



fortiss



Stuttgart, Germany, 2014-11-27

Integrating Workload Specification and Extraction for Model-Based and Measurement-Based Performance Evaluation

An Approach for Session-Based Software Systems

André van Hoorn, Christian Vögele

Eike Schulz, Wilhelm Hasselbring, Helmut Krcmar

Symposium on Software Performance (SOSP) 2014

Problem Statement

Situation

- Workload specification and execution essential to evaluate performance properties of session-based application systems
 - Measurement-based approaches (e.g., Krishnamurthy et al. 2006, Menascé et al. 1999, Arlitt et al. 2001)
 - Model-based approaches (e.g., Becker et al. 2009, Kounev et al. 2014)

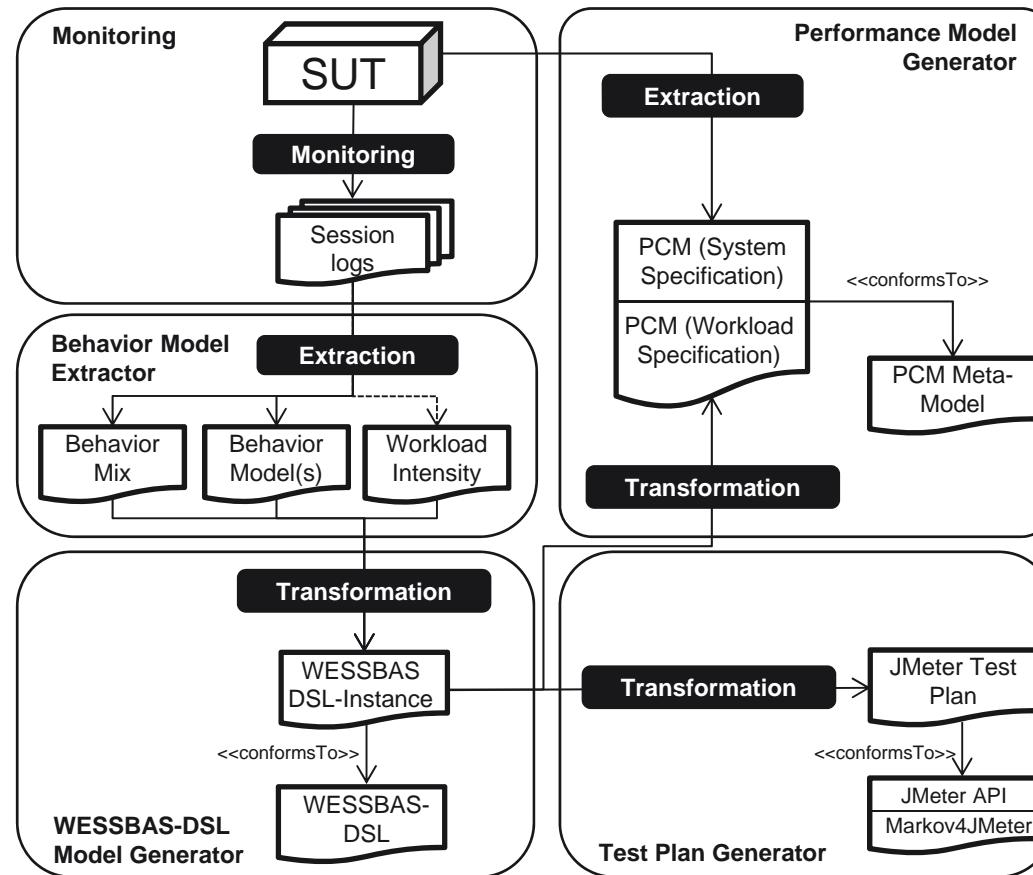
Complication

- Manual creation of representative workload specifications is difficult, time consuming, and error-prone (Shams et al. 2006)
- Nowadays, workload specifications for measurement- and model-based approaches are modelled separately of each other (M-by-N problem)

Resolution: WESSBAS

- 1) System- and tool-agnostic modeling of probabilistic workloads of session-based application systems
- 2) Automatic extraction of these specifications from running systems
- 3) Transformation of these specifications into
 - 1) load test scripts
 - 2) architecture-level workload model specifications

WESSBAS Approach



Content of this talk...

Automatic Extraction of Probabilistic Workload Specifications for Load Testing Session-Based Application Systems

André van Hoorn,¹ Christian Vögele,² Eike Schulz,²
Wilhelm Hasselbring,³ Helmut Krcmar⁴

¹ University of Stuttgart, Institute of Software Technology, 70569 Stuttgart, Germany

² fortiss GmbH, 80805 München, Germany

³ Kiel University, Department of Computer Science, 24118 Kiel, Germany

⁴ Technische Universität München, Chair for Information Systems, 85748 Garching, Germany

ABSTRACT

Workload generation is essential to systematically evaluate performance properties of application systems under controlled conditions, e.g., in load testing environments. The definition of workload specifications that represent the real workload as accurately as possible is one of the biggest challenges in this area. This paper presents our approach for the modeling and automatic extraction of probabilistic workload specifications for load testing session-based application systems. The approach, called WESSBAS, comprising (i.) a domain-specific language (DSL) for specifying and generating workload specifications as well as support for (ii.) automatically extracting instances of the DSL from recorded sessions logs and (iii.) transforming instances of the DSL to workload specifications of existing load testing tools. During the extraction process, different groups of customers with similar navigational patterns are identified using clustering techniques. We developed a tool for load testing including a transformation to probabilistic test scripts for the Apache JMeter load testing tool. The evaluation of the proposed approach using the industry standard benchmark SPECjEnterprise2010 demonstrates its applicability and the representativeness of the extracted workloads.

Categories and Subject Descriptors

C.4 [Performance of Systems]: measurement techniques, modeling techniques

1. INTRODUCTION

For essentially any measurement-based software performance evaluation activity—e.g., load, stress, and regression testing, or benchmarking—it is necessary to expose the system under test (SUT) to synthetic workload [1, 5, 6, 7], i.e., automatically generating requests to system-provided services. Workload generation tools—also called workload drivers—

are used to emulate a multitude of concurrent system users based on workload specifications, ranging from manually defined static workloads to complex, adaptive, and dynamic ones. This paper focuses on analytical model-based workload generation for session-based systems, i.e., systems that are used by users in time-bounded sessions of interleaved requests and think times between subsequent requests [8].

Approaches have been proposed for specifying and generating workloads for this type of systems (e.g., [6, 7, 8]). However, most of these approaches lack the ability to generate workload specifications that produce workload characteristics similar to a system's production usage profile, e.g., arrival rates of sessions and requests to system-provided services. Further, the extraction and specification of workloads strongly depends on the used workload generation tool. Because of this the workload must be extracted for each tool and specified in a specific structure.

In response to these challenges, this paper presents our WESSBAS¹ approach for specifying and extracting probabilistic workloads for session-based application systems. A domain-specific language (DSL), called WESSBAS-DSL, is introduced which enables the system- and tool-agnostic modeling of user workloads. Instances of the WESSBAS-DSL are generated by the system, and the generated instances of the system are used as basis for the automatic extraction of WESSBAS-DSL instances. Different groups of customers showing similar navigational patterns are identified during the creation of these instances. WESSBAS-DSL instances are transformed to workload specifications for load generation tools. Finally, a transformation to the common load testing tool Apache JMeter, including the MarkovUMeter extension, developed by the authors, is performed [14]. The results of this paper are summarized in Figure 1.

To summarize, the contribution of this paper is our WESSBAS approach for automatic extraction of probabilistic workload specifications of session-based application systems, comprising (i.) a DSL for specifying and generating probabilistic workload specifications, (ii.) an automatic extraction of DSL instances from recorded sessions logs including the clustering of navigational patterns, (iii.) transformations from DSL

¹WESSBAS is an acronym for *Workload Extraction and Specification for Session-Based Application Systems*

Automatic Extraction of Session-Based Workload Specifications for Architecture-Level Performance Models

Christian Vögele,¹ André van Hoorn,² Helmut Krcmar²

¹ fortiss GmbH, 80905 München, Germany

² University of Stuttgart, Institute of Software Technology, 70569 Stuttgart, Germany

³ Technische Universität München, Chair for Information Systems, 85748 Garching, Germany

ABSTRACT

Workload specifications are required in order to accurately evaluate performance properties of session-based application systems. These properties can be evaluated using measurement-based approaches such as load tests and model-based approaches, e.g., based on architecture-level performance models. Workload specifications for both approaches are created separately of each other which may result in different workload characteristics. In order to overcome this challenge, this paper extends our WESSBAS approach which defines a domain-specific language (DSL) enabling the layered modeling and automatic extraction of workload specifications, as well as the transformation into load test scripts. This approach will be enabled by the capability of specifying WESSBAS-DSL instances as workload specifications of architecture-level performance models. The transformation demonstrates that the WESSBAS-DSL can be used as intermediate language between system-specific workload specifications on the one side and the generation of the required inputs for performance evaluation approaches on the other side. The evaluation using the standard industry benchmark SPECjEnterprise2010 shows that workload characteristics of the simulated workload match the measured workload with high accuracy.

Categories and Subject Descriptors
C.4 [Performance of Systems]: measurement techniques, modeling techniques

1. INTRODUCTION

In order to validate whether non-functional performance requirements like given throughputs and response times of application systems can be met, measurement- and model-based performance evaluation approaches are applied [16]. Workload specifications are required for both approaches. Workload specifications are input to measurement-based approaches in order to generate synthetic workload to the system under test (SUT), i.e., executing a set of consecutive

and related customer requests within a session [9, 10]. Additionally, these specifications are taken into account in frameworks for model-based approaches to predict performance properties early in the software development cycle [3, 8, 16].

