33. Different Types of Research Hypotheses, Questions, Methods, and Results in Software Engineering

Prof. Dr. Uwe Aßmann Softwaretechnologie
Technische Universität Dresden
2013-0.6, 13-11-7
http://st.inf.tu-dresden.de/asics

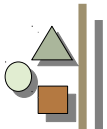
1) Success Criteria for Research
2) Template Abstracts
3) Shaw's classification of Hypothesis and Questions
4) Types of papers
5) The Discussion part
6) Observations, Laws, Theories

[Library of Congress WPA poster]

Obligatory Literature

- [Shaw-Research] Mary Shaw. What makes good research in software engineering? Int. Journal of Software Tools for Technology Transfer (STTT), 4(1):1-7, 2002.
- [Shaw-ETAPS02] Mary Shaw. Slide set of key note at ETAPS 2002. Good summary of [Shaw-Research]
- Mary Shaw's web site http://spoke.compose.cs.cmu.edu/shaweb/
- [Bundy] Alan Bundy. How to Write an Informatics Paper. Web page:
  - http://homepages.inf.ed.ac.uk/bundy/how-tos/writingGuide.html

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

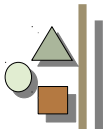


References

- Dieter Rombach. Klaus Endres. A Handbook of Software and Systems Engineering. Addison-Wesley.
- [Xu-Nygard] Dianxiang Xu and Kendall E. Nygard. Threat-driven modeling and verification of secure software using aspect-oriented petri nets. IEEE Trans. Software Eng, 32(4):265-278, 2006.

- Fun:
  - Scientific Balloons
    - http://www.centennialofflight.gov/essay/Dictionary/Scientific_Balloons/DI72.htm

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann



33. Discussion of Last Homework

- Analysis of the essay - "Innovationen sichern den ökonomischen Erfolg". (1996)

Schmidt does it in 4 sections:

- I Problems (with a list)
- II Short-term emergency program for creating innovations (Solutions)
- III Mid-term program (Solutions)
- IV Conclusion (not very sophisticated...)

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

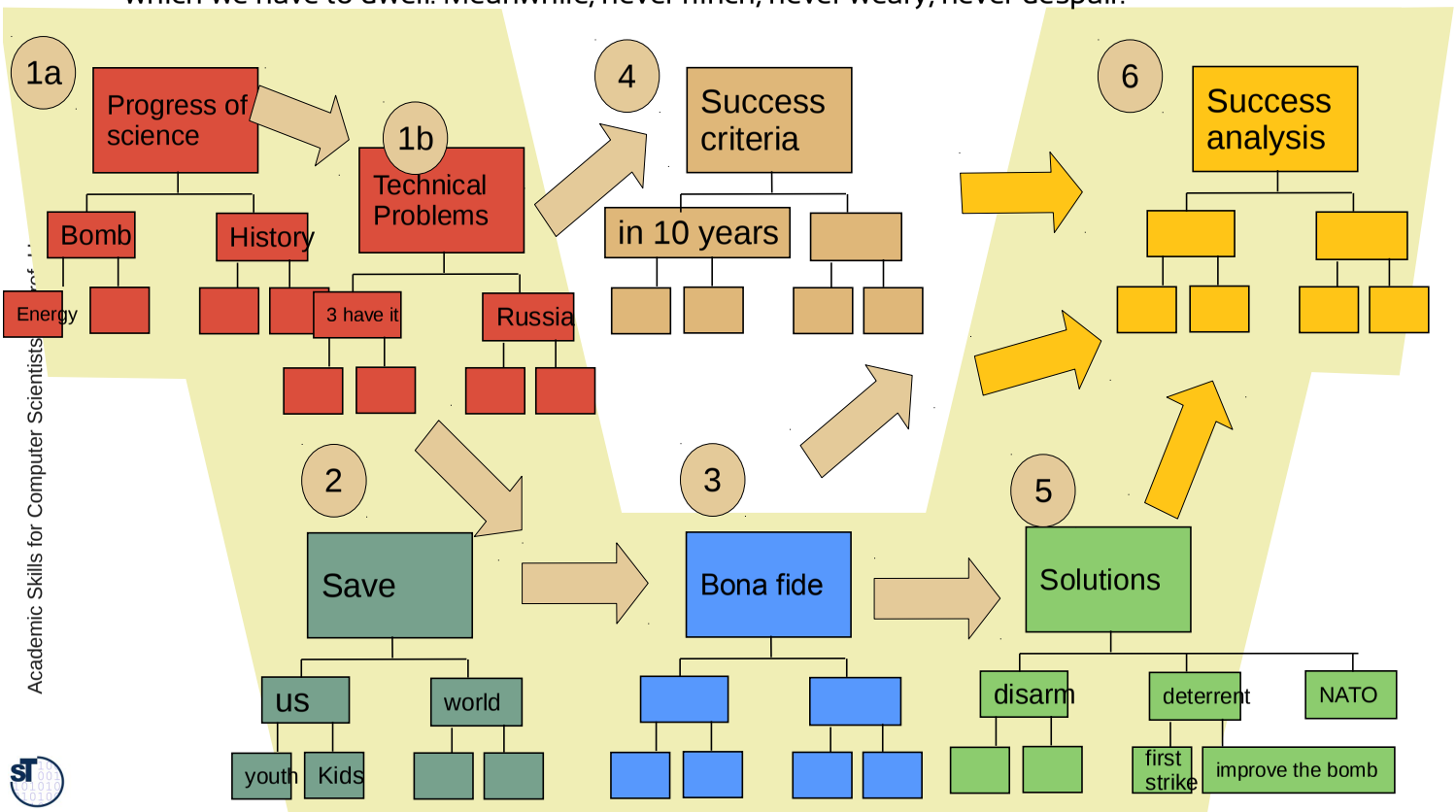
33. Discussion of Last Homework

- BATE-POPP Analysis of Winston Churchill's speech "Never despair".
  - https://www.winstonchurchill.org/learn/speeches/speeches-of-winston-churchill/1946-1963-elder-statesman/102-never-despair

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

BATE-POPP on "Never Despair"

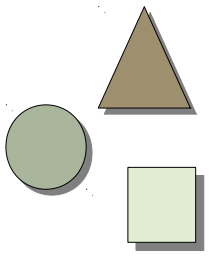
"The day may dawn when fair play, love for one's fellow-men, respect for justice and freedom, will enable tormented generations to march forth serene and triumphant from the hideous epoch in which we have to dwell. Meanwhile, never flinch, never weary, never despair."



Academic Skills for Computer Scientists

33.1 Different Kinds of Research Hypotheses

- [Bundy] The key to successful paper writing is an explicit statement of both a scientific hypothesis and the evidence to support (or refute) it.
- In experimental research, hypotheses typically take one of the following two forms:
  - Technique/system X **automates** task Y for the first time;
  - Technique/system X automates task Y **better**, along some dimension, than each of its rivals;..



Tribute

- The web site of Mary Shaw's research course, its literature link page
  - http://spoke.compose.cs.cmu.edu/ser04/R/bib-meta.htm

Mary Shaw: "A research paper is a purposeful, designed artifact, just like a software system. Apply software design techniques to paper design:

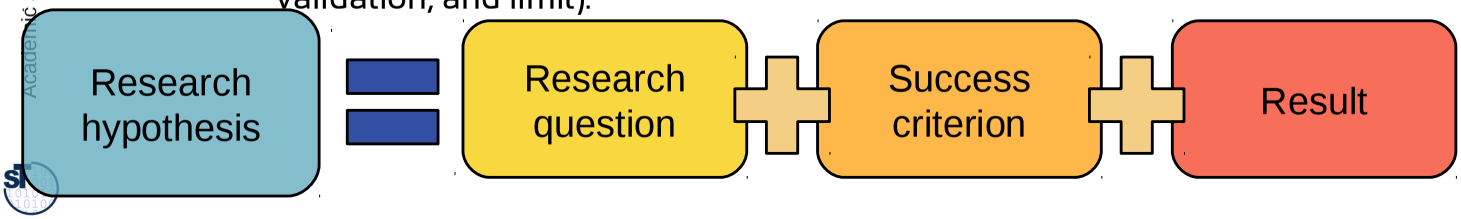
- *Start with the requirement:* read the call for papers
- *Select an architecture:* plan the sections, what they say
- *Plan a schedule:* allow time for review, revision
- *Check consistency:* type-check text like code"

http://spoke.compose.cs.cmu.edu/shaweb/images/mary-shaw.jpg

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Motivation: Displaying Your Research (Make yourself heard)

