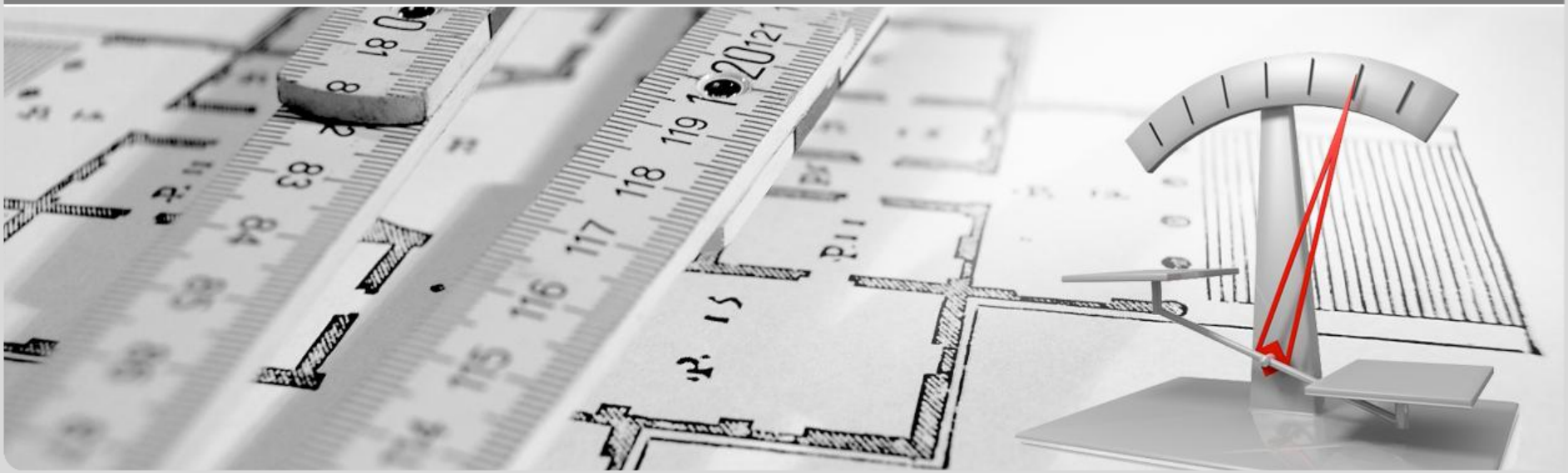
Karlsruhe Institute of Technology

EventSim – An Event-driven Palladio Software Architecture Simulator

Palladio Days '11 | 17th of November

Philipp Merkle, Jörg Henss

SOFTWARE DESIGN AND QUALITY GROUP
INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION, FACULTY OF INFORMATICS



Motivation

- Performance Analyses of Palladio Models
  - analytical approaches → state space explosion
  - therefore, **simulative approaches for complex systems**
- Palladio's simulator SimuCom is mainly process-oriented
- But process-oriented simulations known for inferior performance and scalability (e.g. [Banks 2010])

- **EventSim: tackle performance and scalability issues using event-driven simulation**

Agenda

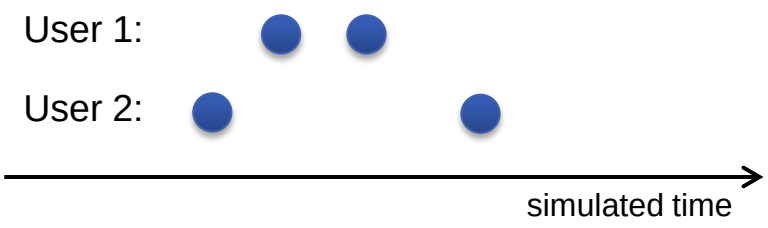
- Foundations
- Related Work
- EventSim
- Validation
- Evaluation
- Conclusion

- Short Demo

Process- vs. Event-orientation in a Nutshell

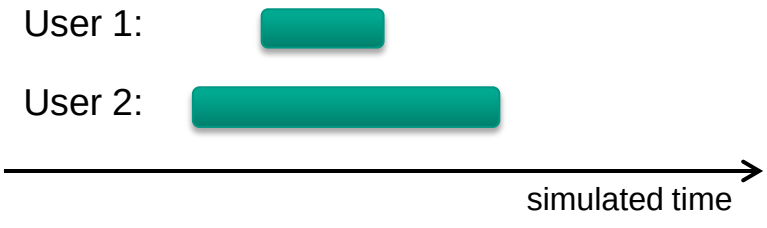
Event-Orientation

- Behaviour modelled by **sequences of events**
  - represent time instant
  - do not overlap each other
  - execute one after another using a **single thread**



Process-Orientation

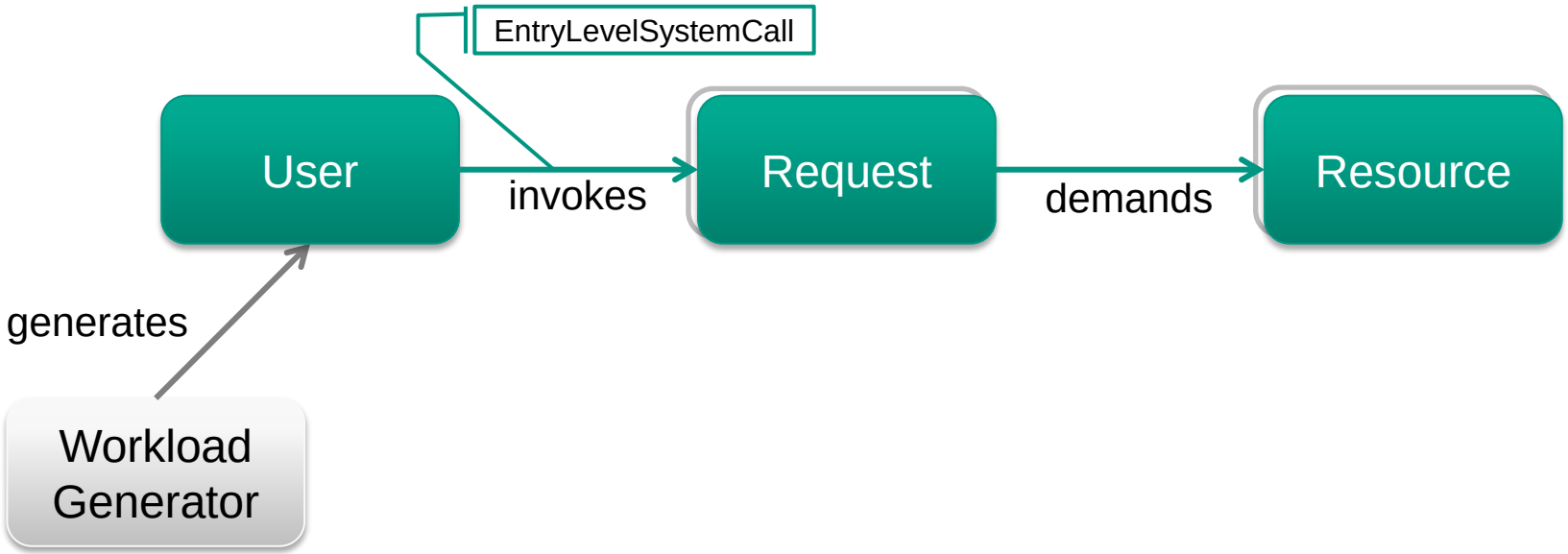
- Behaviour modelled by **processes**
  - represent periods of time
  - may overlap → concurrent control flow
  - **Java: execute using multiple threads**



