TheServerSide JAVA SYMPOSIUM

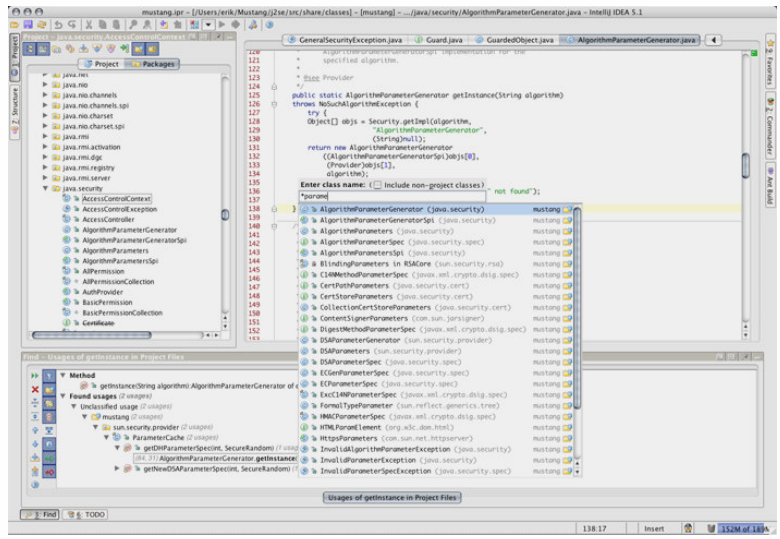
Software Visualization and Model Generation

Erik Doernenburg
Software Developer
ThoughtWorks, Inc.

Gregor Hohpe
Software Engineer
Google, Inc.

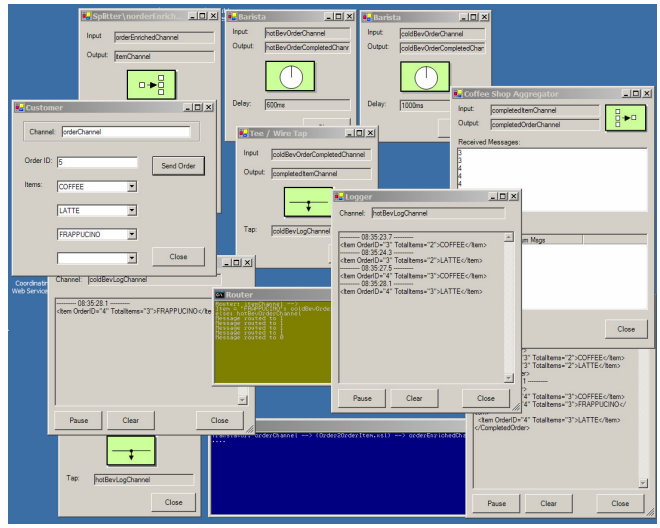
TheServerSide JAVA SYMPOSIUM

Where are the most defects?



TheServerSide JAVA SYMPOSIUM

Which way do the messages flow?



TheServerSide JAVA SYMPOSIUM

A picture says more than 1000 words

- **The amount of information in current systems is beyond what we can handle (understand?)**
- **Often we are only interested in a specific angle**
  - **Relationship between classes – not the entire source**
  - **Number of messages flowing – not the message content**
- **We're good at a spotting patterns in images…**

Being able to control large-scale systems is an illusion.
But we can observe what is happening…

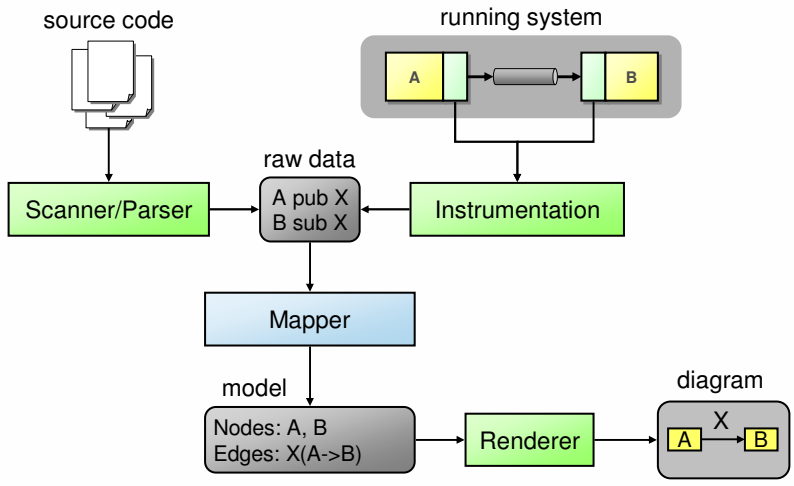
TheServerSide JAVA SYMPOSIUM

Where do we get the picture from?

