

# Towards Actionable Visualisation in Software Development

Leonel Merino, Mohammad Ghafari, and Oscar Nierstrasz  
Software Composition Group,  
University of Bern,  
Bern, Switzerland

## Collection

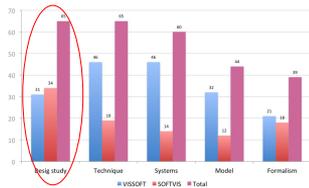


Figure 2. Classification of the 273 SOFTVIS/VISSOFF papers by type.

p. 2 Towards Actional Visualization in Software Development

## Background

**Why is software visualisation not widely used in software development?**

- Out of touch with developer needs

### Prior Work

- Taxonomy and surveys
- Framework to assess tools
- Help users understand tools

## Approach

- **Map** needs to solutions in a problem domain
- finely-grained **developer needs**

### Eng. Where is this method called?

**Question** → **Problem Domain** → **Visualisation Tool**

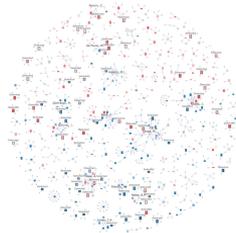
## Research Questions

**What are the goals of this literature review?**

- **RQ1:** What are the characteristics of visualisation techniques that support developer needs?
- **RQ2:** How well are various problem domains supported by visualisation?

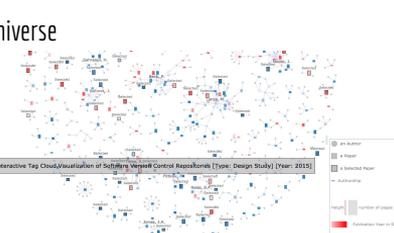
Filter 346 → 65 design study papers

## Universe



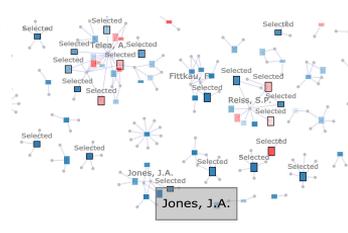
<http://scg.unibe.ch/research/visualisation-review>

## Universe



<http://scg.unibe.ch/research/visualisation-review>

## Universe



<http://scg.unibe.ch/research/visualisation-review>

## Data Extraction

Attributes

## Data Extraction

- **Evaluation:** Task, **Need**, Audience, Data Source
- **Implementation:** **Representation**, Medium, Tool

### How?

- Frequent terms
- Questions
- Goals

## Extraction - Need

75% stated explicit questions (need)

**How were needs identified when questions weren't asked?**

**How many needs?**

Table VII  
VISUALISATION TOOLS AND NEEDS INTRODUCED BY PAPERS.

Building and branching
Debugging
History
Implementing
Implications
Policies
Rationale
Refactoring
Teammates
Testing
Concurrency
Intent and implication
Location
Method properties
Performance

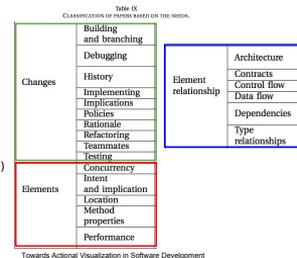
Questions and Goals that Motivate Visualizations

- to get a better insight of the control or data flow inside a program
- are figure readable or representation convenient
- how the computation reached that result
- which applications are displayed on multiple nodes?
- which developers collaborated
- what happened to our system recently
- how different our work appears to different threads?
- to check guidelines and an engineering of coding software
- how the GUI and the underlying code are related
- how a specific code location can be reached via function call?
- how are classes distributed in system structure?
- how the system is actually organized?
- what new files changed compared to source files at the beginning of a project?
- how the dependency relation between a system and its dependencies evolved?
- what other programmers are working on?
- what are coworkers working on?
- what kind of changes have been made?
- how many versions contains annotation classes?
- where and when a thread waits or releases?
- how much time is spent blocking on a specific lock?

