Design science research in information systems and software systems engineering

Prof. Dr. Roel Wieringa
University of Twente,
The Netherlands
www.cs.utwente.nl/~roelw
Research methodology accross the disciplines

• Do these disciplines have the same methodology?
  – Technical science? Build cool stuff; test it; iterate
  – Social science? Observe people, interpret what they do or say; or select a sample, do a lot of statistics; iterate.
  – Physical science? Build instruments, create phenomena, analyze data, create theories; iterate.
  – Mathematics? Read, think, write, think; iterate.
Mutual lack of appreciation

• Do they appreciate each other’s methodology?
  – For social scientists, engineers are slightly autistic tinkerers
  – For technical scientists, social scientists are chatterboxes
  – For physicists, statistics is stamp collecting
  – Mathematicians think that they provide the foundations of civilization
Our approach

• All research in all disciplines is **problem-solving**

• The problems in design science research are design problems
  – Goal is to design something useful
  – Research method is the design cycle

• The problems in empirical research are knowledge questions
  – Goal is to acquire theoretical knowledge
  – Research method is the empirical cycle

• Wieringa, R.J. (2014) *Design science methodology for information systems and software engineering*. Springer Verlag
Outline

1. What is design science
   - Research goals and problems
   - The design and engineering cycles

2. Theories
   - Scientific inference
   - Research design
What is design science?

• Design science is the **design** and **investigation** of artifacts in context
Two kinds of research problems in design science

To design an artifact to improve a problem context

- Design software to estimate Direction of Arrival of plane waves, to be used in satellite TV receivers in cars
- Design a Multi-Agent Route Planning system to be used for aircraft taxi route planning
- Design a data location regulation auditing method

To answer knowledge questions about the artifact in context

- Is the DoA estimation accurate enough in this context?
- Is it fast enough?
- Is this routing algorithm deadlock-free on airports?
- How much delay does it produce?
- Is the method usable and useful for consultants?

Is the artifact useful?

Is the answer true?
Reality check

• What research problem(s) are you investigating?
  – Artifact and context

• NB
  – The title of your thesis is the shortest summary of your research project.
  – Often, it mentions the artifact and the context.
Framework for design science

Social context:
Location of stakeholders

• Source of relevance.
• Relevance, and money, comes and goes

Goals, budgets

Design science

Improvement design

Answering knowledge questions

Existing problem-solving knowledge, Old designs

New problem-solving knowledge, New designs

Existing answers to knowledge questions

New answers to knowledge questions

Knowledge context:
Mathematics, social science, natural science, design science, design specifications, useful facts, practical knowledge, common sense, other beliefs

• Source and destination of theories
• Theories are forever
(Dis)similarity to Hevner et al. framework

- Hevner et al. want to identify these two activities
- But the methodology of these two activities is totally different
Outline

1. What is design science
   - Research goals and problems
   - The design and engineering cycles

2. Theories
   - Scientific inference
   - Research design
Goal structure: example

Social context

To improve a problem context: To provide mobile home care for the elderly

Contribution

To achieve stakeholder goals: Reduce national health care cost

Contribution

Design research

To (re)design a research instruments: a questionnaire, the setup of a field experiment

Contribution

To (re)design an artifact: A remote health monitoring system

Contribution

To answer knowledge questions: Is it usable? Does it save time? What quality of care is experienced?
Goal structure

Social context

To achieve stakeholder goals: Utility (sponsor), fun (designer), curiosity (empirical researcher)

To improve a problem context

Contribution

Design research

To (re)design an artifact

To (re)design a research instrument

Contribution

To answer knowledge questions

Contribution

Contributions

Utility (sponsor), fun (designer), curiosity (empirical researcher)
Three kinds of design research questions

1. **Design research problems** (a.k.a. *technical research questions*)
   - To improve some kind of artifact in some kind of context.

2. **Empirical knowledge questions**
   - To ask questions about the real world.

3. **Analytical knowledge questions**
   - To ask questions about the logical consequences of definitions
Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- Reduce my headache
- by taking a medicine
- that reduces pain fast and is safe
- in order for me to get back to work
Template for design problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- **Reduce my headache**
  - by taking a medicine
  - that reduces pain fast and is safe
  - in order for me to get back to work

Problem context and stakeholder goals.

Stakeholder language
Template for design problems

• Improve <problem context>
• by <treating it with a (re)designed artifact>
• such that <artifact requirements>
• in order to <stakeholder goals>

• Reduce my headache
• by taking a medicine
• that reduces pain fast and is safe
• in order for me to get back to work

Artifact and its desired properties.