- **Models created upfront convey a vision but usually don't reflect reality**
- **Generating a complete model for large systems is nearly impossible**
- **Systems evolve locally, often uncontrolled**
  - **Particularly true for loosely coupled, dynamic systems (SOA)**
- **The best picture very much depends on the question you are trying to answer**
  - **We need tools that make it easy to create ad-hoc models**

TheServerSide JAVA SYMPOSIUM

Visualizing Software



TheServerSide JAVA SYMPOSIUM

Five Steps

1. **Select Meta-model**
2. **Inspection / Instrumentation**
3. **Mapping to Model**
4. **Visualization / Output**
5. **Validation / Analysis**

TheServerSide JAVA SYMPOSIUM

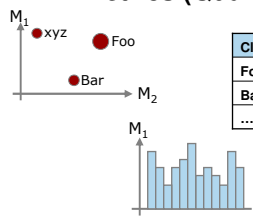
1. Select Meta-model

- **"A model that describes a model"**
  - **The elements a model can be composed of**
  - **How to combine these elements**
- **Example: meta-model for a class diagram**
  - **A class is a box with name, methods, fields,…**
  - **Available connectors: association, inheritance, aggregation…**
  - **Rules: no circles in inheritance etc.**
- **Sounds more scientific than it really is**
- **Usually pick from a few popular candidates**

TheServerSide JAVA SYMPOSIUM

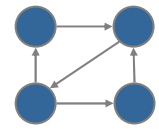
Common Metamodels

Metrics (Quantitative)

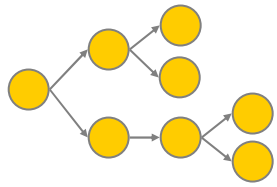


Class	M1	M2	M3
Foo	3.4	3.8	6.5
Bar	5.0	0.0	10.0
...	...	...	...

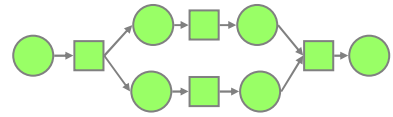
Directed Graph



Tree



Process Model (e.g. Petri Net)



TheServerSide JAVA SYMPOSIUM

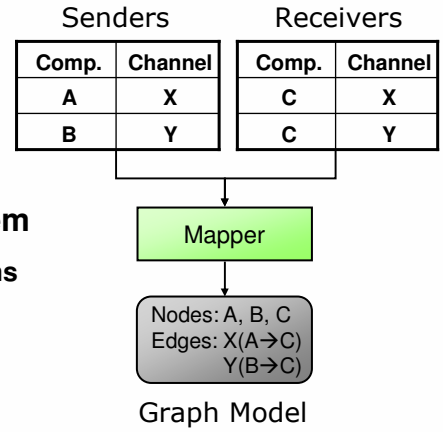
2. Inspection / Instrumentation

Static Analysis	Dynamic Analysis
Inspect System Design	Inspect Running System
Source code	Profiling
Configuration repository	Listen to messages
Scan / Parse into intermediate format	Log files
	Network sniffer
	Compiler decorator

TheServerSide JAVA SYMPOSIUM

3. Mapping to Model

- **Map the gathered data onto the meta-model**



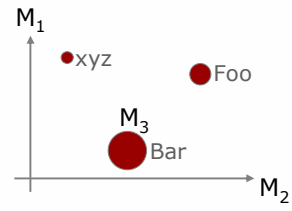
- **Example: Messaging System**
  - **Capture send / receive actions**
  - **Map onto directed graph**

