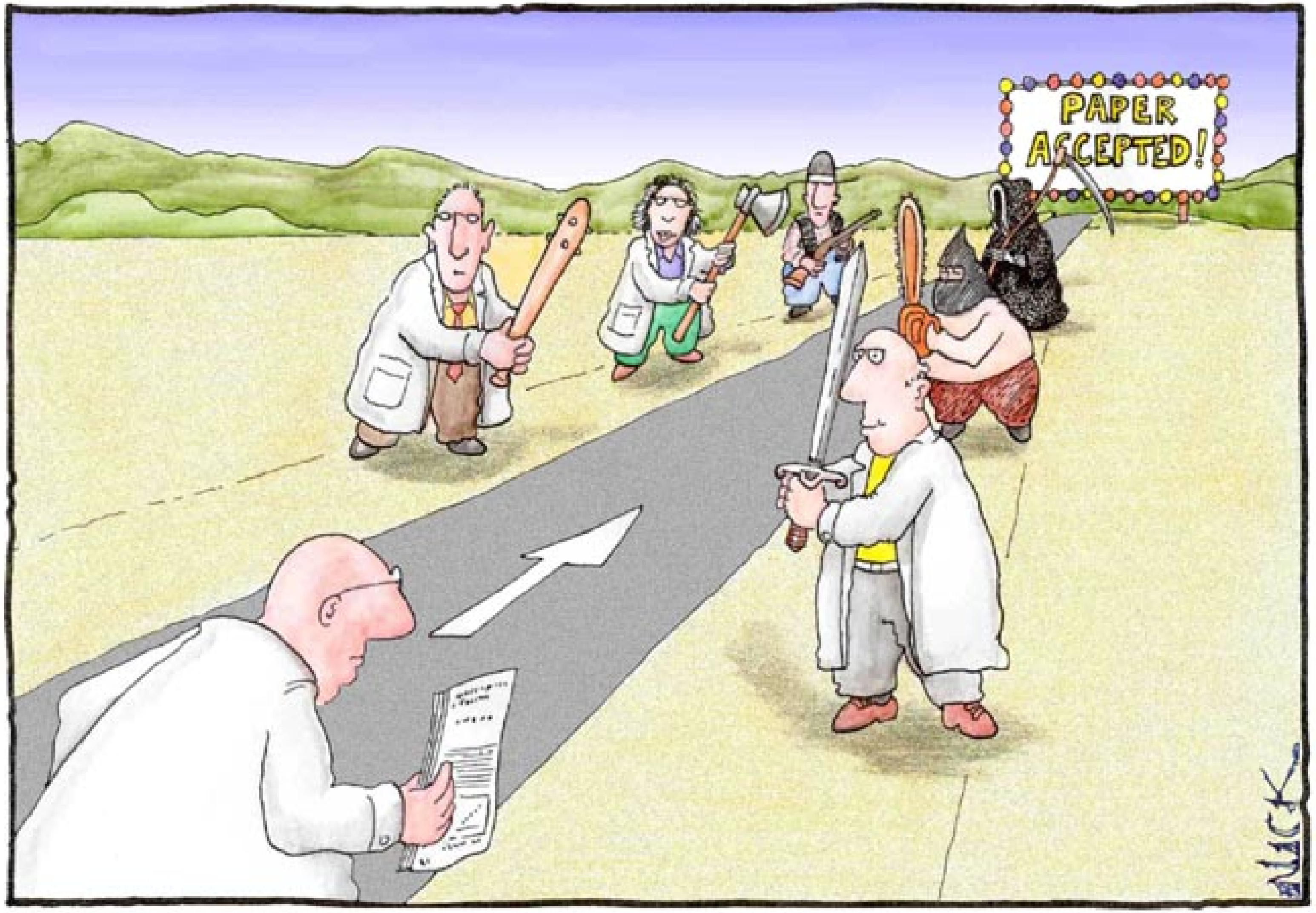
Petri Nets in Software Technology

Scientific Writing

Hauptseminar (SS 19)
Wednesday, 2. DS, APB/3105
Thomas Kühn (thomas.kuehn3@tu-dresden.de)

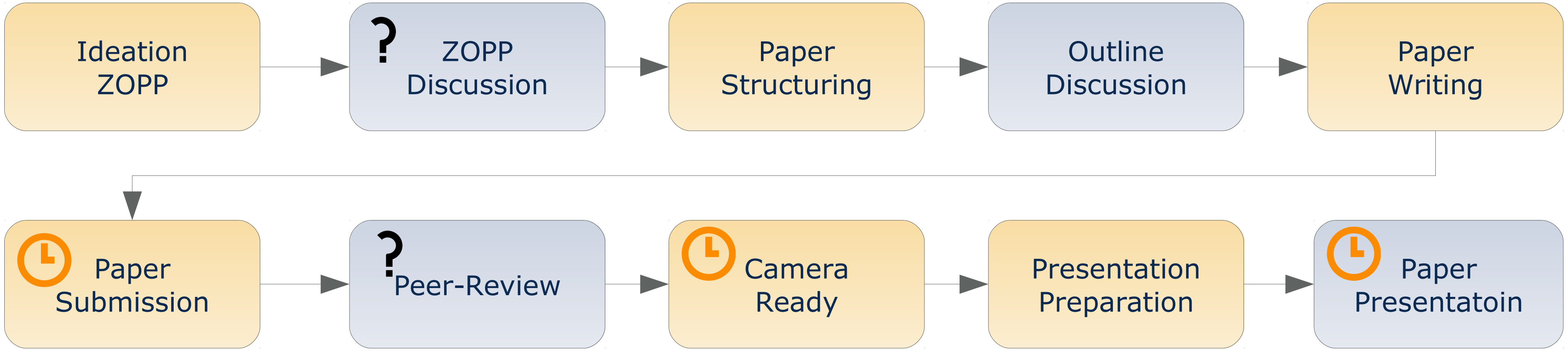
Doing Science is Hard?



