Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 264 (2010) 19–35
www.elsevier.com/locate/entcs

A Polyhedron Approach to Calculate Probability Distributions for Markov Chain Usage Models

Winfried Dulz[a] Stefan Holpp[b] Reinhard German[a]

[a] *Department of Computer Science 7*
University of Erlangen-Nuremberg
Erlangen, Germany

[b] *sepp.med gmbh*
Roettenbach, Germany

Abstract

Statistical usage testing of hardware/software systems is based in the main on a Markov chain usage model. This kind of model represents the expected use of the system by a usage profile, i.e. appropriate probability values that are attached to the state transitions. In this paper we present a constraint-based polyhedron approach to calculate the probability distribution for the MCUM from a given set of usage constraints. Comparing the computed probability distributions of our polyhedron approach with the maximum entropy technique shows that our result is much closer to the intented constraint semantics. Using the polyhedron method, customer profiles can be calculated so that they reflect the intended system usage of different customers or customer types much better. In order to demonstrate the applicability of our approach, workflow testing of a complex RIS/PACS system in the medical domain was carried through and yielded very promising results.

Keywords: statistical usage testing, Markov chain usage model, profile generation, metrics, medical application domain

1 Introduction

MBT (Model-based Testing) techniques apply formal descriptions (models) of either the SUT (System under Test) or the expected usage behavior of SUT customers. In the former case, a behavioral specification is derived from the requirement definitions and serves as a starting basis to automatically generate test cases in order to test the SUT [9]. In the latter case, usage models are deduced from the requirements

[1] Email: dulz@cs.fau.de
[2] Email: german@cs.fau.de
[3] Email: Stefan.Holpp@seppmed.de

doi:10.1016/j.entcs.2010.12.012

and may be considered as independent of the specification. Because exhaustive testing of real systems is infeasible in practice, an appropriate set of test cases is derived for accomplishing a given test goal. At this point, Markovian statistics come into play.

Statistical software testing has experienced a large growth of interest during the last years [4,6,3]. In contrary to other testing approaches, the statistical testing approach benefits from usage-driven testing and the possibility to make statistical inferences about the system, based on a set of (statistically correct) sampled test cases (Fig. 1).

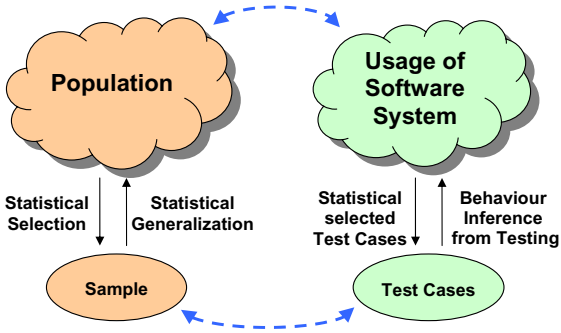

Fig. 1. The Statistical Testing Approach.

The core objective of statistical testing is not to improve the quality of the system by finding more software defects, but to provide an estimate of the achieved quality metrics. Cleanroom software engineering [11] uses statistical testing as a reliability certification method.

Statistical testing relies on Markov chain usage models [14] that characterize the estimated distribution of possible uses of a given software in its intended environment. A Markov chain consist of states and transitions between states [15]. This directed graph structure describes all possible usage scenarios for a given SUT derived by randomly traversing the arcs between dedicated start and end states. Transitions are augmented by probabilities from a usage profile that reflects usage choices users will have when they interact with the SUT (Fig. 2). Providing more than one usage profile for a given Markov chain structure is also possible. Hereby, the statistical selection of test cases reflects distinct properties of different roles or user classes, e.g. experts or normal users, in interaction with the SUT.

Markov chains are mathematically represented by transition matrices [4]. Here, the i-th row of the j-th column of the transition matrix holds the transition probability for leaving state i and entering state j as depicted in Fig. 3.

Each path of the MCUM represents a possible use of the system, and corresponding transition probabilities define the mean rate at which a costumer will traverse this part of the system during the application.

Analytical computations can be applied even before test cases are executed in order to obtain characteristic test metrics. Examples are the Kullback-Leibler diver-

[4] The transition matrix is often called stochastic matrix since the matrix elements are probability values.

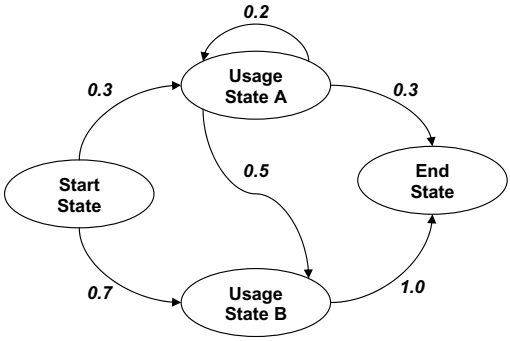

Fig. 2. Example Markov Chain.