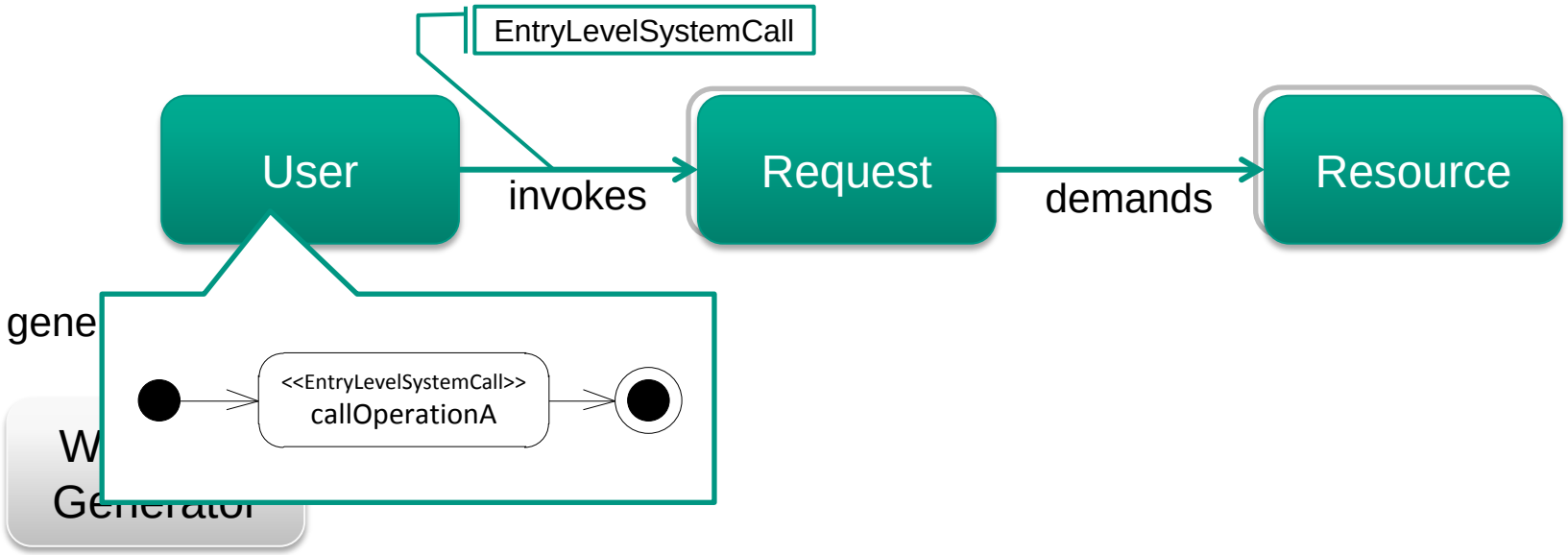
Related Work – PCM Simulators

- **SimuCom** [Becker 2008]
  - Palladio‘s default simulator
  - process-oriented behaviour simulation
  - event-oriented resource simulation
- **SLAstic.SIM** [von Massow 2010]
  - focus on runtime reconfiguration
  - fully event-oriented, but no stochastic expressions
- **SimQPN** [Kounev and Buchmann 2006]
  - simulates queuing Petri nets (QPN)
  - Meier [Meier et al. 2011] developed a PCM to QPN transformation
  - fully event-oriented, but lower prediction accuracy

