional mean sojourn times $\overline{\mathbf{\Sigma}}_{i,j}$ first. The mean sojourn times $\overline{\boldsymbol{\sigma}}_{i,j}$ are calculated subsequently.

$$\overline{\mathbf{\Sigma}}_{i,j} = diag\left(\int_0^\infty \overline{\mathbf{F}}_{i,j}(t)\right) + \overline{\mathbf{\Delta}}_{i,j}\overline{\mathbf{\Sigma}}_{i,j} \tag{4}$$

$$\overline{\boldsymbol{\sigma}}_{i,j} = \overline{\mathbf{v}}_{i,j}(0)\overline{\mathbf{\Sigma}}_{i,j} \tag{5}$$

In case a composite state s_i contains multiple parallel regions, i.e., $r_i > 1$, a transient analysis has to be performed for each region separately. If a region is described by a CTMC, the transient analysis is concerned with solving the transient probabilities $\overline{\mathbf{v}}_{i,j}(t) = \overline{\mathbf{v}}_{i,j}(0)e^{\overline{\mathbf{Q}}_{i,j}t}$ of that region. If a region is described by a DTMC, the transient analysis has to consider the branching probabilities in the calculation of the transient probabilities $\overline{\mathbf{v}}_{i,j}(k\tau) = \overline{\mathbf{v}}_{i,j}(0)\overline{\mathbf{\Delta}}^k_{i,j}$. In case a region is described by an SMP, its transient probabilities $\overline{\mathbf{v}}_{i,j}(t)$ are retrieved based on Markov renewal theory [6] by calculating the conditional transient probabilities $\overline{\mathbf{V}}_{i,j}(t)$ first, with global kernel matrix $\overline{\mathbf{E}}_{i,j}(t) = diag(\overline{\mathbf{F}}_{i,j}(t))$, local kernel $\mathbf{K'}_{i,j}(t) = diag(\overline{\mathbf{f}}_{i,j}(t))\overline{\mathbf{\Delta}}_{i,j}$ and $*$ as convolution operation.

$$\overline{\mathbf{V}}_{i,j}(t) = \overline{\mathbf{E}}_{i,j}(t) + \mathbf{K'}_{i,j}(t) * \overline{\mathbf{V}}_{i,j}(t) \tag{6}$$

$$\overline{\mathbf{v}}_{i,j}(t) = \overline{\mathbf{v}}_{i,j}(0)\overline{\mathbf{V}}_{i,j}(t) \tag{7}$$

The effort required to calculate the transient probabilities in case of an SMP is given by $O(|s_{i,j}|^2)$. It can be reduced by using supplementary variables [6]. Note that, in call cases (CTMC, DTMC, SMP) it is possible to derive a closed form solution if the topology is acyclic.

The mean sojourn times $\overline{\boldsymbol{\sigma}}_{i,j}$ spent in non-absorbing states in region $s_{i,j}$ can be derived directly from the transient probabilities.

$$\overline{\boldsymbol{\sigma}}_{i,j} = \int_0^\infty \overline{\mathbf{v}}_{i,j}(t)dt \tag{8}$$

The complement of the sum over all transient distributions for non-absorbing states yields the distribution $G_{i,j}(t)$, which specifies the time to absorption in isolation within region $s_{i,j}$.

$$G_{i,j}(t) = 1 - \overline{\mathbf{v}}_{i,j}(t)\mathbf{e} \tag{9}$$

Due to the synchronization upon leaving a composite state s_i, its mean sojourn time $E[X_i]$ is given as maximum over the distributions that specify the time to absorption in isolation for each region.

$$E[X_i] = \int_0^\infty 1 - \prod_{j=1}^{r_i} G_{i,j}(t)dt \tag{10}$$

The value derived for $E[X_i]$ is used as entry $c_{i,i}$ in matrix $\mathbf{C}$ from (2).

With the solution to (2), the steady-state probability can be calculated for any sub-state at composite state level. It represents a fraction of the steady-state probability for the composite state and depends on the mean sojourn time spent in the sub-state with regard to the mean sojourn time in the composite state:

$$\boldsymbol{\pi}_{i,j} = \pi_i \frac{\boldsymbol{\sigma}_{i,j}}{c_{i,i}}. \tag{11}$$

Note that, the mean sojourn time for the final state of a region is given as $\sigma_{i,j,m} = E[X_i] - \overline{\boldsymbol{\sigma}}_{i,j}\mathbf{e}$, the difference between the mean sojourn time spent in the composite state s_i and the sum of mean sojourn times spent in non-absorbing states in region $s_{i,j}$.