	Start State	Usage State A	Usage State B	End State
Start State	0	0.3	0.7	0
Usage State A	0	0.2	0.5	0.3
Usage State B	0	0	0	1
End State	0	0	0	0

Fig. 3. Corresponding Transition Matrix.

gence with respect to the given profile, the mean number of test cases necessary to reach a certain confidence level, or the expected reliability of the future system [7].

Because all computations of MCUM metrics heavily rely on the probability distribution of the given profile, methods for deriving correct probabilities are a critical part in creating a MCUM [13]. In this paper we focus on the computation of probability distributions from a given set of probability constraints by applying a polyhedron approach.

2 Derivation of Usage Profiles

2.1 Motivation

After all structural components, i.e. states and arcs of the MCUM, have been identified [1], appropriate transition probabilities have to be obtained. Ideally, for every state of the MCUM, the exact probability distribution of the outgoing arcs of each state is known and fixed. Unfortunately, this is often not the case. For example, in the medical domain very large and complex RIS/PACS systems have to be tested, where the usage of the system is subject to diverse influences [5]. American Image Centers, for example, show a typical similar system usage and can be treated as one customer type. The behavior of this customer type is therefore mapped to a single MCUM.

2.2 General Strategies for Profile Derivation

In [12], Whittaker proposes three possible techniques to compute MCUM transition probabilities, namely:

- *uninformed:* probabilities are uniformly assigned over all outgoing arcs of a state,
- *informed:* probabilities are assigned according to collected field data and
- *intended:* probabilities are assigned according to the intended use of the system.

2.2.1 Uninformed Strategy

This is the most simple but also least suitable strategy for creating usage profiles. All outgoing arcs of a state within the usage profile are assigned equal probability values. While this approach does not presume any information about system usage, it can be applied if no such information is available.

2.2.2 Informed Strategy

Informed probability generation presumes collecting field data, which is then mapped to a usage profile. Hence, this method represents the most accurate way of gaining information about system usage. Field data may be derived from direct logging of user interaction use or screen recording, and can afterwards be evaluated or directly mapped to the model structure.

2.2.3 Intended Strategy

Probability values are defined by estimating the intended use of the system and strongly depend on the quality of the information sources. We preferred this strategy because neither changes on the system nor any additional technical resources are necessary. Development teams often possess a multitude of information sources for the required customer specification. In addition, application specialists, customer support and application trainers generally know the different customers and their specific way of working very well.

3 Constraint-Based Strategy

Since it can not be assumed that customer experts do also have expert knowledge about statistics and MCUM theory, the process of information acquisition must be arranged in the most informal way as possible. This implies that customer experts are allowed to describe their knowledge in terms of *constraints* [13]. Instead of assigning fixed numbers to the MCUM transition probabilities, they have to postulate relationships between them. These constraints can further be automatically transformed to fixed probabilities by using algebraic techniques.

Reffering to the simple state example shown in Fig. 4, a customer expert has several possibilities to formulate his knowledge.

(i) **Area Definition**

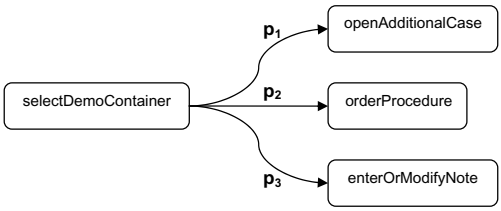

Fig. 4. MCUM State Example for Constraint Descriptions.

"Having selected a demo container, the fraction of time a radiologist would order a procedure is definitely bigger than 20%."
This would result in $p_2 > 0.2$.

(ii) **Interval Definition**
"Having selected a demo container, the fraction of time a radiologist would open an additional case is in between 50% and 80%."
This would result in $p_1 \geq 0.5$ and $p_1 \leq 0.8$.

(iii) **Exact Definition**
"Having selected a demo container, the fraction of time a radiologist would enter of modify a note is exactly 25%."
This would result in $p_3 = 0.25$.

(iv) **Dependent Definition**
"Having selected a demo container, the fraction of time a radiologist would order a procedure is twice as big as opening an additional case."
This would result in $p_2 = 2 \cdot p_1$.