Most scientists regarded the new streamlined peer-review process as "quite an improvement."

Cartoon by Nick D Kim strange-matter.net. Used by permission.

Organization
Overview



- *Give scientific presentations* (20 min + 10 min discussion)
  - Individual presentations **05.06.2019**
- *Write a research paper* (>=5 pages ACM Style)
  - Paper submission[1] **12.07.2019**

[1] Per mail

Organisation

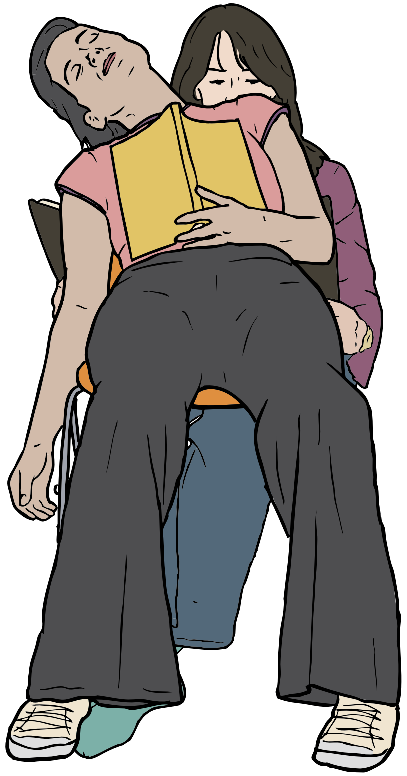
Overarching Task

- Pick a research topic within **Petri Nets in Software Technology**
- Work alone
- Give a presentation on your topic
  - 30 minutes presentation
  - Practice giving presentations
  - Follow the advise provided throughout the next lecture
- Writing a research paper
  - >=5 pages two column paper
  - Preferably use LaTeX
  - Style: *ACM SIGPLAN Conference Format* (`acmart`)[1]
  - Apply the skills learned throughout this lecture

[1] http://www.sigplan.org/Resources/Author/

Scientific Research
Daily Business

Reading **Writing** **Organizing**



Images from OpenClipart.org (Creative Commons by Steve Lambert)

Common Tasks

- Iterative process from idea to written paper
- Develop *“Ziel-orientierte Projektplan”* (ZOPP)
- Write an **abstract** early
- **Structure** your paper *(chapters, sections, paragraphs)*
- Write **outlines** for each chapter, section, ...

Paper Writing Process
“Ziel-orientierte Projektplan” (ZOPP)

Mindmap with four arcs capturing the main idea of your paper

What *problems* does your paper address?

- Pick one main problem and add detailed subproblems

What are the corresponding *goals* of your paper?

- Align goals and subgoals to problems you address

What *solution* does your paper present?

- List your solution and subsolutions aligned to goals

What are *success criteria* or how to *evaluate* your solution?

- Specify *functional* and *non-functional* requirements
- *Evaluate* whether solution achieves goals

Paper Writing Process

Problem Definition

- Derived from a ZOPP
- State the *problem* in one or two sentences including your *goal*
- Highlight your *solution* in one or two sentences
- State your *success criteria* or how you *evaluate* your solution

> "Currently, CROM models can be created textual or graphically without taking the well-formedness rules for these models into account. Hence, the goal of this work is to create a Eclipse-based plugin, which validates CROM models with respect to well-formedness. Additionally, the plugin should be easy to integrate into the existing editors."
>
> – *Kühn (2015)*

Paper Writing Process
Abstract

- Includes the *problem definition* as crucial part
- Prepends a background of this research *(Why is it important?)*
- Summarizes the major problems and goals
- Appends a description of your evaluation *(success proof)*

Example Abstract

“Modelling context-dependent domains is hard, as capturing multiple context-dependent concepts and constraints easily leads to inconsistent models or unintended restrictions. However, current semantic technologies not yet support reasoning on context-dependent domains. To remedy this, we introduced ConDL, a set of novel description logics tailored to reason on contextual knowledge, as well as JConHT, a dedicated reasoner for ConDL ontologies. ConDL enables reasoning on the consistency and satisfiability of context-dependent domain models, e.g., Compartment Role Object Models (CROM). We evaluate the suitability and efficiency of our approach by reasoning on a modelled banking application and measuring the performance on randomly generated models.“

– Böhme et al. (2017)