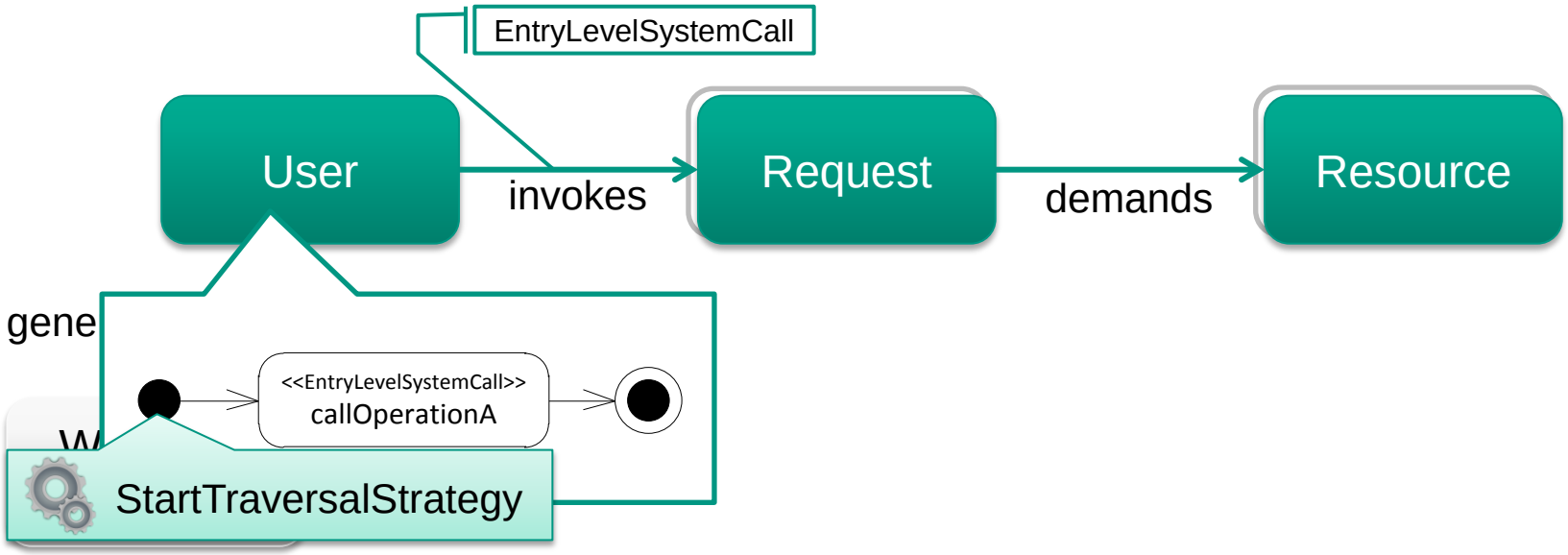
EventSim Overview



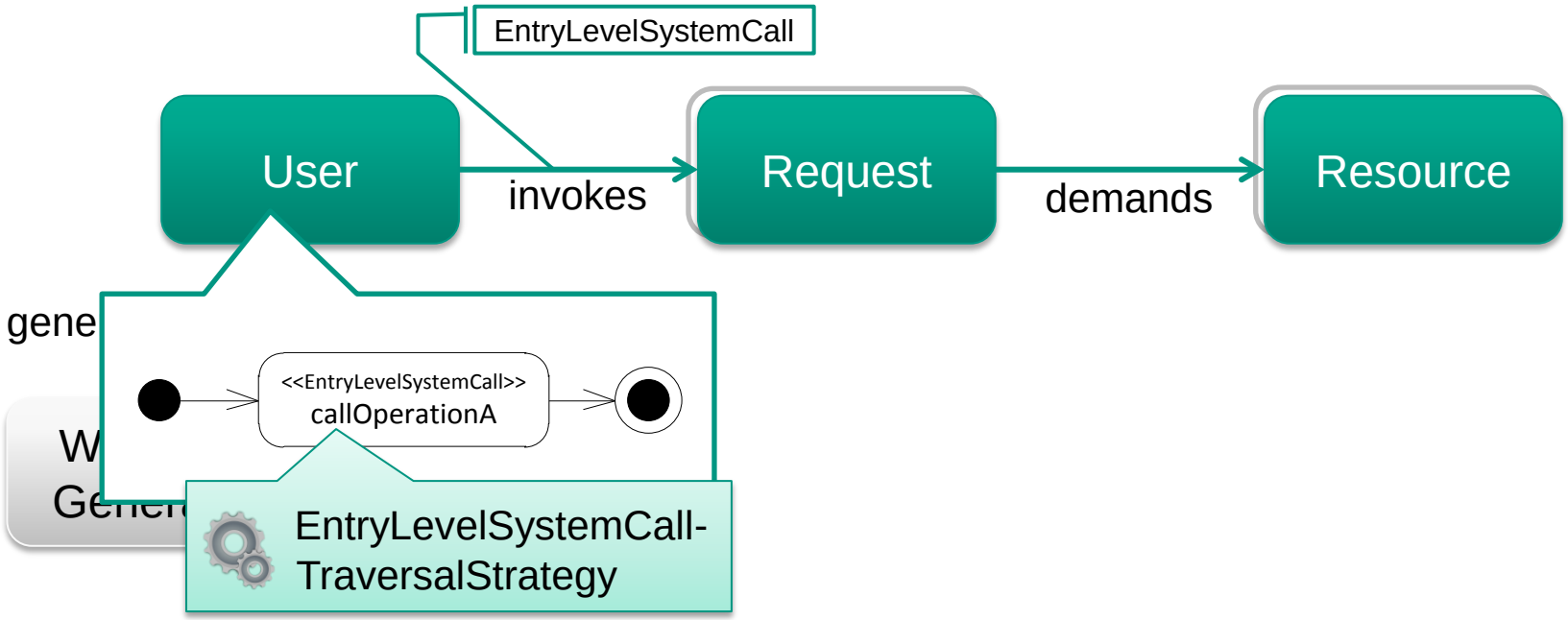
EventSim Overview



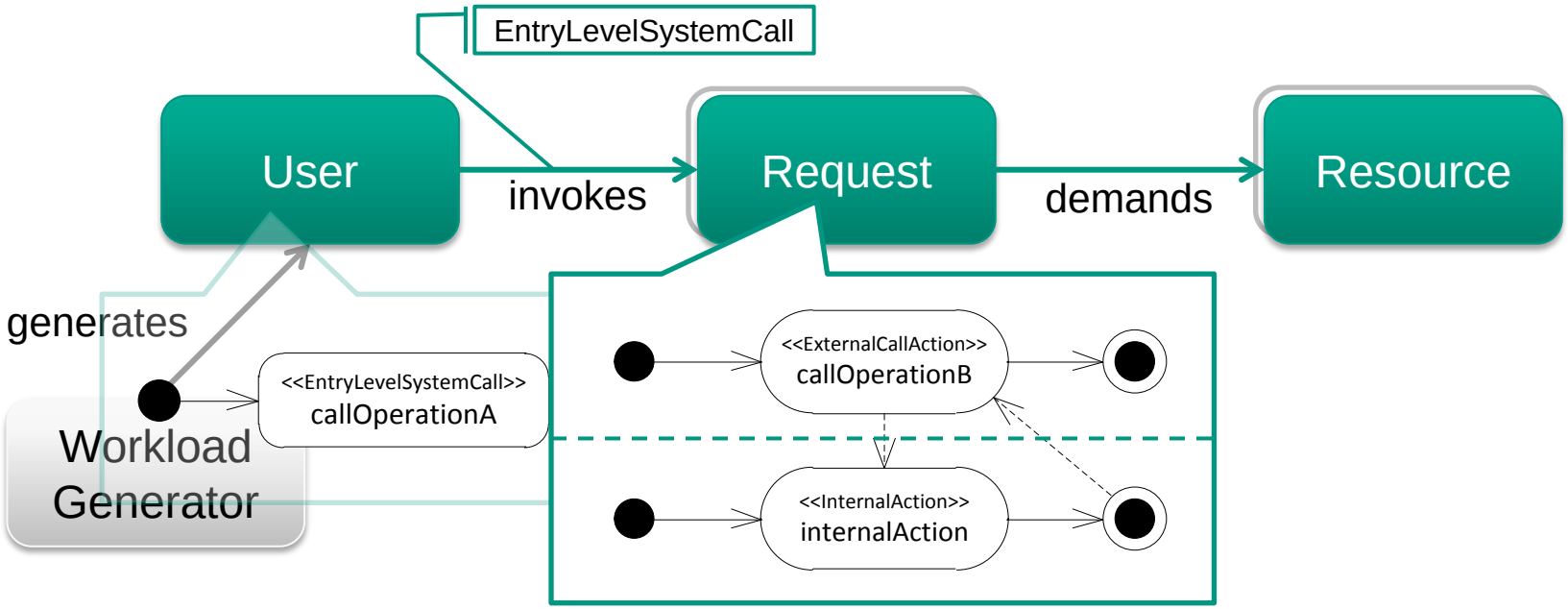
EventSim Overview



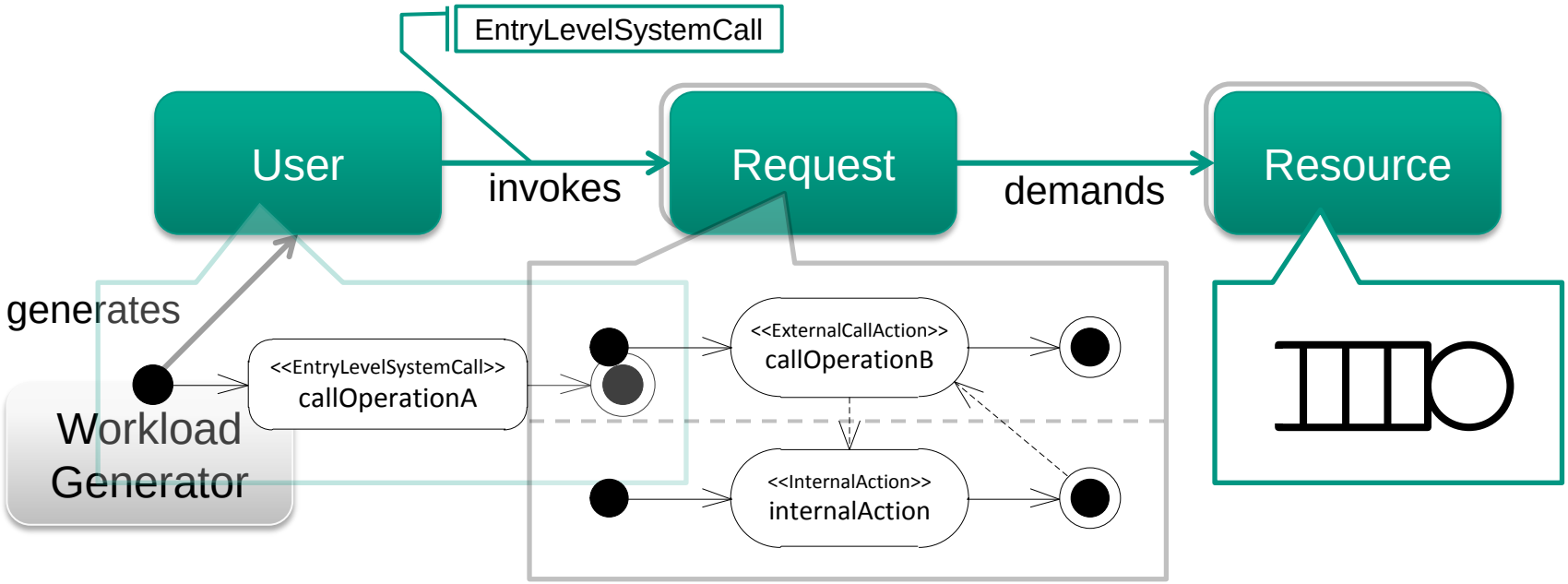
EventSim Overview



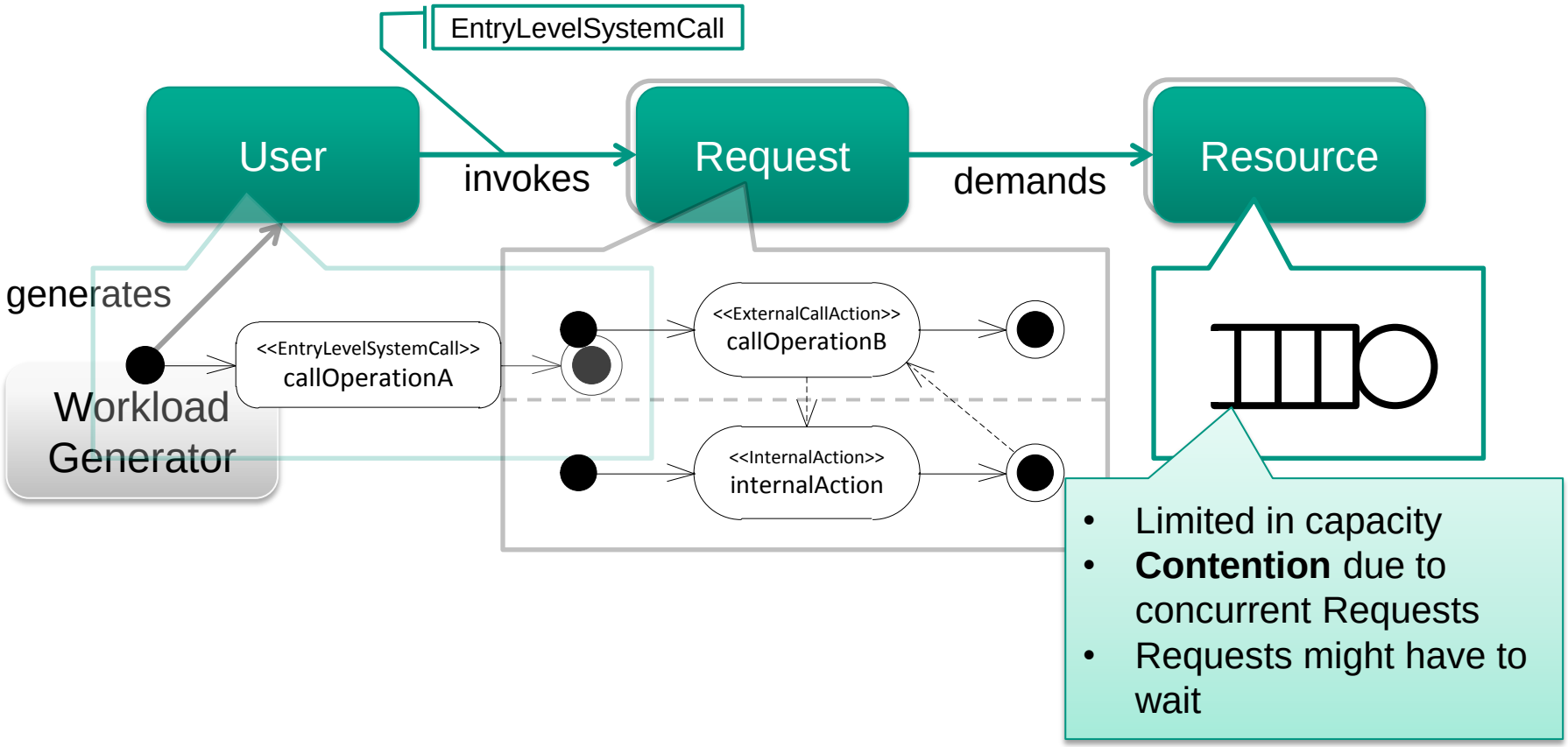
EventSim Overview



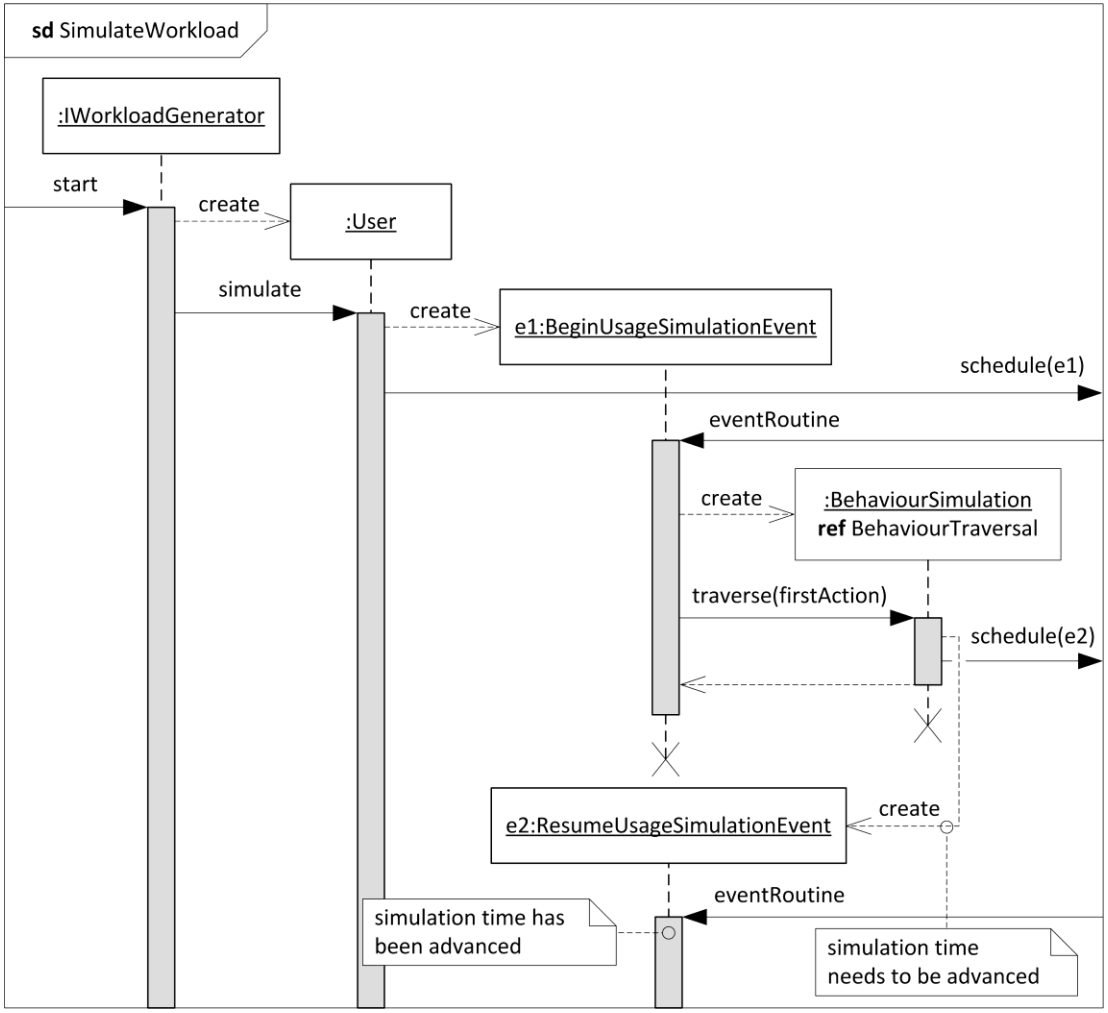
EventSim Overview



