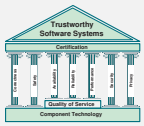
Generating Probabilistic and Intensity-varying Workload for Web-Based Software Systems*

André van Hoorn, Matthias Rohr, and Wilhelm Hasselbring
Contact: van.Hoorn@Informatik.Uni-Oldenburg.DE

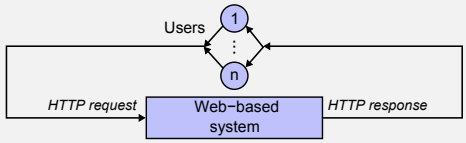
Graduate School TrustSoft and Software Engineering Group
University of Oldenburg, Germany

SPEC International Performance Evaluation Workshop (SIPEW '08)
June 27, 2008 @ Darmstadt, Germany



*This work is supported by the German Research Foundation (DFG), grant GRK 1076/1

Web-Based Software System



- Provides services (i.e., use cases) through a Web Server
  - E.g, "Sign On", "Add Item To Cart", and "Purchase"
  - Web protocols like HTTP
- Service invocation made up by ≥ 1 lower-level (HTTP) requests
- Users alternate between (ON/OFF model by Barford and Crovella (1998))
  1. Submitting requests and
  2. Waiting for a response (+ "thinking")

Workload Generation

- Load tests for performance evaluation of Web-based software systems (Menascé, 2002):
  1. Workload generator mimics users behavior
  2. System performance monitored for later analysis

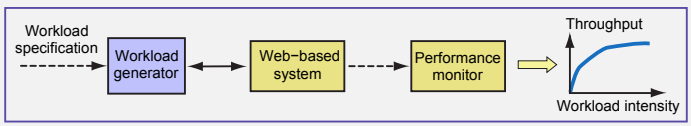


Figure based on (Menascé, 2002)

Workload Generation

- Load tests for performance evaluation of Web-based software systems (Menascé, 2002):
  1. Workload generator mimics users behavior
  2. System performance monitored for later analysis

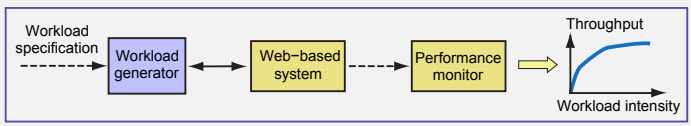


Figure based on (Menascé, 2002)

- Workload generation approaches (Barford and Crovella, 1998):
  - Replay requests from recorded real-world workload
  - Generate requests based on mathematical models

Our Requirements for a Workload Generator

Empirical evaluation of our research in

- Software performance evaluation (van Hoorn, 2007)
- Timing behavior anomaly detection and automatic fault localization (Rohr, 2008)
- Runtime reconfiguration (Matevska and Hasselbring, 2007)

Desired features

- Workload specification should be
  - Maintainable,
  - Reusable, and
  - Application-generic
- Probabilistic user behavior (i.e., interactions with the system)
- Specification of intensity-varying workload intensity

Session and Workload Intensity
Web-Based Software System (cont'd)

Session (Menascé et al., 1999)

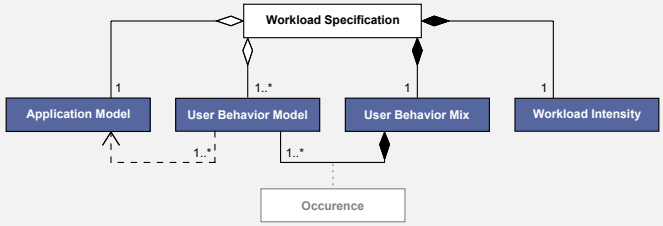
Sequence of related request or service invocations issued by the same user (i.e., during a single visit).

Workload Intensity (in this context)

- Number of active sessions, i.e., no. of concurrent users
- (Implicitly: think time)

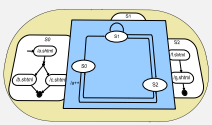
Outline of this Talk

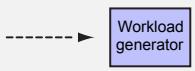
Workload Specification
Elements and Relations



Workload Specification (cont'd)
Overview

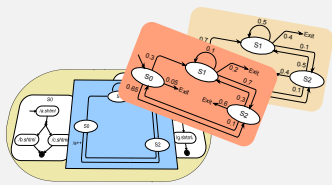
- Application model
  1. Allowed sequences of system interactions within a session
  2. Protocol details required to generate valid requests