Constraints can be further classified as follows:

- **Structural constraints** are implicitly given by the model structure, i.e.
  · The probability values $p_{i,j}$ of all outgoing arcs $a_{i,j}$ of a state s_i must sum to one: $\sum_j p_{i,j} = 1 \quad \forall i$
  · Each probability value $p_{i,j}$ must be in the interval [0..1]: $p_{i,j} \geq 0, \quad p_{i,j} \leq 1 \quad \forall i, j$
- **Usage constraints** are given to define the intended system usage and cannot be treated automatically.

Constraints reduce the solution space when searching for a suitable probability distribution for the MCUM. This is demonstrated by re-using the previous constraints definitions referring to Fig. 4:

- Taking the constraints (iii) and (iv) ($p_3 = 0.25, \quad p_2 = 2 \cdot p_1$) in addition to the structural constraints ($p_1 + p_2 + p_3 = 1, \quad p_1 \geq 0, p_2 \geq 0, p_3 \geq 0, \quad p_1 \leq 1, p_2 \leq 1, p_3 \leq 1$), the correct probability distribution is implicitly given with $p_1 = 0.25, p_2 = 0.5, p_3 = 0.25$.
- Considering constraints (i) and (ii) instead of (iii) and (iv), there exists an infinite set of possible solutions. The distribution $p_1 = 0.7, p_2 = 0.25, p_3 = 0.05$ is valid as well as $p_1 = 0.5, p_2 = 0.3, p_3 = 0.2$.

Obviously, objective functions have to be defined so that they deliver a single probability distribution in case that the number of possible solutions is greater than one.

3.1 Maximum Entropy Approach

In [8], it was argued to use the entropy of the probability distribution as objective function and to maximize this entropy afterwards.

The state entropy $h(s_i)$ of a state s_i with n outgoing arcs $a_{i,j}$ and their corresponding probabilities $p_{i,j}$ with $j = 1..n$ is defined as:

$$h(s_i) = -\sum_{j=1}^{n} p_{i,j} \cdot \log_2(p_{i,j}) \tag{1}$$

This seems reasonable because the entropy is a measure of uncertainty and therefore guarantees that the information content is limited exclusively by the defined constraints. Considering the entropy function without defining constraints, the maximum is found when the probability values represent a uniform distribution, which also matches the idea of the uninformed strategy 2.2.1.

3.1.1 Constraints as Equalities

Let us consider a constrained, non-linear optimization problem with equality constraints first. Optimization problems of the form

$$\textbf{max}\ f(p_1, ..., p_k) \tag{2}$$

with constraints

$$\begin{aligned} g_1(p_1, ..., p_k) &= 0 \\ &\cdots \\ g_m(p_1, ..., p_k) &= 0 \end{aligned} \tag{3}$$

can be solved using the Lagrange Method as proposed in [8]. Constraints as discussed in section 3 can easily be converted to this form by simple transformations. Having the operational function

$$f(p_1, ..., p_k) = -\sum_{i=1}^{k} p_i \cdot \log_2(p_i)\,, \tag{4}$$

the structural constraint

$$g_1(p_1, ..., p_k) = -1 + \sum_{i=1}^{k} p_i\,, \tag{5}$$

and $m-1$ usage constraints

$$g_j(p_1,...,p_k) = 0, \quad j = 2..m\,, \tag{6}$$

the Lagrange function Λ can be defined as follows

$$\Lambda(p_1,...,p_k,\lambda_1,...,\lambda_m) = f(p_1,...,p_k) + \sum_{j=1}^{m} \lambda_j \cdot g_j(p_1,...,p_k) \tag{7}$$

where λ_j are Lagrange Multipliers. The gradient of Λ is given by

$$\nabla_{p_i,\lambda_j}(\Lambda) = \frac{\partial \Lambda}{\partial(p_i,\lambda_j)} = \begin{pmatrix} \log_2 p_i \sum\limits_{t=1}^{m} \frac{\partial \lambda_t g_t}{\partial p_t} \\ g_j \end{pmatrix} \tag{8}$$

Assuming only linear constraints, the constraint functions can be written as

$$g_t(p_1,...,p_k) = \sum_{i=1}^{k} (a_{t,i} \cdot p_i) + c_i\,, \quad \text{with} \quad a_{t,i}, c_i \in \mathbb{R} \quad \text{and} \quad t \in [1..m] \tag{9}$$

where $a_{t,i}, c_i$ are fixed constants, resulting from the previous constraint definitions. The gradient of Λ can be simplified to

$$\nabla_{p_i,\lambda_j}(\Lambda) = \frac{\partial \Lambda}{\partial(p_i,\lambda_j)} = \begin{pmatrix} \log_2 p_1 + \sum\limits_{t=1}^{m} a_{t,1} \cdot \lambda_t \\ \vdots \\ \log_2 p_k + \sum\limits_{t=1}^{m} a_{t,k} \cdot \lambda_t \\ \sum\limits_{t=1}^{k} (a_{1,t} \cdot p_t) + c_t \\ \vdots \\ \sum\limits_{t=1}^{k} (a_{m,t} \cdot p_t) + c_t \end{pmatrix} \tag{10}$$

