05.09.2014

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

Christian Vögele, André van Hoorn, Eike Schulz

SPEC RG DevOps Performance WG

fortiss GmbH
An-Institut Technische Universität München

Challenge: Specifying Representative Workoads

Situation

- Workload generation is essential to systematically evaluate performance properties of (session-based) software systems

Complication

- Manual creation of representative workload specifications is difficult, time consuming and error-prone
- Extraction and specification of workloads strongly depends on the used workload generation tool

Resolution

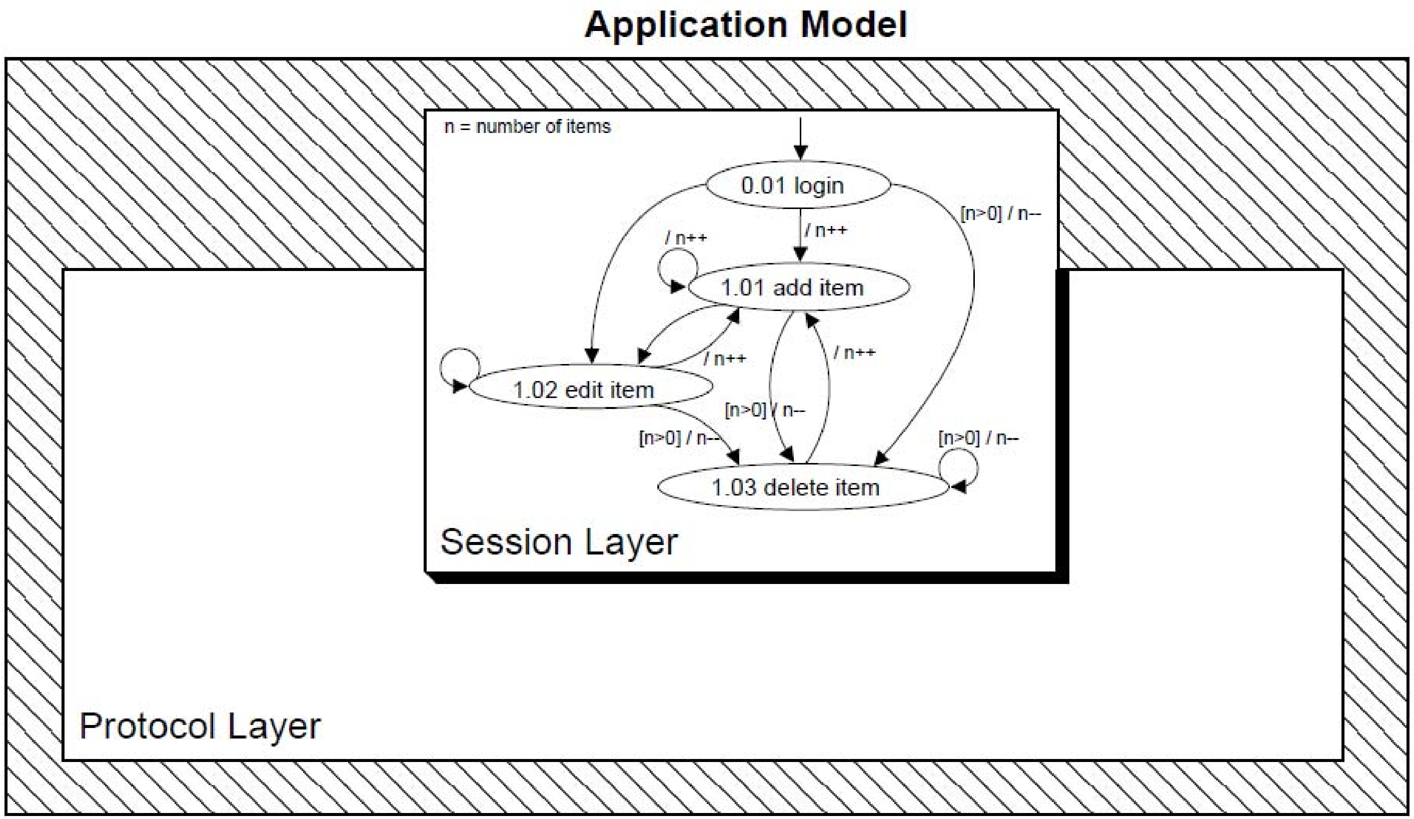
- Approach for systematically extracting probabilistic workload specification for session-based software systems from production usage profiles

Agenda

- Motivation
- Background
- Approach
  1. Overview of M4J-DSL
  2. Behavior Model Extraction
  3. Clustering of Customer Groups
  4. M4J-DSL Model Generation
  5. Apache JMeter Test Plan Generation
- Evaluation
- Future Work