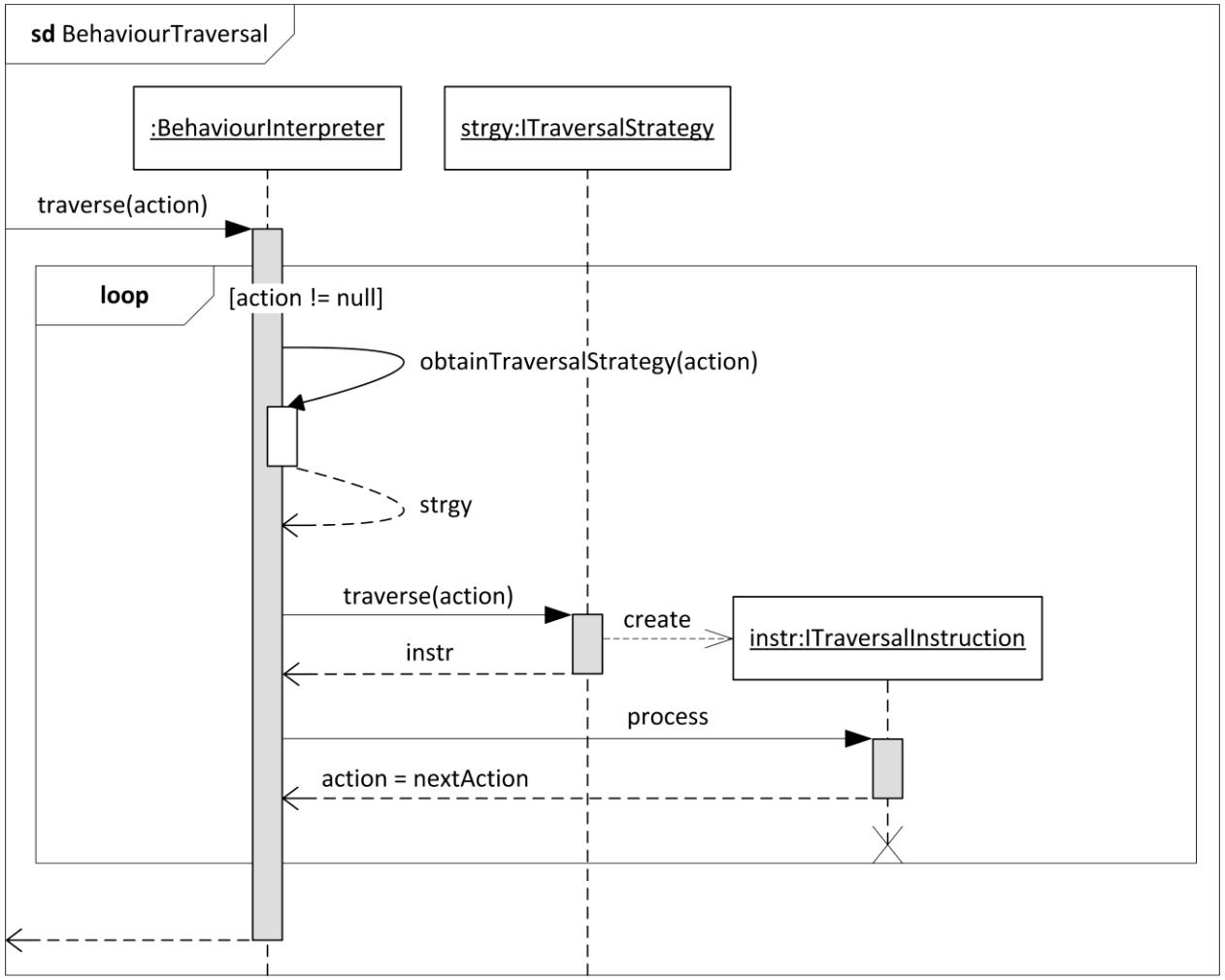
EventSim Overview



Usage of Events



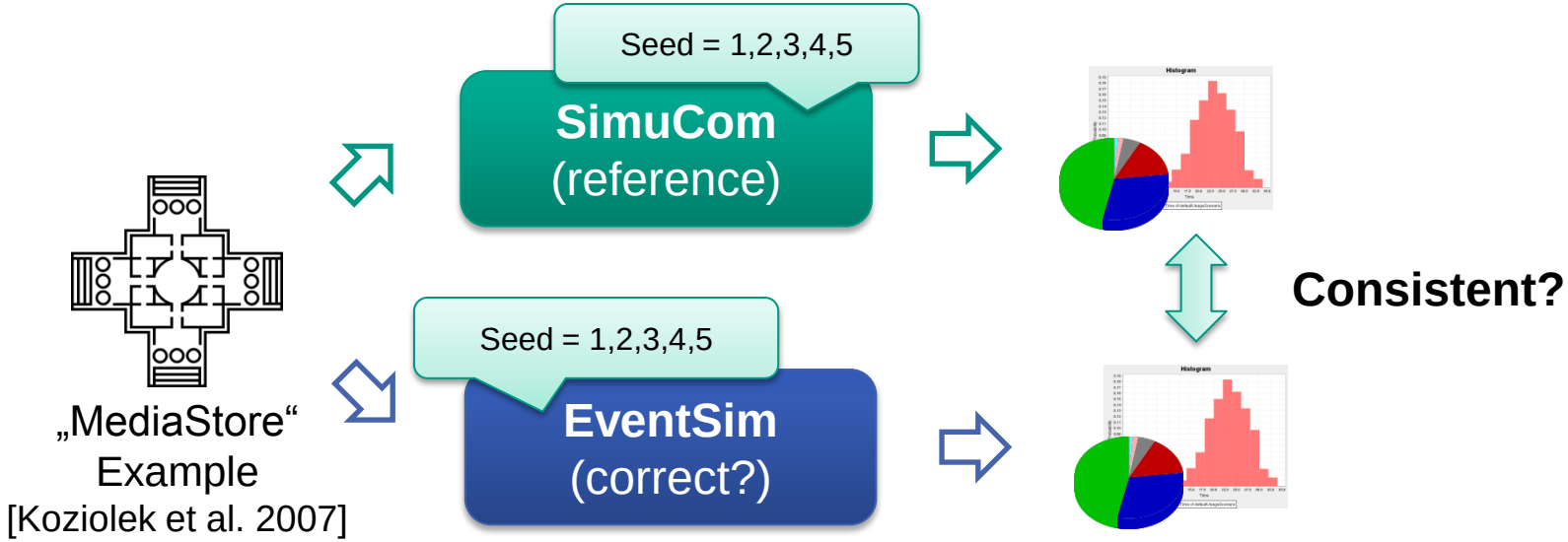
Behaviour Interpreter



Extensibility

- Traversal strategies allow for flexible simulator extension
- Examples
  - additional control flow element
    → new strategy
  - adjust semantics of control flow element
    → change existing strategy
