Supporting Backward Transitions within Markov Chains when Modeling Complex User Behavior in the Palladio Component Model

Maximilian Barnert, Helmut Krcmar
{maximilian.barnert, helmut.krcmar}@tum.de
Technical University of Munich, 85748 Garching, Germany

Abstract

The specification of complex user behavior as accurate as possible is required in order to evaluate performance characteristics for application systems. Approaches exist to model probabilistic aspects within user behavior for session-based application systems using Markov chains. To integrate these approach into performance prediction activities, the authors transform the workload specifications of WESSBAS into performance model instances of the Palladio Component Model (PCM). In doing so, existing elements of the meta-model are reused to represent Markov chains. However, the simulation of these performance models is limited to Markov chains without backward transitions. To deal with this use case, this paper presents our approach to enable backward transitions within Markov chains using available elements of the PCM meta-model. By extending the existing approach, further complexity within workload for application systems is supported during performance modeling.

1 Introduction

Performance evaluation activities require high accuracy when representing workload specifications [4]. This includes reproducing real user behavior adequately to predict performance for application systems precisely. In order to reflect the user behavior sufficiently in performance models, Vögele et al. [1] present an approach to transform workload specifications for session-based application systems into performance model instances of the Palladio Component Model (PCM). However, the simulation of these models is limited to workloads where service calls appear only once within user sessions. Besides, Vögele et al. [4] propose an alternative approach extending the PCM meta-model to reflect complexity in user behavior. As this requires changes to the meta-model, we re-use the approach of Vögele et al. [1] in this paper to enable simulating activity cycles within workloads by using available elements of PCM. In doing so, the presented adaptions broaden the applicability of the original approach by supporting further workload characteristics. In addition, this prevents changing the PCM meta-model.

The remainder of this paper is organized as follows: Section 2 details the background of this work. The addressed problem is stated in section 3. Afterwards, section 4 presents our approach to solve this issue. In section 5, we depict limitations of our approach. Section 6 discusses related work. Finally, the conclusion and future work are outlined in section 7.

2 Background

The approach described in this paper is based on previous works for specifying and extracting workload for session-based application systems [2, 3]. The workload modeling formalism, called WESSBAS[1], uses behavior models specified as Markov chains for providing a probabilistic representation of the user behavior. To integrate probabilistic workload characteristics into performance prediction activities, Vögele et al. [1] transform workload specifications of WESSBAS to performance model instances of PCM.[2] For simplification purposes, we exclude elements of the original modeling approach in this paper. In particular, we do not integrate (1) the concept of *Guards and Actions* (GaAs) and (2) external operation calls for system components, e.g, a HTTP call. This allows to state the problem in section 3 using a reduced model complexity without touching the underlying concept for representing behavior models in PCM.

For representing WESSBAS behavior models (i.e., Markov chains) in PCM, Vögele et al. [1] use the PCM repository model as the PCM usage model does not allow to reflect several incoming or outgoing edges. In doing so, the authors integrate a basic component with a corresponding interface. A component operation as RDSEFF is attached for each Markov state. A probabilistic branch within each operator's RDSEFF models the transitions between different Markov states. Each branch contains a transition's think time modeled as *InternalAction* using a normal distribution with mean and standard deviation. An *ExternalAction* is added to reflect the transition to another Markov state. In behavior models two artificial elements are used to tag the initial (I) and final

[1] Acronym for Workload Extraction and Specification for Session-Based Application Systems

[2] https://github.com/Wessbas/wessbas.wessbas2pcm

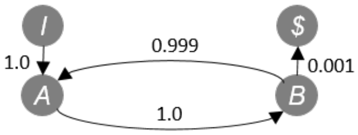


Figure 1: Example of a behavior model specified as Markov chain containing a backward transition