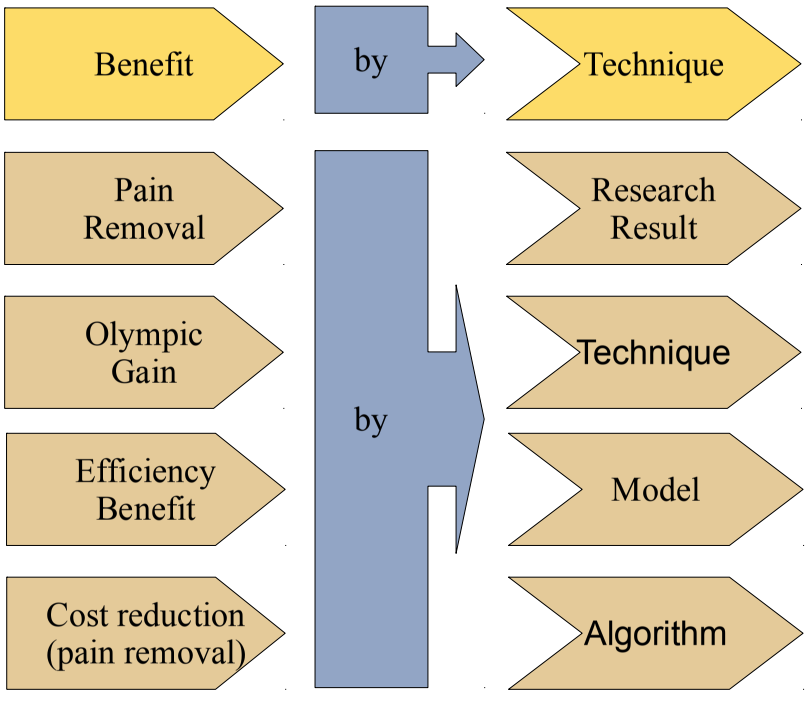
- The relationship of a research question to a research hypothesis is similar to the relationship of a thesis question and a thesis of a text block.
  - Text Question creates interest, thesis answers it. Text Thesis has topic and controlling idea
  - **Research hypothesis** has a *research result* (topic) and a *success criterion, research result*
  - **Controller hypothesis** has additionally a *research method* and *research valuation* (development scheme), and *a limit*
- Important:
  - Your research hypothesis and research question has to be found *crystally clear* in your **introduction** and your **abstract**
    - Every time, you refine the hypothesis, you have to rewrite the introduction and the abstract
  - Your **slides of the defense** also have to display them crystally clear
  - Make **one** slide to present hypothesis (question, success criterion, result, validation, and limit).



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

A Technical Science Hypothesis

- A **Technical Science Hypothesis (Technik-Hypothese)** is an hypothesis about *achieving a benefit (solving a problem, solving a research problem or reaching an objective) with a technique (technical science research result)*
- It forms the basis of a technical science paper, Master or PhD thesis



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Bundy's Dimensions of Enhancement (Optimization)

The dimensions of enhancement are typically [Bundy]:

- **Behaviour**: X has a higher success rate than Y
  - X produces better quality outputs than Y
  - X is shorter, is easier to understand, is easier to write, is more similar to human outputs...:
- **Coverage**: X is applicable to a wider range of examples then Y.
- **Efficiency**: X is giving more utility with less cost
- **Olympic**:
  - **Utility**: X is faster, X is more precise
  - **Cost**: X uses less space or energy then Y
- **Dependability**: X is more reliable, safe or secure than each of its competitors
- **Maintainability**: Developers find X easier to adapt and extend than its alternatives.
- **Useability**: Users find X easier to use than Y.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Important Classes of Research Hypotheses (and Corresponding Success Criteria) in Technical Science

- How can I automate a technique?
- What is an engineering technique for a problem?

Constructive Existential (Automating)

- What is an engineering process for this problem?

- How can we be faster, go farer, higher?

Farer, Higher, Wider (Olympic, absolute)

Optimizing (better)

- How can we be more efficient (better cost-utility function)?

Efficiency (Quality: Utility vs Resources)

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Important Classes of Research Hypotheses (and Corresponding Success Criteria) in Technical Science

- How much better is this method/techique in (industrial or daily) practice? (according to usability criteria)

Empirically better

- Which classes of users, companies can benefit?

- Does feature F hold?

Existential

- Can we predict P?

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Important Classes of Research Hypotheses (and Corresponding Success Criteria) in Technical Science

- Where does method/techique M fail?
- Under which conditions does it not work?
- Assumptions for result

Limit

- Where is a gap?
- Where is an open research question?

- I discovered the following problems with a well-established scientific method.

Limit removal
(can apply to all others)

- I show how to remove them.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Different Kinds of Research Hypotheses in a Technical Science (Summary)

The *benefit* of a technical science hypothesis may be reached in different ways.

- **Existential hypothesis**: something exists.
- **Automation hypothesis:** something can be automated the first time [Bundy]
  - Then, you have to show that
  - It is assumed that automation helps
- **Optimizing hypothesis (Enhancement hypothesis):** something can be automated in a better way than with other methods [Bundy]
  - **Olympic hypothesis:** something can be done faster, wider, higher
  - **Efficiency hypothesis:** something can be done faster, wider, higher with less cost and resource consumption (cost – utility function or relation, enonomic hypothesis)
  - **Comparison hypothesis:** something A is better than something B. Comparison can be olympic or economic (efficiency-based)
- **Limit hypothesis:** some other result has its limits
- **Limit removal hypothesis:** my research removes the limits of another method

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

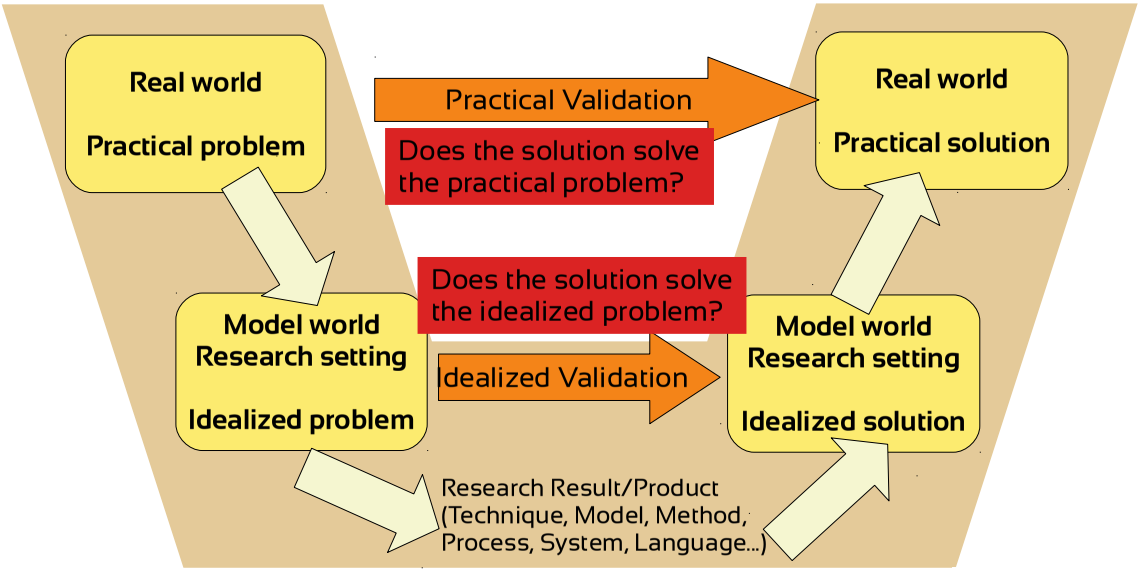
Important Classes of Research Questions in Engineering

Type of question/criterion	Examples
Existence?	Does X exist? Does X hold under Y? What is property X of artifact/method Y?
Documenting	What is the current state of X / practice of Y?
Automatable?	What is an automatic way to do/create X? How can we do/create (or automate doing) X?
Olympic? (Quantitative)	How can run X faster? How does X use less memory? How does X spend less energy? How can deliver X more utility? What is a clearer, simpler, more structured design or implementation for application X?
Efficient? (Quality, economics)	How do cost and utility of X relate? How can I increase utility while freezing cost? (better utility) How can I achieve utility while sinking cost?
Comparison	How does X compare to Y?
Limiting	Where does result X not hold? Where is solution X deficient? What are the assumptions under which X holds? What is an open question?