Technical language
Template for design research problems

- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- Reduce patients’ headaches
- by treating it with a medicine
- that reduces pain fast and is safe
- in order for them to function as they wish

The problem is now to design an artifact that helps a class of stakeholders achieve a class of goals.
Goal structure again

- The design problem template links the artifact to the problem context and stakeholder goals

Social context

To achieve stakeholder goals: Utility (sponsor), fun (designer), curiosity (empirical researcher)

To improve a problem context

Contribution

Design research

To (re)design an artifact

Contribution

To (re)design a research instrument

Contribution

To answer knowledge questions

Contributions
Discussion

• Who are the stakeholders of your project?
  – Real or hypothetical: Stakeholders may not know they are stakeholders

• What is/are your top-level design problem(s), using our template?
  – Improve <problem context>
  – by <treating it with a (re)designed artifact>
  – such that <artifact requirements>
  – in order to <stakeholder goals>

• NB some parts may be currently uncertain, fuzzy, or unknown.
• But surely, some parts are currently known!
There is no single “correct” problem statement

• A good problem statement forces the reader to think focused about the artifact while remaining aware of the intended problem context

• Next two examples extracted from two M.Sc theses
  – http://essay.utwente.nl/67945/
  – http://essay.utwente.nl/69399/
• **BPMN Plus: a modelling language for unstructured business processes.**

• The objective of this study is
  – To investigate the way through which unstructured business processes can be modelled and managed without limiting their run-time flexibility.

• Research questions
  – Q1 What are the differences between structured and unstructured business processes?
  – Q2 What are the differences between Business Process Management and Case Management in dealing with unstructured business processes?
  – Q3 What are the capabilities of existing modelling notations to deal with unstructured business processes?
  – Q4 How to model an unstructured business process while providing run-time flexibility?

  – Improve <problem context in which unstructured business process is to be modelled> by <introducing a modeling language for unstructured business processes> such that <requirements such as run-time flexibility, and ... learnability etc?> in order to <stakeholder goals, e.g. provide better process improvement advice to clients>
• **Automated generation of attack trees by unfolding graph transformation systems.**
  
  – RQ1: Can graph transformation be used as a modeling paradigm to specify systems and organizations as input models for the attack tree generation approach?
  
  – RQ2: Can partial-order reduction, and specifically the unfolding of a graph transformation model, be used to reduce the state-space explosion problem that occurs during the automated exploration of a model?
  
  – RQ3: How can the set of attacks be converted into an attack tree, what are the trade-offs and how can additional information such as sequential AND's be included in the tree?

  • Improve <attack tree generation>
  
  • by <graph transformation system>
  
  • such that <artifact requirements, e.g. faster generation of bigger attack trees>
  
  • in order to <stakeholder goals, e.g. security risk assessment is more complete>
Three kinds of design research questions

1. Design problems (a.k.a. *technical research questions*)
   - To improve some artifact in some context.

2. Empirical knowledge questions

3. Analytical knowledge questions (math, conceptual, logical). We ignore these in this course.
Empirical knowledge questions

• **Descriptive** knowledge questions:
  – What happened?
  – How much? How often?
  – When? Where?
  – What components were involved?
  – Who was involved?
  – Etc. etc.

• **Explanatory** knowledge questions:
  – Why?
    1. What has *caused* the phenomena?
    2. Which *mechanisms* produced the phenomena?
    3. For what *reasons* did people do this?

Journalistic questions. Yield facts.

Beyond the facts. Yields theories.
Discussion

• What descriptive and explanatory knowledge are you searching for in your project?
• **BPMN Plus : a modelling language for unstructured business processes.**

• The objective of this study is
  – To investigate the way through which the unstructured business processes can be modelled and managed without limiting their run-time flexibility.

• Research questions
  – Q1 What are the differences between structured and unstructured business processes?
  – Q2 What are the differences between Business Process Management and Case Management in dealing with unstructured business processes?
  – Q3 What are the capabilities of existing modelling notations to deal with unstructured business processes?
  – Q4 How to model an unstructured business process while providing run-time flexibility?
• **Automated generation of attack trees by unfolding graph transformation systems.**
  
  – RQ1: Can graph transformation be used as a modeling paradigm to specify systems and organizations as input models for the attack tree generation approach?
  
  – RQ2: Can partial-order reduction, and specifically the unfolding of a graph transformation model, be used to reduce the state-space explosion problem that occurs during the automated exploration of a model?
  
  – RQ3: How can the set of attacks be converted into an attack tree, what are the trade-offs and how can additional information such as sequential AND's be included in the tree?
Summary

1. **Design research problems** (a.k.a. *technical research questions*)
   - Improve <problem context>
   - by <treating it with a (re)designed artifact>
   - such that <artifact requirements>
   - in order to <stakeholder goals>.

