35. Writing a Literature Overview Paper

Prof. Dr. Uwe Aßmann
Softwaretechnologie
Fakultät Informatik
Technische Universität Dresden

2013-0.4, 13-11-12
http://st.inf.tu-dresden.de/acse

1) Writing

Obligatory Literature

- Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. ACM Trans. Auton. Adapt. Syst., 4(2):14:1-14:42, May 2009.
- Wayne Wolf, Cyber-physical Systems. IEEE Computer, 2009
- [OpenImp] Kiczales Gregor, Lamping John, Christina Videira Lopies, Chris Maeda, Anurag Mendhekar, and Gail Murphy. Open implementation design guidelines. In Proceedings of the 1997 International Conference on Software Engineering, pages 481-490. ACM Press, 1997.

35.1 Literature Analysis Paper

Rpt.: A Survey Paper *(Literature Analysis)* is an Enhanced Model Paper

- A Survey Paper presents a survey of work in an area F.
  - Characterization criteria (comparison criteria) are used to structure the field.
  - Every approach is characterized or classified according to the criteria
  - Features of every approach are *analyzed*
- The results are research questions, research limits, success criteria, i.e., if the literature analysis does not end in a good research hypothesis, it is too shallow
- Ex. First chapters of "Invasive Software Composition"

Attention: every Bachelor/Master/PhD thesis needs at least one chapter of Literature Analysis ("related work")

Characterizing	Documenting	Enhanced Descriptive or Analytic Model	Analysis or Examples
Under the following criteria, how can we structure field F?	Structuring	Enhanced structural model of field; comparability of appr.	Analysis of open research questions
		Limits of field, open research questions **research hypothesis**	Analysis of limits of field
		Success criteria for research	

Rpt.: Content of an "Overview" - Paper

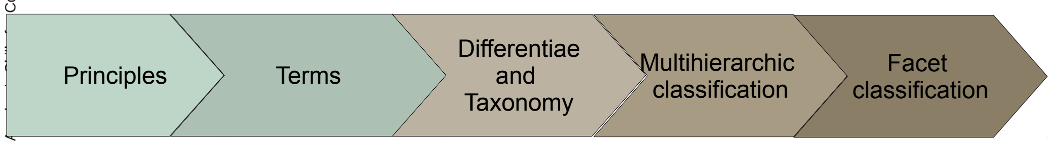
- In a research field, you have read a lot of papers. You produce sections on:
- **Principles** and **basic terminology**
- **Taxonomy or Facet classification** of the field
- **Research landscape** with portfolio diagrams or kiviat diagrams
  - **Research project list** of European, DFG, BMBF projects
  - **Technology list**
  - **Technology hierarchy**
- **Qualitative comparison model** with qualitative comparison criteria
  - one- or multidimensional (Kiviat graphs)
- **Quantitative comparison model** with **scales**
  - **School grading:** simple school grades to evaluate approaches in different dimensions (Kiviat graph)
- **Problem model:** Use a ZOPP, PROBLOSS, or GQM to describe the problems of the field
- **Variability model:** describe the variations points of the technology, as well as the main variants. Develop a feature model.

Other Content

- **Value chain:** which products exist with which components? who has to collaborate? which technologies are important? which suppliers exist? who is the OEM?
- **Research map:** collect the main research questions
- **Research roadmap:** collect a prospective path for the future. What will be in 3, 5, 10 years?
- **Strategy analysis:** do a strategic analysis, e.g., SWOT, Value Proposition Analysis