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Remember: Practical Research vs. Idealized Research

- [Shaw-ETAPS02] Many research papers and solutions require a *model of reality* in which their result is valid. A **model of reality** is an idealized abstraction of reality
- An **idealized research problem** is a research problem in a model of reality, a **complete (practical) research** result solves a practical research problem
- Structural science (mathematics, theoretical computer science, computer science) works in idealized model worlds
- Technical science (engineering science), also Software Engineering, works for **practical problems** and must research **practical solutions**
- **Technical scientists and Engineers** have to produce **practical solutions**

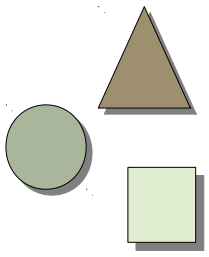


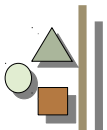
Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

In theory, there is no difference between theory and practice.
But, in practice, there is.
Jan L.A. van de Snepsheut (1953-1994)

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

33.2 Paper Patterns: Newman's Template Abstracts





Patterns of Research Papers

- A **template abstract (pro forma abstract, abstract pattern, paper pattern)** is a semantic development scheme for an abstract containing several template sentences. [Newman]
- A template abstract specializes the "technical science hypothesis" in a specific way.
- Newman explored them with 5 schemes for the field of Human-Computer Interaction (HCI), but they can be generalized to all disciplines.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

The Template Abstracts of Newman for Classes of Research Papers

- Several template abstracts of Newman suggest olympic or efficiency success criteria.

Enhanced Model (EM) (Generalized model, ZOPP-like):

Problem: Existing <model-type> models are deficient in dealing with <properties> of <solution strategy>.

Result and Solution: An enhanced <model-type> is described, capable of providing more accurate analyses / predictions of <properties> in <solution strategy> designs.

Validation: The model has been tested by comparing analyses / predictions with empirically measured values of <properties>.

Enhanced Solution (ES): (Better: more olympic or efficient, ZOPP-like)

Problem: Studies of existing <artefact-type> have shown deficiencies in <property>.

*Solution:*An enhanced design for an <artefact-type> is described, based on <solution strategy>.

*Result:*In comparison with existing solutions, it offers enhanced levels of <property>, according to analyses based on <model-type>.

Validation: These improvements have been confirmed / demonstrated in tests of a working <artefact-type> based on the design.

Enhanced Tool (ET): (Better: more olympic or efficient, MOPARC-like)

Motivation: The effectiveness of <model-type> / <solution strategy> in supporting the design of <artefact-type> has been demonstrated.

Result: An enhanced tool / method is described for the design of <artefact-type> based on <model-type> / <solution strategy>.

Validation: Examples are provided confirming the effectiveness of its support for <model- type> / <solution strategy> in design.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

The Template Abstracts of Newman

Radical Solution (RS, MOP):

Solution: A radical solution to the problem of <problem definition> is described, based on <solution strategy>.

Motivation: In comparison with <existing normal solutions> it offers <advantages>, which have been demonstrated in preliminary tests, but it leaves a number of side-effects to be addressed including <list of side-effects>.

Optimization: Strategies are suggested for addressing these side-effects.

Experience and/or Heuristic (XH)

Background: Studies reported here of <application> supported by <supporting technology> generate a number of findings concerning <issues>, including <list-of-findings>.

Limit/Deficiency/Problem: They indicate that <requirement> is / is not met by <design-heuristic>.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

POPP Template Abstracts Similar to Enhanced Solution

- Instead of EnhancedSolution, we can use B-POPP

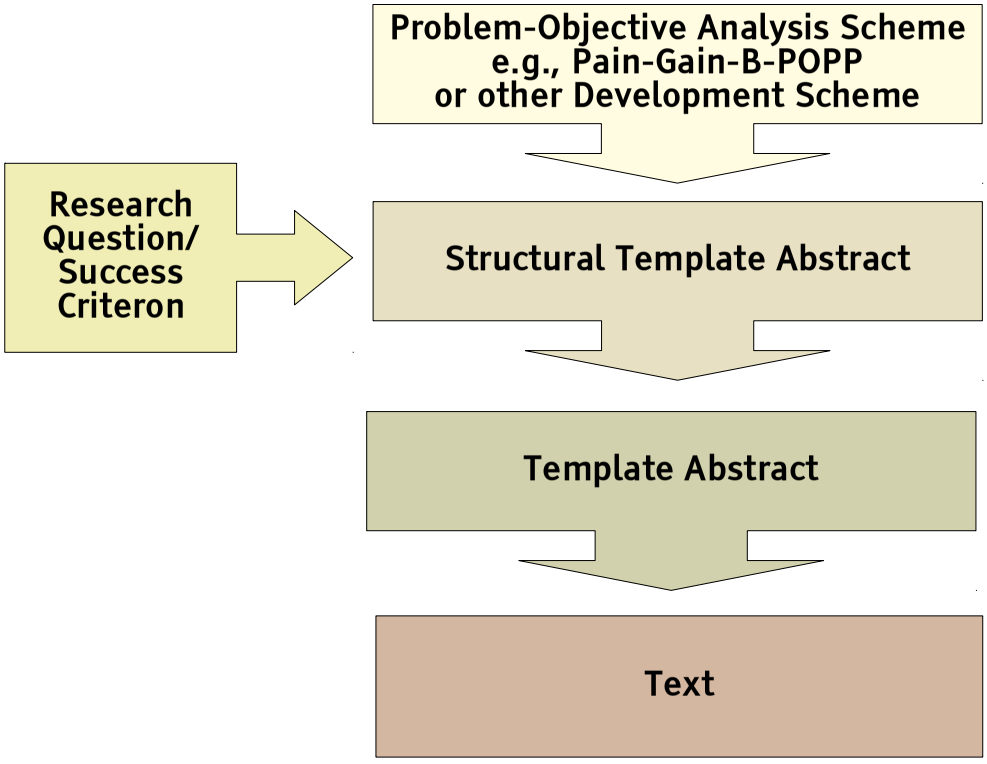
Enhanced Solution (ES): (Better: more olympic or efficient, ZOPP-like)
Problem: Studies of existing <artefact-type> have shown deficiencies in <property>.
*Solution:*An enhanced design for an <artefact-type> is described, based on <solution strategy>.
*Result:*In comparison with existing solutions, it offers enhanced levels of <property>, according to analyses based on <model-type>.
Validation: These improvements have been confirmed / demonstrated in tests of a working <artefact-type> based on the design.

B-POPP:
Problem: Studies of existing <artefact-type> have shown deficiencies in <property>.
Goal: <stakeholder group> needs the following <olympic improvments | efficiency improvements | automation>
Blocking factor: So far, the <limit> of <artefact-type> could not be removed.
Success criterion: If <stakeholder-group> can get 20 % of improvement, it will be satisfied.
*Solution:*An enhanced design for an <artefact-type> is described, based on <solution strategy>.
Result: In comparison with existing solutions, it offers enhanced levels of <property>, according to analyses based on <model-type>.
Validation: These improvements have been confirmed / demonstrated in tests of a working <artefact-type> based on the design.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Combination of POA and Template Abstracts

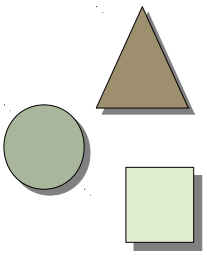
- "Enhanced Solution" Template Abstracts can be produced by any problem-goal-analysis scheme, or any development scheme, if combined with a olympic or efficiency research question and success criterion.