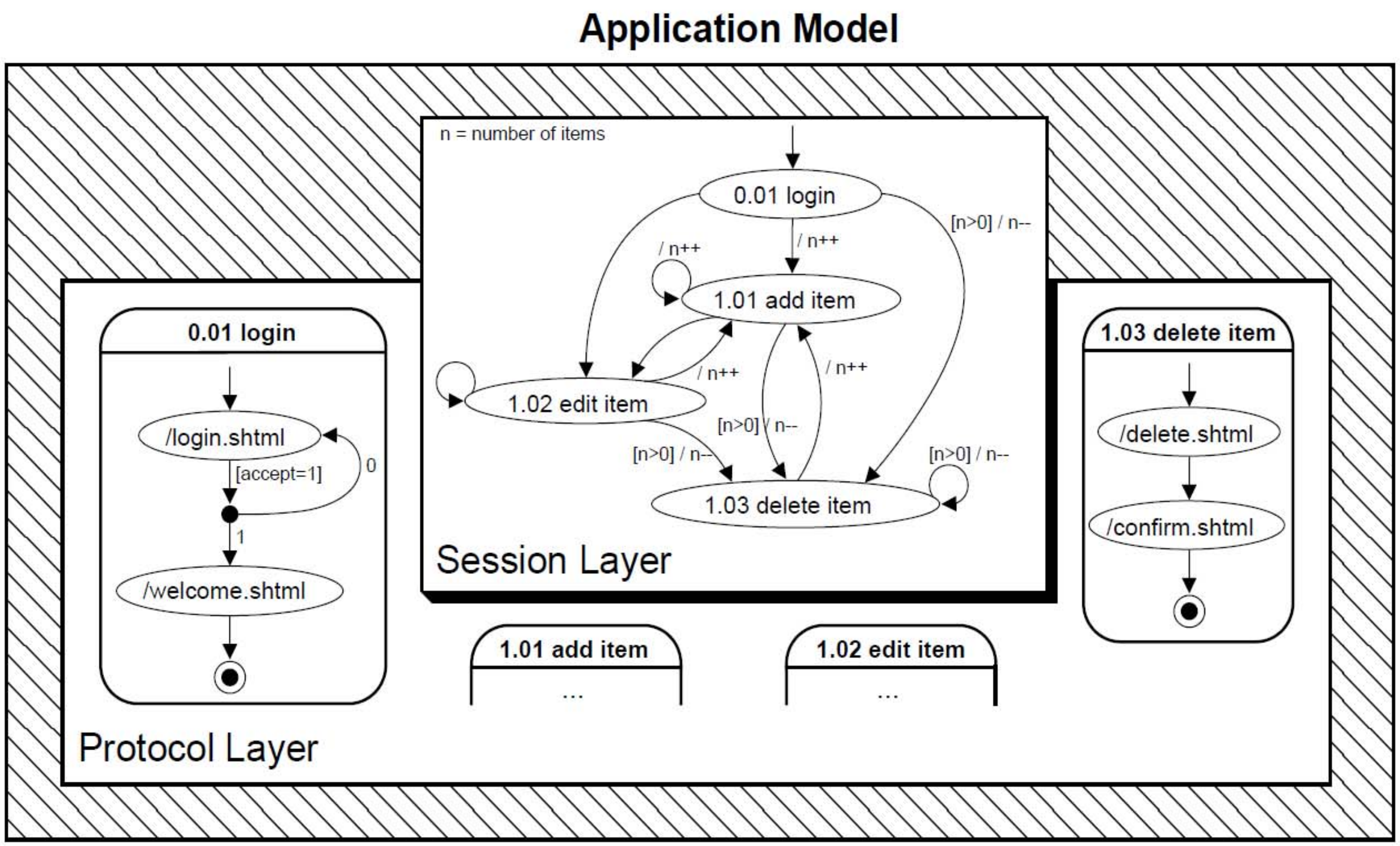
Background – Markov4JMeter [1]

Example of an Application Model



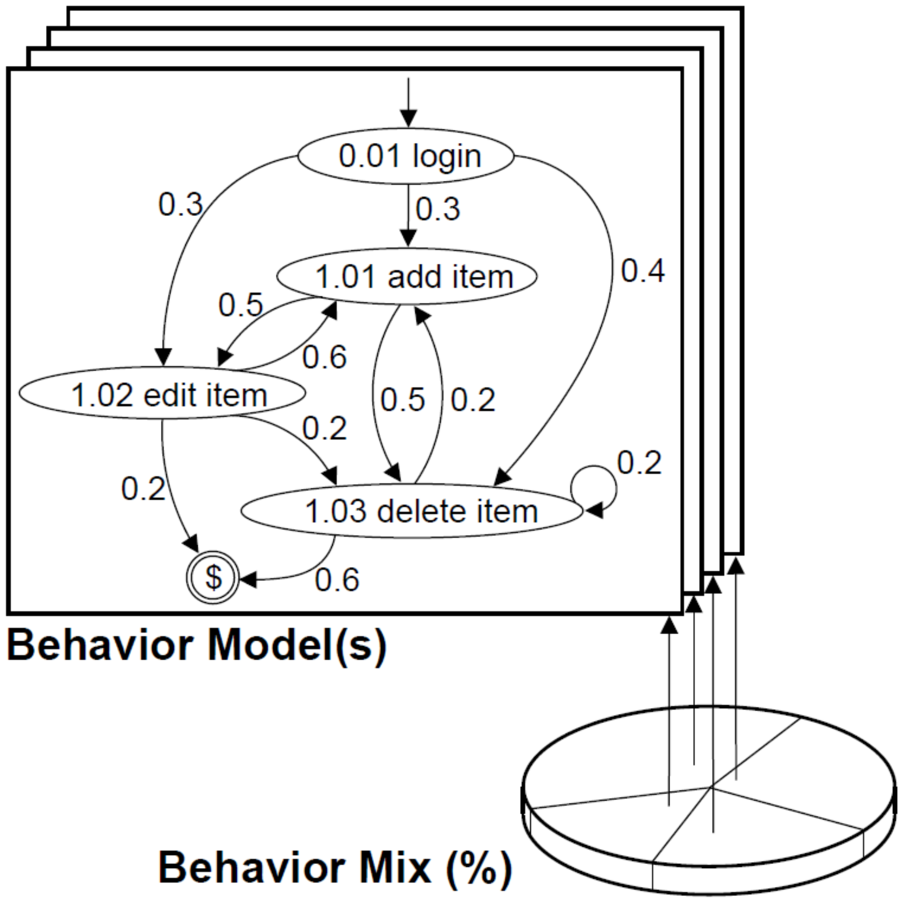
Background – Markov4JMeter [1]