To obtain the critical values of Λ, $\nabla(\Lambda)$ is set to zero. This leads to a system of equations

$$\begin{aligned} \log_2 p_1 + a_{1,1} \cdot \lambda_1 + \cdots + a_{m,1} \cdot \lambda_m &= 0 \\ &\vdots \\ \log_2 p_k + a_{1,k} \cdot \lambda_1 + \cdots + a_{m,k} \cdot \lambda_m &= 0 \\ a_{1,1} \cdot p_1 + \cdots + a_{1,k} \cdot p_k &= 0 \\ &\vdots \\ a_{m,1} \cdot p_1 + \cdots + a_{m,k} \cdot p_k &= 0 \end{aligned} \tag{11}$$

with $(k+m)$ equations and $(k+m)$ unknowns. Thus, solving this equation system yields the probability distribution $(p_1, ..., p_k)$ with maximum entropy.

3.1.2 Constraints with additional inequalities

The method of Lagrange Multipliers can be extended to handle additional inequality constraints. The Karush-Kuhn-Tucker Conditions (KKT [2]) describe a generalization of the method of Lagrange Multipliers. Let us consider the same optimization problem as before extended with additional inequalities:

$$\begin{aligned} h_1(p_1, ..., p_k) &\leq 0 \\ &\vdots \\ h_n(p_1, ..., p_k) &\leq 0 \end{aligned} \tag{12}$$

The KKT conditions state that p is a global minimum [5] of $f(p)$ if

$$\nabla f(p) + \sum_{i=1}^{m} \mu_i \nabla g_i(p) + \sum_{j=1}^{n} \nu_j \nabla h_j(p) = 0 \tag{13}$$

under the conditions

(i) $g_i(p)$ are affine functions,

(ii) $h_j(p)$ are convex functions,

(iii) $\nu_j \geq 0$ and μ_i exist $\forall\ i = 1..m,\ j = 1..n$,

(iv) $\nu_j \cdot h_j(p) = 0 \quad \forall\ j = 1..n$.

Solving this can be done analogously to the method of Lagrange Multipliers.

4 Polyhedron Approach

4.1 Motivation

The maximum state entropy strategy faces a major drawback: while the state entropy is maximized, the resulting probability distribution for outgoing arcs of each state is always as uniform as it could be. For example, referring to Fig. 4 and using maximum state entropy, the single constraint $p_1 \geq 2 \cdot p_2$ would lead to the same result as the constraint $p_1 = 2 \cdot p_2$ would do.

However, the customer expert who formulated the constraint probably had in mind p_1 to be *truly* bigger than twice the amount of p_2. Thus, maximizing the state entropy may lead to inadequate solutions in certain cases. We therefore propose a more sophisticated method for generating constrained probability distributions while assuring the most probable constraint semantics.

[5] To obtain the maximum of $f(p)$, the minimum of $-f(p)$ can be calculated.

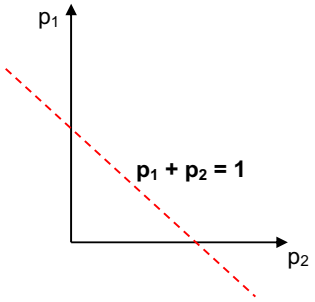

Fig. 5. Equality constraint, limiting the solution space by an affine subspace.

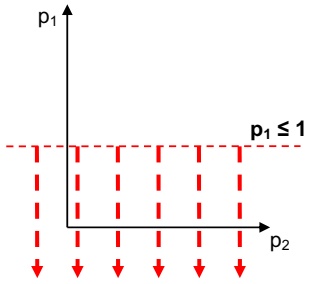

Fig. 6. Inequality constraint, limiting the solution space by a halfspace.

4.2 Computing the Probability Distribution

The basic idea is as follows. Without any constraints, the solution space of the searched probability distribution of a state with k outgoing arcs would be $\mathbb{R}^k$. Of course, at least the structural constraints and, if stated, any usage constraints, restrict the solution space.