3.2 Calculation of Parameters

The SMP can be used to calculate parameters related to durations. For example, the mean time $E[d]$ until the final state of the SMP is reached. It is calculate as the average waiting time W by means of Little's Law $L = \lambda W$ [7]. As long-term average number of customers L in our process we use the fraction of time spent in non-absorbing states of the SMP in the long run, which is given by $1 - \pi_n$ with π_n as steady-state probability of the final state at top level. The long-term arrival rate is given by $\lambda = \frac{\pi_n}{E[X_n]}$.

$$E[d] = W = \frac{L}{\lambda} = \frac{1 - \pi_n}{\pi_n} E[X_n] \tag{12}$$

The introduction of parallel regions to an SMP enables the calculation of two new parameters. The mean active time within a region $s_{i,j}$ of a composite state s_i can be derived as $E[a_{i,j}] = \overline{\boldsymbol{\sigma}}_{i,j}\mathbf{e}$, the sum over the mean sojourn times of its non-absorbing states. The mean sojourn time $\sigma_{i,j,m}$ that is spent in the final state of a region $s_{i,j}$ can be interpreted as mean waiting time $E[w_{i,j}]$ within that region.

4 Timed Usage Models with Parallel Regions

A TUM with parallel regions consists of states $s_i \in S$ and transitions $a_{i,j} \in A$ that originate in state s_i and have state s_j as target. A state represents the externally visible state of the SUT while it is used and is either a simple or a composite state. Each simple state has a distribution $F_i(t)$ annotated that describes the sojourn time. Each composite state contains parallel regions, i.e., it specifies concurrent streams of use. No distribution is added to a composite state, as its sojourn time depends on the evolution within its regions. Each region specifies again a TUM that may utilize composite states with parallel regions. This allows to nest concurrent streams of use. Final states are used to terminate a region. A transition specifies a stimulus y, i.e., an input that is applied to the

SUT. Additionally, each transition consists of a probability $p_{i,j}$ and a distribution $F_{i,j}(t)$ describing the sojourn time. An example TUM with parallel regions is depicted in Figure 2a. It is explained in more detail in Section 5.

Test cases can be derived automatically from a TUM with parallel regions. The process is straightforward: A random walk is applied, beginning at the start state and ending at the final state at top level. If a simple state is reached, a sojourn time will be sampled from its distribution. Subsequently, an outgoing transition is selected with regard to the branching probabilities together with a respective firing time from the annotated distribution. If a composite state is encountered, a path is sampled from each of its regions analog to the top level.

Parameters can be calculated prior to any test case generation. Therefore, it is required to map the model to an SMP with parallel regions.

4.1 Mapping to an SMP with Parallel Regions

The mapping represents a natural extension of the mapping known for classical TUMs. Each simple state s_i is mapped to a simple state s'_i in the SMP. The distribution $F_i(t)$ is added to state s'_i. Each composite state s_i is mapped to a composite state s'_i in the SMP. No distribution is attached to composite state s'_i, as there is none available at the original state s_i.

Each transition $a_{i,j}$ is mapped as follows: First, a new state s'_{ij} is introduced in the SMP with distribution $F_{i,j}(t)$. A transition $a'_{i,ij}$ is created with transition probability equal to $p_{i,j}$. Afterwards, a new transition $a'_{ij,j}$ with probability 1 is created, it leads from state s'_{ij} to state s'_j. Note that, it is not required to consider stimuli in the mapping. They are not required for the analysis.

Finally, the parallel regions of each composite state s_i are mapped to corresponding parallel regions of composite state s'_i in the SMP. The same mapping rules apply to the contents of each region as for the top level of the TUM.

4.2 Calculation of Parameters

The analysis of the SMP underlying a TUM with parallel regions yields the steady-state probabilities and the mean sojourn times (see Section 3). They can aid the test process. The mean active time $E[a_{i,j}]$ and mean waiting time $E[w_{i,j}]$ of a region $s_{i,j}$ in a TUM, under the condition that the respective composite state is entered, and the mean test case duration $E[d]$ are direct results of the analysis of the underlying SMP. Note that, the mean sojourn time of the final state is not included in the test case duration.