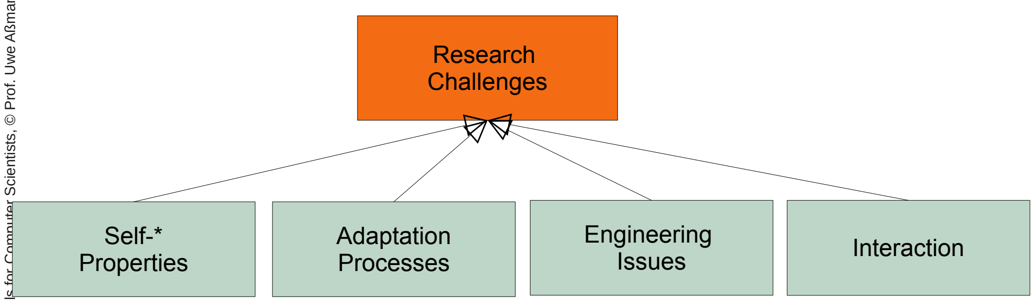
Principles, Terminology, Classification

- Basic concepts (terms) of a field are defined and explained by examples.
- Definitions are made in
  - definitory sentences
  - definitory paragraphs
- From differentiae of terms, a **Taxonomy (hierarchical classification)** of the field can be constructed
- A Multihierarchy (multitaxonomy) uses multiple inheritance and leads to an **acyclic classification**
- If the attributes of a concept do not form differentiae, a Facet classification can be made
  - Facets are independent orthogonal partitions of the concept's attributes



A Simple Taxonomy of Research Challenges

- From [Salehie, Fig. 5]



Facet Classification of Field

- The following taxonomy is really a facet classification [Salehie, Fig. 3]

Research Landscape

- A Research Landscape collects several lists:
  - **Related discipline list** enumerates all research disciplines treating the research problem from different angles
  - **Research project list** of European, DFG, BMBF projects [Salehie Table III]
  - **Technology list** (Register allocation by linear scan, Chaitin graph coloring, attribute evaluation) with examples and citations [Salehie Table i]
  - **Technology hierarchy** [Salehie Fig. 1]
- **Relational Matrix analyses** compare lists or hierarchies with other lists or hierarchies, e.g.,
  - (research project list x facet classification) [Salehie Table VI]
  - (research project list x taxonomy)
  - (technology list x taxonomy)
  - (technology list x research project list) [Salehie Table V]
- **Formal concept analysis** is a specific relational matrix analysis. It compares lists of objects with lists of attributes
  - (research project list x technology list) [Salehie Table IV]

Qualitative Comparison Model with Qualitative Criteria

- **The criteria list (criteria table, attribute list)** collects a simple table to compare technologies, approaches, objects
- Qualitative comparison is usually done then in a boolean matrix, from which an FCA can be started. [Salehie Table IV]

Quantitative Comparison Model using Weighted Scales

- Multi-criteria Attribute Analyses
  - e.g., 2-dimensional attribute analysis of objects (2 criteria), with portfolio diagrams
  - n-dimensional attribute analysis with kiviat diagrams
- **Metrics:**
  - **School grading:** simple school grades are given to a list of objects or approaches, to evaluate approaches in different dimensions
  - This can be displayed by a Kiviat graph
- Other scales can be used

Problem Model of the Field

- Use a problem-objective analysis (ZOPP, B-POPP, BATE-POPP, or GQM) to describe the problems of the field
- ZOPP uses hierarchical problem models
- GQM acyclic problem models

Variability Model of a Technology

- describe the variations points of the technology, as well as the main variants. Develop a **feature model**.
- (see course Software technology II)

Value chain

- which products exist with which components? who has to collaborate? which technologies are important? which suppliers exist? who is the OEM?

Research Challenge Map

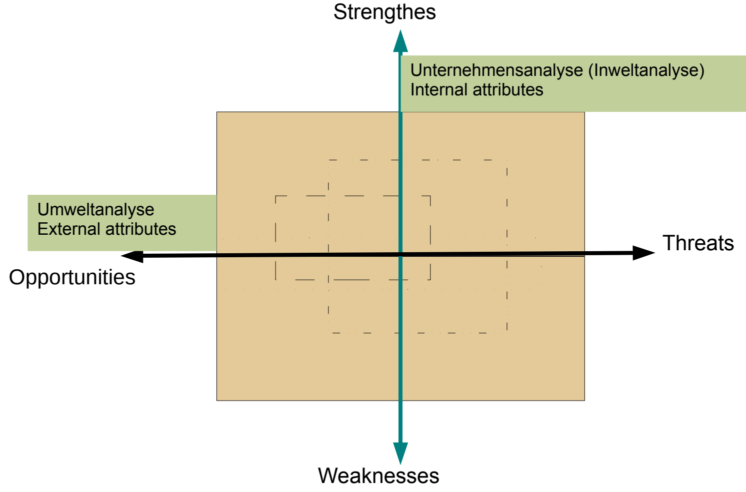
- Collect the main research questions
- Example [Wolf-CPS] (only 2 pages)
  - 2 short introduction paragraphs
  - Theoretical underpinnings
  - Efficiency Boost
  - Contrl theory issues
  - Cyber-physical roadmap
  - Conclusion paragraph

Research Roadmap

- Based on a Research Challenge Map, collect a prospective path for the future. What will be in 3, 5, 10 years?
  - Use the national roadmap's circular scheme
- Research Roadmap with Strategy Analysis
  - Do a strategic analysis for the research field, e.g., SWOT, or a BSC
  - Do a Value Proposition Analysis with the field, e.g., PAIN-GAIN POPP
- How should the research field develop? What should be done? Which risks exist?