Equality constraints can be considered as restrictions to affine subspaces of lower dimensions within $\mathbb{R}^k$ (see Fig. 5) while inequalities result in restrictions to half spaces, as shown in Fig. 6. Regarding lots of constraints will lead to a convex polyhedron representing the constrained solution space. Based on this polyhedron, a suitable probability distribution is obtained. Again, let $g_k(p)$ be the set of equality constraints, and $h_l(p)$ the set of inequality constraints where p is the vector of all outgoing state transition probabilities p_i.

$$\begin{aligned} g_1(p) &= c_1 \\ &\vdots \\ g_m(p) &= c_m \\ h_1(p) &\leq c_{m+1} \\ &\vdots \\ h_n(p) &\leq c_{m+n} \end{aligned} \tag{14}$$

A polyhedron can be defined as a system of linear equations. To obtain such a poly-

hedron, the inequality equations $h_l(p)$ must be transformed to equality constraints by inserting additional variables s_l for each inequality:

$$h_l(p) \leq c_l \rightarrow h_l(p) + s_l = c_l, s_l \geq 0 \quad \forall l \in [(m+1)..(m+n)] \tag{15}$$

This leads to the following system of linear equations:

$$\begin{aligned} g_1(p) &= c_1 \\ &\vdots \\ g_m(p) &= c_m \\ h_1(p) + s_1 &= c_{m+1} \\ &\vdots \\ h_n(p) + s_n &= c_{m+n} \end{aligned} \tag{16}$$

with the solution vector $x = (p_1, ..., p_k, s_1, ..., s_n)^T$. This can also be written in matrix form for a $(m+n) \times (k+n)$-Matrix $A = (a_{i,j}) \in \mathbb{R}^{(m+n)\times(k+n)}$:

$$A \cdot x = c, \quad \text{where} \quad c = (c_1, ..., c_{m+n})^T \tag{17}$$

It is obvious that

$$k + n \geq m + n \tag{18}$$

holds since the dimension of the solution space (k) is always greater than or equal the number of linear constraints (m). For each linear constraint, the solution space is limited by a further affine subspace and therefore loses (at least) one degree of freedom. In the special case where $k + n = m + n$, the vector x is implicitly given and no further inequality constraints can affect the result. Let b_r^s be the r-th basis vector in column s of A defined as

$$b_r^s = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_{m+n} \end{pmatrix}, \qquad \text{where} \qquad \beta_r = 1, \quad \beta_i = 0 \quad \forall\, i \neq r\,. \tag{19}$$

Assume that

$$(\alpha_1, ..., \alpha_{m+n}), \quad \alpha_i \in \mathbb{Z}^+ \tag{20}$$

is a possible arrangement of basis vectors within A and α_k is corresponding to $b_k^{\alpha_k}$. All possible combinations of basis vectors are given by the set

$$\mathcal{B} := \left\{ (t_1, ..., t_{m+n}) \mid t_i \in \{\alpha_1, ..., \alpha_{k+n}\}, t_i \neq t_j \; \forall i, j = 1...(m+n) \right\}. \tag{21}$$

Each element $z_i \in \mathcal{B}$ represents a combination of basis vectors, which leads to a valid basis solution of A. To obtain a solvable order of these basis vectors, one can choose a permutation of this combination for which A is solvable. The set of possible permutations for a given combination of basis vectors is given by the set

$$\mathcal{P}(z_i) := \left\{ (t_1, ..., t_{m+n}) \mid t_i \in z_i \setminus \{t_1, ..., t_{i-1}\} \right\}. \tag{22}$$

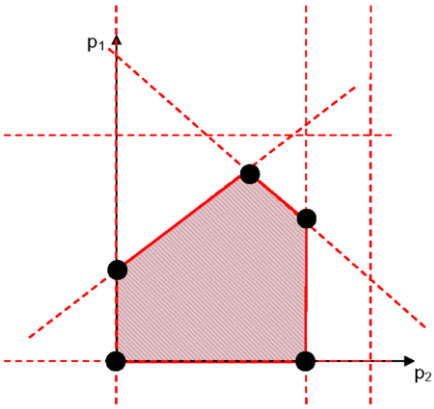

Fig. 7. Valid Solution Space with respect to Constraint Definitions.

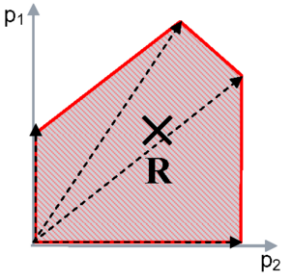

Fig. 8. Center of Mass of a Polyhedron.

