Seminar Software Composition: Project P6 How are Software Visualizations Evaluated?

Presented by: Lukas Imstepf lukas.imstepf@unifr.ch

Department of Informatics University of Fribourg

June 27, 2017







Systematic Literature Review

Research Method



 Following Kitchenham's guidlines for systematic literature reviews in software engineering.

B. Kitchenham (2004): "Procedures for Performing Systematic Reviews"

Search Strategy and Data Sources

Research Method

1. Search for:

"software visualization" OR "software visualisation"

Search Strategy and Data Sources

Research Method

1. Search for:

"software visualization" OR "software visualisation"

- 2. in three scientific online databases:
 - ACM Digital Library
 - ► IEEE Xplore DL
 - ScienceDirect

Search Strategy and Data Sources

Research Method

1. Search for:

"software visualization" OR "software visualisation"

- 2. in three scientific online databases:
 - ACM Digital Library
 - IEEE Xplore DL
 - ScienceDirect
- **3.** Download the search results as BibTeX/CSV files, converting CSV to BibTeX with *bibsani*



Research Method



1289 search results



- 1289 search results
- Exclusion criteria: data sanitization (incomplete entries, duplicates: -100)



- 1289 search results
- Exclusion criteria: data sanitization (incomplete entries, duplicates: -100)
- Exclusion criteria: scraping failures (-66, +7 manual downloads)

Research Method

Subtotal: 1130 scrapped PDF files

Research Method

Subtotal: 1130 scrapped PDF files

Further exclusion criteria:

1. Fewer than five pages

(-279, subtotal = 851)

Research Method

Subtotal: 1130 scrapped PDF files

Further exclusion criteria:

- **1.** Fewer than five pages (-279, subtotal = 851)
- 2. InfoVis (medical/geographical) papers (-318, subtotal = 533)

Research Method

Subtotal: 1130 scrapped PDF files

Further exclusion criteria:

- **1.** Fewer than five pages (-279, subtotal = 851)
- **2.** InfoVis (medical/geographical) papers (-318, subtotal = 533)
- 3. Exclusion by paper type:
 - 3.1 Technique papers (novel algorithms)
 - 3.2 Design study papers (particular domain problems)
 - 3.3 Systems papers (architectural choices)
 - 3.4 Evaluation papers
 - **3.5** Model papers (taxonomy, formalisms, commentary)

Paper types by Munzner (2008): "Process and Pitfalls in Writing Information Visualization Research Papers".

Exclusion Criteria: Paper Type

SoftVis papers from 1992 to 2017

























62 Design studies

Venues

Design studies, N = 62



Target audience

Design studies, N = 62



Programming Paradigms

Design studies, N = 62



Visualizations and evaluations

Overview



121 Visualizations of 62 design studies

Usage of visualization frameworks

Design study papers, N = 62



*other visualization frameworks: Graphplace, Flatland, Sovis

Presentation of a new visualization tool?

Design study papers, N = 62



Who needs visualizations, and why?

Programming paradigms vs. programming languages



Who needs visualizations, and why?

Programming paradigms vs. programming languages


Programming paradigms vs. programming languages













Target audience vs. data visualization questions



Target audience vs. data visualization questions



Target audience vs. data visualization questions

	Implications (changes): 10
	Refactoring (changes): 18
	Rationale (changes): 1 _ Building and branching (changes): 5
	Concurrency (elements): 9
	Method properties (elements): 1 _
professional: 159	Dependencies (relationships): 12
	Contracts (relationships): 2
	Intent and Implementation (elements): 6
	History (changes): 18
	Location (elements): 19
	Architecture (relationships): 10
	Performance (elements): 20
academic: 2	Type relationships (relationships): 5
academic/professional: 29	Debugging (changes): 26
education: 23	Control flow (relationships): 18
	Data flow (relationships): 17
	Teammates (changes): 5 Testing (changes): 3 Implementing (changes): 8

Target audience vs. data visualization questions

	Implications (changes): 10
	Refactoring (changes): 18
	Rationale (changes): 1 Building and branching (changes): 5
	Concurrency (elements): 9
	Method properties (elements): 1 _
	Dependencies (relationships): 12
	Contracts (relationships): 2
	Intent and Implementation (elements): 6
professional: 159	History (changes): 18
	Location (elements): 19
	Architecture (relationships): 10
	Performance (elements): 20
academic: 2	Type relationships (relationships): 5
academic/professional: 29	Debugging (changes): 26
education: 23	Control flow (relationships): 18
	Data flow (relationships): 17
	Teammates (changes): 5
	Testing (changes): 3
	Implementing (changes): 8