Recurring structure of scientific papers in computer science

- Introduction / Motivation
- Background / Preliminaries / Contemporary Approaches
- Concept / Methodology
- Implementation / Realization
- Evaluation / Case Study / Illustration / Discussion
- Related work
- Conclusion / *Contributions*

Paper Writing Process

How it feels like

Images from https://parske-shop.de/Gartentechnik/Haecksler/Erco-Holzhaecksler-GHX-CH1900-Zapfwelle-oder-Benzin.html

Paper Writing Process
How it should be

Image from MaxPixel.net (Creative Commons by Markus Baumeler)

Common Tasks

- Structuring paper
  - Define chapters, sections, subsections
  - Summarize outline of each part
- Write individual paragraphs
  - Include individual artifact *images, tables, listings*
  - Outline points become individual paragraphs
  - Write a structured paragraph
  - Finalize paragraphs and transitions

Structuring Papers

- Employ *recurring structure* of scientific papers in computer science
- Define the structure by means of **headings** for

  parts, chapters, sections, subsections, ...
- Write short outlines/summaries for each heading
  - Use bullet points and short statements outlining the intended content
  - Which questions are answered?
  - What arguments are provided?
  - What solutions/conclusions are described?

Outlines

- *Not* written for the *reader*, but for **you**
- Summarizes intended content of *chapter*, *section*, *subsection*, and …
  - What concepts/ideas must be introduced/discusses?
  - Which parts in your text cover which parts of your ZOPP?
  - What questions should be raised and answered?
- Helps to focus writing and avoiding running off the topic
- Useful to track writing progress

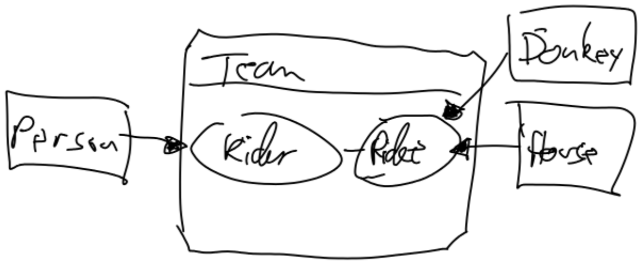
Outline Example

2.3. Graphical Editor Frameworks

- *Give a short overview on typical frameworks for the development of graphical editors*
  - *Provide a clear description of the typical aspects that need to be implemented in such a framework, e.g. Language Concern (Metamodel), Editor Concerns, Edit Policies.*
  - *Describe GEF, GMF, Graphiti, Sirius, and EuGENia*

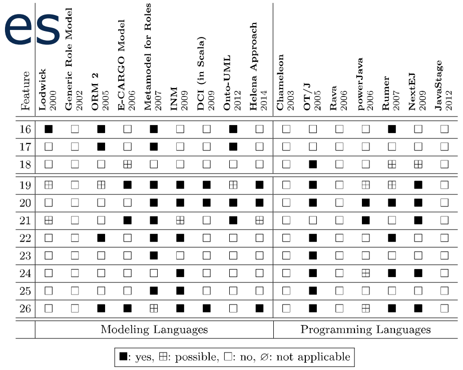
Include Artifacts

Images



- Start with sketches, low resolution images
- Later create scalable vector images *(time consuming)*

Tables

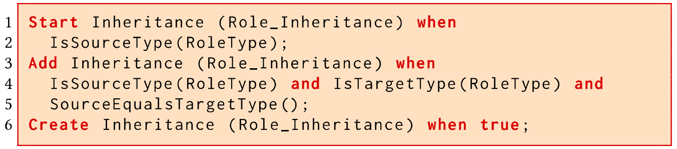


Feature	**Lodwick** 2000	**Generic Role Model** 2002	**ORM 2** 2005	**E-CARGO Model** 2006	**Metamodel for Roles** 2007	**INM** 2009	**DCI (in Scala)** 2009	**Onto-UML** 2012	**Helena Approach** 2014	**Chameleon** 2003	**OT/J** 2005	**Rava** 2006	**powerJava** 2006	**Rumer** 2007	**NextEJ** 2009	**JavaStage** 2012
16	■	□	■	□	■	□	□	■	□	□	□	□	□	■	□	□
17	□	□	■	□	■	□	□	■	□	□	□	□	□	□	□	□
18	□	□	□	⊞	□	□	□	□	□	□	■	□	□	⊞	⊞	□
19	⊞	□	⊞	■	■	■	■	⊞	■	□	■	□	⊞	⊞	■	□
20	□	□	□	□	■	■	■	■	■	□	■	□	■	■	■	□
21	⊞	□	□	■	■	⊞	□	■	⊞	□	□	□	■	□	■	□
22	□	□	■	□	■	■	□	□	□	□	■	□	□	■	□	□
23	□	□	□	□	■	□	□	□	□	□	■	□	□	□	□	□
24	□	□	□	□	□	■	□	□	□	□	■	□	⊞	■	■	□
25	□	□	□	□	■	■	□	□	□	□	■	□	□	□	□	□
26	□	□	■	■	⊞	■	■	□	■	□	■	□	⊞	■	■	□
	Modeling Languages									Programming Languages						