Towards Actional Visualization in Software Development

## Needs Classification

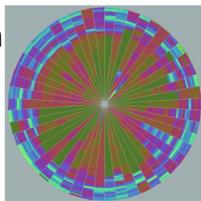
- Classify questions based on general set
- Filter **needs** (finely-grained) within **domain** (coarsely-grained)



## Extraction - Representation

### Dense Pixel

- From Keim's Taxonomy
- Massive sets of data



Dense Pixel Displays: Circle Segments Technique  
from Keim

## Extraction - Representation

### Transform

- Node links
- Explore relationships

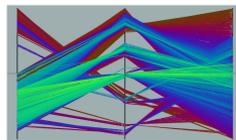
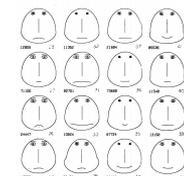


Fig. 2. Parallel Coordinate Visualization @HERE  
from Keim

## Extraction - Representation

### Iconic

- Attribute values mapped to icon features



H. Chernoff, "The use of faces to represent points in k-dimensional space graphically," Journal Amer. Statistical Association, vol. 68, pp. 361-368, 1973.

## Extraction - Representation

### Stacked

- Hierarchical displays
- tree maps

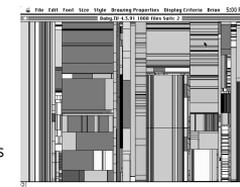


Figure 8. Treemap with 1000 Files

B. Shneiderman, "Tree visualization with treemaps: A 2D space-filling approach," ACM Transactions on Graphics, vol. 11, no. 1, pp. 92-99, 1992.

# Extraction - Representation

- Standard**
- Bar charts
  - X y plots

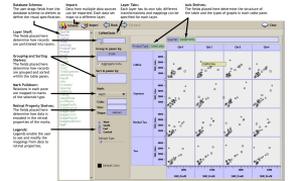


Figure 5: The Polaris user interface. Another overview when based display of relational data by dragging Data into the database browser and clicking through the display of group relationships of Data on display window to view specific data. The browser view is highlighted within the display and relationship operation is performed by the system in general (see display).

D. Tang, C. Stolte and P. Harrahan, "Polaris: A system for query, analysis and visualization of multi-dimensional relational databases," Transactions on Visualization and Computer Graphics, 2001.

# Conclusion

Analysis of Results

# Conclusion

- Covered:**
- History, performance, concurrency, dependencies
- Call to action:**
- Rationale, Intent, Implementation, Refactoring
  - Metaviz (demo at <https://www.youtube.com/watch?v=qe5qIS1cmzs>)

# Critique

# Problem Domain Mapping

## Space filled (hierarchical)

- **what: data** multiple attribs
- **what: derived** sum of papers
- **how: encode** area marks and containment for hierarchy and visualization category, rectilinear layout
- **how: reduce** aggregate
- **what: task** view distribution **RQ1**

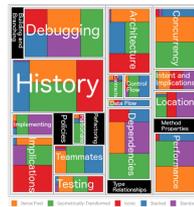


Figure 6: Mapping type of visualization used by studies to problem domains. Towards Actional Visualization in Software Development

# Threats to Validity

- Bias in paper selection
  - Data extraction
- Others not mentioned**
- 'primary contribution' selection
  - single source for developer needs (Latoza)

# Problem Domain Mapping

## Double Bar Chart

- **what: data** categorical and quantitative
- **what: derived** sum of papers and needs
- **how: encode** line marks, color
- **how: reduce** aggregate
- **what: task** compare problem domain with needs **RQ2**

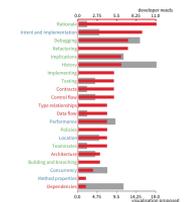


Figure 7: Comparing the degree of importance of developer needs vs. their research papers to problem domains. Towards Actional Visualization in Software Development

# Questions?