2. **Empirical knowledge questions**
   - Descriptive: what, how, when, where, who, etc. → **Facts**
   - Explanatory: Why → explanations

3. **Analytical knowledge questions**
   - Yields definitions, assumptions, theorems.
Outline

1. What is design science
   - Research goals and problems
   - The design and engineering cycles

2. Theories
   - Scientific inference
   - Research design
Implementation evaluation = Problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

Design implementation

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?
Implementation is introducing the treatment in the intended problem context

• If problem context is a real-world context…. implementation of a solution is technology transfer to the real world.
  – Not part of a research project

• If the problem is to learn about the performance of a design ... Implementation of a solution is the construction of a prototype and test environment.
  – Part of a research project
## Nesting of cycles

<table>
<thead>
<tr>
<th>Research project: design cycle</th>
<th>Problem investigation</th>
<th>Treatment design</th>
<th>Treatment validation</th>
<th>Implementation (tech transfer)</th>
<th>Implementation evaluation (in the field)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Problem investigation (How to do the validation?)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Experiment design &amp; validation (design and validate a prototype &amp; test environment)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation (construction of prototype &amp; test environment, lab or field)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluation (analyze results)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is a very special engineering cycle, called the **empirical cycle**.
Design cycle

Real-world implementation evaluation = Real-world problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

Real-world problem-oriented research

Real-world design implementation

Treatment validation

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?

Treatment design

- Specify requirements!
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

Solution-oriented research

ICWE 8th June 2016
© R.J. Wieringa
Two kinds of design science research projects

• Problem-oriented: social-science-like research
  – Investigate real-world implementations
    • E.g. How is the UML used in small and medium sized companies?
    • What is the cause if large SE projects being late?
    • How is RE done in large-scale agile projects?

• Solution-oriented: technical research
  – Design and validate an artifact
    • Design a multi-agent system for autonomous route planning
    • Design a system for remote health monitoring for the elderly
    • Design a requirements engineering technique for agile global software engineering projects
• **BPMN Plus: a modelling language for unstructured business processes.**
• The objective of this study is
  – To investigate the way through which the unstructured business processes can be modelled and managed without limiting their run-time flexibility.
• Research questions
  – Q1 What are the differences between structured and unstructured business processes?
  – Q2 What are the differences between Business Process Management and Case Management in dealing with unstructured business processes?
  – Q3 What are the capabilities of existing modelling notations to deal with unstructured business processes?
  – Q4 How to model an unstructured business process while providing run-time flexibility?
• “The practical usefulness of newly proposed modelling notation is investigated by demonstrating it with the help of an example.
• Moreover, the proposed modelling notation is validated by conducting interviews with experienced practitioners.”
Problem

- Stakeholders? Goals? : BiZZDesign consultants. To provide high-quality consultancy.
- Effects? Positive/negative goal contribution? Limits to consultancy advice.

Treatment

- Specify requirements! Omitted research question. May be part of Q2.
- Requirements contribute to goals? Omitted too.
- Available treatments? See Q3.
- Design new ones! See Q4.

Validation Omitted questions, but done by means of interviews.

- Context & Artifact → Effects? Does it work?
- Effects satisfy Requirements? Does it work as desired?
- Trade-offs for different artifacts? Performance of different languages, similar cases?
- Sensitivity for different Contexts? Does it work in different cases?
Automated generation of attack trees by unfolding graph transformation systems.

- RQ1: Can graph transformation be used as a modeling paradigm to specify systems and organizations as input models for the attack tree generation approach?
- RQ2: Can partial-order reduction, and specifically the unfolding of a graph transformation model, be used to reduce the state-space explosion problem that occurs during the automated exploration of a model?
- RQ3: How can the set of attacks be converted into an attack tree, what are the trade-offs and how can additional information such as sequential AND's be included in the tree?
**Problem** *Implied, no further details.*

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

**Treatment**

- Specify requirements! *Omitted RQ, presumably scalability (RQ2).*
- Requirements contribute to goals?
- Available treatments?
- Design new ones! *RQ1, RQ2, RQ3.*

**Validation** *Omitted RQs*

- Context & Artifact → Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?
Design research problems

• Improve <problem context>
• by <treating it with a (re)designed artifact>
• such that <artifact requirements>
• in order to <stakeholder goals>.

Design cycle
• Problem investigation
• Treatment design
• Treatment validation

2. Empirical knowledge questions
   – Descriptive: what, how, when, where, who, etc. → Facts
   – Explanatory: Why → Theories

3. Analytical knowledge questions
   – Yields definitions, assumptions, theorems.
Questions?
Outline

1. What is design science
   - Research goals and problems
   - The design and engineering cycles

2. Theories
   - Scientific inference
   - Research design
Facts, generalizations, explanations

• Descriptive knowledge questions:
  – What happened?
  – How much? How often?
  – When? Where?
  – What components were involved?
  – Who was involved?
  – Etc. etc.

• Explanatory knowledge questions:
  – Why?
    • What caused this phenomenon?
    • What mechanisms produced it?
    • Why did people do this?