■: yes, ⊞: possible, □: no, ∅: not applicable

- Use generator for latex tables
- Refine/optimize table later

Listings



```
Start Inheritance (Role_Inheritance) when
  IsSourceType(RoleType);
Add Inheritance (Role_Inheritance) when
  IsSourceType(RoleType) and IsTargetType(RoleType) and
  SourceEqualsTargetType();
Create Inheritance (Role_Inheritance) when true;
```

- Start with source code snippets
- Later remove all unnecessary statements

Structured Paragraph

- Write paragraph for each major point of the outline
- Typical structure of paragraphs
  - **Thesis question**
  - **Thesis statement** topic, purpose or development scheme
  - Supporting/opposing **arguments**, claims, evidence or warrants
  - **Thesis conclusion** and transition
- Enumerate arguments

Structured Paragraph Example

2.3. Graphical Editor Frameworks

There exists several graphical editor frameworks for all platforms.

As our prototypical GEPL is based on Eclipse, we focus on corresponding frameworks.

1) The Graphical Editing Framework *(GEF) is the basis for most other frameworks, as it facilitates means to implement rich graphical Java applications.*
2) The Graphical Modeling Framework *(GMF) is a model-driven editor generator, where the various concerns are specified in interrelated models, e.g., the domain model, the graphical definition, and the tooling definition.*
3) EuGENia *and* Sirius *are both frameworks for textual respectively visual specification of GMF editors.*
4) Graphiti *utilizes EMF models to provide a uniform pictogram model linked to a custom domain model, whereas visualizations, behaviors, and edit policies must be manually implemented in IPattern and IFeature classes..*

None of them natively supports the modular definition of language features.

Finalize Paragraphs and Transitions

- Remove enumeration from structured paragraph
- Improve **wording**
- Link thesis question, thesis statement, and arguments with **transitions**
    - Additions: *Moreover, furthermore, especially, in detail, …*
    - Cause & Effect: *Thus, accordingly, as a result, consequently, hence, …*
    - *(More in the appendix)*
- Introduce **transition** and **pivot** sentences
- Add **controlling idea** to thesis conclusion

Finalized Paragraph Example

2.3. Graphical Editor Frameworks

There exists a ***plethora*** *of graphical editor frameworks for all platforms,* ***yet*** *as our prototypical GEPL targets Eclipse, we focus on* ***associated*** *frameworks.*

In general*, the* Graphical Editing Framework *(GEF) is the basis for most other frameworks, as it facilitates means to implement rich graphical Java applications.*

On top of GEF, there exists both model-driven and model-based frameworks.

For the former*,* the Graphical Modeling Framework *(GMF) is a model-driven editor generator, where the various concerns are specified in interrelated models, e.g., the domain model, the graphical definition, and the tooling definition.*

Moreover*,* EuGENia *[@kolovos2017eugenia] and* Sirius *[@viyovic2014sirius] are both frameworks for textual respectively visual specification of GMF editors.*

By contrast*,* Graphiti *utilizes EMF models to provide a uniform pictogram model linked to a custom domain model, whereas visualizations, behaviors, and edit policies must be manually implemented in IPattern and IFeature classes.*

Although these frameworks significantly simplify the design of graphical editors, *none of them natively supports the modular definition of language features.*

Typical Transitions

Cause and Effect

- accordingly
- as a result
- consequently
- hence
- it follows, then
- since
- so
- then
- therefore
- thus

Conclusion

- as a result
- consequently
- hence
- in conclusion, then
- in short
- in sum, then
- it follows, then
- so
- the upshot of all this is that
- therefore
- thus
- to sum up
- to summarize

Typical Transitions

Comparison

- along the same lines
- in the same way
- likewise
- similarly

Addition

- also
- and
- besides furthermore
- in addition
- in fact
- indeed
- moreover
- so too

Contrast

- although
- but
- by contrast
- conversely
- despite the fact that
- even though
- however
- in contrast
- nevertheless
- nonetheless
- on the contrary
- on the other hand
- regardless
- whereas
- while
- yet

Typical Transitions

Concession

- admittedly
- although it is true that
- granted
- I concede that
- of course
- naturally
- to be sure

Example

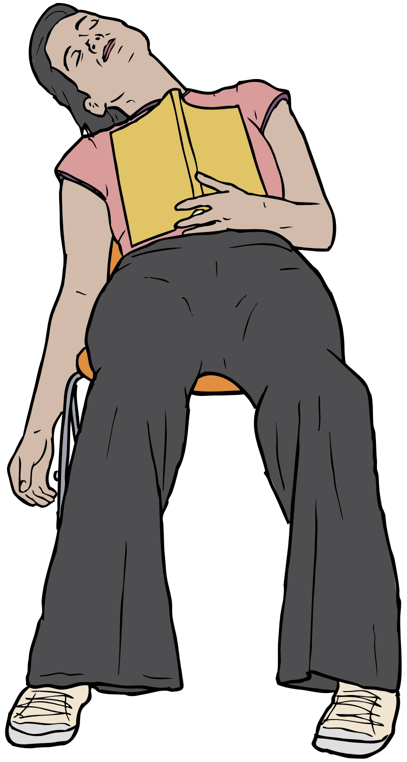
- after all
- as an illustration
- consider
- for example
- for instance
- specifically
- to take a case in
- point