To ensure that the measured and predicted performance characteristics of the SUT are similar, the same workload specification must be used. However, there is a lack of approaches that support the automatic extraction of workload specifications resulting in the same workload characteristics for both approaches. The extraction and specification of workloads is done separately for each approach and each tool. This results in additional specification and maintenance effort. The reason for this are, that these approaches are designed for specific tools and their workload is defined on a different level of detail. Measurement-based approaches need detailed system-specific information like protocol data, whereas model-based approaches are specified on a more abstract level.

In response to these challenges, this paper extends our WESSBAS approach² [13, 14], originally developed to automatically extract probabilistic workload specifications for load testing session-based application systems. The WESSBAS-DSL, a domain-specific language (DSL) created for this task, extracts modeling of workload specifications, will be used as intermediate language for creating workload specifications of architecture-level performance models. This work focuses on architecture-level performance models as they allow to model system architecture, execution environment and system behavior in a more abstract way [1, 2, 10, 14]. Thus, the possibility of extracting workload specifications can be reused by model-based approaches. The architecture-level performance model Palladio Component Model (PCM) [3] will be used to demonstrate the applicability of the proposed approach. The developed tools, models, and results of this paper are publicly available online.³

2. RELATED WORK
In order to automate the extraction of workload specifications for session-based application systems, several measurement-based approaches [1, 2, 10] were introduced. However, these approaches generate workload specifications in tool-specific formats which are not envisaged for model-based approaches as well.

¹WESSBAS is an acronym for *Workload Extraction and Specification for Session-Based Application Systems*

²<http://markov-jmeter.net/b115>

To appear at VALUETOOLS 14

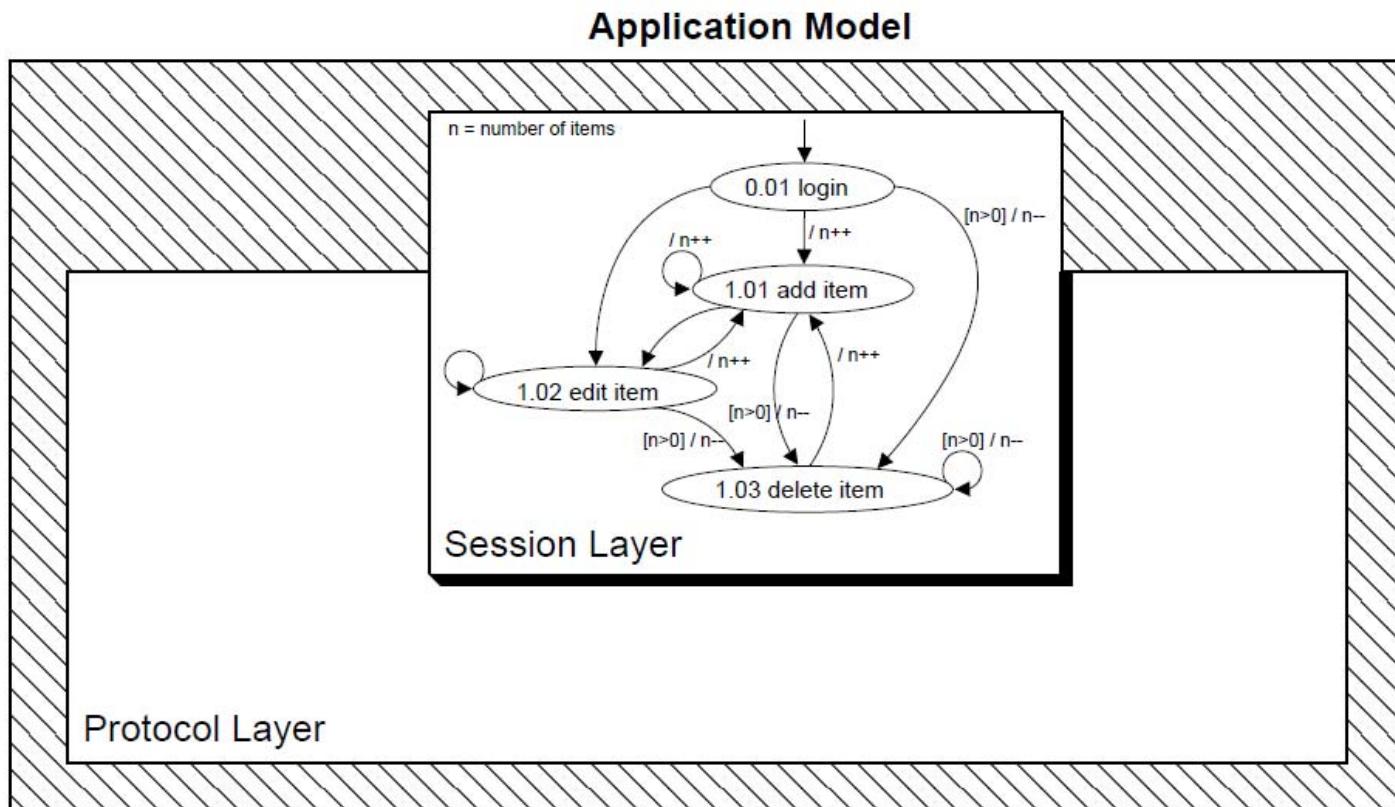
To appear at LT 2015 @ ICPE 2015

Agenda

- Problem Statement and Overview of Approach
- WESSBAS Approach
 1. WESSBAS-DSL
 2. Extraction of WESSBAS-DSL Instances
 3. Clustering of Customer Groups
 4. Transformations
 1. Apache JMeter
 2. Palladio Component Model
- Evaluation
- Future Work

Background – Markov4JMeter

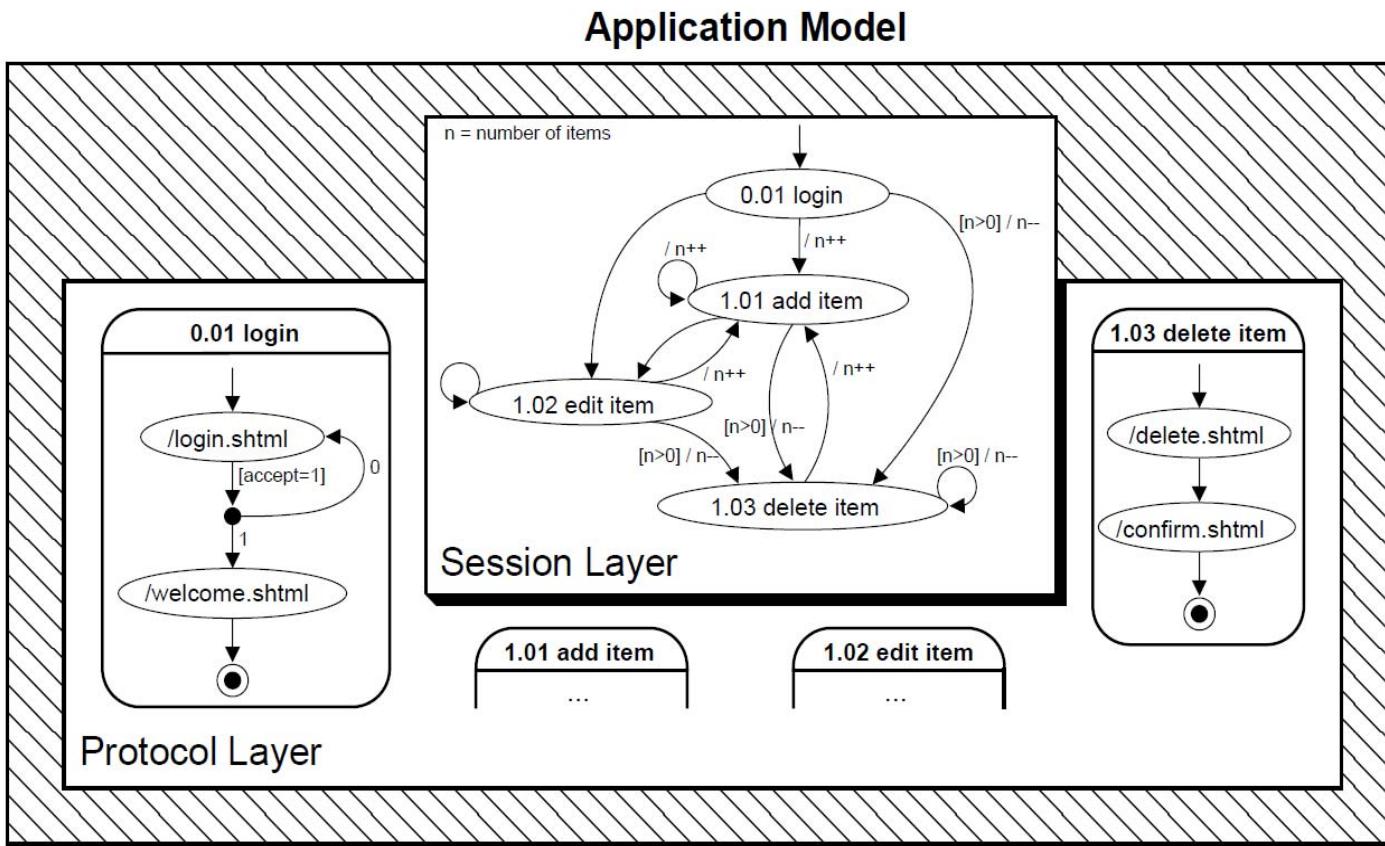
Example of an Application Model



[van Hoorn et al. 2008]