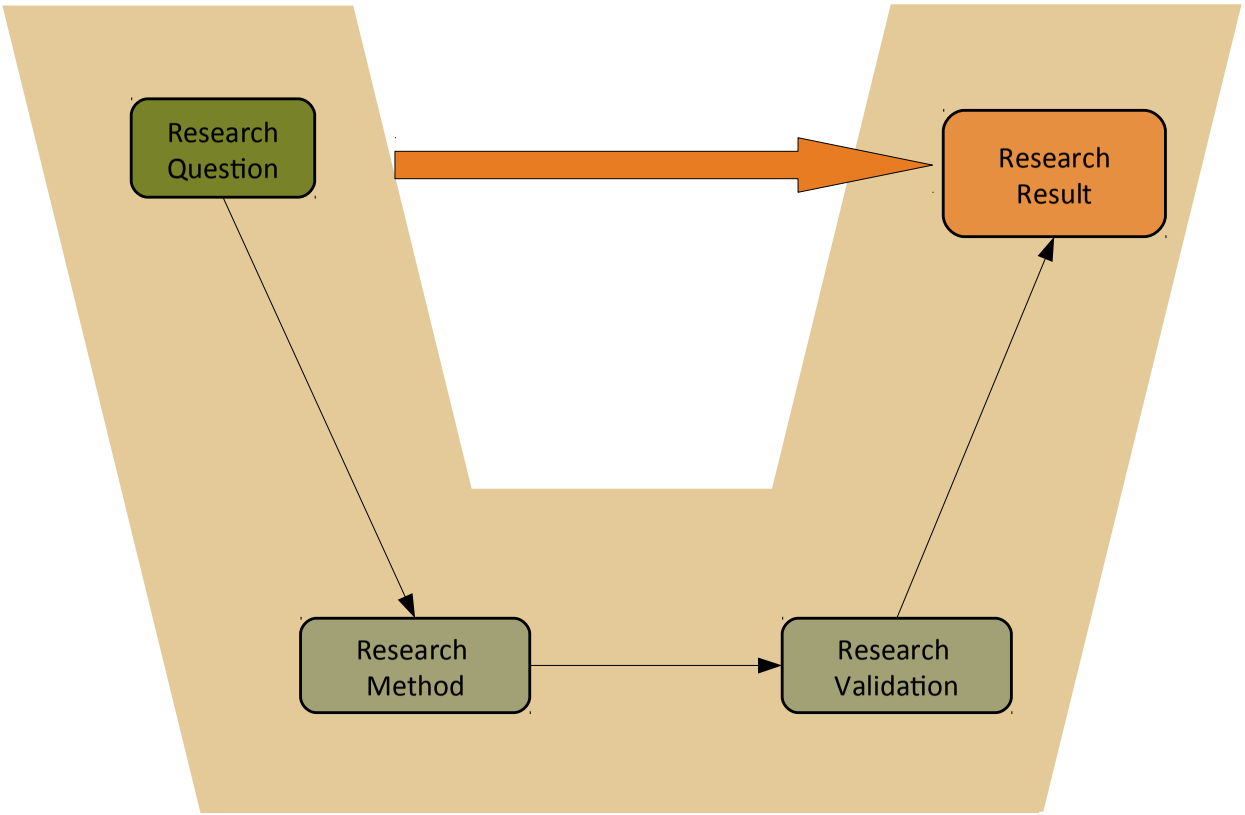
Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

33.3 Shaw's Classification of Research Hypotheses

.. and how to make more template abstracts out of the classes

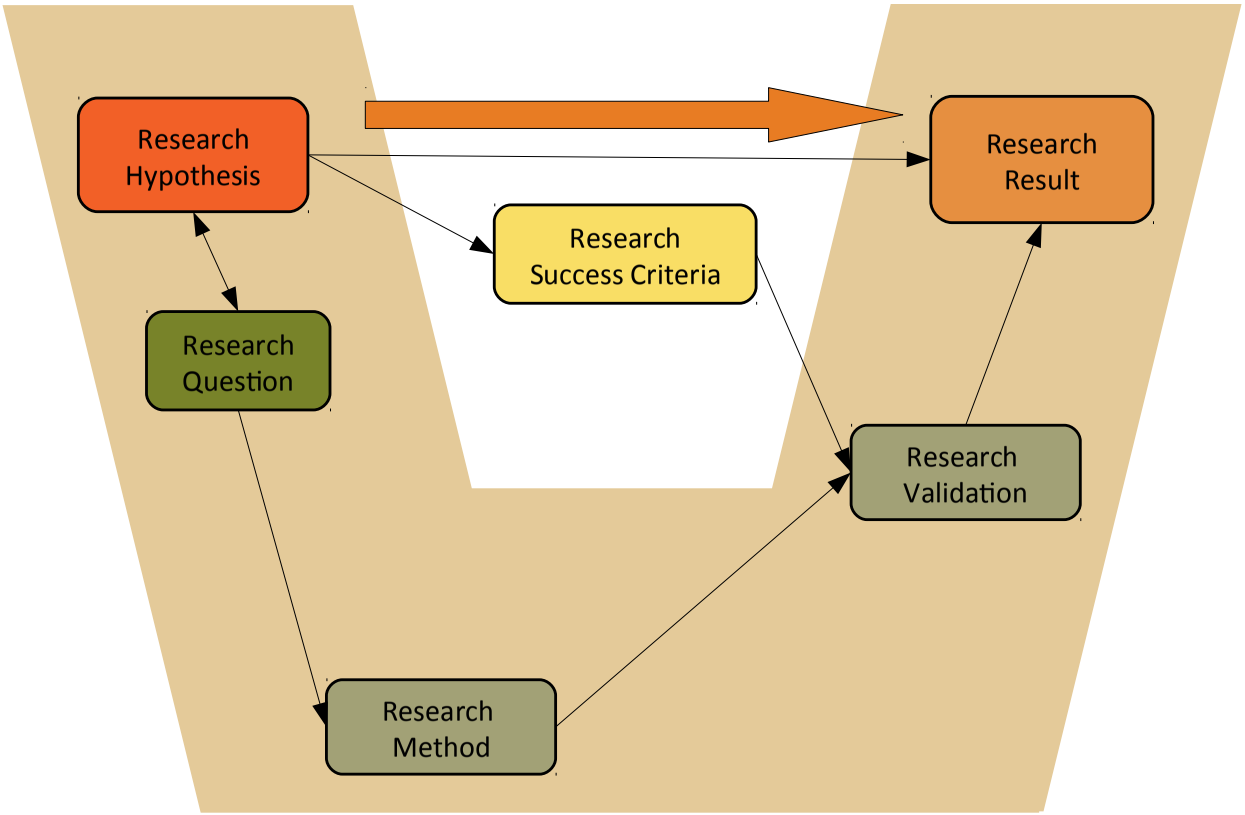


The Shaw Model of Research in Software Engineering



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

The Extended Shaw Model for Research Hypothesis



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Shaw's Original Facet Classification

Research Question	Research Result	Research Validation
Method	Procedure / technique	Analysis
Development Method/ means of design	Model	Evaluation
Design pattern	Qualitative or descriptive model	Experience
Method for analysis	Analytic model (quantitative, continuous)	Example
Method for comparison	Empirical model	Persuasion
Design, evaluation, analysis of a particular instance	Tool / System / Notation (language)	
Generalization or characterization	Specific solution	
Feasibility	(Experience) Report	

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Research Questions

Type of Question		Examples of Research Questions
New Development Method or means of development		How can we do/create (or automate doing) X? Is there a best practice how to do X? A design pattern?
	Optimized Development Method	What is a better way to do/create X?
Method for analysis		How can I evaluate the quality/efficiency/correctness of X? How do I choose between X and Y?
	Method for comparison	How do I systematically compare between X and Y? What are the criteria for comparison and contrast?
Design, evaluation, or analysis of a particular instance		What is a (better) design or implementation for application X? What is property X of artifact/method Y? How does X compare to Y? What is the current state of X / practice of Y?
Generalization or characterization		Given X, what will Y (necessarily) be? What, exactly, do we mean by X? What are the important characteristics of X? What is a good formal/empirical model for X? What are the varieties of X, how are they related?
	Advantages of classifications	Investigate the special features of all classes of a classification. Find criteria to test membership in these classes and then apply the special features. Example: AG hierarchy, XGRS classes
Feasibility		Does X even exist, and if so what is it like? Is it possible to accomplish X at all?