Each basis solution x_i represents an intersection of affine subspaces and half spaces, while basis solutions with $x_i > 0$ represent the corners of the convex polyhedron. Let $\{s_i\} \subset \mathcal{B}$ be the set of w unique permutations of z_i which lead to solvable basis solutions x_i with $x_i > 0, \quad i = 1..w$. The polyhedron itself represents the valid solution space given by the defined constraints. Every point within this solution space leads to a valid probability distribution (see Fig. 7). Of course, one of these valid solutions must be picked now and there might be a multitude of possible approaches for doing so. Intuitively, a probability distribution which matches the semantics of the defined constraints must be found somewhere in the center of the polyhedron. A suitable approach for choosing a distribution is to calculate the polyhedron's *center of mass* $\mathbf{R}$ as follows (see also Fig. 8). Let x_i^p be the vector of k upper elements of x_i, with $x_i^p(p_1, ..., p_k)$, which holds the searched probability distribution. The center of mass $\mathbf{R} \in \mathbb{R}$ is given by

$$\mathbf{R} = \frac{1}{w} \cdot \sum_{i=1}^{w} x_i^p \tag{23}$$

In Table 1 and Table 2, we compare the solutions for both polyhedron and maximum state entropy approaches for different constraint definitions.

Table 1
Maximum Entropy vs. Polyhedron Approach for a constraint $p_1 \geq p_2 \geq p_3 \geq p_4$.

	p_1	p_2	p_3	p_4
Max. Entropy	0.25	0.25	0.25	0.25
Polyhedron	0.52	0.27	0.15	0.06

In our opinion, the probability distribution obtained from the polyhedron approach is more closer to the intended constraint semantics than the probabilties that are derived by maximizing the state entropy function: instead of always delivering values that are as close as possible to a uniform distribution, i.e. as random as possible, the polyhedron approach is focussing on providing the center of all constraints inequalities and not on returning marginal distributions by considering only the edge values.

It should also be remarked that in the limiting case of using only equalities both techniques produce the same solution set for the probabilities.

Table 2
Maximum Entropy vs. Polyhedron Approach for a constraint $p_1 \geq p_2 + p_3,\ \ p_3 \leq p_4$.

	p_1	p_2	p_3	p_4
Max. Entropy	0.36	0.18	0.18	0.27
Polyhedron	0.45	0.13	0.09	0.33

4.3 Constraint Modeling

For reasons of compactness, constraints can be placed directly within the model. The combination of both constraints and models result in a MCUM usage profile.

To ensure an unambiguous mapping, each arc of the MCUM is assigned a unique name. Then, constraints can be placed in the model as a simple text field, as depicted in Fig. 9.

5 Usage Profile Comparison

"How much does the expected usage of two customers differ?"

Usage profiles may be compared[6] with respect to the represented expected system usage. The resulting value indicates the magnitude of deviation of two usage profiles. Thus, high values prove that two customers use the system very differently.
The Kullback-Leibler divergence KL [7] is an information theoretic measure of the difference between two probability distributions P and Q:

$$KL(P,Q) = \sum_i p(i) \cdot \log\left(\frac{p(i)}{q(i)}\right) \tag{24}$$

[6] Usage profiles have to be based on the same MCUM structure to be comparable.

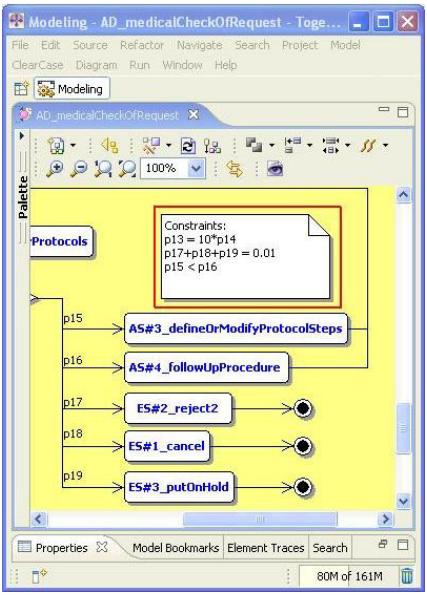

Fig. 9. Modeling of Constraints in Together Architect.

In order to compare Markov chain usage profiles, this metric is extended in the following to consider steady state information, i.e. the rate $\pi(s_i)$ a user will chose state s_i of P during the average system usage:

$$KL(P,Q) = \sum_i \pi(s_i) \sum_j p_{i,j} \cdot \log\left(\frac{p_{i,j}}{q_{i,j}}\right) \tag{25}$$

Obviously, $KL(P,Q)$ is not symmetric, i.e. *in general*,

$$KL(P,Q) \neq KL(Q,P) \tag{26}$$

The KL divergence assumes that P represents a precisely calculated distribution, whereas Q is an approximation of P. Since usage profiles have to be treated equivalently, some kind of average KL divergence is needed.