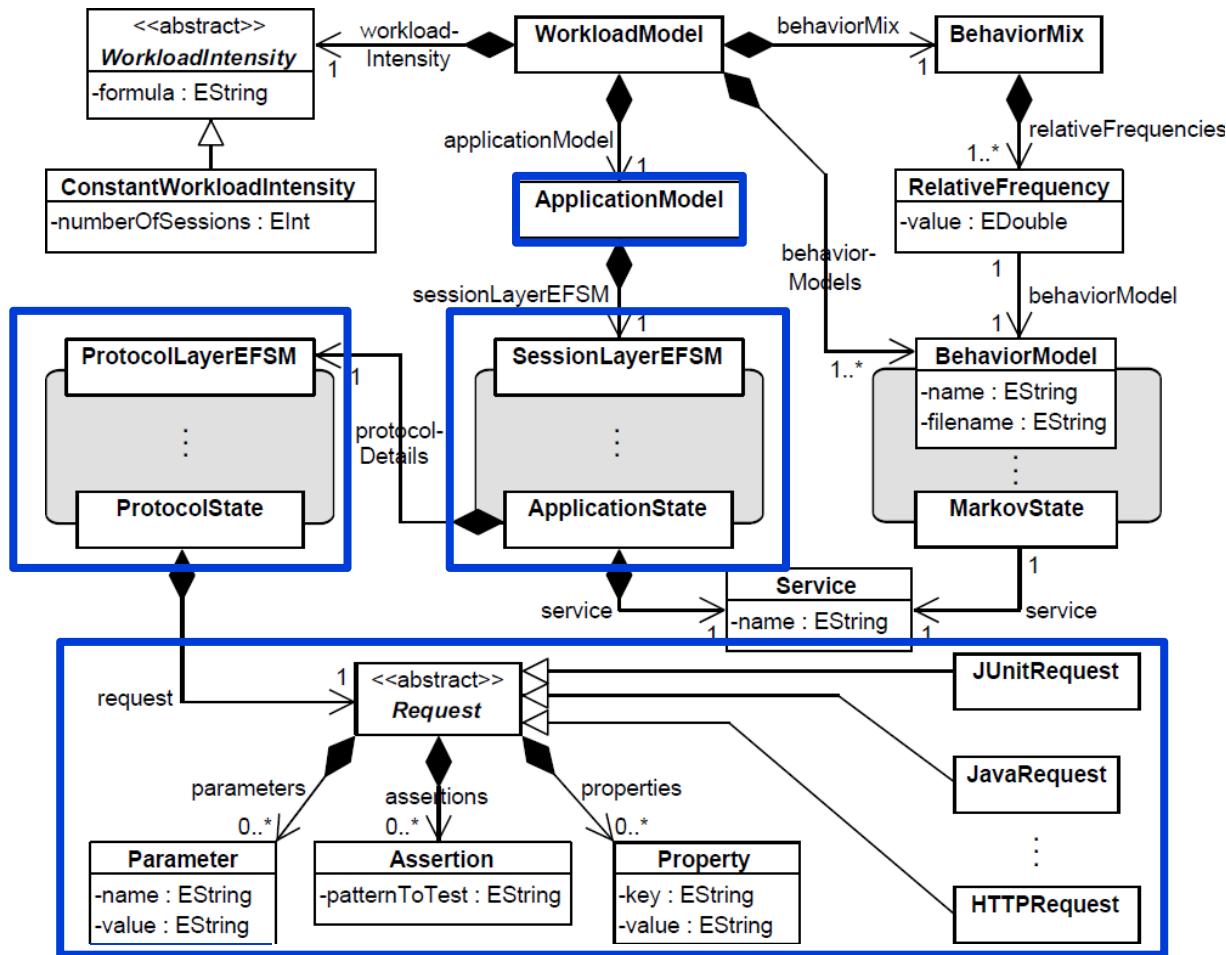
Background – Markov4JMeter

Example of an Application Model



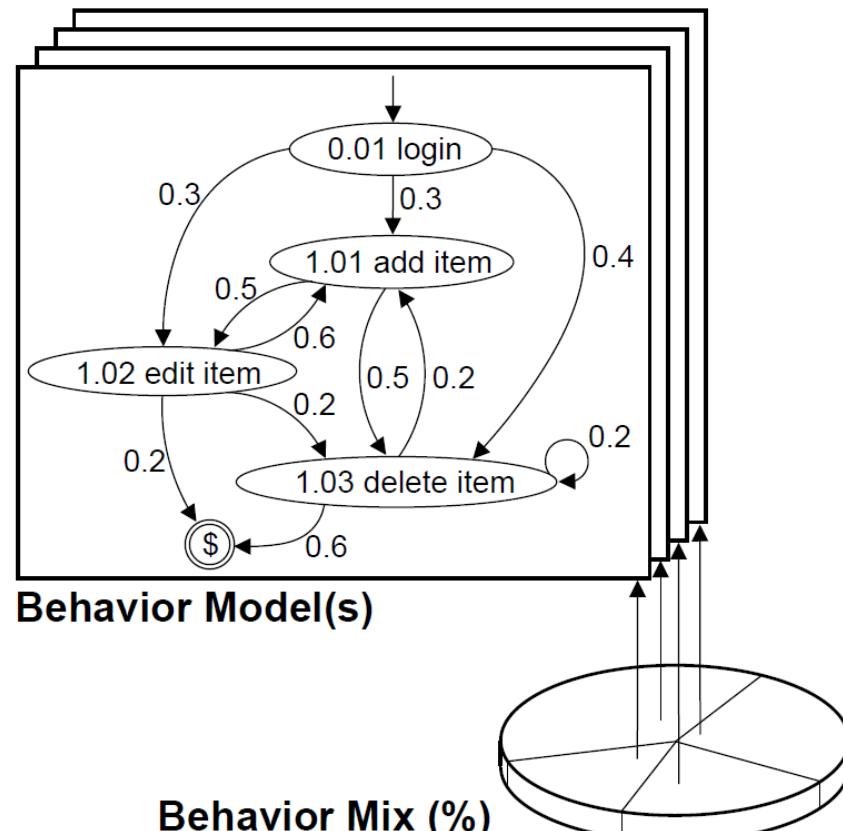
WESSBAS-DSL

Application Model



Behavior Model + Behavior Mix

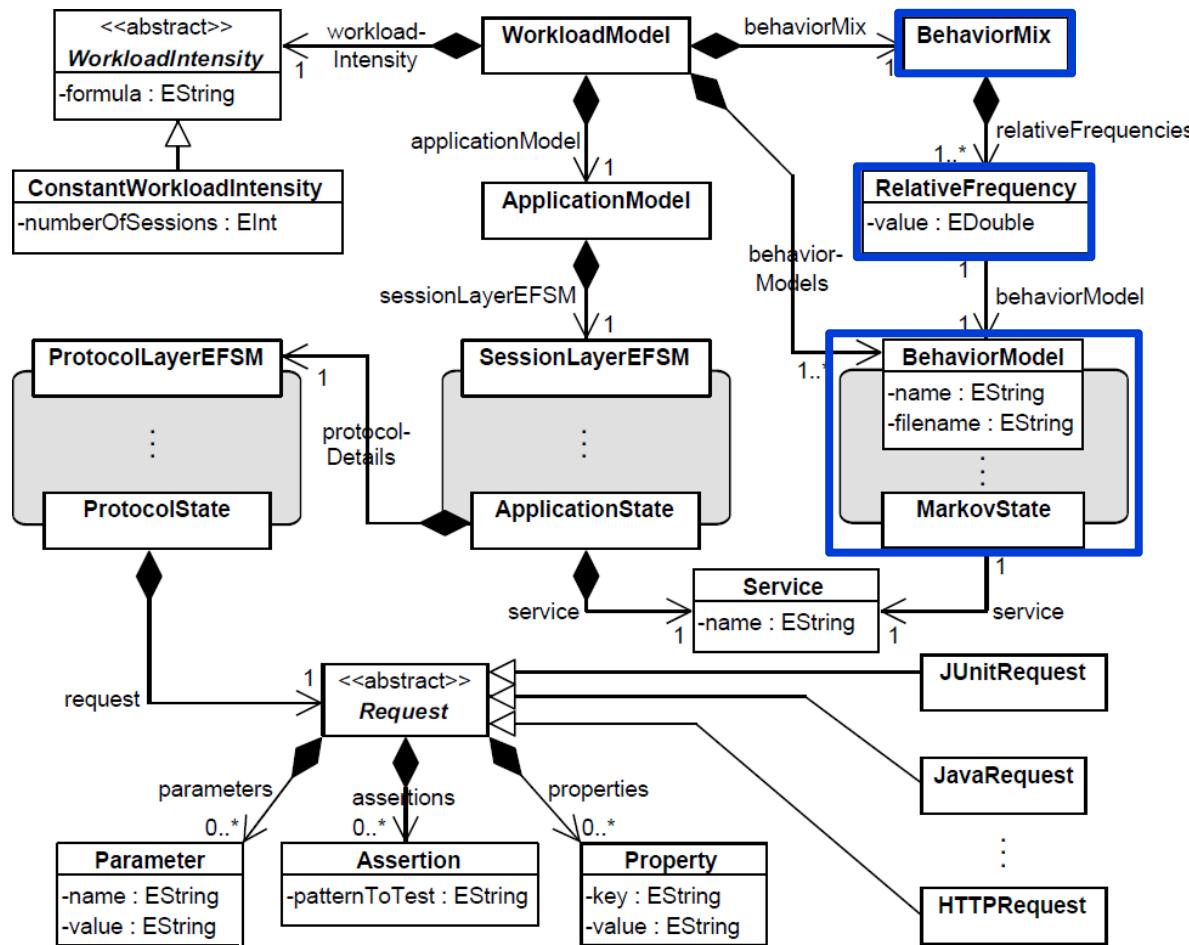
Background – Markov4JMeter



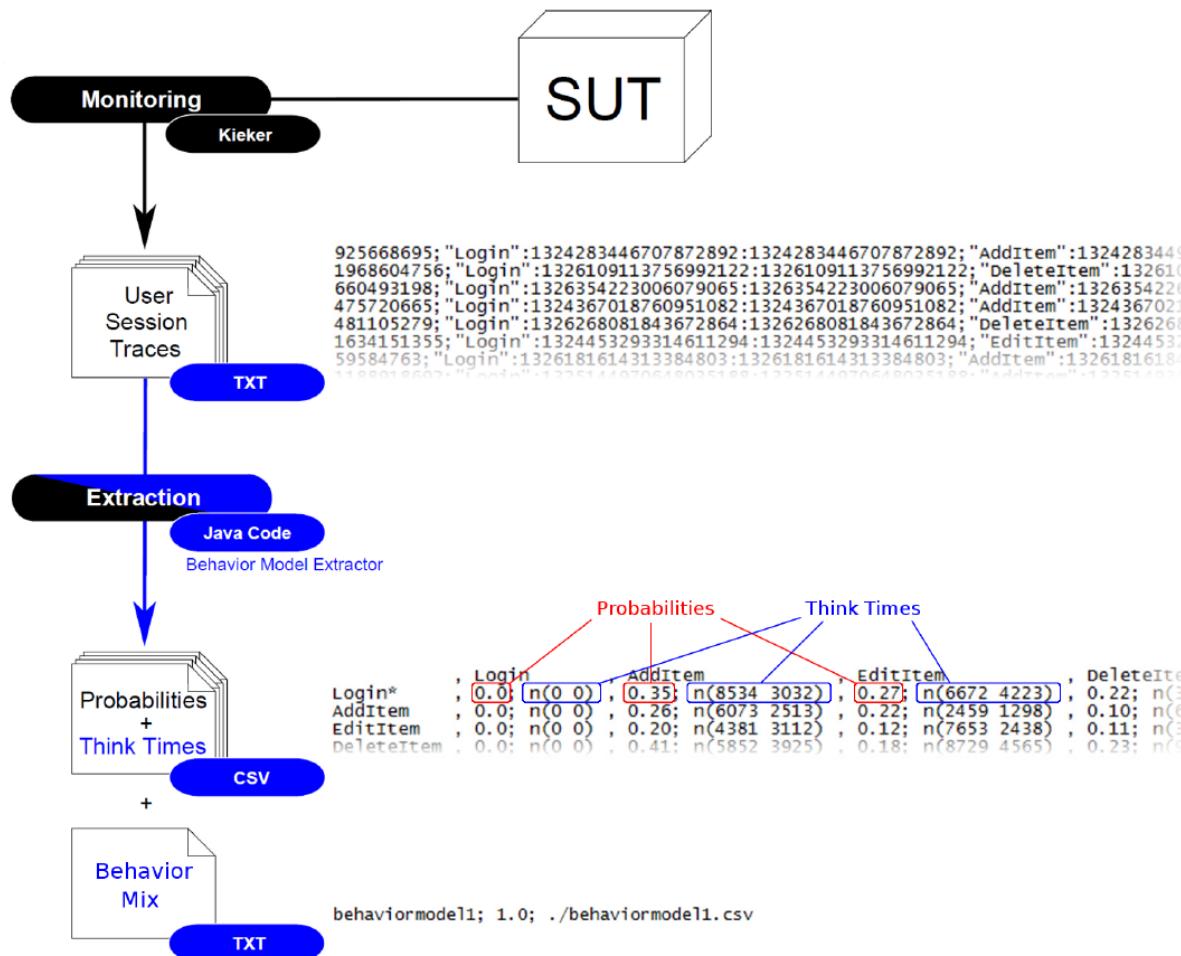
[van Hoorn et al. 2008]

WESSBAS-DSL

Behavior Mix and Behavior Model(s)