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Research Results

Types of Research Results		Example of Research Result
Procedure / Technique / Process		New/better ways to do development/analysis tasks
Model	Qualitative or descriptive model	Structure/taxonomy/ontology for problem area; framework Informal guidance, informal domain analysis
	Analytic model	Structural model that permits formal analysis, automation
	Empirical model	Empirical predictive models based on real data
Tool / System		Tool that embodies model or technique
Notation (language)		New language with better X. Ex.: Gradual typing;
Specific solution		Solution to application problem applying SE principles, or result of specific analysis
(Experience) Report		Interesting observations, rules of thumb, heuristics best practices, case studies, industrial case studies
Theorem		New theorem in an existing model. Ex: Register allocation with graph cliques is polynomial (complexity), equivalence

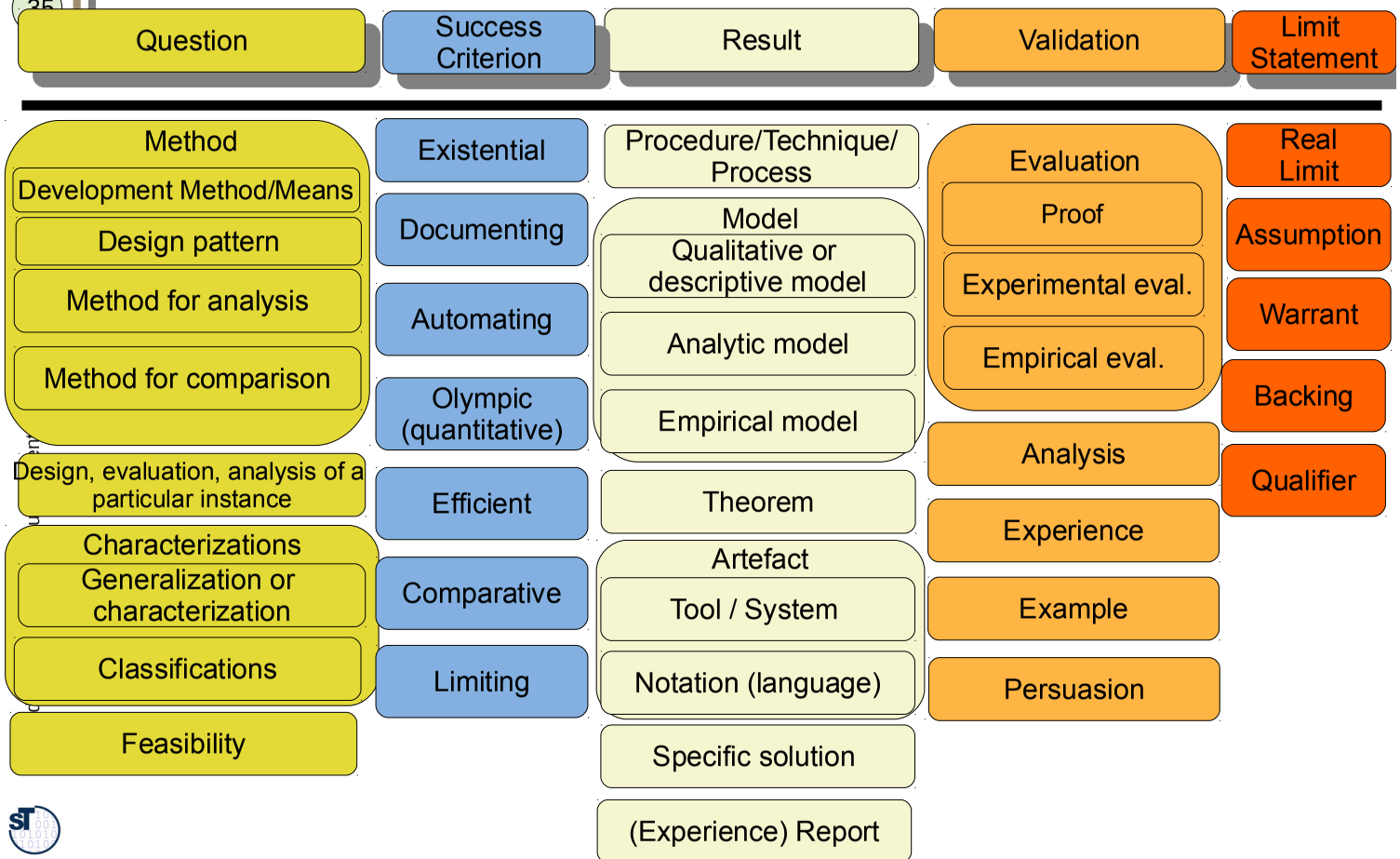
Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Research Validation (Evaluation)

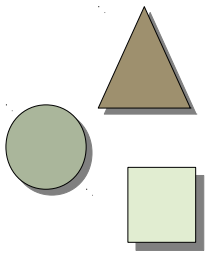
Type of validation		Examples of Phrases
Analysis		I have analyzed my result and find it satisfactory through … • for a empirical model: ..data on controlled use • for a controlled experiment: ...a carefully designed statistical experiment
	Experience	My result has been used on real examples by someone other than me, and the evidence of its correctness / usefulness / effectiveness is … • for a qualitative model: ….narrative • for a empirical model, tool: … some data, usually statistical, on practice • for a notation, technique: … a comparison of this with similar results in actual use
	Example	Here's an example of how it works on... • for a toy example: perhaps motivated by reality • for a slice of life: a system that I have been developing
	Evaluation	Given the stated criteria, my result... • for a descriptive model: .. adequately describes the phenomena of interest • for a qualitative model: ...accounts for the phenomena of interest... • for an empirical model: ...is able to predict ... because ..., or ... gives results that fit real data … Includes feasibility studies, pilot projects
Persuasion		I thought hard about this, and I believe that... • for a technique: ..if you do it the following way... • for a system: ... a system constructed like this would... • for a model: … this model seems reasonable... • for feasibility: … my working system is persuasive, even without analysis
Blatant assertion		No serious attempt to evaluate result

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

The Shaw Facet Classification, Slightly Extended with Success Criterion and Limit Statement



33.4 Types of Papers based on the Shaw Facets



Generalization Paper, Based on Experience

Generalization	Olympic (quantitative)	Report	Experience
What do we mean by X? How to generalize X? How can X also treat Y?	more abstract more general	Report on generalized X covering more use cases	Experience Report about Use; showing more use cases

Academic Skills for Computer Scientists, © Prof.

"Solution Pattern" Paper: Special form of Generalization Paper

- See course "Design Patterns and Frameworks"
- How can I solve a standard problem in a specific context with a standard solution?

Generalization	Olympic (qualtitative)	Qualitative or Descriptive Model	Example
How to provide a standard solution for a standard problem?	more flexible more general more extensible more variable better evolvable less costly	Pattern description in an abstracted form, so that it can be instantiated to other scenarios	Pattern description must have several examples where the pattern already occurs in systems, processes, methods, literature

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Design Pattern Papers

- Design papers need to discuss well-known *design* solutions for well-known problems
  - The criteria of a pattern catalogue (e.g., Gamma)
  - The forces under which they apply
  - Solution patterns
- The research hypothesis is "**documenting**" because a design pattern should not be new, but well-experienced
  - There must be **several examples,** because the pattern must be well-experienced

Design pattern	**Documenting**	Descriptive model of architectural scenario	**Several** Examples
Which micro-architecture should be chosen under a set of design forces?	extensible systems	Descriptive model of object scenarios	Look, the structure has the following advantages

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Enhanced/Improved Method (Optimization Hypothesis)

- Special subclass of "Enhanced Solution"

Optimized Development Method	Olympic	Procedure / technique / Method	Experience
Can Method do better?	more abstract more general	Improved Method	Experience Report about Use showing olympic improvement
Optimized Development Method	Efficiency	Procedure / technique / Method	Experience
Can Method do better, yield more and cost less?	more utility less cost	Improved Method	Experience Report about Use showing more efficiency

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Optimization Technology Paper

- Present an optimization technology (more than an optimized algorithm)
- Show why the current technology is too slow or inefficient
- Show metamodels of optimizing technology
- Give a systems' component diagram
- Give some central algorithms
  - Prove termination
  - Analyze complexity
  - Prove quality features
- Show a case study which proves that your stuff is more efficient

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

New Method *(with Automation Hypothesis)*, Validated with Examples

Can Method do sth? Can Method tell you X? Can Y solve X?	Automating	New Method	Carefully chosen examples
Can register allocation be done with game theory?	X realized by technique Y	2-player games to allocate registers	Register allocation for benchmark suite on simulator

- A **combination result** shows that a so far uncorrelated method from another branch in science can solve problem X
  - Ex.: Graph rewrite systems can describe program optimizations
  - How to use Datalog to solve traffic problems

Academic Skills for Computer Scientists, © Pro

Empirical Validation by Statistics

Method for analysis	Existential	Analytic model	Analysis
Can we predict cost?	Prediction	Cost Estimation model	Statistical comparison

- Empirical validation is possible by
  - statistics
  - controlled experiments with user groups
  - field studies
- Example: [Xu-Nygard] reduces attack trees to aspect-oriented PetriNets and verifies absence of intrusions: first time automating intrusion checking

Academic Skills for Computer Scientists, © Prof. Uwe A

Not Easy to Publish: Persuasion for Optimized Method

- Hard to Publish:

Optimized Method?	Olympic or Efficency	New Method	Persuasion
How can we do X better?	Prediction	Better X	Look, it works...