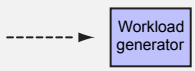




Workload Specification (cont'd)
Overview

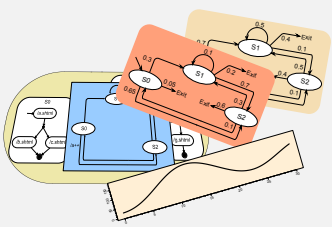
- Application model
  1. Allowed sequences of system interactions within a session
  2. Protocol details required to generate valid requests
- Probabilistic user behavior models (Markov chains)

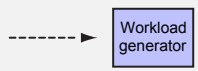




Workload Specification (cont'd)
Overview

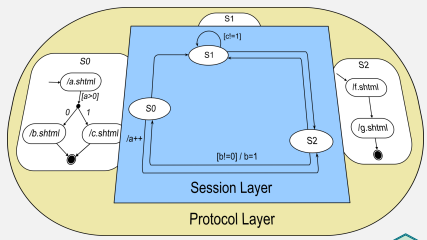
- Application model
  1. Allowed sequences of system interactions within a session
  2. Protocol details required to generate valid requests
- Probabilistic user behavior models (Markov chains)
- Workload intensity specifies number of active sessions as function of elapsed experiment time





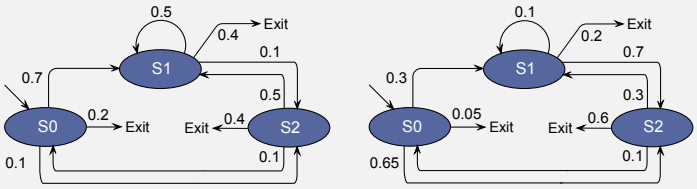
Application Model
Workload Specification (cont'd)

- 2-layered hierarchical finite state machine
- **Session layer**
  allowed sequences of service calls within a session
- **Protocol layer**
  protocol-specific (e.g., HTTP) request details



User Behavior Model
Workload Specification (cont'd)

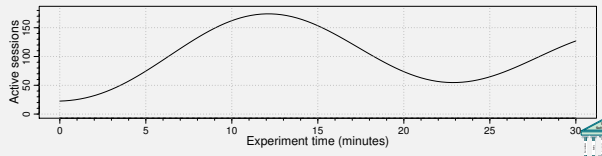
- Markov chains model probabilistic behavior within a session
- States correspond to states (services) of session layer ▸ Appl. Model



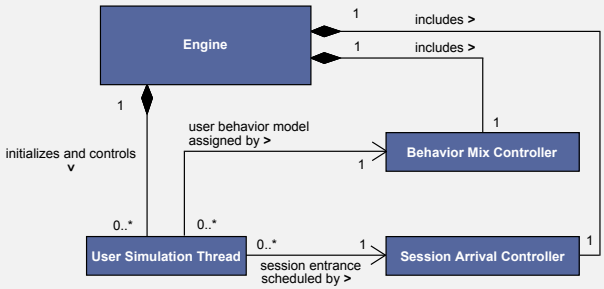
- Application model and user behavior model will be combined into probabilistic session model

User Behavior Mix and Workload Intensity
Workload Specification (cont'd)

- User Behavior Mix
  - Probability of occurrence for each user behavior model
  - Formally, a set $\{(\mathcal{B}_{A,0}, p_0), \ldots, (\mathcal{B}_{A,n-1}, p_{n-1})\}$, $\sum_{i=0}^{n-1} p_i = 1$
- Workload Intensity
  - Number of active sessions, i.e., no. of concurrent users
  - Relative to elapsed experiment time: Function $\mathbb{R}_{\geq 0} \rightarrow \mathbb{N}$



Workload Generation
Architecture of Conceptual Workload Generator



Probabilistic Session Model
Workload Generation (cont'd)

- Application model and user behavior model related by application states and the states of the Markov chain
- Enriching application transitions with probabilities of Markov chain
- Workload generation:
  1. Start with entry state of user behavior model
  2. Next state:
     a. Determine outgoing transitions guards evaluate to *true*
     b. Select transition based on assigned probabilities (scaled)
  3. Execute assigned action
  4. Issue requests according to related protocol layer state machine
  5. Session ends when Exit state reached

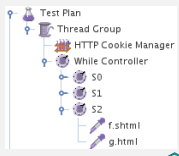
Structure

Tool for Generating Probabilistic and Intensity-Varying Workload

Markov4JMeter

Implementation of our workload generation approach (extension for JMeter).
⟶ http://markov4jmeter.sourceforge.net

- Apache JMeter[1]
  - Popular workload generation tool
  - Workload specified in *Test Plan*
    - (Ordered) tree of *Test Elements*
    - Control flow: *Logic Controllers*
    - Requests: *Samplers* (HTTP, FTP, ...)
  - Test Plan instantiated for each thread



