TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Laura Bassi Centre of Expertise
Centre for Visual Analytics Science & Technology

Visual Debugging Aids for eclipse

Bilal Alsallakh, Peter Bodesinsky, Alexander Gruber, Silvia Mikisch, Dorna Nasseri

Program Tracing

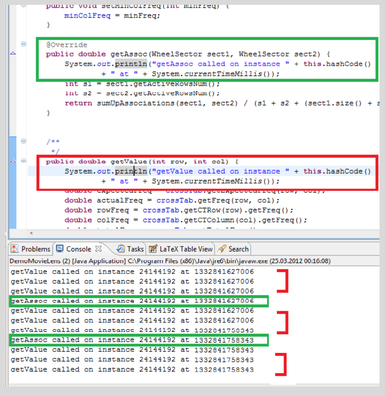
Ad-hoc Tracing in Eclipse

Status Quo

- Developers insert code snippets in desired location
- Snippets print out values that need to be inspected
- Checking textual output for post-mortem analysis

Drawbacks

- Time consuming to check the textual traces
- not possible in compiled code (available source is read-only)



old methods

Array Inspection

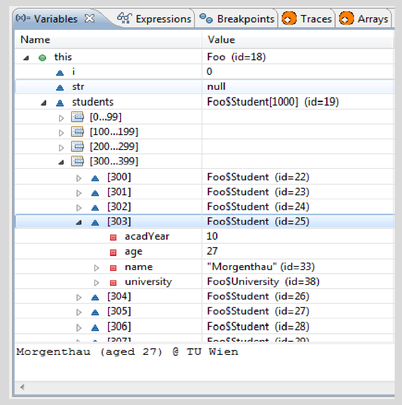
Eclipse built-in variable inspection

Features

- generic indented list for all variables & fields
- can change variable values

Not suitable for large arrays

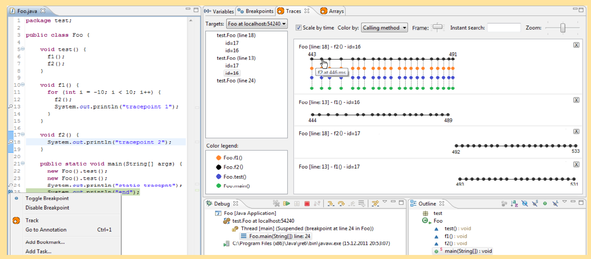
- No overview
  Item need to be expanded and inspected individually.
- Hard to navigate & search
  Example: locating certain items at an unknown index

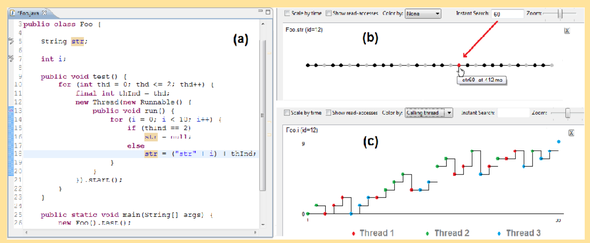


Eclipse Tracer

Features

- Can make trace-points out of normal breakpoints
- tracepoints record information about the program state when they are hit:
  - current timestamp, **`this`** instance and variable value (for watch-points)
  - calling thread, its stack frames and their code locations (source and line number)
- a time-line view of the trace-point hits on a specific **`this`** instance
- ability to show the hit information via color or line charts
- a double-click on a hit opens its source file and jumps to the respective line number





new methods

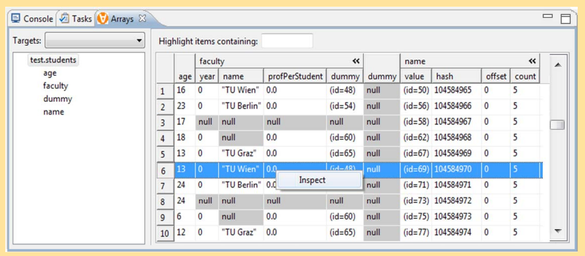
Array Explorer

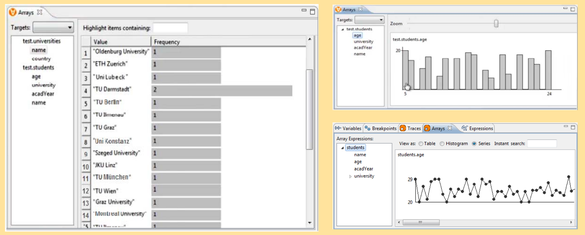
Features

- provides overview using tables
- table columns show the fields and can be expanded to check the sub-fields
- enables flexible searching, sorting, and filtering
- charts reveals distributions
- Handles variables or expressions of
  - normal Java arrays []
  - `Collections` (List, Set, etc.)
  - `Maps` (as key-value pairs)

Limitation

- shows information from the generic type. Manual inspection needed to see fields from sub-classes





status

- available implementation is experimental (can cause deadlocks)
- needs better integration with Eclipse JDT to avoid registering own listeners that cause the deadlocks
- will be released to open-source community in April 2012

- positive feedback from Java developers
- needs stabilization, professionalization of the views, and better handling of special cases.
- will be released to open-source community in April 2012

Tools downloadable at http://www.cvast.tuwien.ac.at/projects/visualdebugging