- Idea paper, is more interesting and sometimes published:

Feasibility	Automating	New Method	Example
Can X be automated?	Prediction	Realization X	It works in these cases under these frame conditions

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Change Assumptions Paper ("..dennoch..")

- Weak change assumptions paper:

Specific instance	*	New Model	Persuasion
Under new assumptions or frame conditions, how can we do X?	Automating	Now working X	Look, it works...

- Stronger change assumptions paper:

Specific instance	*	New Model	Experience
Under new assumptions or frame conditions, how can we do X?	Automating	Now working X	Look, it worked in the following industrial projects

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

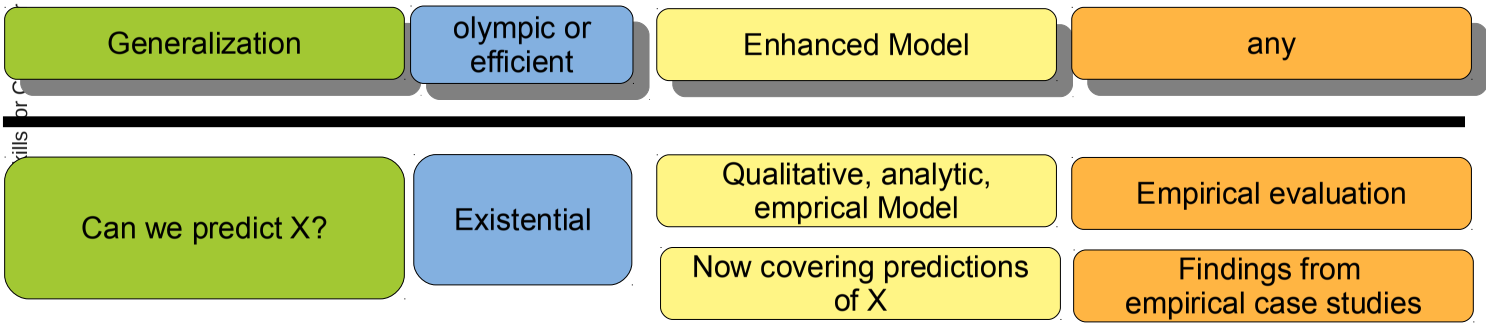
Enhanced Model (EM) (Generalized model)

- All Newman template abstracts can be entered into the Shaw classification.

Problem: Existing <model-type> models are deficient in dealing with <properties> of <solution strategy>.

Result and Solution: An enhanced <model-type> is described, capable of providing more accurate analyses / predictions of <properties> in <solution strategy> designs.

Validation: The model has been tested by comparing analyses / predictions with empirically measured values of <properties>.



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Model Presentation Paper

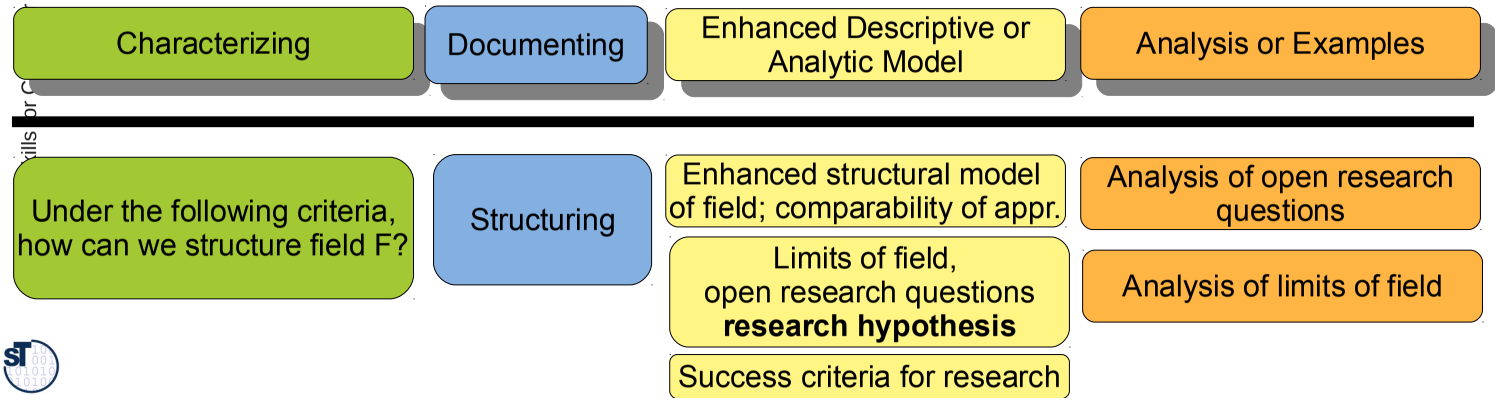
- [Atkinson/Kühne 2003, A Foundation for Metamodeling] presents a 2-dimensional metamodeling scheme for metamodeling.
  - Classification in 2 dimensions, different instance-of-relationships

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

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")



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

"Overview" - Paper

- In research field, you have read a lot of papers. You produce a:
- **Facet classification** of the field
- **Research landscape** with portfolio diagrams or Kiviat diagrams
- **Qualitative comparison model** with qualitative comparison criteria
- **Quantitative comparison model** with
  - **School grading:** simple school grades to evaluate approaches in different dimensions (Kiviat graph)
  - **Metrics:** Use a GQM to evaluate quantitatively
- **Problem model:** Use a ZOPP, B-POPP, 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.
- **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? Use de Bono's strategy scheme
- **Strategy map:** do a strategic analysis, e.g., SWOT

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Classification Paper

- Facet classification of Requirements
  - Martin Glinz: Rethinking the Notion of Non-Functional Requirements
- Classifikation of a domain (domain model presentation)
  - Mens, Czarnecki, Van Gorp. A Taxonomy of Model Transformations.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Algorithm Paper

- Papers presenting a new algorithm need to discuss:
  - Correctness
  - Termination
  - Complexity
    - NP-completeness, decidability
    - for practical algorithms: linearity, n log n, quadratic, cubic

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Specific instance	Existential	Theorems	Proof
Is algorithm A correct? Does algorithm A terminate? What is its complexity? Are there optimizations?	Automating	This sorting algorithm is O(n log n)	The proof is done by induction over the size of the set

Algorithm Analysis Paper

- Present a new algorithm, or an optimized algorithm
- Prove termination
- Analyze complexity
  - on a RAM or PRAM
  - on a logp-machine
- Prove quality features
  - memory consumption
  - energy consumption

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

System and Tool Papers

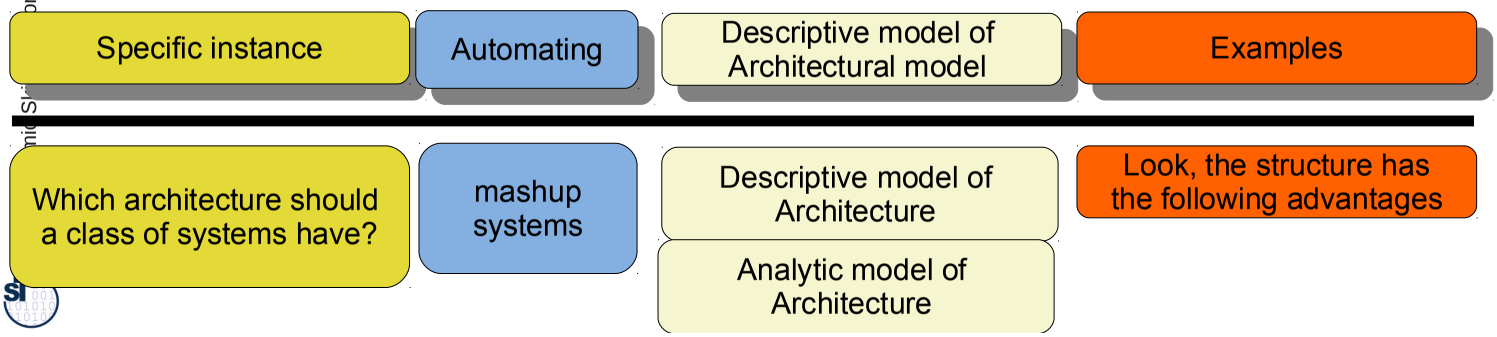
- System papers need to discuss
  - Deficiencies or limits of other systems
    - Market data or studies of economical need
  - Success factors and requirements for the system
  - Unique features not available in other systems
    - Components of the system that contribute to the unique features
    - 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?