SWOT Analysis for Research Relevance

- SWOT is a 4-dimensional attribute analysis for the development of a strategy for of a project [Albert Humphrey]
- For strategic decisions of your thesis and your research



35.2. Writing a Systems Paper

Prof. Dr. Uwe Aßmann
Softwaretechnologie
Fakultät Informatik
Technische Universität Dresden
2014-0.1, 13-11-12
http://st.inf.tu-dresden.de/acse

1) Writing

Obligatory Literature

- Roy Levin and David D. Redell. An Evaluation of the Ninth SOSP Submissions or How (and How Not) to Write a Good Systems Paper. ACM SIGOPS Operating Systems Review, Vol. 17, No. 3 (July, 1983), pages 35-40
- http://infolab.stanford.edu/~widom/paper-writing.html.

Rpt. System and Tool Papers

- System papers need to discuss
  - Deficiencies or limits of other systems
    - Market data or studies of economical need
  - Success factors and requirements for the system
  - Unique features not available in other systems
    - Components of the system that contribute to the unique features
    - why is automation with a tool important?
  - Important use cases
  - Limits of the system
  - Ev. empirical evaluation
- Tools are special systems which automate things that should otherwise be done by hand
  - Aching factors: what aches if the tool is not available?

Specific instance	Automating	System	Experience
What can system S do?	Formalize textual requirements	System components: Requirements editor Requirements checker Requirements parser Formalizer	Look, the tool worked in the following industrial projects

R. Vuduc , J.W. Demmel, K.A. Yelick. OSKI: A Library of Automatically Tuned Sparse Matrix Kernels. SciDAC 2005 (Journal of Physics), UCRL-CONF-213753

Outline:

- 1. Goals and Motivation: Interesting, explicit list of motivations
- 2. An Introduction to the Tuning Interface by Example
  - 2.1. Basic usage: globally migrating applications
  - 2.2. Providing explicit tuning hints
  - 2.3. Tuning based on implicit profiling
- 3. Saving and restoring tuning transformations
- 4. Other features
- 5. Related work
- 6. Conclusions and future work

- Why does this outline work? constructive hypothesis (automation hypothesis):

"Abstract. The Optimized Sparse Kernel Interface (OSKI) is a collection of low-level primitives that provide *automatically tuned* computational kernels on sparse matrices, for use by solver libraries and applications. These kernels include sparse matrix-vector multiply and sparse triangular solve, among others. The primary aim of this interface is to hide the complex decision- making process needed to tune the performance of a kernel implementation for a particular user's sparse matrix and machine, while also exposing the steps and potentially non-trivial costs of tuning at run-time. This paper provides an overview of OSKI, which is based on our research on automatically tuned sparse kernels for modern cache-based superscalar machines."

35.3 New Concepts" Paper (Enhanced Model)

Kiczales, Lamping, et.al. "Open Implementation": Definition Essay

- [OpenImp] Outline:
  - 1. Introduction
  - 2. A Base Case
  - 3. Separation of Use from Implementation Strategy Control
  - 4. Scope Control
    - Choosing the scope control
  - 5. Subject Matter
    - Tradeoffs
  - 6. Style of the ISC code
  - 7. The Design space
- Why does this outline work? problem-solution paper ("enhanced model"):

"Abstract An examination of existing software systems shows that an increasingly important technique for handling this problem is to design the module's interface in such a way that the client can assist or participate in the selection of the module's implementation strategy. We call this approach *open implementation*.

When designing the interface to a module that allows its clients some control over its implementation strategy, it is important to retain, as much as possible, the advantages of traditional closed implementation modules. This paper explores issues in the design of interfaces to open imple-mentation modules. We identify key design choices, and present guidelines for deciding which choices are likely to work best in particular situations.