  - additional performance metric (e.g. reliability)
    → decorate existing strategies

Validation Approach



- Results „consistent“ if
  - no difference at all (sum of differences = 0)
  - no significant difference (fail to reject H_0 of Kolmogorov-Smirnov test)

Validation Results

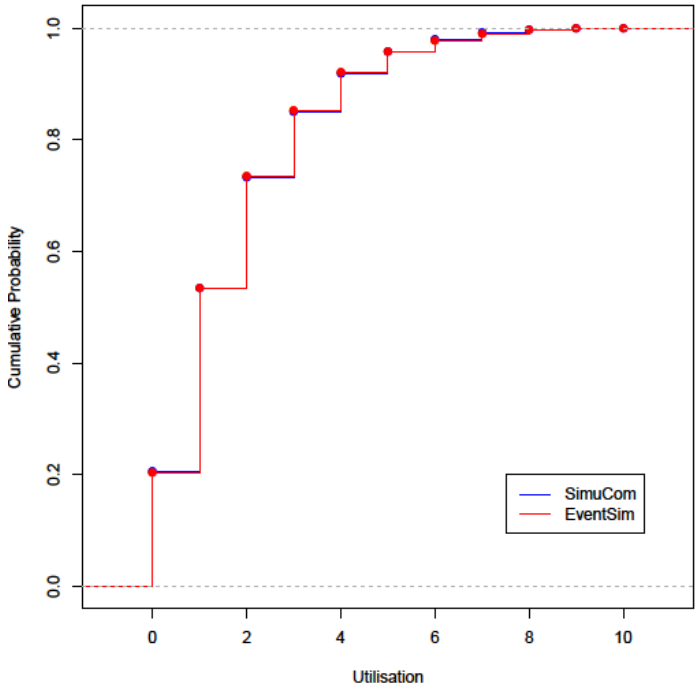
- 3 different experiments using „MediaStore“ Example [Koziolek et al. 2007]