The expected number of occurrences of a state in a test case, given one starts in the start state of the usage model, is no direct result of the SMP analysis. In order to calculate it, we make use of its definition: For a state s_i in a DTMC with final state s_n the expected number of occurrences $E[n_{1,i}]$ within a test case, given one starts in the start state s_1 of the usage model, is obtained via (13) [8]. It holds for DTMCs, as the same fixed time step τ is spent in each state upon entry and the final state occurs exactly once within a test case.

$$E[n_{1,i}] = \frac{\pi_i}{\pi_n} \tag{13}$$

We apply the underlying idea to an SMP with parallel regions. All attached times are ignored and the whole process is considered as DTMC. This also applies to regions of composite states. The expected number of occurrences $E[n_{1,i}]$ for any state s_i within the EMC is obtained via (13) with the steady-state probabilities $\mathbf{u}$ calculated for the EMC (see Section 3.1). In order to calculate the expected number of occurrences for a sub-state $s_{i,j,k}$, we consider a reduced process where only the top level and the respective region $s_{i,j}$ of the composite state s_i are considered, i.e., we blend out all other parallel regions and regions of different composite states. Leveraging (1), (2) and (11), we obtain the expected number of occurrences for a sub-state $s_{i,j,k}$ as

$$E[n_{1,ijk}] = \frac{\pi_{i,j,k}}{\pi_n} = \frac{\pi_i \frac{\sigma_{i,j,k}}{E[X_i]}}{\frac{u_n \cdot c_{n,n}}{\mathbf{uCe}}} = \frac{u_i \cdot c_{i,i} \cdot \sigma_{i,j,k}}{\mathbf{uCe} \cdot c_{i,i}} \frac{\mathbf{uCe}}{u_n \cdot c_{n,n}} = \frac{u_i \cdot \sigma_{i,j,k}}{u_n \cdot c_{n,n}}. \tag{14}$$

The value $E[n_{1,ijk}]$ only depends on: the steady-state probabilities from the EMC which are the same for the original and the reduced process and therefore only have to be calculated once, the mean sojourn time of the final state in the usage model which is equal to the chosen step size τ as we consider the whole process as DTMC, and the mean sojourn time of the respective sub-state which is obtained via (3). Note that, the mean sojourn time for the final state of region $s_{i,j}$ can also be obtained by (3), since we blend out other parallel regions. The expected number of occurrences of a transition in a test case is obtained in the same way, as transitions in a TUM are extended to states/sub-states in the underlying SMP (see Section 4.1).

The mean residence time of a state s_i in a test case is given by $E_r[s_i] = E[n_{1,i}] \cdot E[X_i]$, i.e., its number of occurrences within a test case multiplied with its mean sojourn time. Of course, the mean sojourn time used here is derived with regard to the distribution attached to the state. The mean residence time of a transition in a test case is calculated in the same way. The expected execution time for a stimulus is derived as sum over all expected residence times for transitions, that have the stimulus attached.

5 Validation

We applied our approach to an example system from the automotive domain: the adaptive cruise control (ACC). It provides comfort to the driver by automatically maintaining a defined velocity. If a vehicle in front prohibits to drive at desired speed, it will slow down in order to maintain a safe distance. The ACC allows concurrent streams of use, e.g., the driver may change the speed and/or distance, while the vehicle in front may break or accelerate. We specified a respective TUM with parallel regions. It is depicted in Figure 2a. After initialization, the ACC maintains a safe distance to the vehicle in front at a velocity that is lower than configured by the driver. In the first region of composite state s_2, the driver changes the distance to be maintained to the vehicle in front. The second region specifies the actions of the vehicle in front. Test cases generated from the model examine whether the acceleration induced by the ACC stays below a defined threshold. The aim of this threshold is to avoid abrupt movements. Distributions for sojourn times in states and firing times of transitions are omitted in the figure to keep it clear. We used exponential distributions for non-absorbing states and geometrical distributions for transitions. Of course, different times may be used.

We carried out the analysis on the example usage model and calculated all metrics from Section 4.2. In order to validate the analysis, we built a simulation. Therefore, we used Papyrus[2] to specify the model and Syntony [5] to transform it to the simulation framework OMNeT++[3]. The simulation was run multiple times. The results, some shown in Figure 2b, confirmed the values that had been obtained by our approach. Parameters related to the expected number of occurrences of a state/transition were identical. We measured a marginal deviation between the simulation and analysis results only for parameters related to durations. The difference is depicted in Figure 2b. The value for $E[w_{2,2}]$ stands out. This is due to the non-deterministic synchronization effects in the simulation upon exit of the composite state. However, the absolute value is negligibly small.