Elaboration

- actually
- by extension
- in short
- that is
- in other words
- to put it another way
- to put it bluntly
- to put it succinctly
- ultimately

Scientific Research
Daily Business

Reading



Writing

Organizing

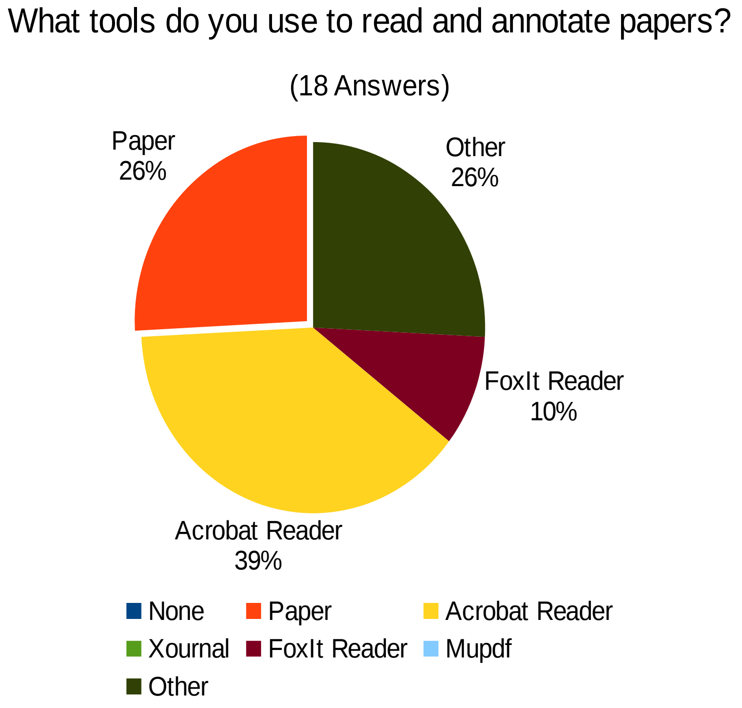
Images from OpenClipart.org (Creative Commons by Steve Lambert)

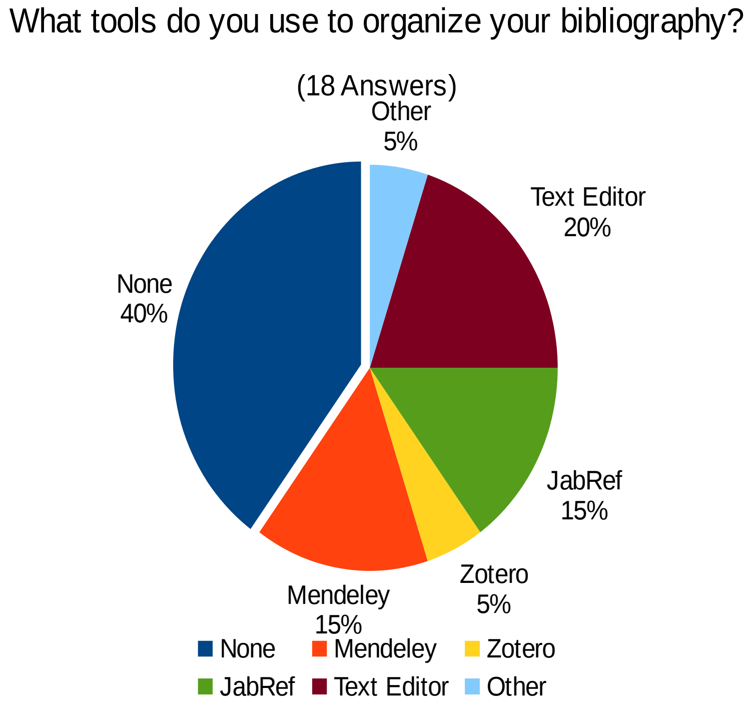
Common Tasks

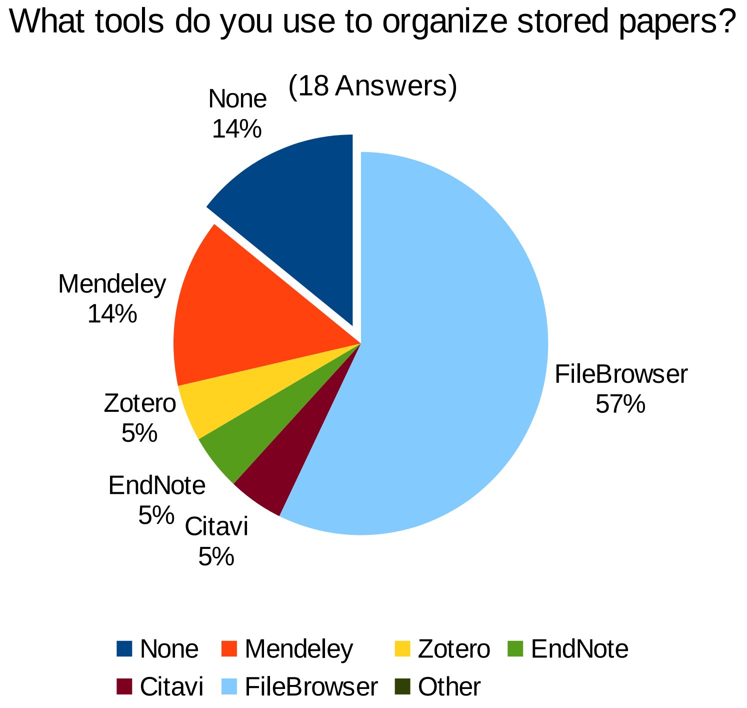
- Find relevant / related publications
  - Query scientific search engines
  - Look up *BibTex* for specific publications from the web
- Investigate found publications
  - Skim papers
  - Make notes and hints
  - Organize downloaded files
  - Maintain a corresponding **bibliography** of *BibTex* entries