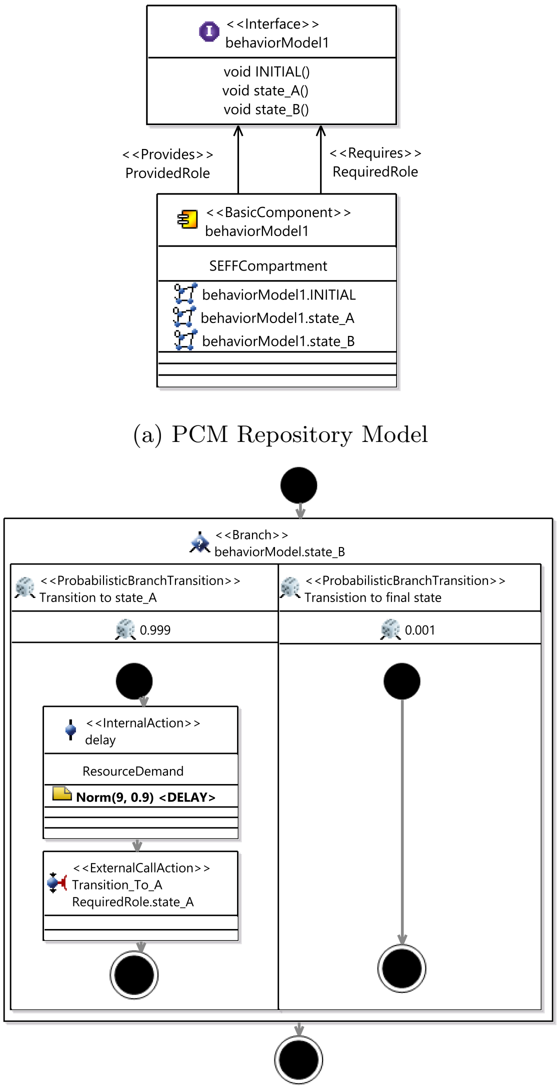
state (*$*) of the chain. Transitions to the final state *$* are represented by a branch consisting of *StartAction* and *StopAction*. The behavior model's initial state *I* is represented by the operation *INITIAL* of the basic component. It contains an *ExternalAction* linked to the entry service of the model. The initial state is called in the closed workload of the PCM usage model to initiate the behavior model's execution.

3 Problem Statement

The applicability of the above stated approach is limited to behavior models (i.e., Markov chains) not containing backward transitions due to simulation constraints. An exemplary Markov chain incorporating a backward transition is depicted in Figure 1. It consists of two services *A* and *B* (e.g., HTTP calls). The initial state *I* is linked to service *A*, which makes it the behavior model's entry point. Markov state *A* has an outgoing transition to state *B*, which again points backward to state *A*. This backward transition is used with a probability of 99.9%. In addition, service B goes to the final state *$* with a probability of 0.1%.

Figure 2a shows the resulting repository model transforming the Markov chain in figure 1 to PCM using the approach of Vögele et al. [1]. The basic component *behaviorModel1* reflects the behavior model. The operations *state_A* und *state_B* are assigned to represent the two Markov states *A* and *B*. The operation *INITIAL* points to the entry point of the Markov chain calling the service *A*. The RDSEFF of *state_A* contains an *ExternalAction*, which calls the component's operator *state_B*. The RDSEFF of *state_B* again calls operator *state_A* with a probability of 99.9% (fig. 2b). The final state is entered with a probability of 0.1%.

To simulate this instance, we created the necessary models on PCM release 4.2. Afterwards, we handed them over to PCM's simulation framework *SimuCom*. However, cycles caused by backward transitions (e.g., the edge from *B* to *A*) prevent simulating such models, because running through a cycle necessitates attaching measurements although the previous series of the same context (i.e., state *A*) is not yet finished.



(a) PCM Repository Model

(b) RDSEFF of operation state_B

Figure 2: Exemplary transformation to PCM

4 Modeling Backward Transitions

In order to simulate such PCM model instances, we adapt the original approach of Vögele et al. [1]. This work proposes two changes to support the simulation of backward transitions to initial and non-initial states within Markov chains (fig. 3):

- Encapsulate outgoing transitions (i.e., *ExternalActions*) with *ForkedBehaviors*
- Replace outgoing transitions to the entry service using a link to the final state

ForkedBehaviors enable writing parallel measurements when simulating performance models containing activity cycles within Markov chains. That is why each *ExternalAction* reflecting a transition to another state is encapsulated by a synchronous *ForkAction*. We consider this modification acceptable as performance data for transitions is typically not needed for performance evaluation, which is mostly based on system component calls within the nodes. After adding *ForkedBehaviors*, we noticed infinite simulation runs with extensive resource consumption for Markov chains with a low probability to attain the