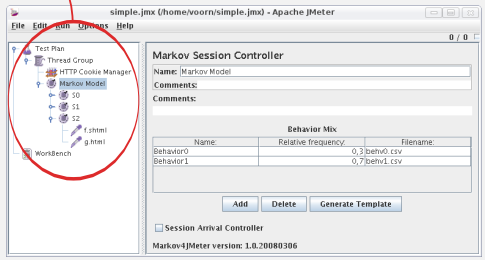
[1] http://jakarta.apache.org/jmeter/

Markov4JMeter
Tool for Generating Probabilistic and Intensity-Varying Workload (cont'd)

- Markov4JMeter allows the definition of probabilistic Test Plans
- Two additional Logic Controllers:
  - Markov Session Controller
    - Root of probabilistic session model within Test Plan
    - GUI dialog: user behavior mix and workload intensity
  - Markov State
    - Corresponds to application state
    - Added underneath Markov Session Controller
    - GUI dialog: transitions with guards and actions
    - Subtree of Test Elements represents protocol layer
- Also:
  - Session Arrival Controller and
  - Behavior Mix Controller
- Markov chains of user behavior models defined in CSV files

Probabilistic Test Plan and Configuration Dialogs
Markov Session Controller

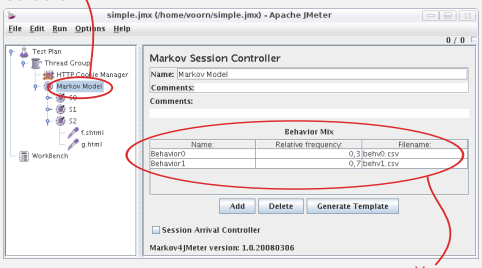
Probabilistic Test Plan



corresponding application model

Probabilistic Test Plan and Configuration Dialogs
Markov Session Controller

Markov Session Controller

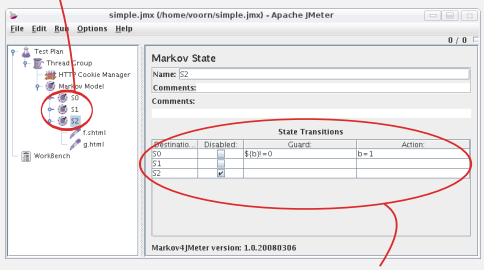


User Behavior Mix

corresponding application model

Probabilistic Test Plan and Configuration Dialogs
Markov State

Markov States



Transitions with guards and actions

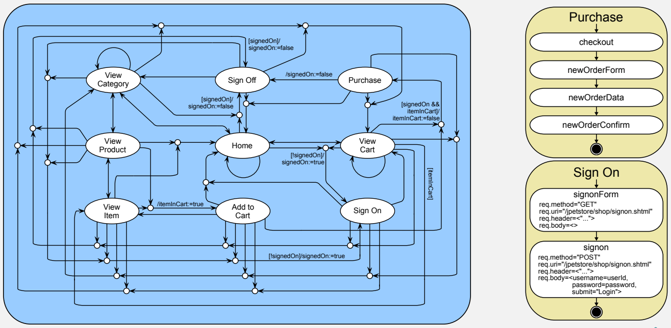
corresponding application model

Structure

Workload Generation for iBATIS JPetStore[1]

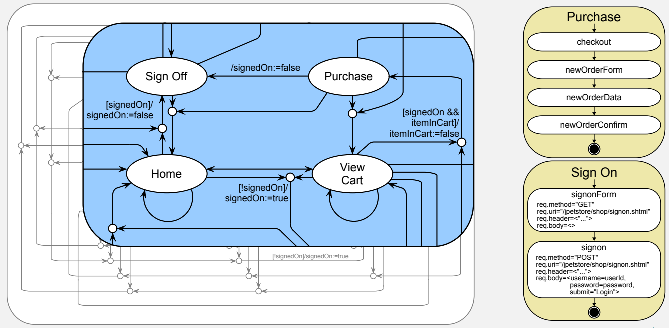
[1] http://ibatis.apache.org

Application Model (Session Layer and 2 Protocol States)
Markov4JMeter Profile for JPetStore (cont'd)