The Jensen-Shannon divergence (JSD) delivers this measure. The JSD is a popular metric to obtain the difference of two equivalent probability distributions and can therefore be used to measure the similarity of two usage profiles P_1 and P_2.

$$JSD(P_1,P_2) = \frac{1}{2}KL(P_1,M) + \frac{1}{2}KL(P_2,M) \tag{27}$$

with

$$M = \frac{1}{2}(P_1 + P_2) \tag{28}$$

Table 3 provides an overview of the deviation of created usage profiles for testing medical workflows [5].

It shows the derivation between a general profile averaged over all hospitals, whereas the USA and Australian profiles characterize two specific hospital type. A

Table 3
Usage Profile Divergence.

	General Profile	USA Hospital	Australian Hospital
General Profile	0	0.1586	0.1242
USA Hospital	-	0	0.1612
Australian Hospital	-	-	0

zero value indicates no profile deviation. It can be seen that an Australian hospital type is more closer to an average hospital than the USA one.

Another possible application of divergence measurements between MCUM profiles is the classification of customers. A customer's profile could be compared and assigned to existing classes, based on the divergence of the usage profiles. Such a class could be "3D User" for customers who use 3D features of the software extensively. After a customer is assigned to a class, comparative data emerges for a variety of parameters. A simple example is the customer's hardware standard. Customers in the class "3D User" are likely to have similar standards as for their hardware equipment. Furthermore, customer trainings could be tailored to these classes to give a certain group of customers specific trainings.

6 Test Planning

We finally present a technique based on the KL divergence, which allows to predefine the number of test cases that have to be executed. This issue is of main interest if the time for test case execution is limited at the end of a software development project.

In this situation, the KL divergence can also be used to identify the quantity of coverage between sampled test cases and the expected usage of the system given by a MCUM profile. Thus, a low value states that testing matches the expected customer usage behavior very closely.

Let s_i be an arbitrary state with n outgoing arcs $a_{i,j}$. Let further $k_{i,j}$ denote the number of times the transition $a_{i,j}$ is covered by a set of sampled test cases. The resulting coverage rate $t_{i,j}$ for the test profile is calculated as

$$t_{i,j} = \begin{cases} \frac{k_{i,j}}{v_i}, & \text{for } v_i \neq 0 \\ 0, & \text{for } v_i = 0 \end{cases} \qquad \forall\, j = 1..n, \qquad v_i = \sum_{j=1}^{n} k_{i,j} \tag{29}$$

Let now T denote a test profile resulting from a test case sample and U an arbitrary usage profile. The KL divergence is defined as

$$K(U,T) = \sum_i \pi_i \sum_j u_{i,j} \cdot \log\left(\frac{u_{i,j}}{t_{i,j}}\right) \tag{30}$$

Obviously, the KL divergence is defined only if all arcs of the usage model are tested at least once. To circumvent this situation, [10] proposes an approximated

KL divergence. For arcs, which are not covered by test cases, $t_{i,j}$ is substituted by a small value $\epsilon > 0$ that is close enough to zero depending on the wanted accuracy in order to avoid a division by zero. The resulting approximated KL divergence is given by

$$\tilde{K}(U,T) = \sum_i \pi_i \sum_j u_{i,j} \cdot \log\left(\frac{u_{i,j}}{\epsilon - \epsilon \cdot (sgn(t_{i,j})) + t_{i,j}}\right) \tag{31}$$

with

$$sgn(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases} \tag{32}$$

Figures 10 and 11 explain the behavior of the KL divergence during test case sampling with respect to different usage profiles. The X-axis is showing the number of test cases already used, whereas the Y-axis represents the corresponding $\tilde{K}(U,T)$ values. Obviously, the KL divergence tends to reach lower values for samples from usage profile from which the test cases are sampled.

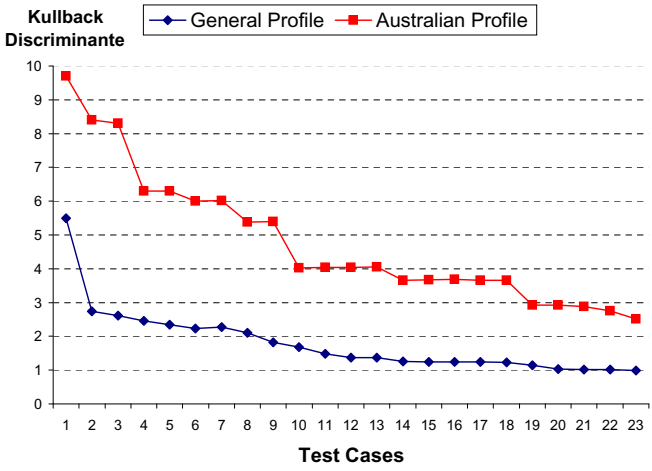

Fig. 10. Test Cases Sampled from the General Profile.