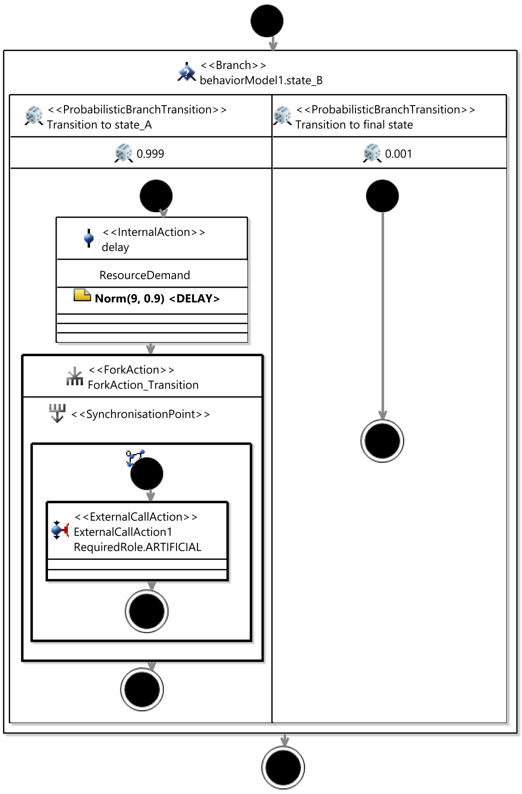


Figure 3: Modified RDSEFF of operation state_B

final state. As a workaround, each backward transition pointing to the entry service (e.g., the outgoing transition from state B to A) is replaced by a link to the final state. This modification is valid since the closed workload in the usage model instantly re-enters the initial state after leaving the former run and again calls the behavior model's entry point (i.e., *state_A*). To preserve information about normal distribution of think times, which is stored in the *InternalAction* of a state's RDSEFF, an artificial state *$'* is introduced. It is represented by the component's operation *ARTIFICIAL* in the repository model. The operation's RDSEFF contains the chain *StartAction* to *StopAction* in order to reach the Markov chain's final state. Backward transitions to the entry service of the chain (i.e., *state_A* in our example) are replaced by a link to the new artificial state *$'*. As a consequence, the *ExternalAction* is linked to the operation *ARTIFICIAL* instead of *state_A*.

5 Limitations

Our approach is subject to limitations. First, a single basic component is used to represent exactly one behavior model in the repository model. In consequence, the PCM usage model is linked to a single behavior model. In contrast, WESSBAS allows to reflect several transactions within a workload using separated behavior models. However, setting clustering to a value of one during the workload extraction process makes it possible to use a single behavior model to represent arbitrary workloads. Second, a distinct transition is defined for the operator *INITIAL*, i.e., exactly one outgoing transition exists for the initial state in the behavior model.

6 Related Work

Two approaches transform behavior models represented as Markov chains to performance models using PCM. As outlined in section 4, Vögele et al. [1] describes an approach to transform workload specifications into PCM model instances. However, the simulation of these models is not possible for activity cycles within user sessions. Furthermore, Vögele et al. [4] propose extensions to the PCM usage model meta-model to reflect complexity in user behavior. However, this approach requires changes to the default meta-model of PCM.

7 Conclusion and Future Work

This work adapts the approach of Vögele et al. [1] to simulate performance model instances representing activity cycles within user sessions. Our adapted approach supports to simulate the performance for further workload running on application systems. At the same time, our extension requires minor changes to the original approach and does not touch the default meta-model of PCM.

For future work, we are going to incorporate the extension presented in this paper to the workload extraction procedure proposed by Vögele et al. [1]. In doing so, we want to support workload containing activity cycles within user sessions for this automatic performance model creation approach.

References

[1] C. Vögele et al. "WESSBAS: extraction of probabilistic workload specifications for load testing and performance prediction—a model-driven approach for session-based application systems". In: *Software and Systems Modeling* 17.2 (2018), pp. 443–477.

[2] A. van Hoorn et al. "Automatic extraction of probabilistic workload specifications for load testing session-based application systems". In: *Porceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2014)*.

[3] A. Van Hoorn, M. Rohr, and W. Hasselbring. "Generating probabilistic and intensity-varying workload for web-based software systems". In: *SPEC International Performance Evaluation Workshop*. Springer, pp. 124–143.

[4] C. Vögele et al. "Modeling Complex User Behavior with the Palladio Component Model". In: *Softwaretechnik-Trends 35(3)*.