Target audience vs. data visualization questions

Implications (changes): 10 Refactoring (changes): 18 Rationale (changes): 1 Building and branching (changes): 5 Concurrency (elements): 9 Method properties (elements): 1 Dependencies (relationships): 12 Contracts (relationships): 2 Intent and Implementation (elements): 6 History (changes): 18 professional: 159 Location (elements): 19 Architecture (relationships): 10 Performance (elements): 20 academic: 2 Type relationships (relationships): 5 academic/professional: 29 Debugging (changes): 26 Control flow (relationships): 18 education: 23 Data flow (relationships): 17 Teammates (changes): 5 Testing (changes): 3 Implementing (changes): 8

Visualzation sources vs. visualization paradigms



Visualzation sources vs. visualization paradigms

Hierarchical and Graph-Based Techniques: 115 Static code analysis data: 95 Icon-based techniques / Icon displays: 17 Version control system data: 37 3D techniques: 19 Timelines: 30 Software execution data: 83 Info graphics: 20 Animation: 18 Issue management data: 3 Mailing list: 6 Pixel-oriented techniques: 5 Sourcecode: 25 Source code highlighting: 22

Visualization Paradigms by Keim, Kriegel (1996): "Visualization Techniques for Mining Large Databases: A Comparison"

Geometric projection techniques: 3 -

Visualzation sources vs. visualization paradigms



Visualzation sources vs. visualization paradigms

Geometric projection techniques: 3



Visualzation sources vs. visualization paradigms

Geometric projection techniques: 3



Visualzation sources vs. visualization paradigms

Geometric projection techniques: 3



Visualzation sources vs. visualization paradigms

Geometric projection techniques: 3



Problem domain visualization contribution vs. visualization paradigms



Problem domain visualization contribution vs. visualization paradigms



Problem domain visualization contribution vs. visualization paradigms



Problem domain visualization contribution vs. visualization paradigms

	Info graphics: 20
Managing software projects: 5	Geometric projection techniques: 4
	Pixel-oriented techniques: 5
Understanding software execution: 85	Timelines: 29
How software is developed: 16	Hierarchical and Graph-Based Techniques: 113
Exploring change in software over time: 33	
Understanding software structure: 81	
	Animation: 17
	3D techniques: 19
	Source code highlighting: 17
Defining and maintaining requirements: 21	Icon-based techniques / Icon displays: 17

Problem domain visualization contribution vs. visualization paradigms



Problem domain visualization contribution vs. visualization paradigms



Problem domain visualization contribution vs. visualization paradigms



79 Evaluations of 62 design studies

To evaluate, or not to evaluate...



percentage of selected design studies (N=62)

Evaluation scope vs. evaluation aspects



Evaluation scope vs. evaluation aspects



Evaluation scope vs. evaluation aspects



Evaluation scope vs. evaluation aspects





Evaluation scope vs. evaluation aspects












Evaluation scope vs. evaluation methods



Evaluation aspects vs. evaluation methods



Evaluation aspects vs. evaluation methods



Evaluation aspects vs. evaluation methods



Evaluation aspects vs. evaluation methods

	Usability test: 2
Evaluating user experience: 30	Questionnaire/Questionary: 17
	Field observation: 18
Understanding environments and work practices: 15	Pilot (or exploratory) study: 10
	Informal evaluation: 21
Evaluating visual data analysis and reasoning: 42	
	Case study: 31
Evaluating communication through visualization: 4	
Valuating functionality of a visualization cool. 4	Laboratory observation: 4 Log analysis: 1
Evaluating visualization algorithms: 22	Comparative study (concurrent control): 12
Evaluating collaborative data analysis: 1	Heuristic evaluation: 2
Evaluating user performance; time and accuracy: 6	Algorithmic performance: 5

Evaluation aspects vs. evaluation methods

	Interview: 1 Usability test: 2
Evaluating user experience: 30	Questionnaire/Questionary: 17
	Field observation: 18
Understanding environments and work practices: 15	Pilot (or exploratory) study: 10
Fuckuting visual data analysis and searching, 1	Informal evaluation: 21
Evaluating visual data analysis and reasoning: 42	
	Case study: 31
Evaluating communication through visualization: 4	
Validating functionality of a visualization tool: 4	Laboratory observation: 4
	Log analysis: 1
Evaluating visualization algorithms: 22	Comparative study (concurrent control): 12
Evaluating collaborative data analysis: 1	Heuristic evaluation: 2
Evaluating user performance; time and accuracy: 6	Algorithmic performance: 5