Concurrent Users	Scheduling Policies	Difference?	Difference Significant? ($\alpha = 0.05$)
1	FCFS, PS	no	n/a
10	FCFS, PS	yes	(no)
10	FCFS	no	n/a

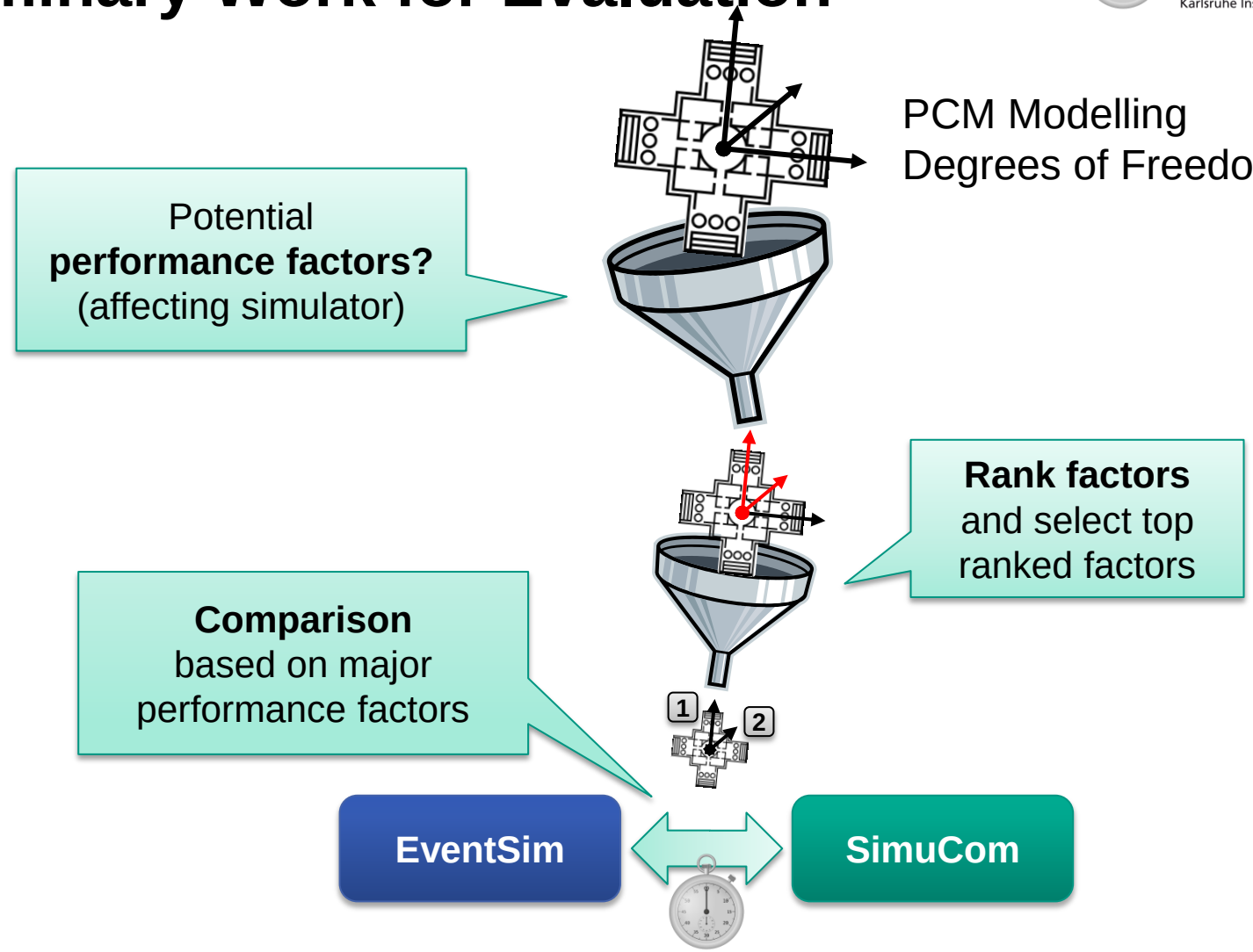
- Difference due to indeterministic scheduler implementation
- **EventSim is correct** (w.r.t the experiments)

Validation Results (cont‘d)

- Significant difference reported in a single case
  (k-s test rejected H_0 „simulation results follow the same cdf“)



Preliminary Work for Evaluation



Evaluation Approach

Repeat each simulation 30 times

Stop simulation at 1000 measurements

{1, 100, 200, …, 1000} elements

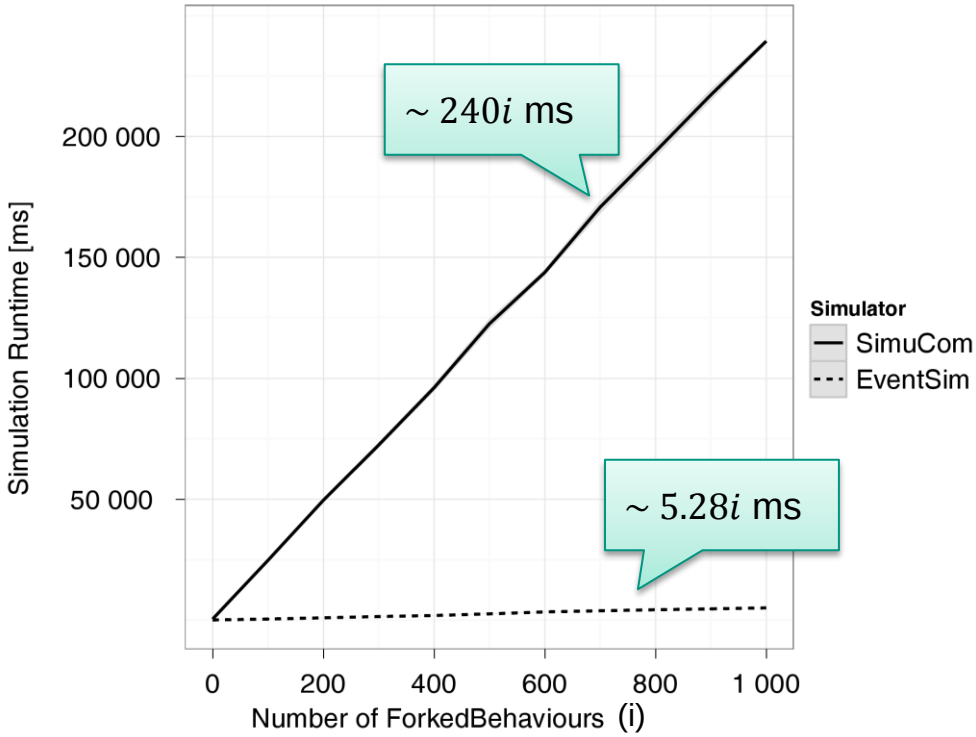
Base Model → Simulate → Add further elements of factor i