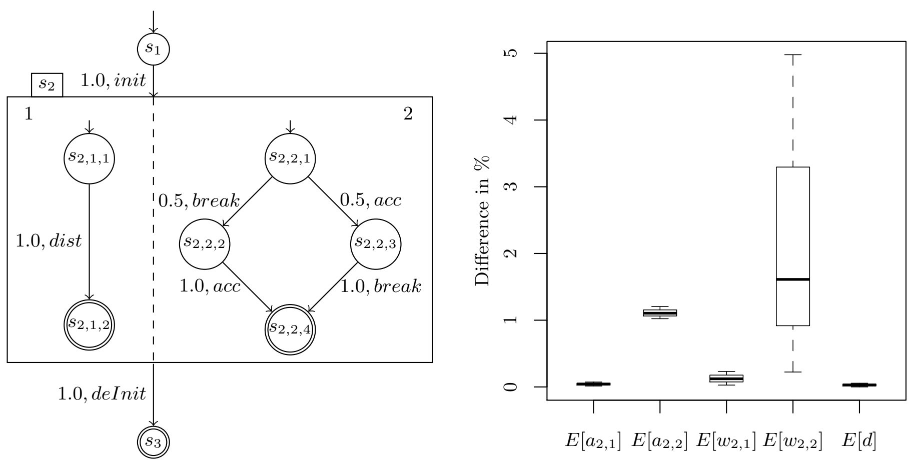

Fig. 2: (a) TUM with parallel regions for ACC example and (b) relative difference between simulation and analysis results.

6 Conclusions and Future Work

With the introduction of parallel regions to TUMs we were able to model both, sequential and concurrent uses of the SUT, with regard to time dependencies. This

[2] Papyrus: Graphical editing tool for UML 2, http://www.eclipse.org/modeling/mdt/papyrus, accessed on May 22, 2015

[3] OMNeT++: An object-oriented modular discrete event network simulation framework, http://www.omnetpp.org, accessed on May 22, 2015

was previously not possible due to restrictions in classical TUMs. Thereby, we preserved the major properties of the model: the automatic test case generation and calculation of parameters that aid the test process. The test case generation stayed a straightforward process. For the analysis, we avoided discretization of time dependencies and introduced an SMP with parallel regions instead.

In our future work we plan to provide an additional test stop criteria that considers the variability of possible uses resulting from concurrent aspects.

References

1. Böhr, F.: Model based statistical testing of embedded systems. In: Proceedings of the 4th International Conference on Software Testing, Verification and Validation Workshops (ICSTW'11). pp. 18–25. Berlin, Germany (March 2011)
2. Bringmann, E., Kramer, A.: Model-based testing of automotive systems. In: Proceedings of the 1st International Conference on Software Testing, Verification, and Validation (ICST'08). pp. 485–493. Lillehammer, Norway (April 2008)
3. Broy, M.: Challenges in automotive software engineering. In: Proceedings of the 28th International Conference on Software Engineering (ICSE'06). pp. 33–42. Shanghai, China (May 2006)
4. Dias Neto, A.C., Subramanyan, R., Vieira, M., Travassos, G.H.: A survey on model-based testing approaches: A systematic review. In: Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies (WEASELTech '07). pp. 31–36. Atlanta, Georgia, USA (2007), http://doi.acm.org/10.1145/1353673.1353681
5. Dietrich, I.: Syntony: A Framework for UML-Based Simulation, Analysis and Test with Applications in Wireless Networks. Dr. Hut, Germany (August 2010)
6. German, R.: Performance Analysis of Communication Systems. Wiley, United Kingdom (June 2000)
7. Little, J.D.C.: A proof for the queuing formula: L= λ W. Operations Research 9(3), 383–387 (May 1961)
8. Prowell, S.J.: Computations for markov chain usage models. Tech. rep., University of Tennessee (2003)
9. Prowell, S.J.: A stopping criterion for statistical testing. In: Proceedings of the 37th Hawaii International Conference on Systems Sciences (HICSS'04). Big Island, Hawaii, USA (January 2004)
10. Siegl, S., Dulz, W., German, R., Kiffe, G.: Model-driven testing based on markov chain usage models in the automotive domain. In: Proceedings of the 12th European Workshop on Dependable Computing (EWDC'09). Toulouse, France (May 2009)
11. Siegl, S., German, R.: Model-driven testing with timed usage models in the automotive domain. In: Proceedings of the 20th International Symposium on Software Reliability Engineering (ISSRE'09). Mysuru, India (November 2009)
12. Siegl, S., Hielscher, K.S., German, R.: Introduction of time dependencies in usage model based testing of complex systems. In: 4th Annual IEEE Systems Conference (SysCon'10). pp. 622–627. San Diego, California, USA (April 2010)
13. Whittaker, J.A., Poore, J.H.: Markov Analysis of Software Specifications. ACM Transactions on Software Engineering and Methodology 2(1), 93–106 (January 1993)