A Small Survey

- Q1:What tools do you use to read and annotate papers?
- Q2:*What tools do you use to organize your bibliography?*
- Q3:*What tools do you use to organize stored papers?*







BibTex

```
@inproceedings{kuehn2015choosy,
  title = {Choosy and picky: configuration of language product lines},
  author = {K{\"u}hn, Thomas and Cazzola, Walter and Olivares, Diego Mathias},
  booktitle = {Proceedings of the 19th International Conference on Software
                    Product Line},
  year = {2015},
  organization = {ACM},
  pages = {71--80},
  citations = {1},
  file = {:./Kuehn/Thomas Kuehn_Choosy and picky - configuration of language
            product lines.pdf:PDF},
  howpublished = {\url{http://dl.acm.org/citation.cfm?id=2791092}},
  owner = {thomas},
  timestamp = {2015.09.07}
}
```

1) http://www.bibtex.org/

Finding Relevant / Related Publications

- Query scientific search engines

  When looking for complex search terms

Google Scholar	(free)	https://scholar.google.com
Elsevier Scopus	(registration)	https://www.scopus.com
Academia	(registration)	https://www.academia.edu
Sci-Hub	(illegal)	http://sci-hub. …

- Query publishers directly

  For a specific journal or conference in computer science

IEEE Xplore	https://ieeexplore.ieee.org
ACM Digital Library	https://dl.acm.org
Springer Link	https://link.springer.com
Elsevier ScienceDirect	https://www.sciencedirect.com

Investigating Found Publications

- Use appropriate reader

  Permit highlighting, comments, and annotations
  - *Xournal, Acrobat Reader, Foxit Reader, Mupdf, …*

- Use tool to manage your bibliography

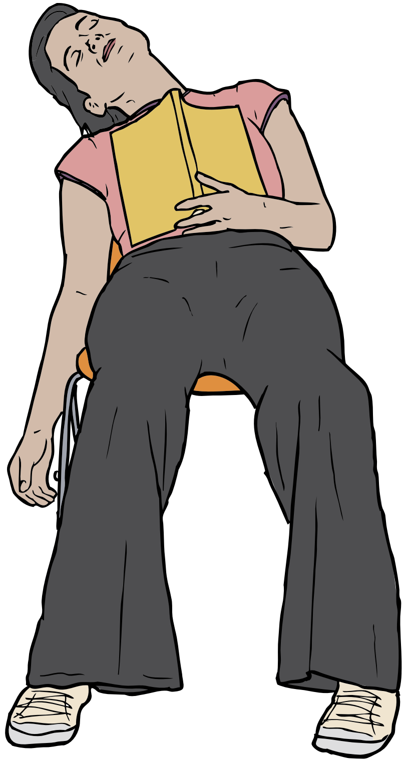
  Organize, search, and annotate your BibTex entries
  - *JabRef, BibDesk, EndNote, …*

- Use one tool for both

  Manage, search, and comment both PDF documents and BibTex entry

• Mendeley	(freemium)	https://www.mendeley.com
• Zotero	(freemium)	https://www.zotero.org
• Citavi	(freemium)	https://www.citavi.com