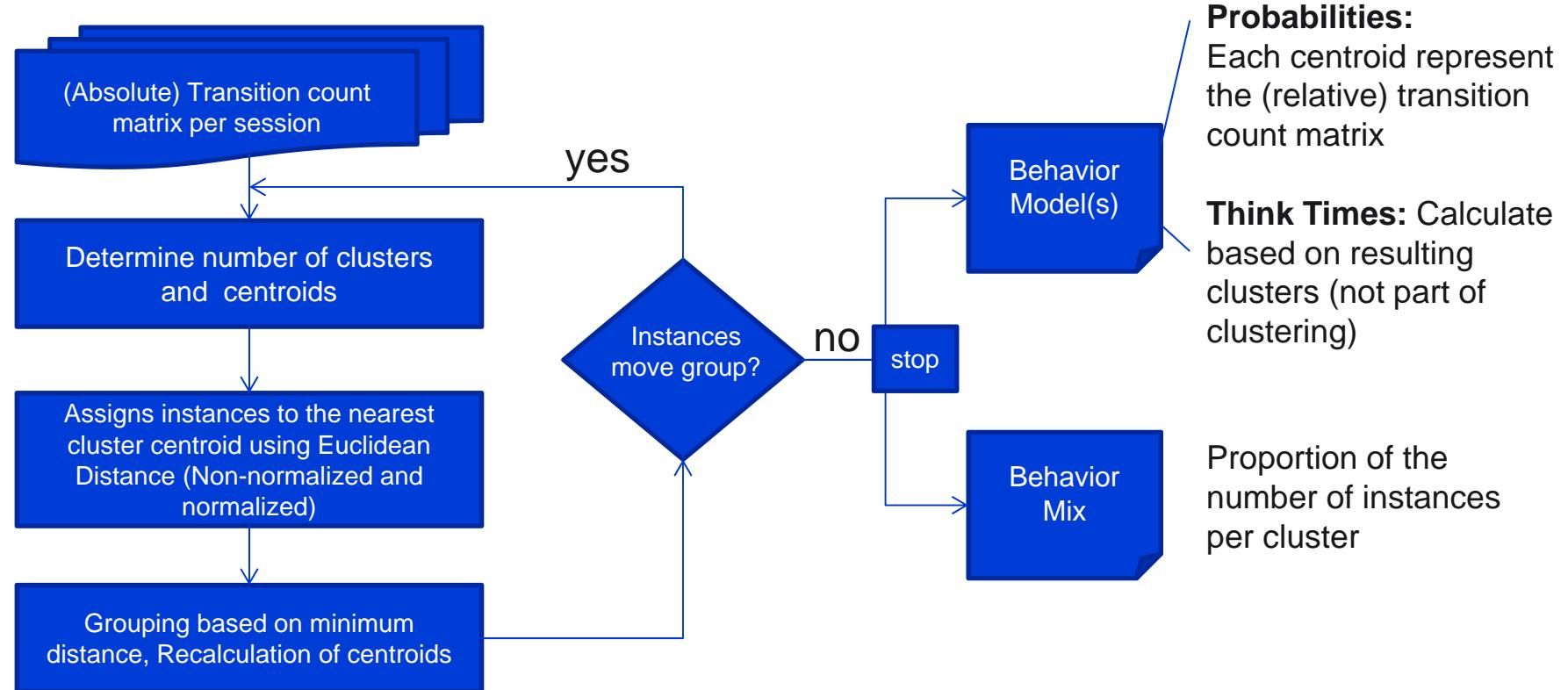


Behavior Model Extraction

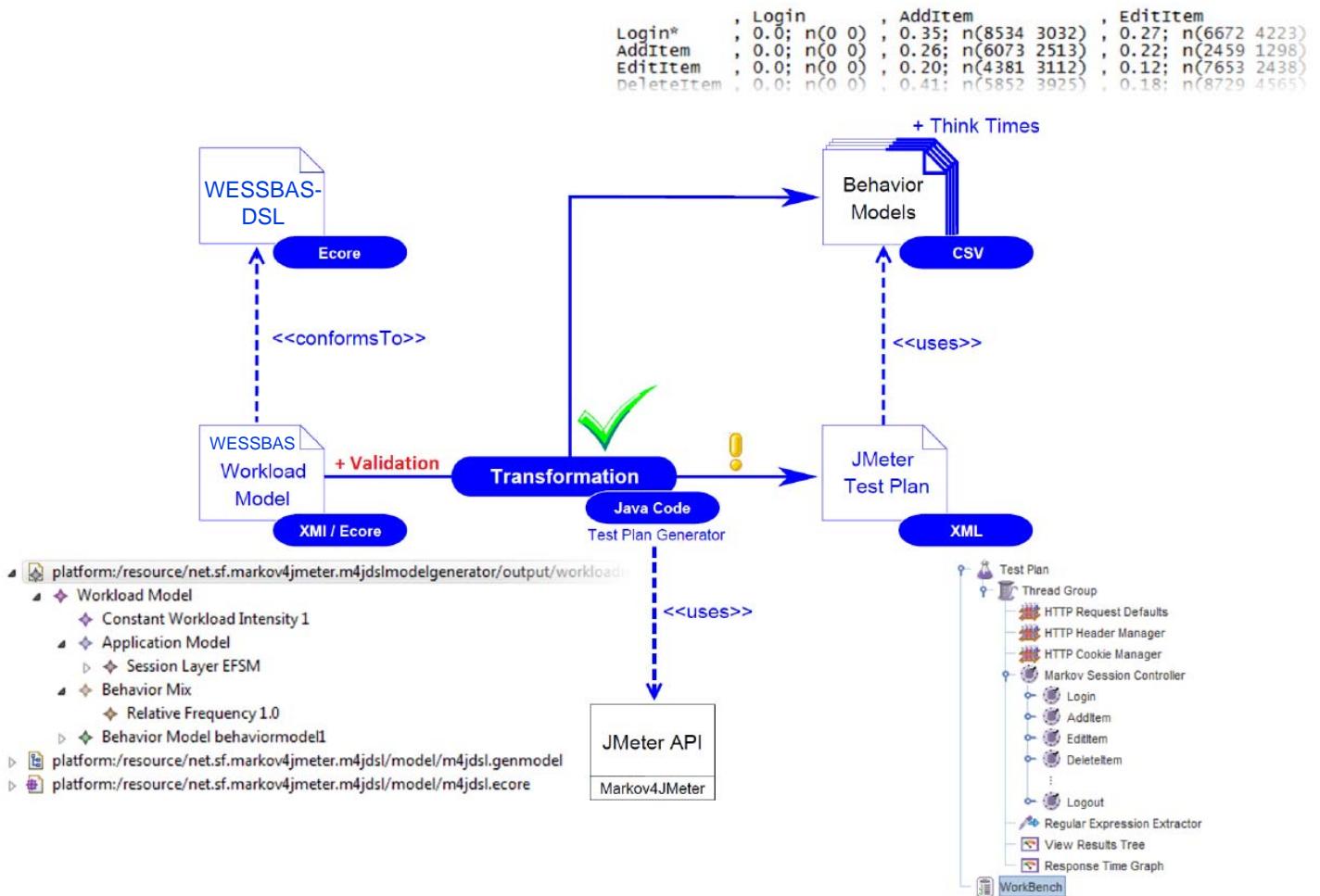


Clustering of Customer Groups

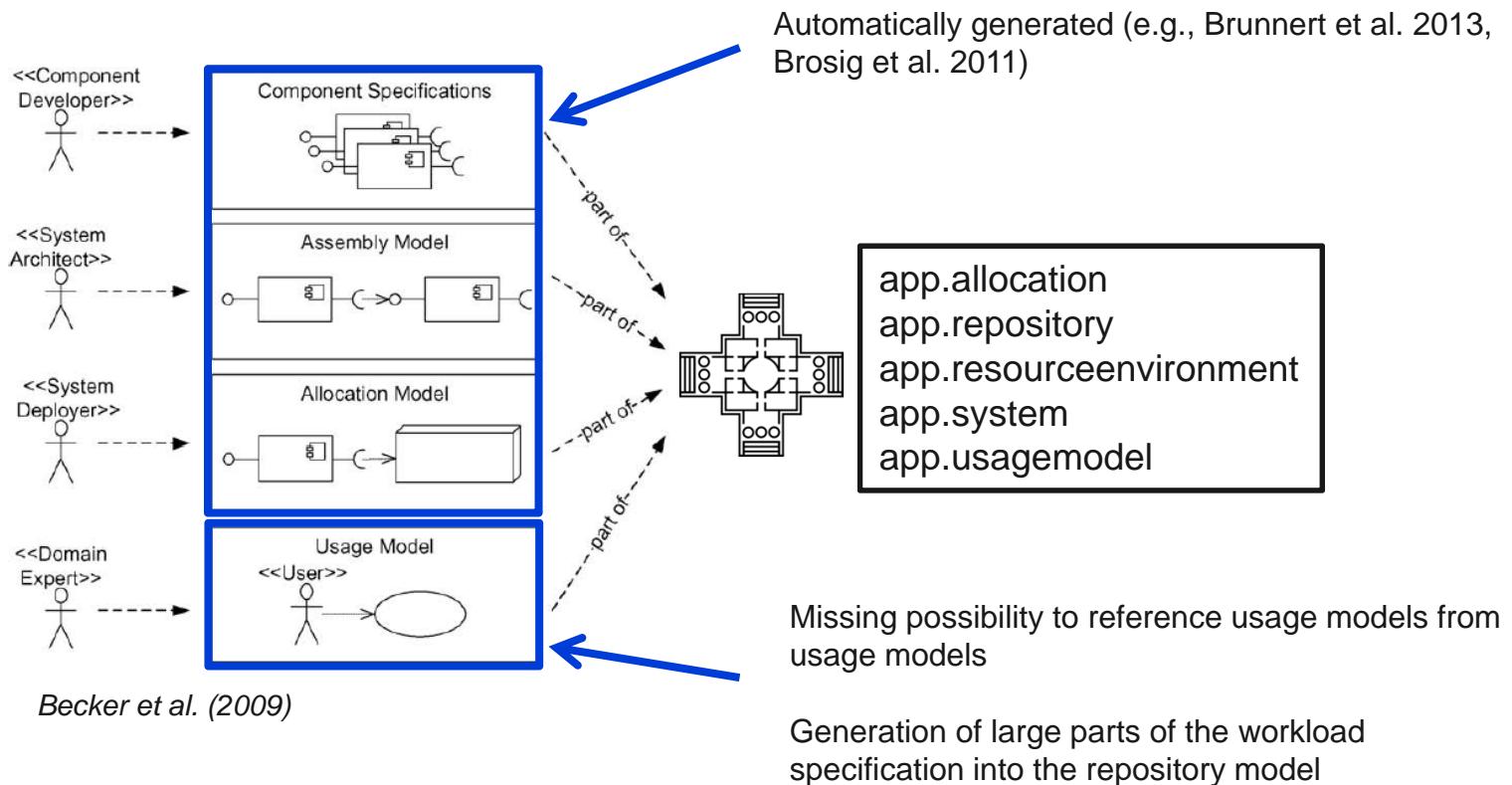
X-means



Transformation into Apache JMeter Test Plans

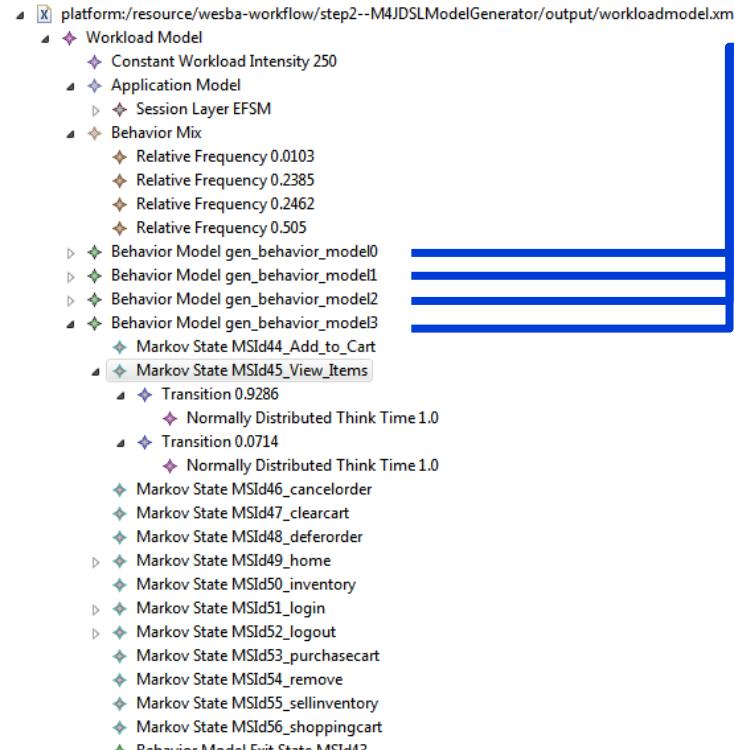


