SynchroVis: 3D Visualization of Monitoring Traces in the City Metaphor for Analyzing Concurrency

Jan Waller, Christian Wulf, Florian Fittkau, Philipp Döhring, and Wilhelm Hasselbring

2013-09-27

http://se.informatik.uni-kiel.de
Parallel systems became a commodity
Addition of concurrent behavior
Leading to further challenges in program comprehension (e.g., debugging non deterministic behavior)
SynchroVis

Introduction

- City metaphor to improve program comprehension
- Visualization of the system’s static structure
- Dynamic behavior is gathered from information collected in monitoring\(^1\) traces
- Focus on providing a detailed visualization of concurrent behavior

\(^1\)http://kieker-monitoring.net
Our City Metaphor

Introduction

Figure 1: Our city metaphor
Figure 2: Thread building visualizing started threads
Figure 3: Synchronization with semaphores / monitors
Figure 4: Synchronization with wait / notify
Figure 5: Visualizing a deadlock in the dining philosophers problem
Related Work

- 3D visualization with the city metaphor
  - Software World [KM00], CodeCity [WL07], realistic 3D city visualization [PBG03], EvoSpace [DA08], hierarchical edge bundles [CZB11]

- Visualizing concurrency
  - UML-based [MW01, LMRR03, MCB07], not UML-based [DPJM+02, TBD10]
Future Work and Conclusions

Conclusions

Future Work:

▶ Space-optimizing layout algorithm
▶ Improve our metaphor by diversifying the look of our buildings
▶ Providing greater scalability of our visualizations

(=> ExplorViz presentation tomorrow)

Conclusions:

▶ SynchroVis 2 enables the visualization of static and dynamic properties of a software system
▶ 3D visualization of monitoring traces with the city metaphor for comprehension of concurrency

http://kieker-monitoring.net/download/synchrovis/
Future Work:

- Space-optimizing layout algorithm
- Improve our metaphor by diversifying the look of our buildings
- Providing greater scalability of our visualizations
  (=> ExplorViz presentation tomorrow)

Conclusions:

- SynchroVis$^2$ enables the visualization of static and dynamic properties of a software system
- 3D visualization of monitoring traces with the city metaphor for comprehension of concurrency

---

http://kieker-monitoring.net/download/synchrovis/
Pierre Caserta, Olivier Zendra, and Damien Bodenes.
3D hierarchical edge bundles to visualize relations in a software city metaphor.

Philippe Dugerdil and Sazzadul Alam.
Execution trace visualization in a 3D space.

Wim De Pauw, Erik Jensen, Nick Mitchell, Gary Sevitsky, John M. Vlissides, and Jeaha Yang.
Visualizing the execution of Java programs.

Claire Knight and Malcolm Munro.
Virtual but visible software.

Hugo Leroux, Christine Mingins, and Annya Réquilé-Romanczuk.
JACOT: A UML-based tool for the run-time inspection of concurrent Java programs.

Giovanni Malnati, Caterina Maria Cuva, and Claudia Barberis.
JThreadSpy: Teaching multithreading programming by analyzing execution traces.

Katharina Mehner and Bernd Weymann.  
Visualization and debugging of concurrent Java programs with UML.  

Thomas Panas, Rebecca Berrigan, and John Grundy.  
A 3D metaphor for software production visualization.  

Jonas Trümper, Johannes Bohnet, and Jürgen Döllner.  
Understanding complex multithreaded software systems by using trace visualization.  

Richard Wettel and Michele Lanza.  
Visualizing software systems as cities.  