Reading



Writing

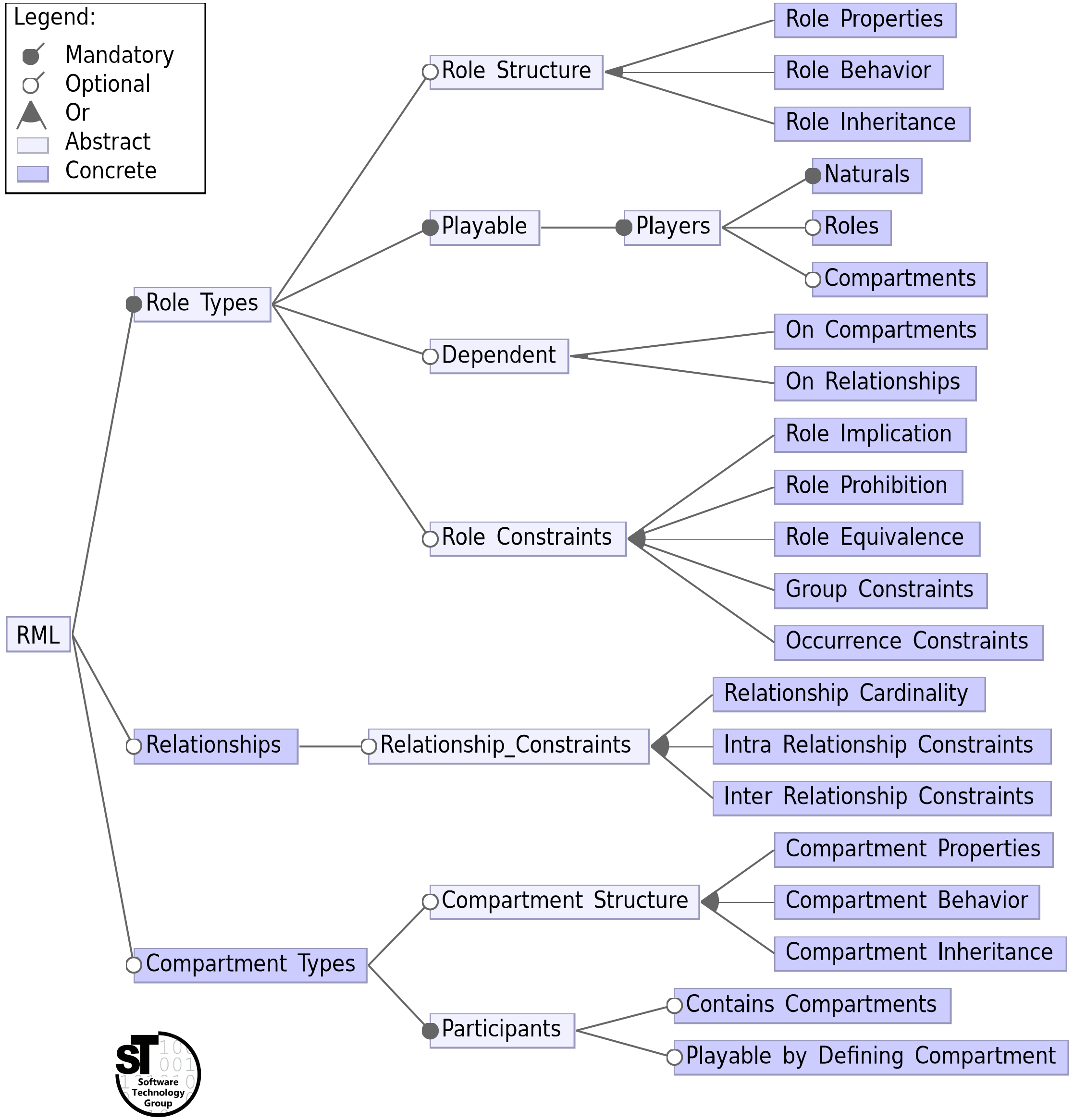
Organizing

Images from OpenClipart.org (Creative Commons by Steve Lambert)

Common Tasks

- Create a classification scheme
  - Identify classifying criteria
  - Set up list, tree, or map of terms, features, requirements, or classes
- Classify papers and approaches
  - Indicate found criteria in papers
  - Maintain classification for each paper or approach
  - Produce diagrams for comparison *tables*, *bubble charts*, or *kiviat graphs*

Classifying Research
Classification Scheme



For Papers

- Taxonomy of terms
- General classification of research papers by Shaw
- Orthogonal dimensions of classes

For Approaches

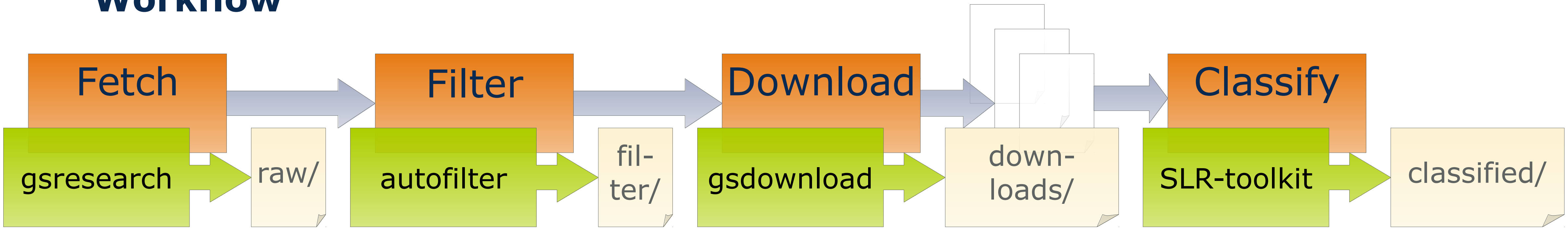
- List of (non-)functional requirements
- List of qualitative and quantitative properties
- Feature model consisting of features and dependencies

Mining a Classification Scheme

- Existence of general classification schemata, *e.g.,*
  Shaw's classification of research [Shaw2002]
- Utilize existing classifications from related **surveys** or **PhD theses**, *e.g.,*
  Feature model for language workbenches [Erdweg et al.2015]
- Creating new classification scheme
  - Start from existing schemata; extend missing dimension
  - Retrieve requirements, goals, or features from publications