Comprises one modelling element for each performance factor

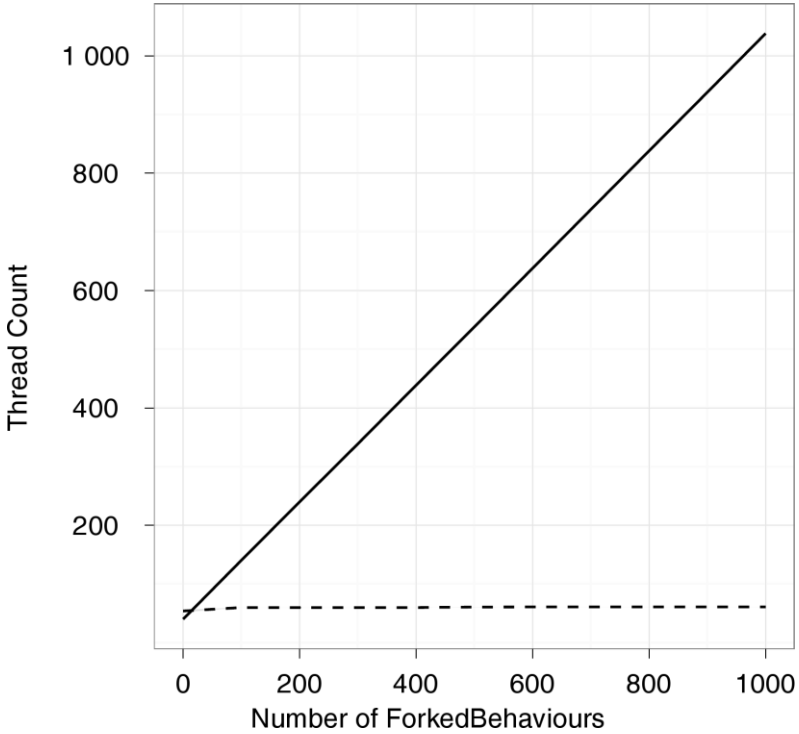
Simulation duration?
Memory? Threads?

Evaluation Results – ForkedBehaviours

- Simulation Duration (1000 repetitions)



- Thread Count



- EventSim around 50 times faster

Evaluation Results Overview

- **Runtime** of both simulators scales linearly
- but at a different rate
  - ForkedBehaviour – EventSim 50 times faster
  - Delay – EventSim 22 times faster
  - InternalAction – EventSim 4 times faster
  - Branch – SimuCom twice as fast
- **Memory consumption** independent of model complexity

Scalability Boundaries

- 64-bit JVM, 4 GB Memory, 1 GB Heap

Factor	SimuCom	EventSim
# ForkedBehaviours	< 820 a)	> 100.000
# InternalActions	< 940 a)	> 100.000
# Delays	< 1.560 a)	> 100.000
# Branches	< 1.250 b)	> 100.000
Workload Population	< 90.000 c)	> 100.000

- Indications of limited scalability
  - a) – StackOverflowError
  - b) – exceeded 64 KB method size limit
  - c) – OutOfMemoryError

Conclusion and Future Work

- Each simulator has its strengths and weaknesses

	SimuCom	EventSim
Scalability:	−	+
Performance (simulation duration)		
• **Simulated Concurrency:** (Forks, intensive workloads)	−	+
• **„Time-advancing“ Actions:** (Delays, InternalActions)	−	+
• **„Control-flow-modifying“ Actions:** (Loops, Branches)	+	−
• Random Number Generator(*):	−	−
• Evaluate Stochastic Expressions(*):	−	−
Feature-Completeness:	+	−

(*) due to identical implementation

- Approach the feature-completeness of SimuCom

EventSim Demonstration

- Input Model: MediaStore Example (for PCM 3.3)
  https://svnserver.informatik.kit.edu/i43/svn/code/Palladio/Core/trunk/Examples/PCM3.3_MediaStore
  - PS scheduling policies replaced with FCFS
  - Increased workload population: 100 (closed workload)
  - No further modifications
- Simulation Settings (default)
  - Stop at 15,000 simulated time units...
    ...or at 10,000 usage scenario measurements (whatever comes first)
  - Fixed Seed (0, 1, ..., 5)
- Modifications of SimuCom
  - Removed simulation of network latency (in calls.xpt)
    ...since not yet implemented in EventSim
  - No further modifications

Bibliography

- **[Banks 2010]** J. Banks, Ed., Discrete-event system simulation, 5th ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2010.
- **[Becker 2008]** S. Becker, “Coupled Model Transformations for QoS Enabled Component-Based Software Design,” Ph.D. dissertation, University of Oldenburg, Germany, Mar. 2008.
- **[Kounev and Buchmann 2006]** S. Kounev and A. Buchmann, “SimQPN – a tool and methodology for analyzing queueing Petri net models by means of simulation,” Performance Evaluation, vol. 63, no. 4-5, pp. 364–394, May 2006.
- **[Koziolek et al. 2007]** “Predicting the Performance of Component-based Software Architectures with different Usage Profiles,” in Proc. 3rd International Conference on the Quality of Software Architectures (QoSA’07), vol. 4880, July 2007, pp. 145–163.
- **[Meier et al. 2011]** P. Meier, “Automated Transformation of Palladio Component Models to Queueing Petri Nets,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 2010, to appear.
- **[von Massow 2010]** R. von Massow, “Performance Simulation of Runtime Reconfigurable Software Architectures,” Master’s thesis, University of Oldenburg, 2010.

Backup Slides

EventSim Features (1)

Dynamic Behaviour	
Usage Modelling	SEFF Modelling
Workload Types	**Actions**
✓ OpenWorkload	✓ StartAction
✓ ClosedWorkload	✓ StopAction
Actions	✓ InternalAction
✓ Start	✓ ExternalCallAction
✓ Stop	✓ SetVariableAction
✓ Delay	✓ AcquireAction
✓ EntryLevelSystemCall	✓ ReleaseAction
✓ Loop	✓ LoopAction
✓ Branch	✗ CollectionIteratorAction
	✓[1] BranchAction
	(✓)[2] ForkAction

[1] both, probabilistic and guarded branch transitions are available

[2] no support of synchronous forks, only asynchronous forks are available

EventSim Features (2)

Static Structure		
Repository	Resource Environment	System
Composition	**Scheduling Policies**	**Miscellaneous**
✓ `BasicComponent`	✓ DELAY	✓ Override Component Parameters
✗ `CompositeComponent`	✓ FCFS	
✗ `SubSystem`	✓ PS	
Miscellaneous	✗[1] EXACT	
✓ Passive Resources	**Network**	
✓ Component Parameters	✗[2] `LinkingResource`	
	✗[2] `Connection`	

[1] the EXACT scheduling policy emulates real schedulers of some operating systems

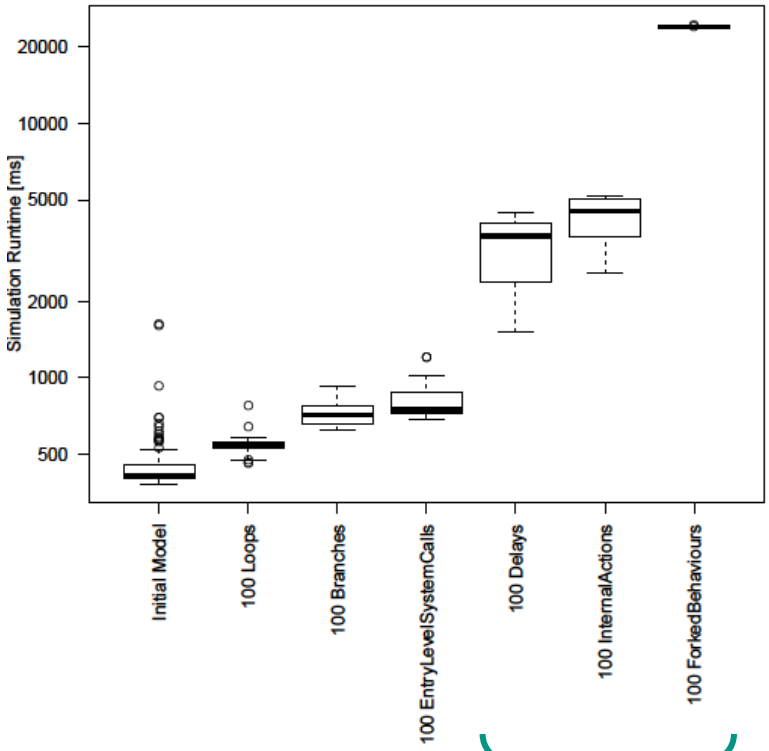
[2] an ideal network connection is assumed to exist between resource containers with an infinite throughput and zero latency.