TheServerSide JAVA SYMPOSIUM

4. Visualization / Output

- **Example: GraphViz Dot**
  - **Automated graph layout tool**
  - **Takes textual input, produces many graphics formats**
  - **Developed by AT&T, Common Public License**

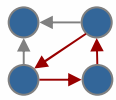
- **Example: Codecrawler**
  - **Object-Oriented reverse engineering**
  - **Polymetric views of metrics**
  - **Hotspots, complexity, inheritance, data storage, etc.**
  - **All views are interactive**



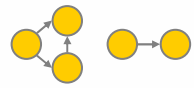
TheServerSide JAVA SYMPOSIUM

5. Validation / Analysis

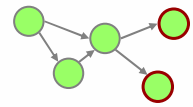
- **A model is useful for more than pretty pictures**
- **Can apply rules against the model**
- **Example: Directed Graph**



Identify circles



Identify islands
(in domain models)



Identify leaf nodes
(in dependency graphs)

TheServerSide JAVA SYMPOSIUM

Example: Object Dependencies

- **Static analysis: Spring bean configuration**
- **SpringViz, a small XSLT sheet, maps bean configuration to input for GraphViz Dot**
- **Mapping and format hard-coded in style sheet**
- **Really simple but really useful**

TheServerSide JAVA SYMPOSIUM

Example: Component Dependencies

- Dynamic Analysis: JAR references
- Decorate compiler with custom Ant task to get dependencies on a JAR level
- Map to a Directed Graph
- Render with GraphViz Dot as clickable SVG
- Navigate through model

TheServerSide JAVA SYMPOSIUM

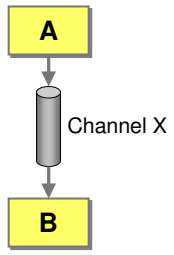
Example: Code Crawler, System Complexity

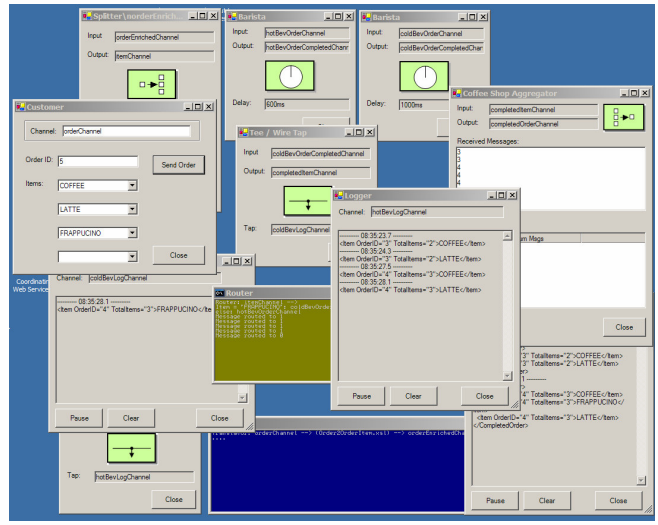
- Static analysis: Source code analysis
- Code Crawler imports XMI and calculates metrics
  - NOA, NOM, WLOC
- Renders polymetric System Complexity view
  - Width, height, color used for metrics
  - Position used for tree layout of inheritance
- Goal of this view is to classify inheritance hierarchies
  - Subsystems
  - Large stand-alone classes
- Can use other views to understand inner workings of specific hierarchies

