TECHNISCHE UNIVERSITÄT DRESDEN

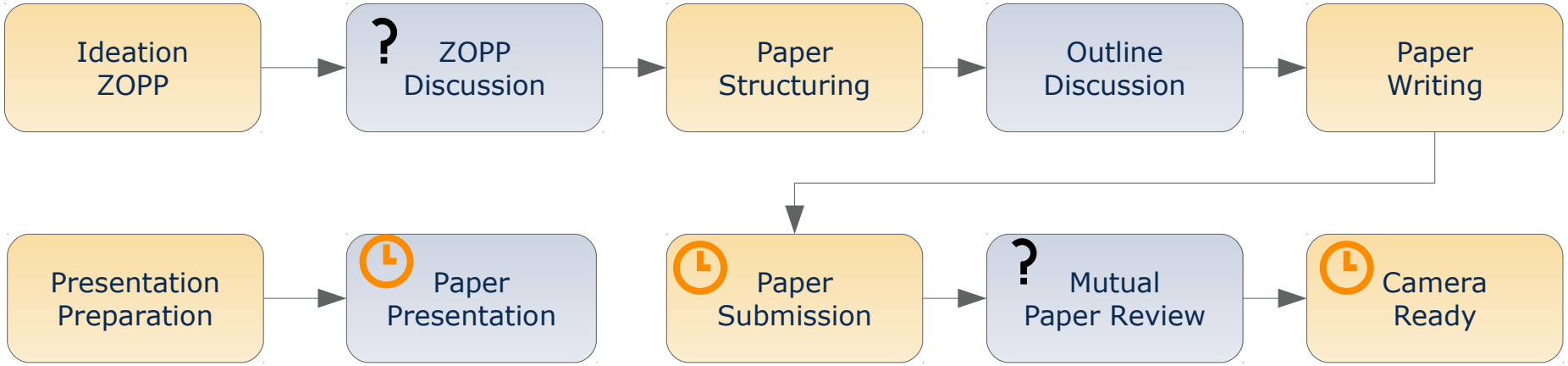
Department of Computer Science Institute for Software and Multimedia Technology, Software Technology Group

Academic Skills in Computer Science (ASiCS)

Structuring, Outlining and Structured Writing

Exercise
Thursday, 6. DS, APB/E001
Thomas Kühn (thomas.kuehn3@tu-dresden.de)

Organization
Paper Writing Process

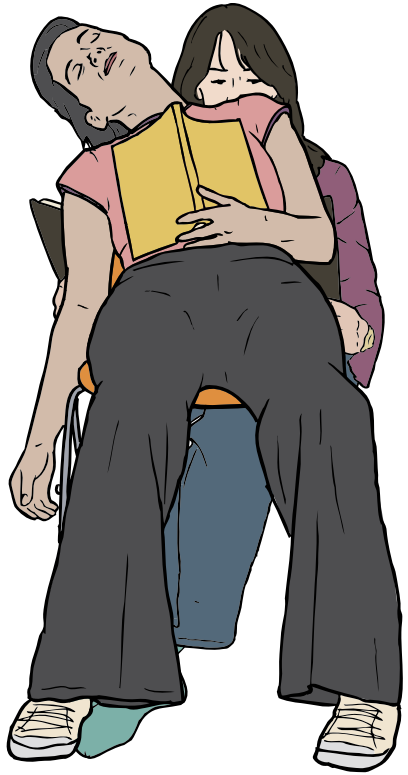


- *Give scientific presentations* (20 min + 10 min discussion)
  - Individual presentations **19.06.2018** and **21.06.2018**
- *Write a research paper* (4-5 pages ACM Style)
  - Paper submission[1] **05.07.2018**
  - Camera ready submission[1] **20.07.2018**

1) easychair.org

Scientific Research
Daily Business

Reading



Writing

Organizing

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

Previous Tasks

1) Create a **Classification Scheme** for your related work.
2) Classify at least 5 papers wrt. this **Classification Scheme**
3) Create and add a comparison table to the *Related Work* section.