Transformation into Palladio Component Models

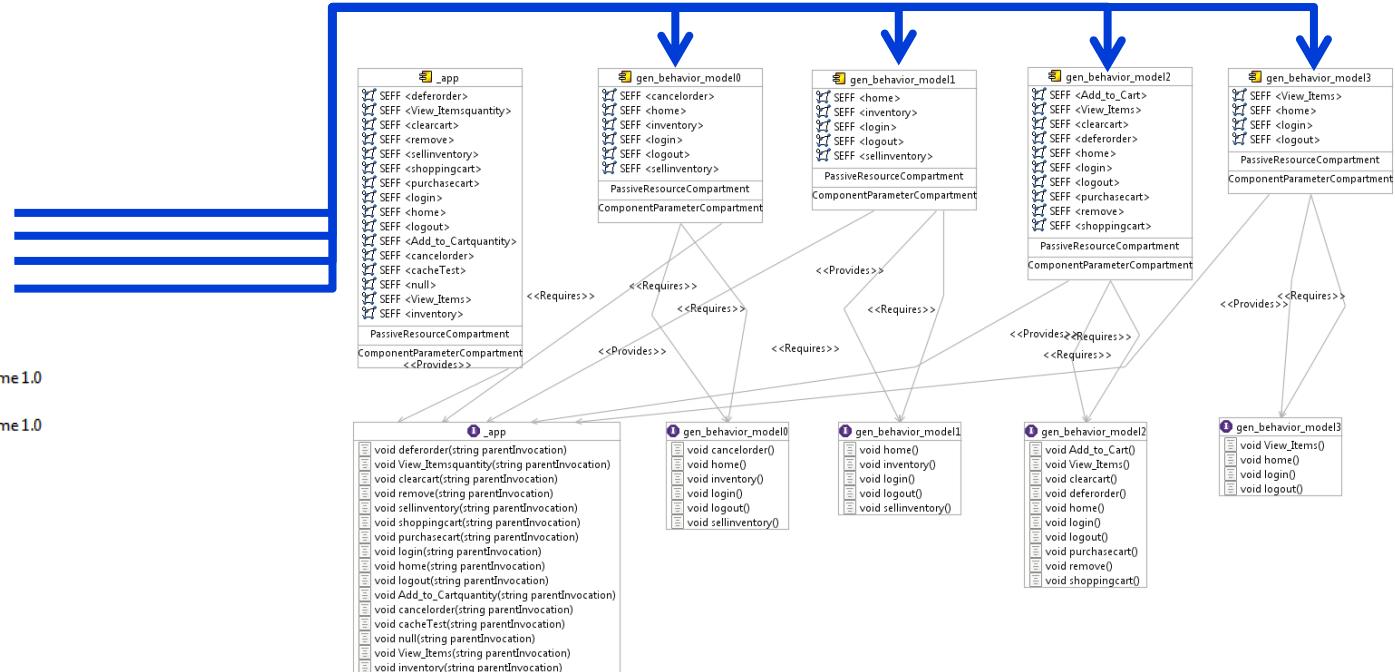


Transformation into Palladio Component Models

WESSBAS-DSL



Properties		Problems	@ Javadoc	Declaration	Search	Console	Call Hierarchy
Property	Value						
EId	MSId45_View_Items						
Service	Service View_Items						