Never use made up classification schemata

Workflow

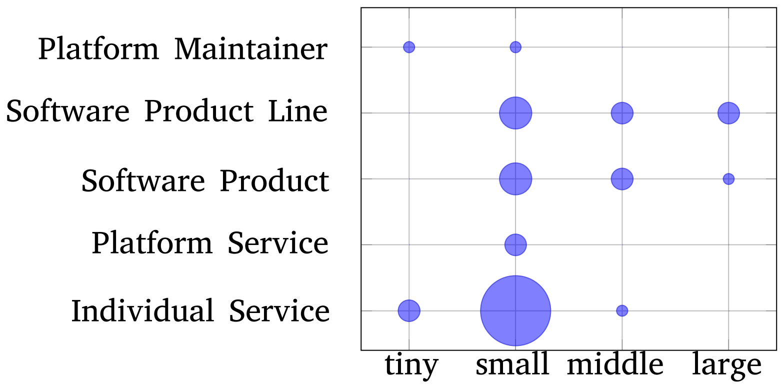


- After selecting relevant papers or approaches
- Investigate each paper annotate mentioned requirements and features
- Use tool support to track annotations for each paper or approach, *e.g.,*

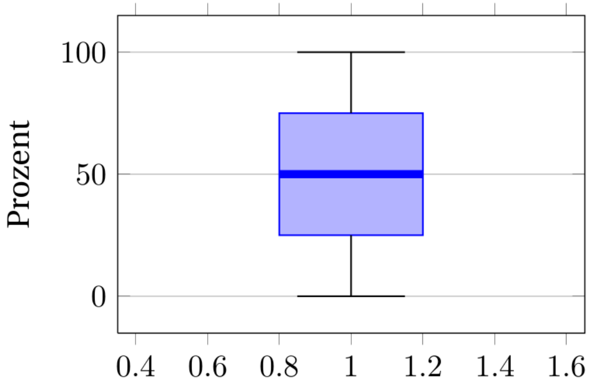
  SLR-Toolkit*[1] uses BibTex annotation and supports arbitrary hierarchical classification schemes*

[1] https://github.com/sebastiangoetz/slr-toolkit

Classifying Research
Showing Comparison Results



Bubble Chart



Box Plot

Qualitative Evaluation

- Comparison tables

  Terms, Icons (○ ◐ ● ◌), …
- Diagrams for detailed comparison

  (2D) *Pie charts, Histograms, …*

  (3D) *Bubble charts, 3D Plots, …*

  (nD) *Kiviatgraphs, Parallel Hierarchies, …*

Quantitative Evaluation

- Tables for basic analysis

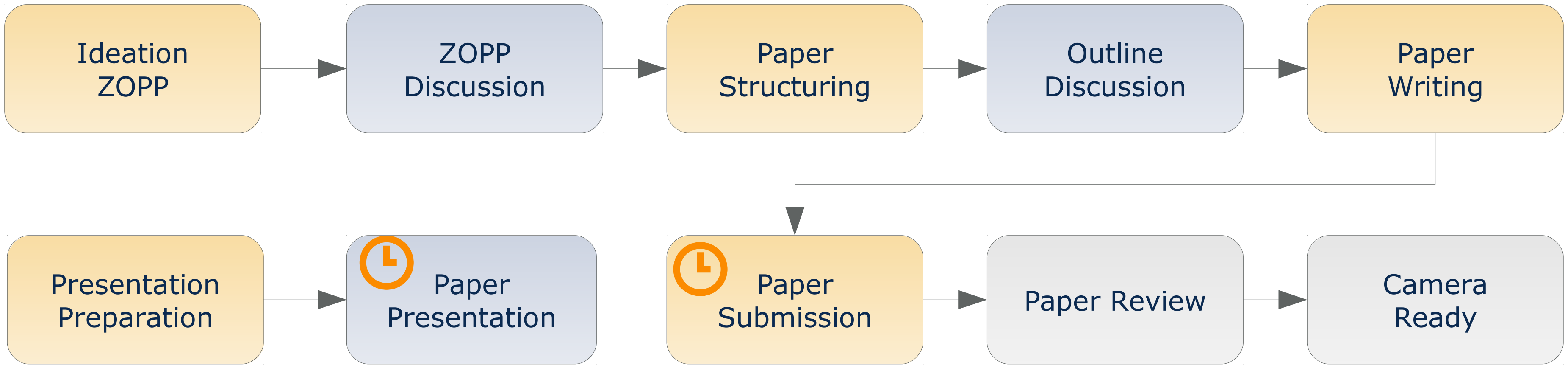
  Standard deviation (+/-), Mean, …
- Plots for more complex analyses

  (2D) *Plots, Box plots, Line chart, …*

  (3D) *Heat Maps, 3D Plots, …*

  (nD) *Parallel Koordinates, …*

Organization
Overview



- *Give scientific presentations* (20 min + 10 min discussion)
  - Individual presentations **05.06.2019**
- *Write a research paper* (>=5 pages ACM Style)
  - Paper submission[1] **12.07.2019**

[1] Per mail

Software for Digital Health

Scientific Writing