Example of an Application Model



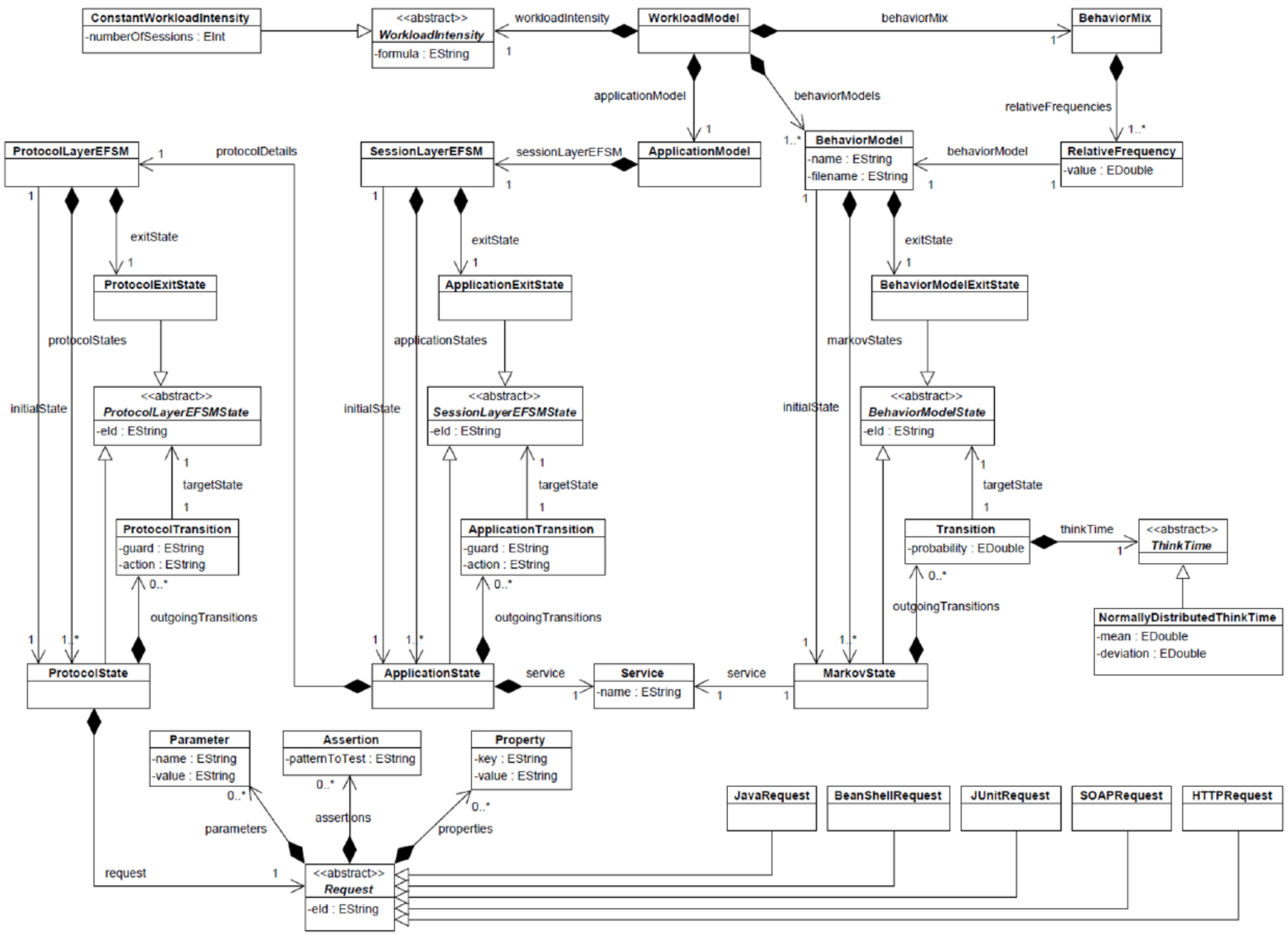
Behavior Model + Behavior Mix

Background – Markov4JMeter



M4J-DSL for Modeling Session-Based Workloads

Approach

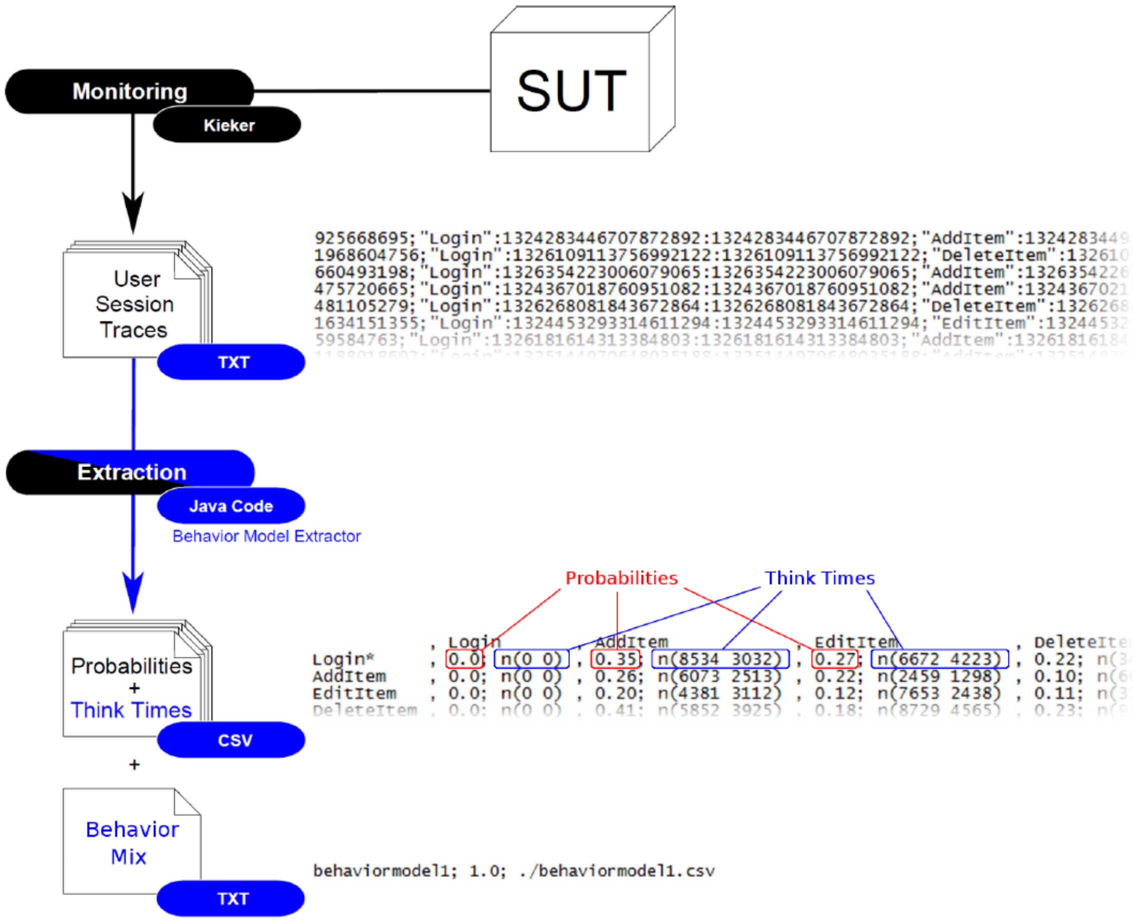


RSS Reliable Software Systems

fortiss

Behavior Model Extraction

Approach



Clustering of Customer Groups

Approach

- Goal: Identification of Behavior Mix
- Advantages of extracting customer groups
  - Comprehensibility of resulting models
  - Understanding navigational pattern of customer groups

    → Optimize navigational structure of application
  - Executing what-if analyses, e.g.,

    How is the performance if the number of heavy buyers increases by X% ?
  - Optimizing paths which are navigated often by heavy buyers

Clustering of Customer Groups

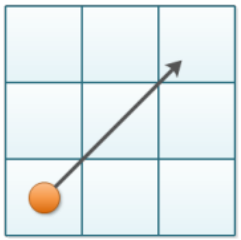
Approach

- Input: Transition count matrix per session
- Think time matrices are not considered for the clustering