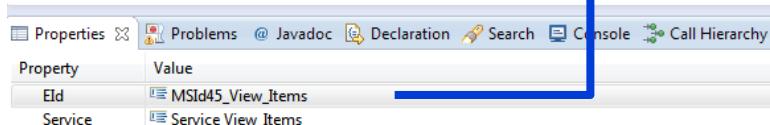
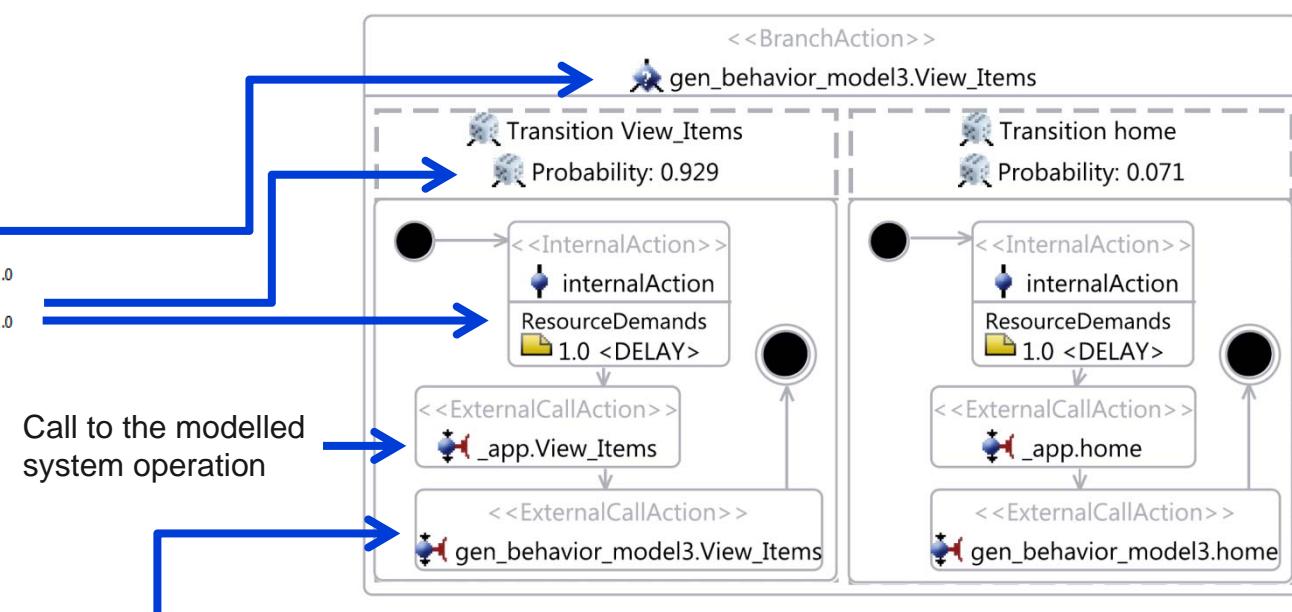
Transformation into Palladio Component Models

WESSBAS-DSL

```

platform:/resource/wesba-workflow/step2--M4JDSLModelGenerator/output/workloadmodel.xmi
  ▾ Workload Model
    ▾ Constant Workload Intensity 250
  ▾ Application Model
    ▾ Session Layer EFSM
  ▾ Behavior Mix
    ▾ Relative Frequency 0.0103
    ▾ Relative Frequency 0.2385
    ▾ Relative Frequency 0.2462
    ▾ Relative Frequency 0.505
  ▾ Behavior Model gen_behavior_model0
  ▾ Behavior Model gen_behavior_model1
  ▾ Behavior Model gen_behavior_model2
  ▾ Behavior Model gen_behavior_model3
    ▾ Markov State MSId44_Add_to_Cart
    ▾ Markov State MSId45_View_Items
      ▾ Transition 0.9286
        ▾ Normally Distributed Think Time 1.0
      ▾ Transition 0.0714
        ▾ Normally Distributed Think Time 1.0
    ▾ Markov State MSId46_cancelorder
    ▾ Markov State MSId47_clearcart
    ▾ Markov State MSId48_deferorder
  ▾ Markov State MSId49_home
  ▾ Markov State MSId50_inventory
  ▾ Markov State MSId51_login
  ▾ Markov State MSId52_logout
  ▾ Markov State MSId53_purchasecart
  ▾ Markov State MSId54_remove
  ▾ Markov State MSId55_sellinventory
  ▾ Markov State MSId56_shoppingcart
  ▾ Behavior Model Exit State MSId43

```



Agenda

- Problem Statement and Overview of Approach
- WESSBAS Approach
 1. WESSBAS-DSL
 2. Extraction of WESSBAS-DSL Instances
 3. Clustering of Customer Groups
 4. Transformations
 1. Apache JMeter
 2. Palladio Component Model
- **Evaluation**
- Future Work

Evaluation Goals and Methodology

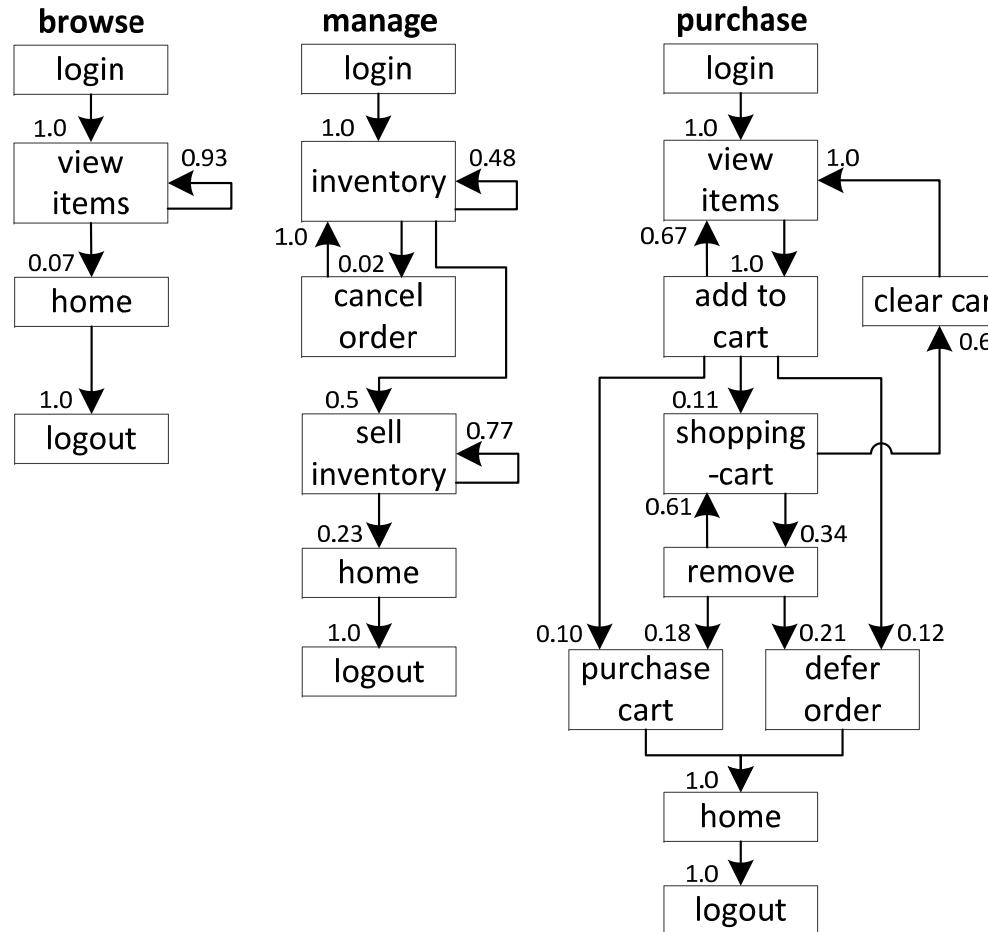
Research Questions

1. How accurately do the clustering results match the input Behavior Mix?
2. What is the impact of the clustering results on the workload characteristics?
3. How well match the predicted workload characteristics the measured workload characteristics?

Methodology

- Instrumentation of SPECjEnterprise2010 using Kieker to obtain session logs
- Extraction of behavior models and behavior mix (includes clustering)
- Extraction of WESSBAS-DSL instances
- Scenario 1: Transformation to JMeter test plans
 - Generation of basic application model (only session layer)
 - No input data, no guards and actions
- Scenario 2: Transformation of WESSBAS-DSL instances to PCM
 - Execute simulation
 - Compare simulation results (request counts) with measured data

Probabilistic Representation of SPECj Workload



Accuracy of Clustering

Research Question 1

		X-means (min 3 cluster, max 3 cluster)						X-means (min 2 cluster, max 20 cluster)							
TM	T	ED			NED			ED			NED			MC	N
		C1	C2	C3	MC	C1	C2	C3	MC	C1	C2	C3	C4		
50	B	0	0	31,060	2.91%	0	31,060	0	0%	0	31,060	0	31,060	24.62%	61,500
25	M	15,298	0	0		15,298	0	0		15,298	0	632	14,666		
25	P	1,789	13,353	0		0	0	15,142		15,142	0	0	15,142		