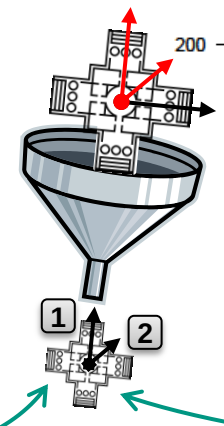
Soft- and Hardware Configuration

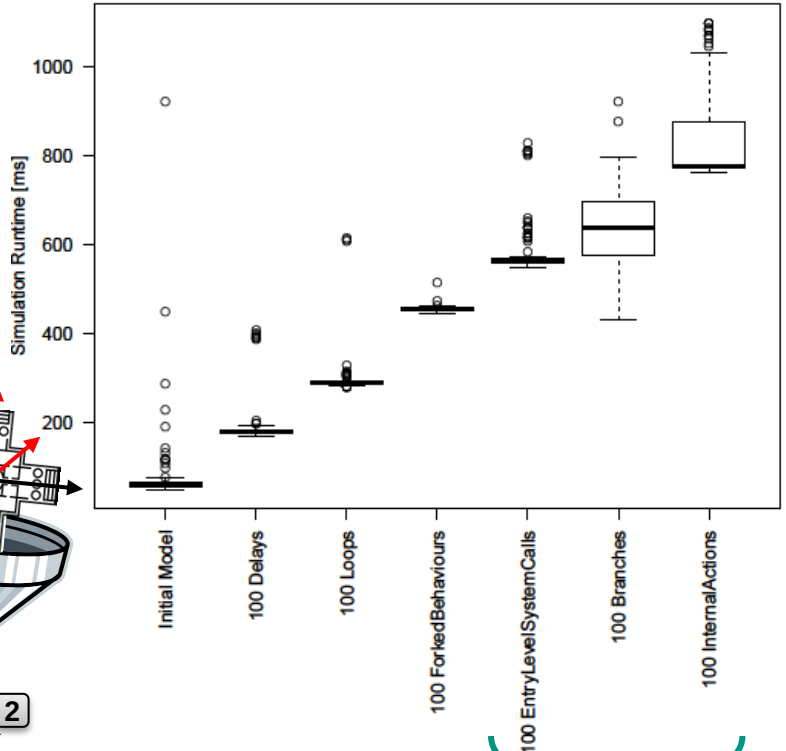
Hardware	
Processor	Intel® Core™2 Quad Q8300 @ 2.50 GHz (4 cores)
Main Memory	4.00 GB @ 800 MHz, single channel mode
Solid-State Drive	OCZ Vertex 2
Operating System	
Version	Windows 7 (Version 6.1)
Java Virtual Machine (JVM)	
Name	Java HotSpot™64-Bit Server VM
Vendor	Sun Microsystems Inc.
Version	20.1-b02
VM Arguments	-Xms512m, -Xmx1024m, -XX:PermSize=256M, -XX:MaxPermSize=512M
Simulators	
Eclipse	Galileo (Version 3.5)
Simulation Library	SSJ (Version 2.1.3)

Factor Ranking Results

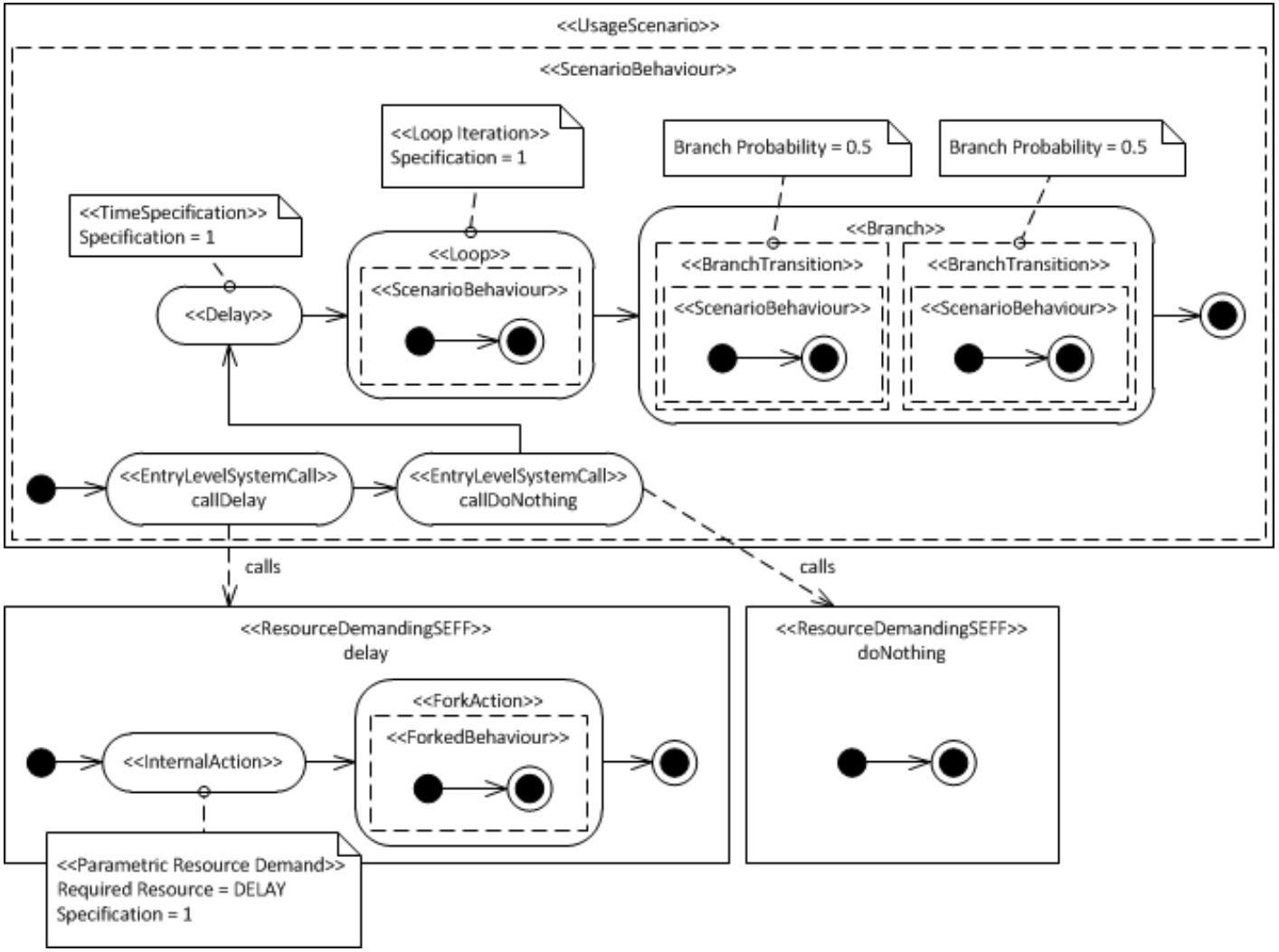
- SimuCom (y-axis in log scale)
- EventSim







PCM Base Model



Executing Simulation Processes in Java

	Threads	Coroutines [Stadler et al. 2010]	Interpretation [Jacobs and Verbraeck 2004]
Context Switching	OS-controlled	manually, suspend() statement	n/a
Synchronisation	yes, preserves execution order among processes	not required	n/a
Exclusion Criteria		JRE extension	slow

- **Threads often unavoidable**
  - important to know induced overhead!