- Usage of centroid-based clustering algorithm X-Means
- Steps of the clustering:
    1. For each cluster a central vector, called centroid, will be determined randomly, which represents the instances of this cluster
    2. Centroid comprises the mean attribute values of the instances it represents
    3. Iterate several times over the dataset and assigns instances to the nearest cluster centroid
    4. Continue until no instance is assigned to another cluster

Clustering of Customer Groups (cont’d)

Approach

- Euclidean distance metric (Non-normalized and normalized)



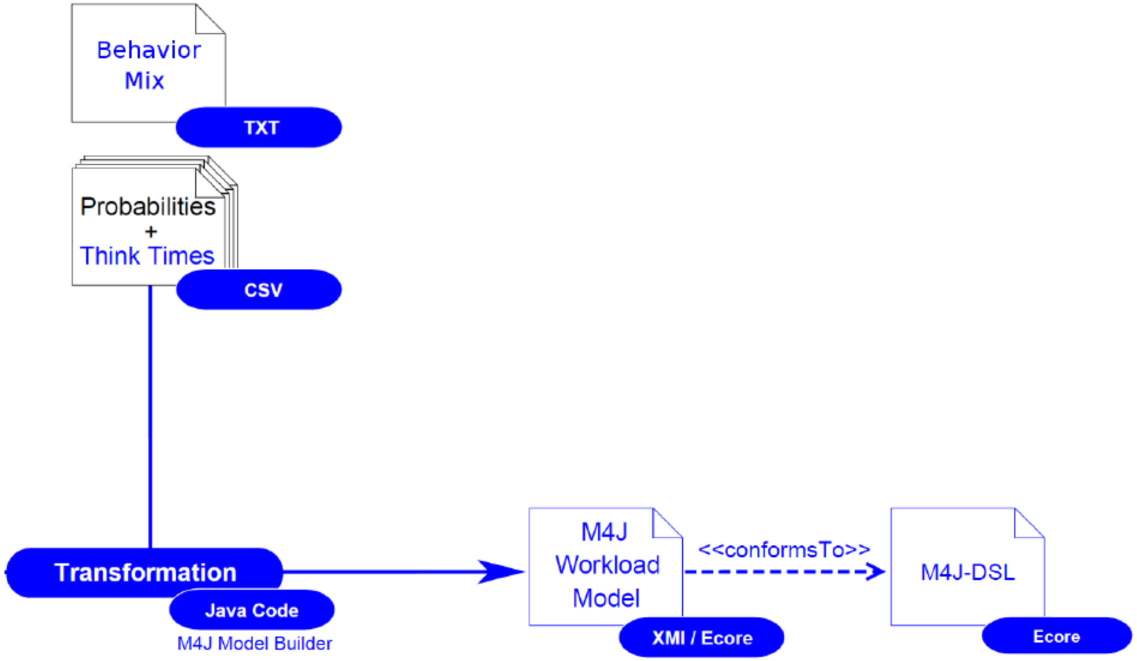
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Intregrate Weka‘s clustering library into our approach

1. Transformation of transition count matrices into a vector
   (Weka cannot handle matrices as input)
2. Execute clustering
3. Transform absolut transition count matrix into relative transition matrix
4. Calculate think time matrices