Evaluation aspects vs. evaluation methods

	Usability test: 2
	Questionnaire/Questionary: 17
Evaluating user experience: 30	Field observation: 18
Understanding environments and work practices: 15	Pilot (or exploratory) study: 10
Evaluating visual data analysis and reasoning: 42	Informal evaluation: 21
	Case study: 31
Evaluating communication through visualization: 4	
Validating functionality of a visualization tool: 4	Laboratory observation: 4
Evaluating visualization algorithms: 22	Log analysis: 1 Comparative study (concurrent control): 12
Evaluating collaborative data analysis: 1	Heuristic evaluation: 2
Evaluating user performance; time and accuracy: 6	Algorithmic performance: 5

Evaluation aspects vs. evaluation methods



Evaluation aspects vs. evaluation methods

	Usability test: 2
Evaluating user experience: 30	Questionnaire/Questionary: 17
	Field observation: 18
Understanding environments and work practices: 15	Pilot (or exploratory) study: 10
Evaluating visual data analysis and reasoning: 42	Informal evaluation: 21
	Case study: 31
Evaluating communication through visualization: 4	
Validating functionality of a visualization tool: 4	Laboratory observation: 4 Log analysis: 1
Evaluating visualization algorithms: 22	Comparative study (concurrent control): 12
Evaluating collaborative data analysis: 1	Heuristic evaluation: 2
Evaluating user performance; time and accuracy: 6	Algorithmic performance: 5

Evaluation aspects vs. evaluation methods

	Usability test: 2
	Questionnaire/Questionary: 17
Evaluating user experience: 30	Field observation: 18
Understanding environments and work practices: 15	Pilot (or exploratory) study: 10
Evaluating visual data analysis and reasoning: 42	Informal evaluation: 21
	Case study: 31
Evaluating communication through visualization: 4	
Validating functionality of a visualization tool: 4	Laboratory observation: 4
Evaluating visualization algorithms: 22	Log analysis: 1 Comparative study (concurrent control): 12
Evaluating collaborative data analysis: 1	Heuristic evaluation: 2
Evaluating user performance; time and accuracy: 6	Algorithmic performance: 5

Who are the evaluation subjects?

Target audience vs. evaluation subjects



And the winner is...

Evaluation score by target audience



Evaluation method ranking roughly based on B. Kitchenham's "Study design hierarchy for Software Engineering"

And the winner is...

Evaluation score by venue



Evaluation method ranking roughly based on B. Kitchenham's "Study design hierarchy for Software Engineering"

Does the evaluation score improve over time?

Evaluation score by year



Evaluation method ranking roughly based on B. Kitchenham's "Study design hierarchy for Software Engineering"

Results

- Widely different software visualization needs based on:
 - ▶ the target audience (e.g. professional vs. education)
 - the programming language/paradigm

Results

- Widely different software visualization needs based on:
 - ▶ the target audience (e.g. professional vs. education)
 - the programming language/paradigm
- Naturally different evaluation aspects and methods depending on evaluation scope (or stage):
 - Informal evaluation methods in early evaluation scopes
 - More formal evaluation methods with later evaluation scopes
 - Overall rather low evaluation effort
 - Minimum standard seems to have been established

Results

- Widely different software visualization needs based on:
 - ▶ the target audience (e.g. professional vs. education)
 - the programming language/paradigm
- Naturally different evaluation aspects and methods depending on evaluation scope (or stage):
 - Informal evaluation methods in early evaluation scopes
 - More formal evaluation methods with later evaluation scopes
 - Overall rather low evaluation effort
 - Minimum standard seems to have been established
- Consolidation from other venues to SOFTVIS/VISSOFT

Discussion

Selection bias (ACM, IEEE, SD)

- Selection bias (ACM, IEEE, SD)
- ► Scope of a seminar project vs. systematic literature review
 - Iterative process (expertise and time)
 - Establishing consensus (peer-review/repeatable?)

- Selection bias (ACM, IEEE, SD)
- ► Scope of a seminar project vs. systematic literature review
 - Iterative process (expertise and time)
 - Establishing consensus (peer-review/repeatable?)
- Paper classification (What is a design study?)

- Selection bias (ACM, IEEE, SD)
- Scope of a seminar project vs. systematic literature review
 - Iterative process (expertise and time)
 - Establishing consensus (peer-review/repeatable?)
- Paper classification (What is a design study?)
- Data extraction

Thanks for listening!

Thanks for listening! Questions?