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

- 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

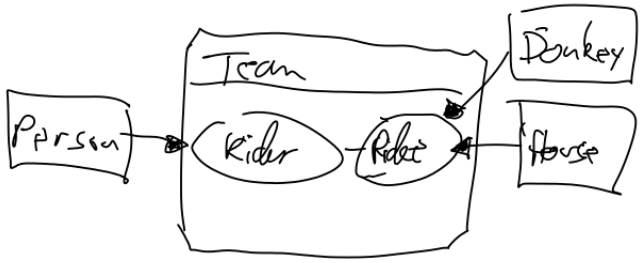
Outline Example

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	Ledwich 2000	Generic Role Model 2002	ORM 2 2005	E-CARGO Model 2006	Metamodel for Roles 2007	INM 2009	DCI (in Scala) 2010	Oto-UML 2012	Helena Approach 2014	Chameleon 2003	OT/J 2005	Rava 2006	powerJava 2006	Reiner 2007	NestEJ 2009	JavaSage 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.*

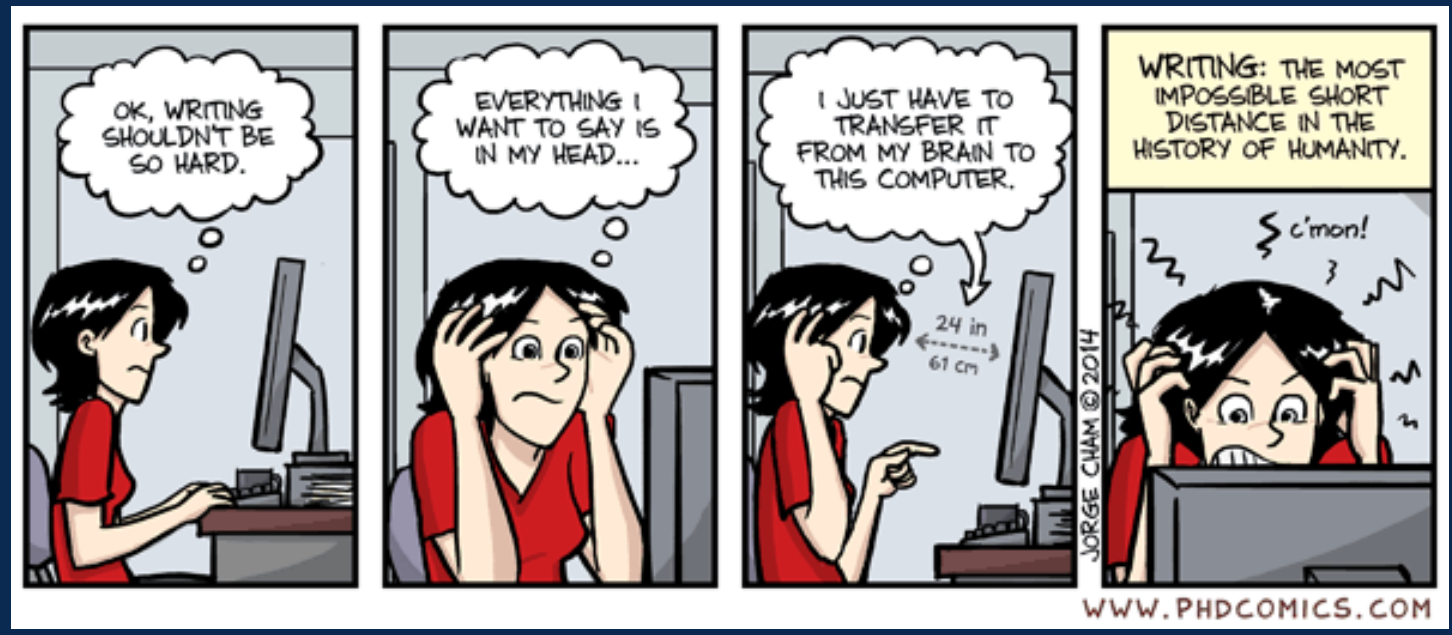
***Although these frameworks significantly simplify the design of graphical editors,** none of them natively supports the modular definition of language features.*

Paper Writing Process
Exercise

1) Revise the structure of your paper.
2) Write an outline/summary for each *section, subsection, subsubsection.*
3) Hand in the *structured* document with *outlines* before the next exercise.

Structuring, Outlining and Writing

Writing



"Piled Higher and Deeper" by Jorge Cham (www.phdcomics.com)
used with permission

Appendix
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

Appendix
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

Appendix
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