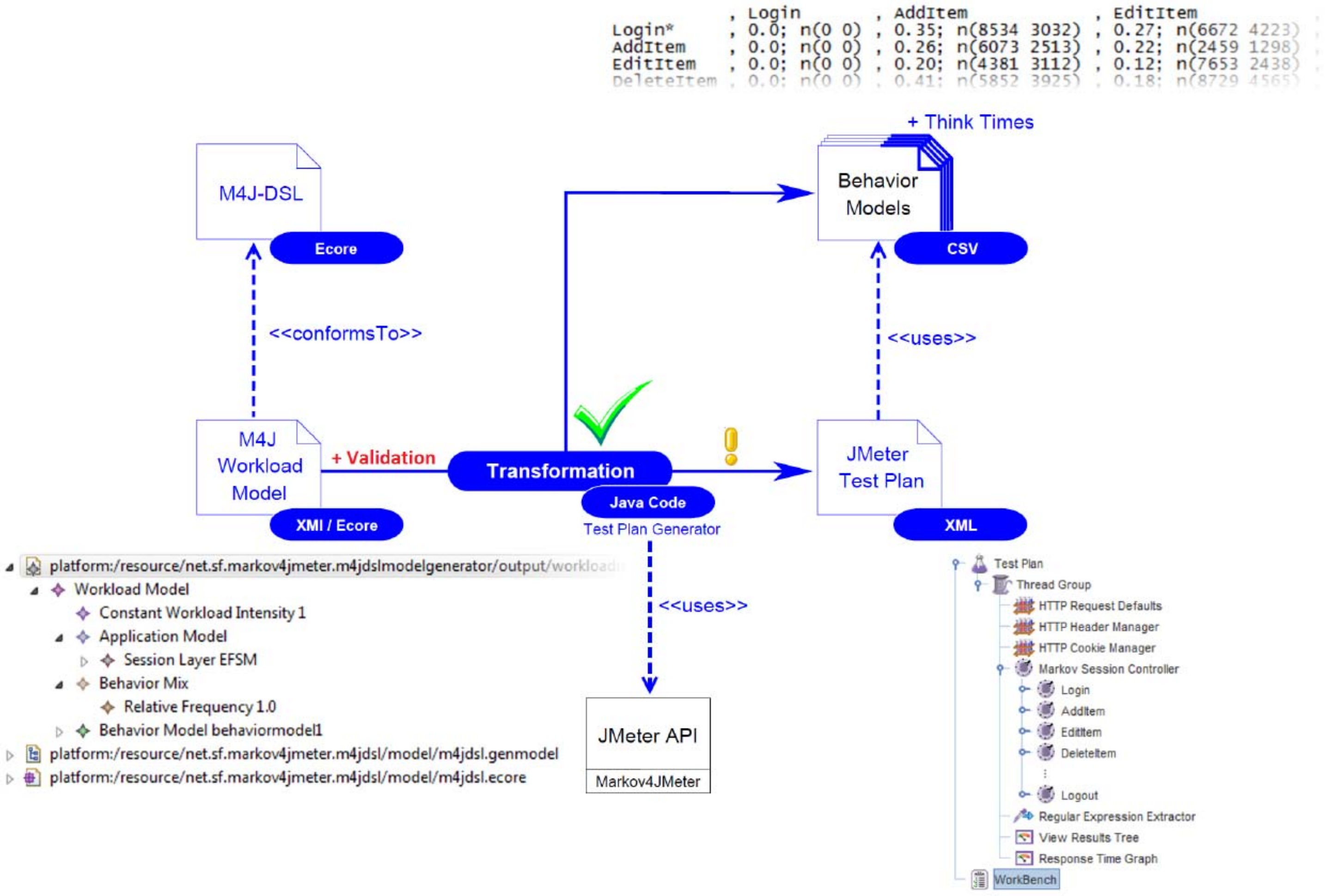
M4J-DSL Model Generation

Approach



Apache JMeter Test Plan Generation

Approach

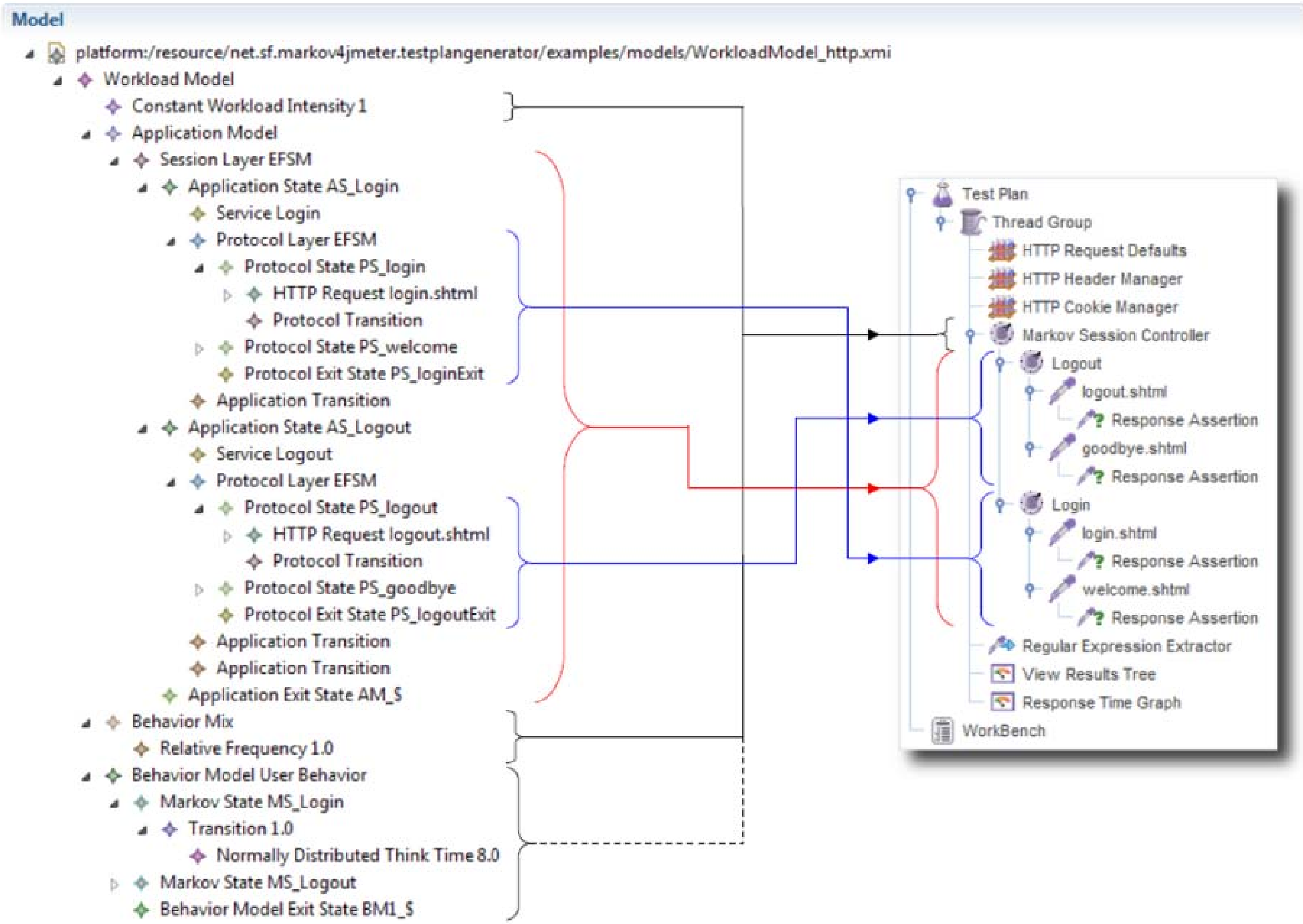


RSS Reliable Software Systems

fortiss

Transformation: M4J-DSL to Apache JMeter Test Plan

Approach



RSS
Reliable Software Systems

fortiss

Evaluation

Methodology

- Instrumentation of SPECjEnterprise2010 with Kieker to obtain session log
- Adapt SPECj dealer driver
  - Login / logout at beginning / end of each transaction
- Extraction of behavior models and behavior mix (includes clustering)
- Transformation of M4J-instances to JMeter test plans
  - Generation of basic application model, only session layer
  - Generation of dummy HTTP requests, e.g., `http://localhost:8080/ActionServlet?type=Add_to_Cart`
  - No input data
  - No guards and actions
- Create dummy web application with actionServlet
- Execute workload on dummy web application and measure workload again with Kieker