7 Conclusions

We have shown that statistical testing based on MCUMs is a promising extension to existing deterministic testing approaches in general, and in the medical domain in particular.

In order to define customer usage profiles, appropriate transition probabilities have to be obtained. The constraint-based approach turned out to be most convenient for this purpose. A polyhedron method for calculating suitable probability distributions has been presented and compared to the maximum entropy approach.

From the usage models instructive metrics we have not explicitly mentioned can be analytically derived to improve test planning or to support management

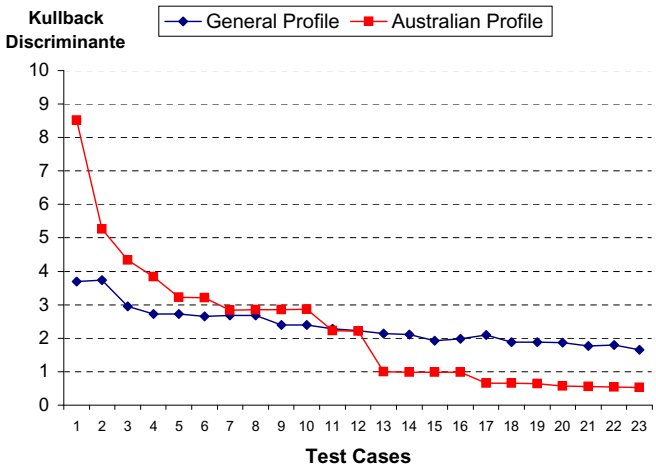

Fig. 11. Test Cases Sampled from an Australian Profile.

decisions. Examples are the mean frequency customers will chose certain usage states or the mean lenght of the test cases for estimating the test exection time. Specific scenarios in order to compare customer classes by means of different MCUM usage profiles have also been identified and discussed.

As for test planning, a method based on the KL divergence to integrate a practicable test stop criteria into existing test case selection procedures has been proposed.

All in all, model-based test case generation using MCUMs and dedicated profile classes enable promising techniques for testing complex systems, for example in the medical or the automotive domain. In the latter case, we are currently extending the test framework of a leading German automobile manufacturer to statistical test case generation techniques.

References

[1] M. Beyer and W. Dulz. 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. Springer Berlin, Heidelberg, 2005.

[2] P. Bosch and J. A. Gomez. Karush-kuhn-tucker theorem for operator constraints and the local pontryagin maximum principle. *Rev. Mat. Apl.*, 18:79–106, 1997.

[3] W. Dulz and F. Zhen. MaTeLo - statistical usage testing by annotated sequence diagrams, Markov chains and TTCN-3. In *IEEE Proc. of Third International Conference on Quality Software (QSIC 2003)*, pages 336–342, 2003.

[4] W. J. Gutjahr. Importance sampling of test cases in Markovian software usage models. *Probability in the Engineering and Informational Sciences*, 11:19–3, 1997.

[5] S. Holpp. Proving the Practicality of the Scenario-based Statistical Testing Approach within the Scope of the System Validation of a RIS/PACS System. Master's thesis, Department of Computer Science, University of Erlangen-Nuremberg, Germany, February 2008.

[6] D. P. Kelly and R. Oshana. Improving software quality using statistical testing techniques. *Information & Software Technology*, 42(12):801–807, 2000.

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

[8] J. H. Poore, G. H. Walton, and J. A. Whittaker. A constraint-based approach to the representation of software usage models. *Information & Software Technology*, 42(12):825–833, 2000.

[9] S. Rosaria and H. Robinson. Applying models in your testing process. *Information and Software Technology*, 42:815–824, 2000.

[10] K. Sayre. *Improved Techniques for Software Testing Based on Markov Chain Usage Models*. PhD thesis, The University of Tennessee, Knoxville, TN, 1999.

[11] C. J. Trammell S. J. Prowell. *Cleanroom Software Engineering: Technology and Process*. Addison-Wesley, 1999.

[12] J. A. Whittaker and J. H. Poore. Markov analysis of software specifications. *ACM Trans. Softw. Eng. Methodol.*, 2:93–106, 1993.

[13] G. H. Walton and J. H. Poore. Generating transition probabilities to support model-based software testing. *Softw. Pract. Exper.*, 30(10):1095–1106, 2000.

[14] G. H. Walton, J. H. Poore, and C. J. Trammell. Statistical Testing of Software Based on a Usage Model. *Software - Practice and Experience*, 25(1):97–108, 1995.

[15] J. A. Whittaker and M. G. Thomason. A markov chain model for statistical software testing. *IEEE Transactions on Software Engineering*, 20(10), 1994.