TM: Transaction Mix

T: Transaction

C_N: Assigned Cluster

MC: Percentage of misclassified

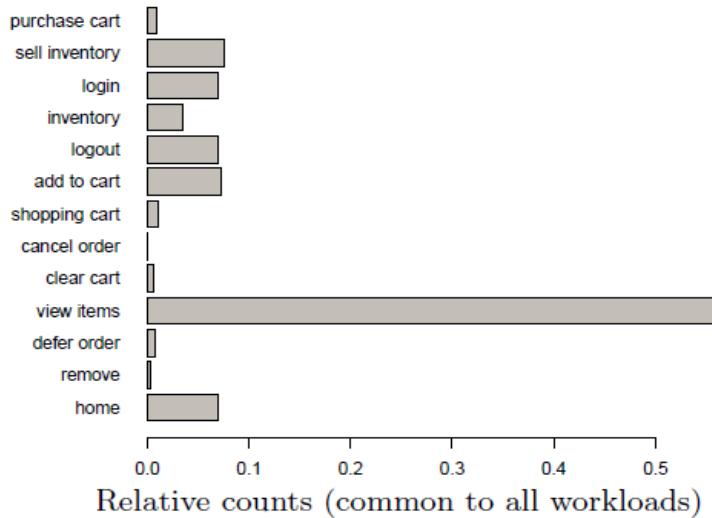
N: Number of instances

ED: Euclidean Distance

NED: Normalized Euclidean Distance

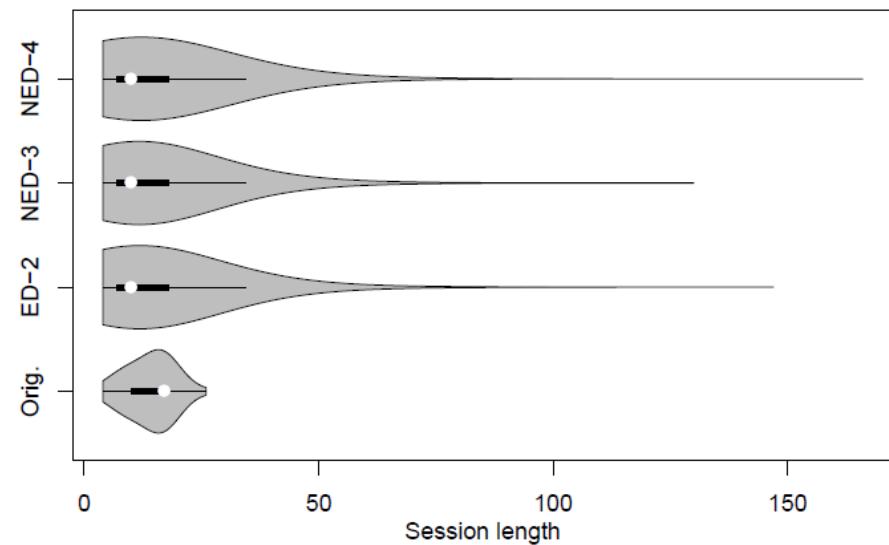
Accuracy of Extracted Workload Specifications

Research Question 2



	Request	Orig.	ED-2	NED-3	NED-4	Rel.
1	add to cart	63,761	63,316	64,250	61,838	0.07
2	cancel order	632	607	634	591	0.00
3	clear cart	6,047	5,941	6,140	5,843	0.01
4	defer order	6,782	6,799	6,863	6,651	0.01
5	home	59,934	60,957	62,054	59,971	0.07
6	inventory	30,596	30,212	31,378	29,808	0.03
7	login	61,500	60,957	62,054	59,971	0.07
8	logout	59,934	60,957	62,054	59,971	0.07
9	purchase cart	8,360	8,328	8,351	8,139	0.01
10	remove	3,027	2,993	3,044	3,064	0.00
11	sell inventory	66,679	65,413	67,691	64,794	0.08
12	shopping cart	9,074	8,934	9,184	8,907	0.01
13	view items	498,601	492,675	499,983	485,611	0.57
	Σ	874,927	868,089	883,680	855,159	1.00

Absolute and relative counts



Violin plot (combination of box and density plot)

Accuracy of PCM Workload Specification

Research Question 3

	Request	Orig.	2 Behavior Models		3 Behavior Models		4 Behavior Models	
		MRC	SRC	PE	SRC	PE	SRC	PE
1	add to cart	63,761	64,943	1.82%	61,812	3.15%	60,986	4.55%
2	cancel order	632	609	3.78%	661	4.39%	625	1.12%
3	clear cart	6,047	6,178	2.12%	5,927	2.02%	5,846	3.44%
4	defer order	6,782	6,873	1.32%	6,524	3.95%	6,606	2.66%
5	home	59,934	61,146	1.98%	58,747	2.02%	58,744	2.03%
6	inventory	30,596	30,539	0.19%	29,574	3.46%	29,405	4.05%
7	login	61,500	61,156	0.56%	58,747	4.69%	58,745	4.69%
8	logout	59,934	61,146	1.98%	58,747	2.02%	58,744	2.03%
9	purchase cart	8,360	8,388	0.33%	7,976	4.81%	7,836	6.69%
10	remove	3,027	2,986	1.37%	2,876	5.25%	2,949	2.64%
11	sell inventory	66,679	66,131	0.83%	63,185	5.53%	63,914	4.33%
12	shopping cart	9,074	9,164	0.98%	8,803	3.08%	8,795	3.17%
13	view items	498,601	491,812	1.38%	470,392	6.00%	475,000	4.97%
	Σ	874,927	871,071	0.44%	833,971	4.91%	838,195	4.38%

MRC: Measured Request Count

SRC: Simulated Request Count

PE: Prediction Accuracy

Future Work

- Automatic generation of application model → Executable load tests
 - Automatic learning of guards and actions
 - Generation of protocol layer
 - Extraction and generation of input data
- Support for workload intensity → LIMBO (Kistowski et al. 2014)
- Additional transformation
 - to alternative workload generators
 - to other architecture-level performance models
(e.g., Descartes Modeling Language) (Kounev et al. 2014)
 - from PCM to WESSION-DSL
- Online clustering to detect evolution of behavior mix
- Industrial case study



References

- M. F. Arlitt, D. Krishnamurthy, and J. Rolia. Characterizing the scalability of a large web-based shopping system. ACM TOIT, 1(1):44-69, 2001.
- S. Becker, H. Koziolek, and R. Reussner. The palladio component model for model-driven performance prediction. Journal of Systems and Software, 82(1):3 22, 2009.
- F. Brosig, N. Huber, and S. Kounev. Automated extraction of architecture-level performance models of distributed component-based systems. pages 183 -192. IEEE., 2011
- A. Brunnert, C. Vögele, and H. Krcmar. Automatic performance model generation for java enterprise edition (ee) applications. In M. Balsamo, W. Knottenbelt, and A. Marin, editors, Computer Performance Engineering, 10th European Workshop on Performance Engineering, volume 8168 of LNCS, pages 74-88. Springer Berlin Heidelberg, 2013.
- A. Ciancone, A. Filieri, M. Drago, R. Mirandola, and V. Grassi. Klapersuite: An integrated model-driven environment for reliability and performance analysis of component-based systems. In Objects, Models, Components, Patterns, volume 6705 of LNCS, pages 99-114. Springer Berlin Heidelberg, 2011.
- D. Krishnamurthy, J. A. Rolia, and S. Majumdar. A synthetic workload generation technique for stress testing session-based systems. IEEE TSE, 32(11):868-882, 2006.
- D. A. Menascé, V. A. F. Almeida, R. Fonseca, and M. A. Mendes. A methodology for workload characterization of e-commerce sites. In Proc. EC '99, pages 119-128, 1999.
- D. Pelleg, A. W. Moore, et al. X-means: Extending K-means with ecient estimation of the number of clusters. In ICML '00, pages 727-734, 2000.
- J. v. Kistowski, N. R. Herbst, and S. Kounev. Modeling variations in load intensity over time. In Proc. LT '14, pages 1-4. ACM, 2014.
- S. Kounev, F. Brosig, and N. Huber. The Descartes Modeling Language. Technical report, Department of Computer Science, University of Wuerzburg, 2014.
- M. Shams, D. Krishnamurthy, and B. Far. A model-based approach for testing the performance of web applications. In Proc. SOQUA 2006, pages 54-61. ACM, 2006.
- C. Smith and C. Llado. Performance model interchange format (pmif 2.0): Xml defintition and implementation. In Proc. QEST '04, pages 38-47, Sept 2004.
- A. van Hoorn, M. Rohr, and W. Hasselbring. Generating probabilistic and intensity-varying workload for Web-based software systems. In Proc. SIPEW '08, pages 124-143, 2008.
- **A. van Hoorn, C. Vögele, E. Schulz, W. Hasselbring, and H. Krcmar. Automatic extraction of probabilistic workload specifications for load testing session-based application systems. VALUETOOLS (to appear), 2014.**
- **C. Vögele, A. van Hoorn, and H. Krcmar. Automatic Extraction of Session-Based Workload Specifications for Architecture-Level Performance Models . LT @ ICPE 2015 (to appear), 2015.**
- M. Woodside, D. C. Petriu, D. B. Petriu, H. Shen, T. Israr, and J. Merseguer. Performance by unified model analysis (puma). In Proc. WOSP '05, pages 1-12, New York, NY, USA, 2005. ACM.