Accuracy of Clustering

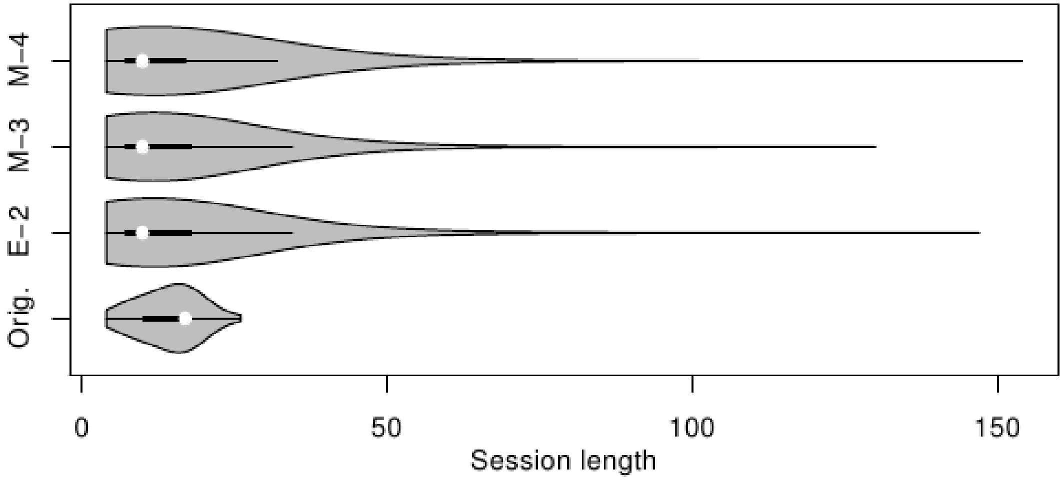
Evaluation

		X-Means (min 3 cluster, max 3 cluster)								X-Means (min 2 cluster, max 20 cluster)								
		Non-Normalized				Normalized				Non-Normalized			Normalized					N
TM	T	C1	C2	C3	MC	C1	C2	C3	MC	C1	C2	MC	C1	C2	C3	C4	MC	
50	B	0	0	31,060		0	31,060	0		0	31,060		0	0	0	31,060		
25	M	15,298	0	0	2.91%	15,298	0	0	0%	15,298	0	24.62%	632	14,666	0	0	1.03%	61,500
25	P	1,789	13,353	0		0	0	15,142		15,142	0		0	0	15,142	0		
25	B	15,091	0	0		15,091	0	0		0	15,091		0	15,091	0	0		
25	M	0	0	15,000	15.98 %	0	15,000	0	0%	15,000	0	24.96%	0	0	707	14,293	15.30%	60,089
50	P	0	20,397	9,601		0	0	29,998		29,998	0		21,513	8,485	0	0		
25	B	0	15,231	0		15,231	o	0		0	15,231		0	0	0	15,231		
50	M	30,510	0	0	2.99%	0	30,510	0	0%	30,510	0	25.16%	29,375	1,135	0	0	1.86%	61,118
25	P	1,824	0	13,553		0	0	15,377		15,377	0		0	0	15,377	0		