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

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

Architecture Papers

- Architecture 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?



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

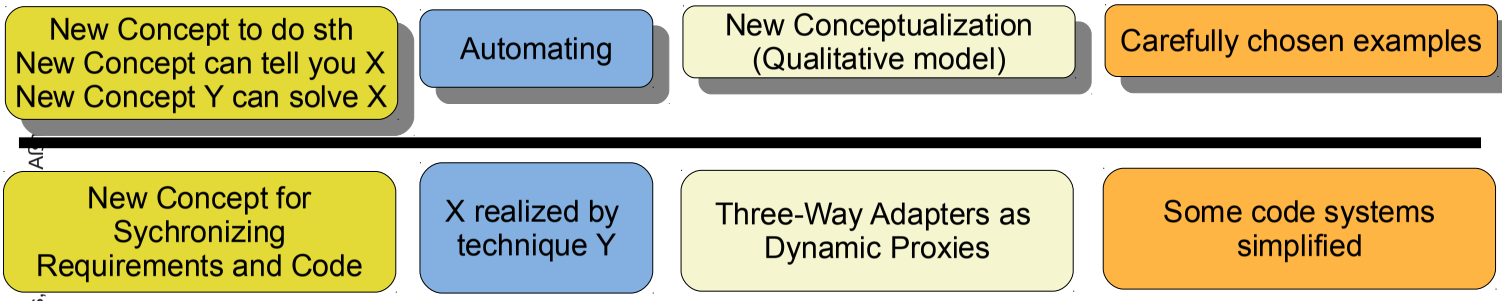
Theorem Paper

- A theorem paper is always working on an idealized research result, based on a model of reality
- LogP Papers of Löwe, Zimmermann, Eisenbiegler discuss the LogP-model of distributing data and computations on distributed machines
  - Much better than the usual PRAM model, because parallel distrubted machine is modeled more realistically
  - L – latency, o - overhead, g - gap
- Wolf Zimmermann and Welf Löwe. Foundations for the integration of scheduling techniques into compilers for parallel languages. IJCSE, 1(2/3/4):99-109, 2005.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Groundbreaking Idea Paper

- In recent years, these are harder to publish
- Contains basically a *conceptualization* of an unknown field (white space)



- Ex.: Uwe Aßmann. Automatic Roundtrip Engineering. In U. Aßmann, E. Pulvermüller, P. Cointe, N. Bouraquadi, and I. Cointe, editors, Proceedings of Software Composition (SC) - Workshop at ETAPS 2003, volume 82 of Electronic Notes in Theoretical Computer Science (ENTCS), Warshaw, April 2003. Elsevier.
- Defines different classes of round-trip systems, such as "bidirectional weaving systems", "partitionable round-trip systems", etc.
- Validation by examples (weak): explains the difference of TeX and Word
- Nevertheless, 30 citations

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Critique Paper (Limitation Paper)

- A **critique paper** contains an analysis
  - why another approach is **deficient**,
  - Bug in proof found
  - why it has its **limits**,
  - limits were not mentioned
  - limits were found
  - why a paper used **unrealistic assumptions**
  - why an idealized research result does not work in practice
  - Invalid assumptions (invalid warrant)
  - why a paper should have used a qualifier, but didn't
- E. W. Dijkstra. Goto statement considered harmful. Communications of the ACM, 11:147-, 1968. Final judgement on unstructured programming in C and C++.
- Per Brinch Hansen. Java's Insecure Parallelism. ACM SIGPLAN Notices, 34 (4):8, April 1999. Brinch Hansen's condemnation of Java, based on his background on monitors:
  - Per Brinch Hansen. Monitors and Concurrent Pascal: a personal history. ACM SIGPLAN Notices, 28(3):1-35, March 1993.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Revision Papers

- A **revision paper** extends a critique paper with a revision proposal
- Friedrich Steimann. A radical revision of UML's role concept. In Andy Evans, Stuart Kent, and Bran Selic, editors, UML 2000 - The Unified Modeling Language. Advancing the Standard. Third International Conference, York, UK, October 2000, Proceedings, volume 1939 of LNCS, pages 194-209. Springer, 2000.
- Friedrich Steimann and Thomas Kühne. A radical reduction of UML's core semantics. Lecture Notes in Computer Science, 2460:34-, 2002.

ntists, © Prof. Uwe Aßmann

Limit of a language concept	Limit removal	New Metamodel	Carefully chosen examples
How can the limits of the Association concept in UML be removed?	Revision of design/ technique/ algorithm	New Model of Roles and their semantics in UML	Some systems simplified

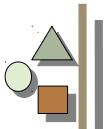
Aca

"Technical Problems" - Paper

- In a well-known approach, you have identified a **technical problem**
  - a deficiency
  - a limit
  - a prerequisite or precondition
- In your paper, you cure the technical problem, remove the limit, generalize the preconditions:
- **Limit discussion:** discuss the limits of the well-known technology.
  - D. W. Wall. Limits of instruction-level parallelism. In Conference on Architectural Support of Operating Systems IV, pages 176-188. ACM, 1991.
  - Wall's paper showed that on instruction level, many programs have only up to 6 threads, which limits parallelism

nputer Scientists, © Prof. Uwe Aßmann

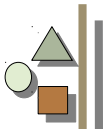
Limit of parallelism	Limit	Numerical threshold on parallelism	Significant Benchmark study
How large is the average possible number of threads?	Revision of design/ technique/ algorithm	Limit: number of threads<=6	Measurements of possible amout of parallelism



Experiment Papers

- Experimental papers measure with benchmarks olympic or efficiency features of programs, processes, techniques
- Benchmark suites, such as:
- Java Grande Benchmark
- Spec benchmark
- Java Qualitas Corpus
  - Ewan D. Tempero, Craig Anslow, Jens Dietrich, Ted Han, Jing Li, Markus Lumpe, Hayden Melton, and James Noble. The Qualitas Corpus: A curated collection of java code for empirical studies. In Jun Han and Tran Dan Thu, editors, APSEC, pages 336-345. IEEE Computer Society, 2010.
  - Roberto Tonelli, Giulio Concas, Michele Marchesi, and Alessandro Murgia. An analysis of SNA metrics on the Java Qualitas Corpus. In Arun Bahulkar, K. Kesavasamy, T. V. Prabhakar, and Gautam Shroff, editors, ISEC, pages 205-213. ACM, 2011.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann



Problem-Objective Analysis Papers

- Use ZOPP, B-POPP, AOPA to analyze the problems and goals of
  - a stakeholder
  - a domain
  - a method
- Define success factors for possible future solutions
- Indicate how solutions could look like

- SWOT Strategic Analysis Paper
  - For research areas or technologies, strategic analytic papers along the SWOT analysis are possible.

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

Tutorial Paper

- A good tutorial paper contains:
  - A set of running examples
  - Bottom-up explanation of concepts and ideas
  - Precise definitions of concepts
  - Classifications of concepts
  - Illustrative figures
  - Some theorems (idealistic research)
  - or case studies (practical research)
- In the SEW course, we use
  - Markus Müller-Olm, David Schmidt, Bernhard Steffen. Model-Checking. A Tutorial Introduction. Springer LNCS, Volume 1694, 1999, p 848ff
    - http://www.springerlink.com/content/l437dulbgk67jl6m/
  - [BW04] Timed Automata: Semantics, Algorithms and Tools, Johan Bengtsson and Wang Yi. In Lecture Notes on Concurrency and Petri Nets. W. Reisig and G. Rozenberg (eds.), LNCS 3098, Springer-Verlag, 2004
    - http://www.it.uu.se/research/group/darts/papers/texts/by-lncs04.ps
  - [BDL04] A Tutorial on Uppaal, Gerd Behrmann, Alexandre David, and Kim G. Larsen. In proceedings of the 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM-RT'04). LNCS 3185.
    - http://www.cs.auc.dk/~adavid/publications/21-tutorial.pdf