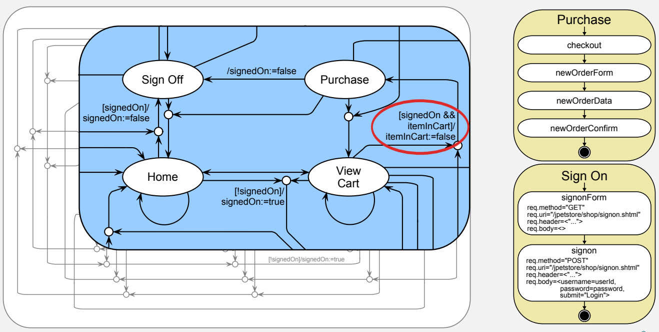
application model

Application Model (Session Layer and 2 Protocol States)
Markov4JMeter Profile for JPetStore (cont'd)



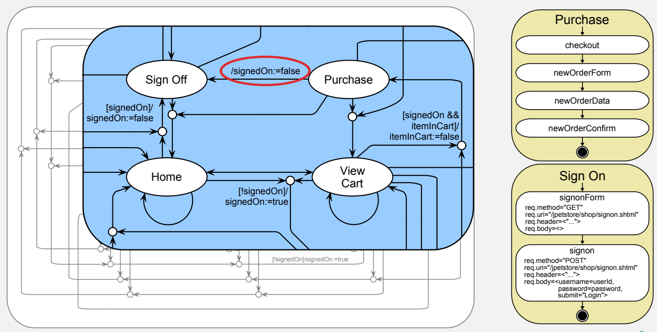
application model

Application Model (Session Layer and 2 Protocol States)
Markov4JMeter Profile for JPetStore (cont'd)



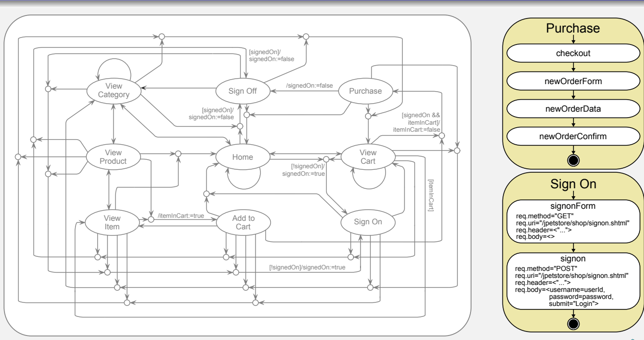
application model

Application Model (Session Layer and 2 Protocol States)
Markov4JMeter Profile for JPetStore (cont'd)



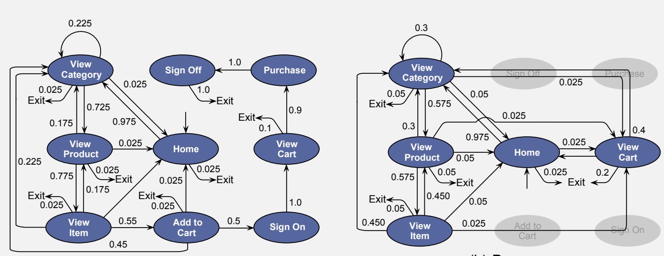
application model

Application Model (Session Layer and 2 Protocol States)
Markov4JMeter Profile for JPetStore (cont'd)



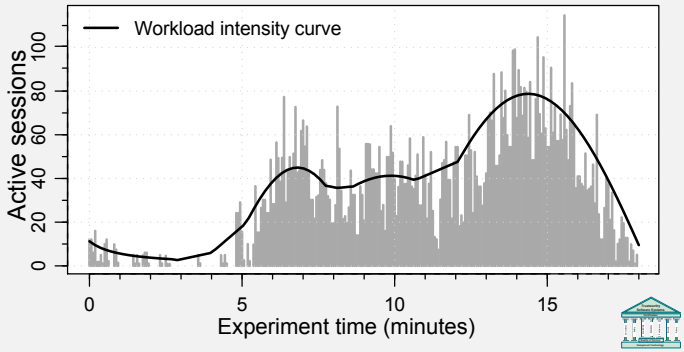
application model

User Behavior Models (Browser and Buyer)
Markov4JMeter Profile for JPetStore (cont'd)

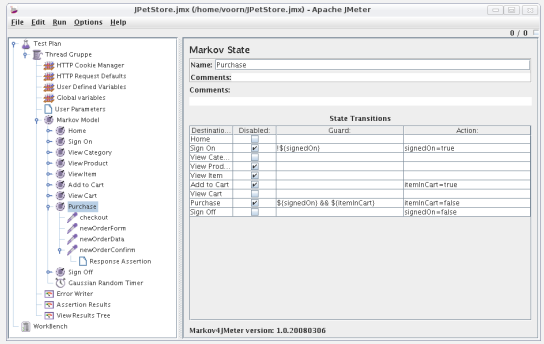


(a) Buyer (b) Browser

Workload Intensity Curve
Markov4JMeter Profile for JPetStore (cont'd)