TM: Transaction Mix
T: Transaction
C_N: Assigned Cluster
MC: Percentage of misclassified
N: Number of instances

Accuracy of Extracted Workload Specifications

Evaluation



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

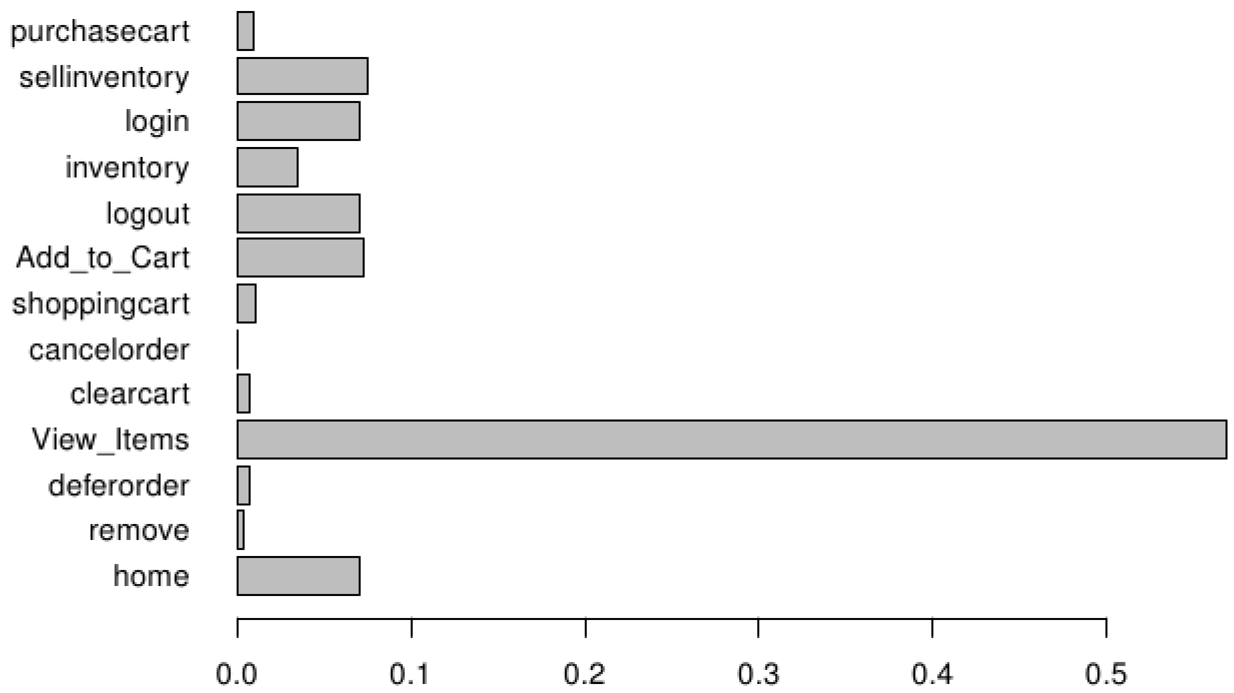
	Min.	Q_1	*Med.*	*Mean*	$CI_{0.95}$	Q_3	*Max.*	*N*
Orig.	4	10	17	14.23	[14.19,14.26]	17	26	61,500
E-2	4	7	10	14.24	[14.15,14.33]	18	147	60,976
M-3	4	7	10	14.24	[14.15,14.33]	18	130	62,054
M-4	4	7	10	14.31	[14.21,14.40]	17	154	60,084

(b) Summary statistics

Figure 7: Session length statistics for the original workload (Orig.) and the synthetic workloads (E-2, M-3, M-4)

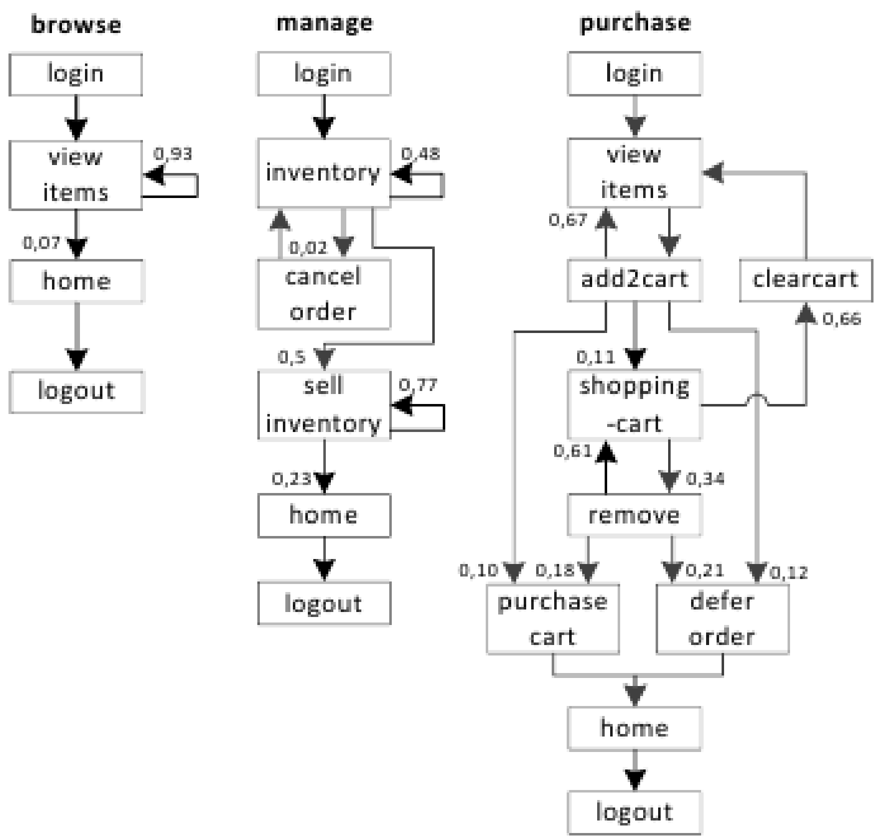
Accuracy of Extracted Workload Specifications