TheServerSide JAVA SYMPOSIUM

Example: Asynchronous Messaging

Processes communicate by sending messages across channels

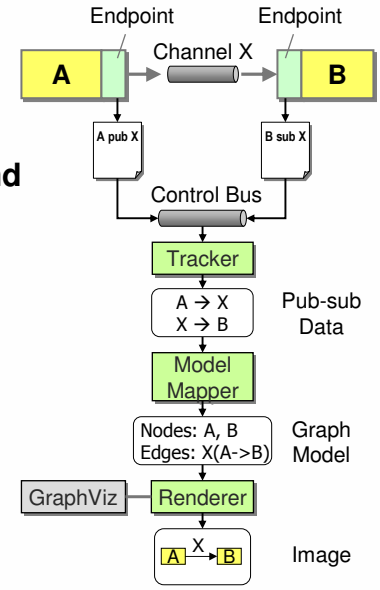




TheServerSide JAVA SYMPOSIUM

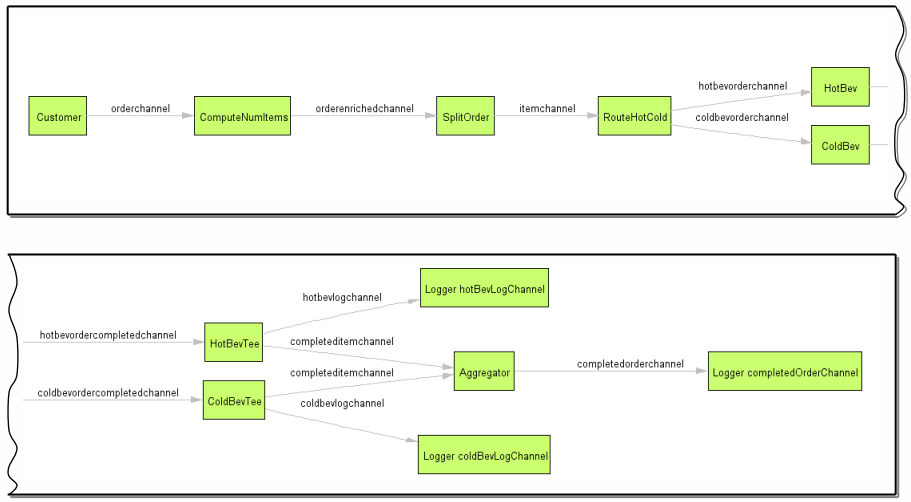
Visualization

- **Dynamic Analysis: Instrument Message Sender and Receiver**
- **Collect publication and subscription data centrally**
- **Map to a Directed Graph model**
- **Render with GraphViz Dot**
- **Re-render as model changes**



TheServerSide JAVA SYMPOSIUM

Message Flow Graph



TheServerSide JAVA SYMPOSIUM

Example: Process Mining

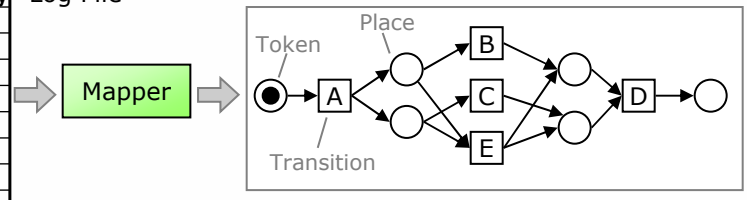
- **A system performs a series of activities for each case (process instance)**
- **Goal: create a process model from running system**
- **Approach: Analyze activity logs and create process model**
- **Meta-model: Petri Net**
- **Variety of algorithms, e.g. α-Algorithm (Wil v.d. Aalst)**

TheServerSide JAVA SYMPOSIUM

Process Mining Example

Case	Activity
1	A
2	A
3	A
3	B
1	B
1	C
2	C
4	A
2	B
2	D
5	A
4	C
1	D
3	C
3	D
4	B
5	E
5	D
4	D

Log File



Process Model

Process-Aware information Systems, Dumas et al.

TheServerSide JAVA SYMPOSIUM

What's Next?

- **Applies to many levels**
  - **Single module source code**
  - **Multiple modules**
  - **Whole systems**
- **Diagrams hard to scale to huge systems**
  - **Interactive zoom / drill-down**
  - **Annotation-aided visualization**
- **As systems become larger and more dynamic tools like this become a necessity**

TheServerSide JAVA SYMPOSIUM

Resources

- **Tools**
  - http://www.graphviz.org **(Dot)**
  - http://www.samoht.com/wiki/wiki.pl?SpringViz
  - http://www.eaipatterns.com **(Messaging visualization)**
- **Wil v.d. Aalst: Process Aware Information Systems, Wiley, 2005**
  - www.processmining.org **(Process Mining Tool)**
- **Michele Lanza's work (CodeCrawler)**
  - http://www.inf.unisi.ch/faculty/lanza

TheServerSide JAVA SYMPOSIUM

Questions ?