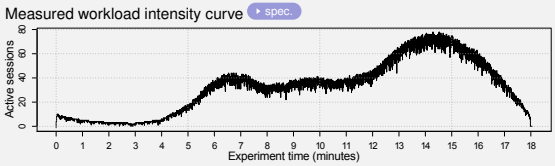


Probabilistic Test Plan
Markov4JMeter Profile for JPetStore (cont'd)



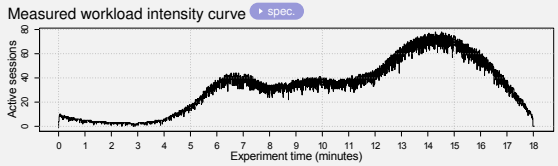
Measurement Results

1. Measured workload intensity curve ▸ spec.

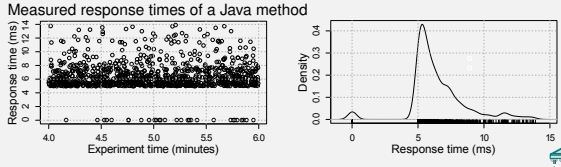


Measurement Results

1. Measured workload intensity curve ▸ spec.



2. Measured response times of a Java method



Structure

Related Work

- Workload specification extends prior work by
  - *Barford and Crovella (1998)* : ON/OFF, virtual users
  - *Menascé et al. (1999)* : CBMGs
  - *Shams et al. (2006)*: EFSMs
- *Ballocca et al. (2002)*: Workload based on CBMGs
- *Lee and Tian (2003)*:

  "Markov chains provide fairly accurate models of Web usage"
- Workload generation tools (*Peña-Ortiz et al., 2005*)

Summary

- Conceptual approach for specifying and generating
  - probabilistic and
  - intensity-varying workload
- Markov4JMeter: Implementation as JMeter extension
- Demonstrated applicability of approach in case study

Markov4JMeter Web Site

http://markov4jmeter.sourceforge.net

References

G. Ballocca, R. Politi, G. Ruffo, and V. Russo (2002). Benchmarking a Site with Realistic Workload. In *Proceedings of the 5th IEEE International Workshop on Workload Characterization (WWC-5)*, pages 14–22. IEEE.

P. Barford and M. Crovella (1998). Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Proceedings of the ACM SIGMETRICS*, pages 151–160. ACM.

Z. Li and J. Tian (2003). Testing the Suitability of Markov Chains as Web Usage Models. In *Proceedings of the 27th International Conference on Computer Software and Applications (COMPSAC '03)*, pages 356–361. IEEE.

J. Matevska and W. Hasselbring (2007). A Scenario-based Approach to Increasing Service Availability at Runtime Reconfiguration of Component-based Systems. In *Proceedings of the 33rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 137–144. IEEE.

D. A. Menascé (2002). Load Testing of Web Sites. *IEEE Internet Computing*, 6(4):70–74.

D. A. Menascé, V. A. F. Almeida, R. Fonseca, and M. A. Mendes (1999). A Methodology for Workload Characterization of E-Commerce Sites. In *Proceedings of the ACM Conference on Electronic Commerce (EC '99)*, pages 119–128. ACM.

R. Peña-Ortiz, J. Sahuquillo, A. Pont, and J. A. Gil (2005). Modeling continuous changes of the user's dynamic behavior in the WWW. In *Proceedings of the 5th International Workshop on Software and Performance (WOSP '05)*, pages 175–180. ACM.

M. Rohr (2008). *Workload-sensitive Timing Behavior Anomaly Detection for Automatic Software Fault Localization.* PhD thesis, Department of Computing Science, University of Oldenburg, Oldenburg, Germany. work in progress.

M. Shams, D. Krishnamurthy, and B. Far (2006). A Model-based Approach for Testing the Performance of Web Applications. In *Proceedings of the International Workshop on Software Quality Assurance (SOQUA '06)*, pages 54–61. ACM.

A. van Hoorn (Sept. 2007). Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems. Master's thesis (Diplomarbeit), Department of Computing Science, University of Oldenburg, Germany.

Generating Probabilistic and Intensity-varying Workload for Web-Based Software Systems*

André van Hoorn, Matthias Rohr, and Wilhelm Hasselbring

Contact: van.Hoorn@Informatik.Uni-Oldenburg.DE

Graduate School TrustSoft and Software Engineering Group
University of Oldenburg, Germany

SPEC International Performance Evaluation Workshop (SIPEW '08)
June 27, 2008 @ Darmstadt, Germany

* This work is supported by the German Research Foundation (DFG), grant GRK 1076/1