Evaluation



Probabilistic Representation of SPECj Workload

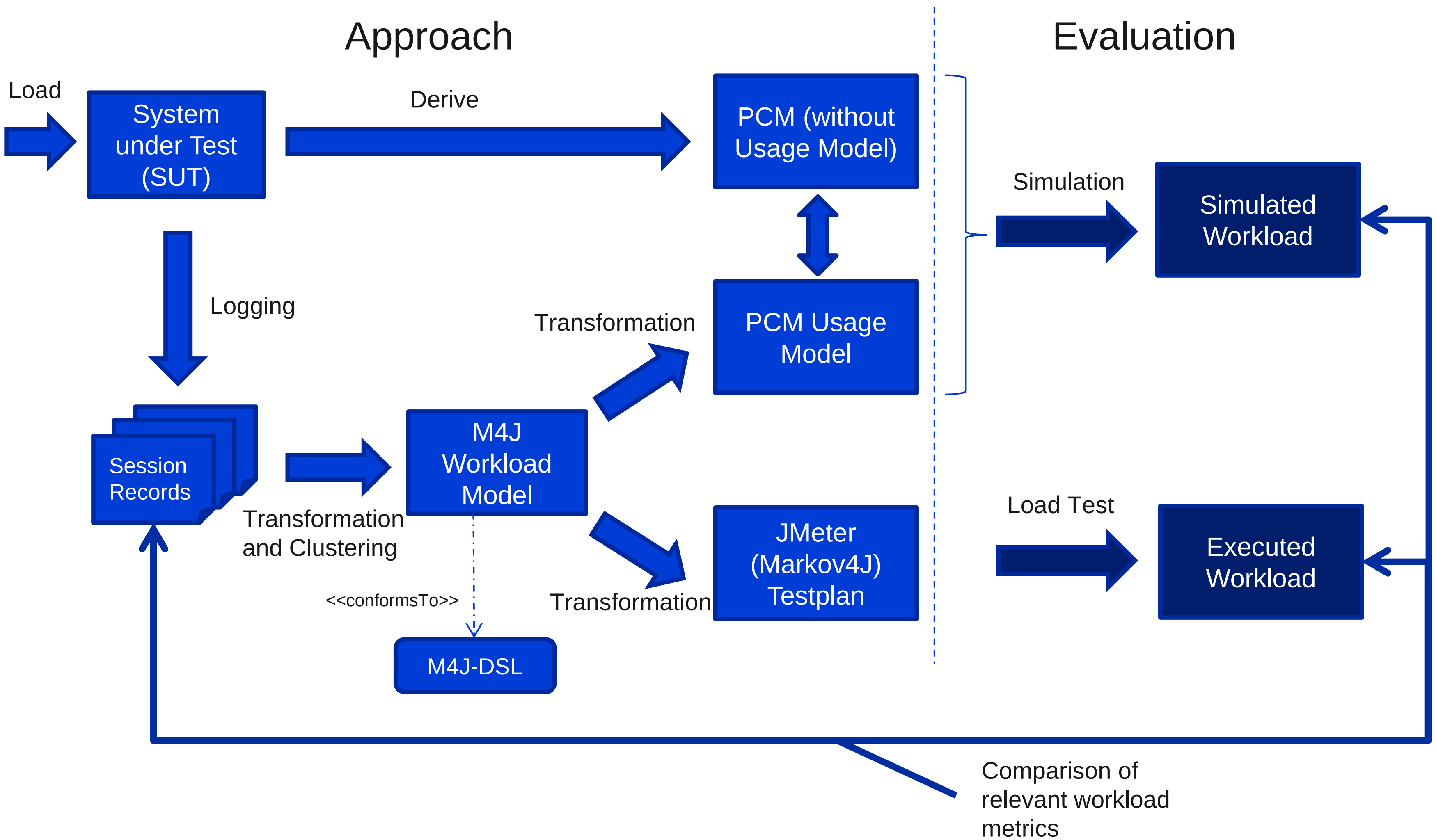
Evaluation



Future Work

- Automatic generation of application model → Executable load tests
  - Automatic learning of guards and actions
  - Generation of protocol layer
  - Generation of input data
- Transformation of DSL instance to PCM
- Support for workload intensity → LIMBO
- Evaluate impact of number of clusters on performance
- Evaluation of other clustering approaches
- Transformation to alternative workload generators
- Online clustering to detect evolution of behavior mix

Future Work


Approach
Evaluation
Load
System under Test (SUT)
Derive
PCM (without Usage Model)
Logging
Transformation
PCM Usage Model
Simulation
Simulated Workload
Session Records
Transformation and Clustering
M4J Workload Model
<<conformsTo>>
M4J-DSL
Transformation
JMeter (Markov4J) Testplan
Load Test
Executed Workload
Comparison of relevant workload metrics

References

1. 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.
2. E. Schulz. Integrating performance tests in a generative software development platform, June 2014. Master’s Thesis, Kiel University, Germany.
3. C. Vögele, A. Brunnert, A. Danciu, D. Tertilt, and H. Krcmar. Using performance models to support load testing in a large SOA environment. In Proc. LT ‘14, 2014.

Backup

Evaluation

Research Questions

RQ1: How practical is the proposed approach for realistic systems

RQ2: How accurately do cluster results match the input workload mix?

RQ3: What is the impact of the clustering results on the session-based workload characteristics ?

RQ4: How accurately do the non-session-based input workload characteristics match the resulting non-session based workload characteristics?