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

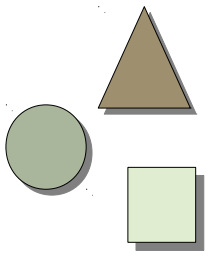
Statistics on Types of Papers

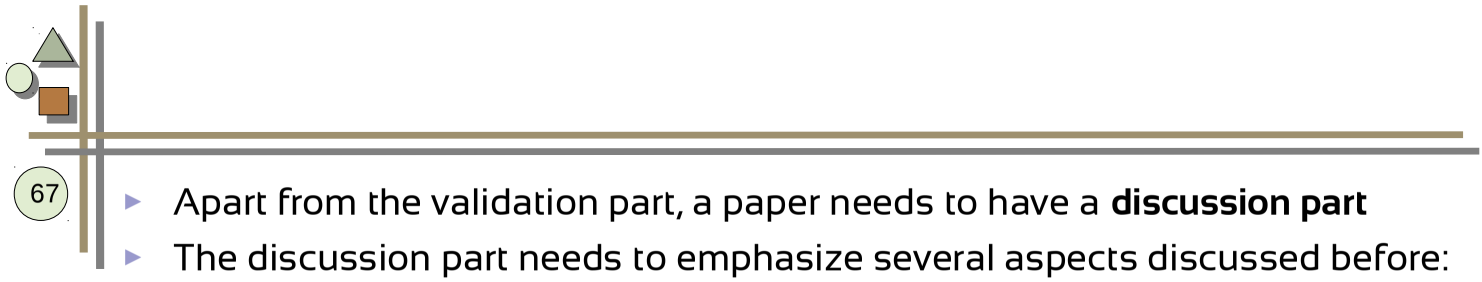
- Shaw's findings on papers submitted to ICSE 2002

Question	Result	Validation	Count
Development method	Procedure	Analysis	3
		Experience	4
		Example	7
	Qualitative model	Experience	2
		Persuasion	1
	Analytic model	Experience	3
	Notation/tool	Analysis	1
		Experience	1
		Example	2
Analysis method	Procedure	Analysis	1
		Experience	3
		Example	2
	Analytic model	Analysis	1
		Experience	1
		Example	2
	Tool	Example	1
Evaluation of instance	Specific analysis	Analysis	3
		Example	1
	Answer	Analysis	1

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

33.5 The "Discussion" Part

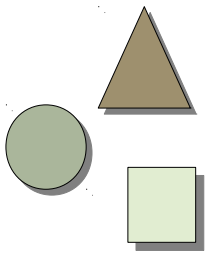




- Apart from the validation part, a paper needs to have a **discussion part**
- The discussion part needs to emphasize several aspects discussed before:
  - **Limits** (real limits, scope and assumptions)
  - **Unique selling points** other research results do not have
  - **Key performance indicators (metrics)** and how they could be improved
  - At least **advantages** and **disadvantages**

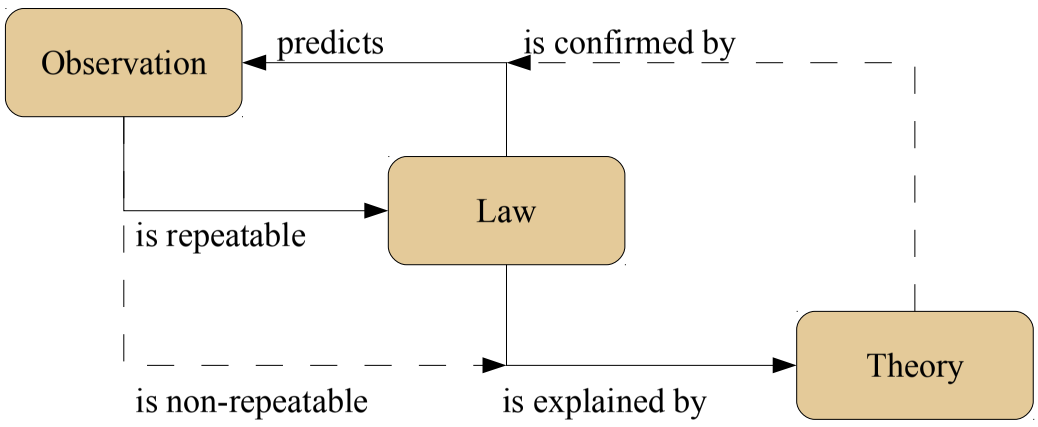
Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

33.6. Different Kinds of Research Results: Observations, Laws, Theories



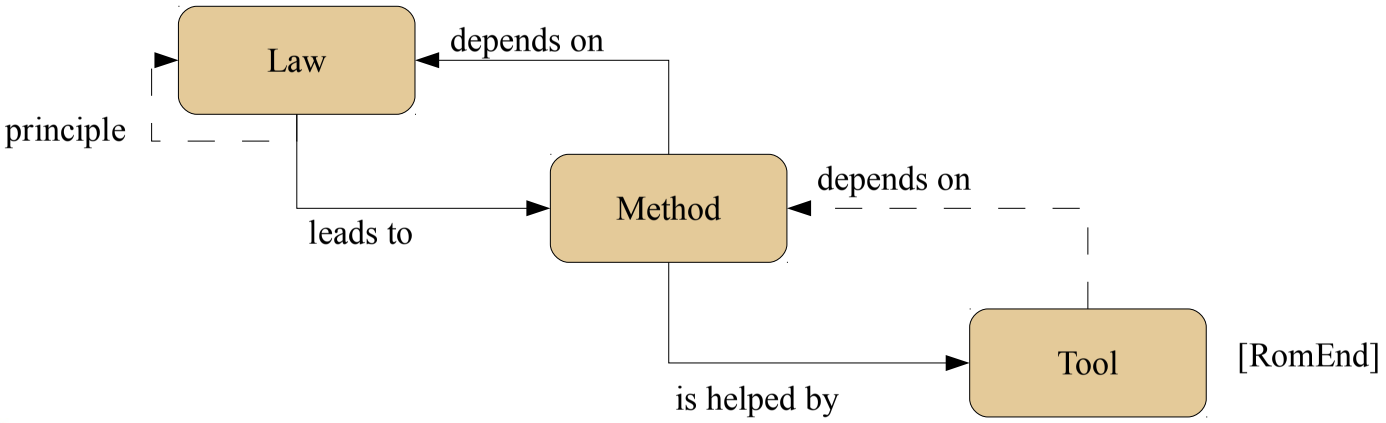
Laws

- [RomEnd] collects many research results in software engineering since the 60s. The book suggests also a division of research results (mainly descriptive, analytical and empirical models) into *observations, laws, and theories*.
- A **law** must lead to the same observation, over and over again.
- A law does not explain why an observation can be made, instead, a **theory** should explain a law.
  - Theories can be improved over time (see falsificationalism).
  - A theory can consist of a descriptive, analytical or empirical model.



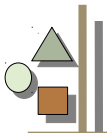
Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

- A **law** is a claim that leads to repeatable observations, and hence, leads to firm and objective knowledge.
- A **hypothesis** is a proposition that is tentatively accepted.
- A **conjecture** is a guess.
- **A principle** is a basic concept of designing, development, engineering
- **Techniques** are technical ways to support the work of the software engineer.
- Processes (procedures) behavioral instructions for the work of the software engineer.
- **(Best) Practices** are behavioral recommendations to support the work of the software engineer.
- **Methods** are procedures, techniques or practices.



Academic Skills for Computer Scientists, © Prof. Uwe Aßmann

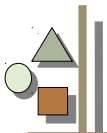
33.7 Different Kind of Research Results



What You Can Expect from a SE Researcher

- Remember:
  - An engineer works out systems to solve problems
  - a technical scientist works out methods and techiques for engineers
- Papers (examples):
  - Solution Pattern descriptions/papers
  - HOWTO-Papers
  - Literature analysis studies
  - SWOT analyses
- Artefacts (demonstrators often in 1st, 2nd and 3rd generation, most often not for industrial use):
  - Code Libraries and Frameworks helping other people doing work
  - Model frameworks
  - Tools for automation, for specific languages
  - Composition systems and reuse langauges
  - Interpreters and compilers for languages
  - Books overviewing a subject area or method
- Processes
  - Methods to engineer
  - Method frameworks for method engineering

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann



The End

Mary Shaw: "A research paper is a purposeful, designed artifact, just like a software system. Apply software design techniques to paper design:

- *Start with the requirement:* read the call for papers
- *Select an architecture:* plan the sections, what they say
- *Plan a schedule:* allow time for review, revision
- *Check consistency:* type-check text like code"

Academic Skills for Computer Scientists, © Prof. Uwe Aßmann