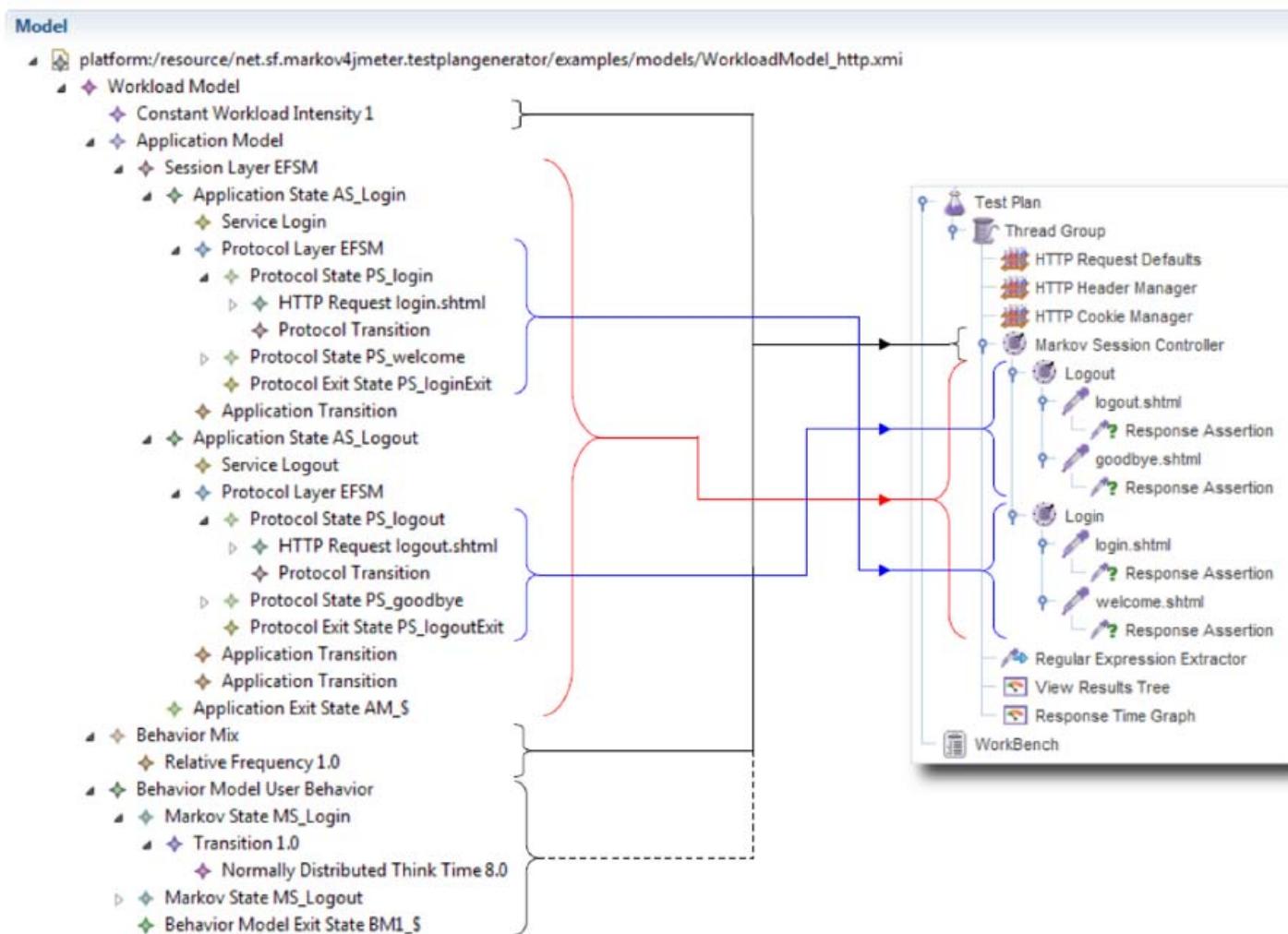
BACKUP

Related Work

- Several approaches for the extraction of workload specifications for session-based applications systems already exist (Arlitt et al. 2001, Krishnamurthy et al. 2006, Menascé et al. 1999) but formats are not envisaged for model-based approaches
- Approaches for the automatic extraction of performance models (Brunnert et al. 2013, Brosig et al. 2011) focus on the generation of system specific details of the SUT
- Several intermediate languages introduced to reduce the complexity of generating different kinds of performance models (Ciancone et al. 2011, Smith et al. 2004, Woodside et al. 2005) provide no support for complex workload specifications

WESSION Approach

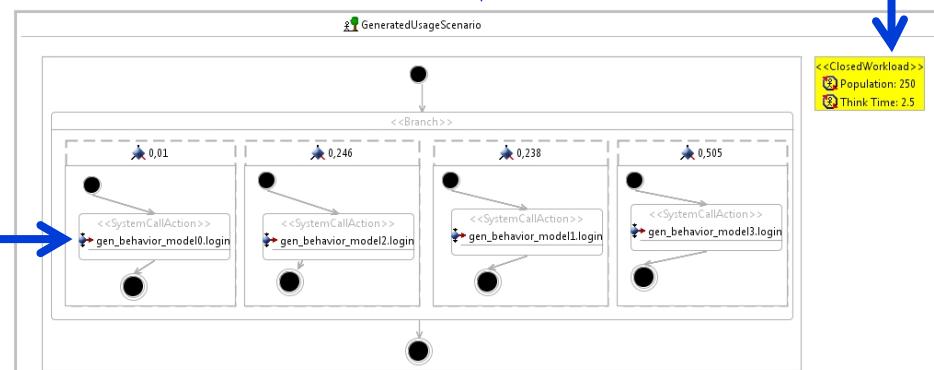
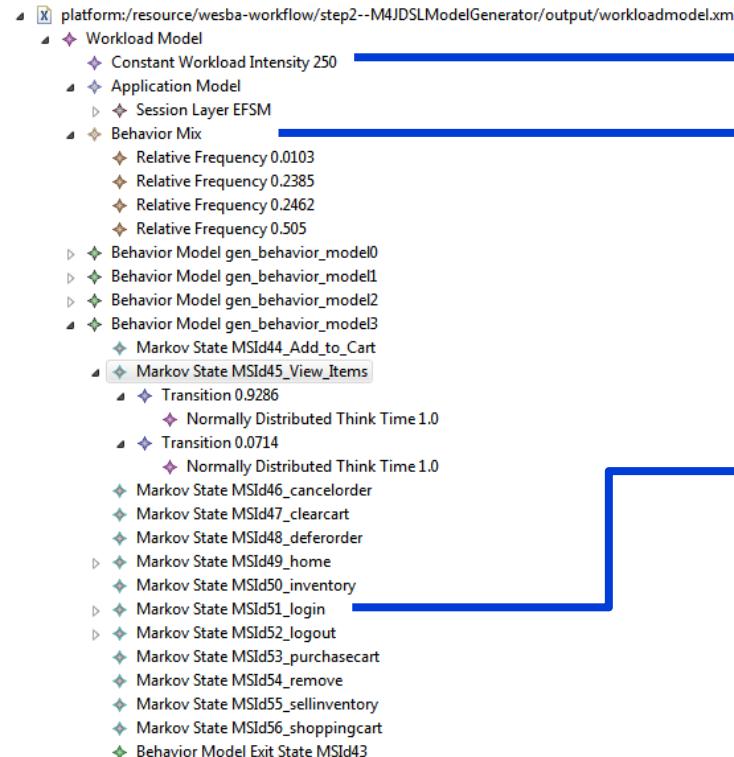
Transformation: WESSION-DSL to Apache JMeter Test



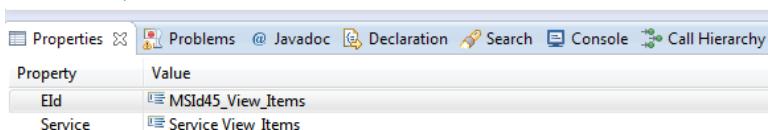
WESSION Approach

Transformation into Palladio Component Models

WESSION-DSL

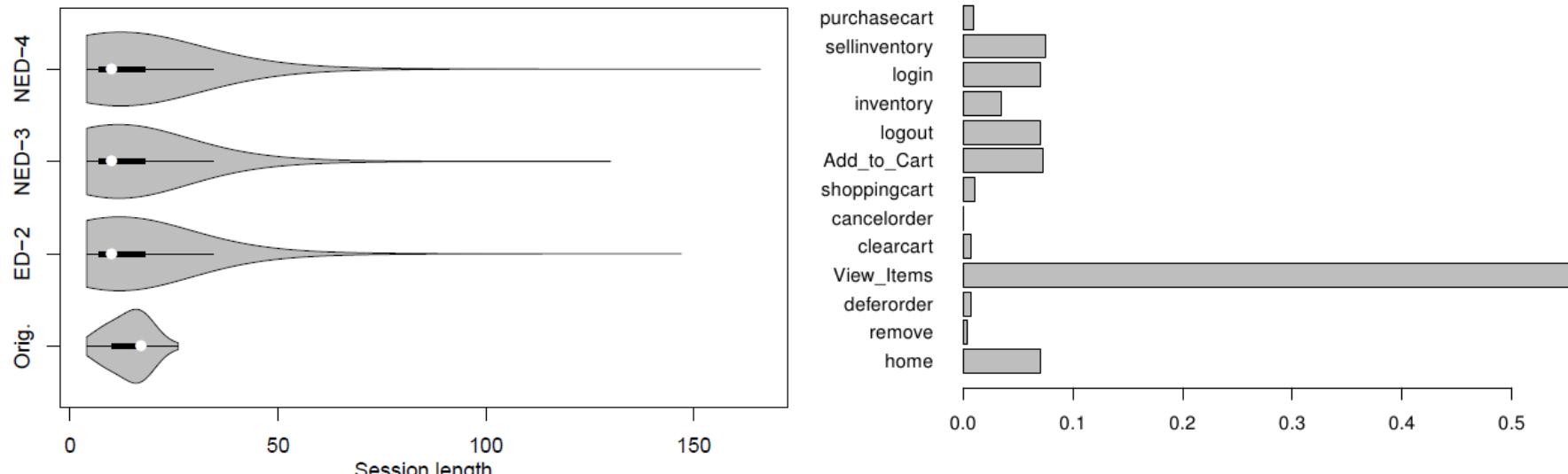


app.usagemodel



Accuracy of Extracted Workload Specifications

Research Question 2



(a) Violin plot (combination of box and density plot)

	Min.	Q ₁	Med.	Mean	CI _{0.95}	Q ₃	Max.	N
Orig.	4	10	17	14.23	[14.19,14.26]	17	26	61,500
ED-2	4	7	10	14.24	[14.15,14.33]	18	147	60,957
NED-3	4	7	10	14.24	[14.15,14.33]	18	130	62,054
NED-4	4	7	10	14.26	[14.17,14.35]	18	166	59,971

(b) Summary statistics