• Yield facts about cases or samples.
• May be generalized beyond the facts, to descriptive theories about a population

• Beyond the facts: explanatory theories about cases/samples.
• May be generalized to explanatory theories about a population
Two ways to go beyond the facts

Observed sample of cases

• What happens in these cases?
• What average, variance in this sample?

Unobserved population

• What happens in all cases?
• What average, variance in this population?

Generalize

Explain

• Why?

Explain

• Why?
Facts versus theories

Facts

- Observed sample of cases
  - What happens in these cases?
  - What average, variance in this sample?

Explanatory theory of the case/sample

- Why?

Generalize

Descriptive theory of the population

- Unobserved population
  - What happens in all cases?
  - What average, variance in this population?

Explanatory theory of the population

- Why?
What is a theory?

• A **theory** is a belief that there is a pattern in phenomena.
  – Idealizations: “Merging two faculties reduces cost.” “This works in theory, but not in practice.”
  – Speculations: “The NSA is monitoring all my email.”
  – Opinions: “The Dutch lost the soccer competition because they are not a team.”
  – Wishful thinking: “My technique works better than the others.”
  – **Scientific theories:** Theory of electromagnetism
Scientific theories

• A scientific theory is a belief that there is a pattern in phenomena, that has survived
  – Tests against experience:
    • Observation, measurement
    • Possibly: experiment, simulation, trials
  – Criticism by critical peers:
    • Anonymous peer review
    • Publication
    • Replication

• Examples
  – Theory of electromagnetism
  – Technology acceptance model
  – Theory of the UML

• Non-examples
  • Religious beliefs
  • Political ideology
  • Marketing messages
  • Most social network discussions
Scientific design theories

• A scientific design theory is a belief that there is a pattern in the interaction between an artifact and its context

• Examples:
  – Theory of the UML in software engineering projects
  – Theory of your design in the intended problem context
The structure of scientific theories

1. Conceptual framework
   - Definitions of concepts.

2. Generalizations
   - Express beliefs about patterns in phenomena.
Theory of electromagnetism

• Conceptual framework (concepts defined to describe and explain the relevant phenomena):
  – Definitions of electric current, electric charge, potential difference, electric resistance, electric power, capacitance, electric field, magnetic field, magnetic flux density, inductance, ..., ... and their units.

• Generalizations
  – Electric charges attract or repel one another with a force inversely proportional to the square of their distance.
  – Magnetic poles attract or repel one another in a similar way and always come in North-South pairs.
  – An electric current inside a wire creates a corresponding circular magnetic field outside the wire.
  – A current is induced in a loop of wire when it is moved towards or away from a magnetic field
Technology Acceptance Model

• Conceptual framework
  – Definitions of perceived usefulness, perceived ease of use, perceived resources, attitude towards using, behavior intention to use, actual system use

• Generalization

• Example design theory
  
  – Concepts: definitions of concepts to specify a direction-of-arrival recognition algorithm, and of concepts to describe antenna array, and of accuracy and execution time
  
  – Generalization: (Algorithm MUSIC) x (antenna array, plane waves, white noise) \rightarrow (execution time less than 7.2 ms.)
• **Descriptive UML theory**
  – Concepts: UML concepts, definitions of software project, of software error, project effort.
  – Descriptive generalization: (UML) X (SE project) → (Less errors, less effort than similar non-UML projects)

• **Explanatory UML theory:**
  - Concepts: definition of concept of domain, understandability
  - Explanatory generalizations:
    o UML models resemble the domain more than other kinds of models;
    o they are easier to understand for software engineers;
    o So they they make less errors and there is less rework (implying less effort).
The use of theories in the design cycle

Design theory describes and possibly explains interaction between A and C.

- Context & Artifact $\rightarrow$ Effects?
- Effects satisfy Requirements?
- Trade-offs for different artifacts?
- Sensitivity for different Contexts?

Problem theory describes and explains the problem. Symptoms and diagnosis.

Treatment design

- Specify requirements
- Requirements contribute to goals?
- Available treatments?
- Design new ones!

Implementation evaluation = Problem investigation

- Stakeholders? Goals?
- Conceptual problem framework?
- Phenomena? Causes, mechanisms, reasons?
- Effects? Positive/negative goal contribution?

Treatment validation

- Effects?
All theories can be used to make predictions

- **Problem theory** describes and explains the problem. Symptoms and diagnosis.
- **Design theory** describes and possibly explains interaction between A and C.

- Both theories may be used to predict
  - What will happen if the problem is untreated
  - What will happen if the treatment is applied
Discussion

• Which theory do you hope to produce?
  – (Theory about real-world problems/implementations, or theory about newly designed artifact in a context.)
  – Explanations, generalizations provided by the theory?

• What evidence do you have, and what do you still intend to produce?
Outline

1. What is design science
   - Research goals and problems
   - The design and engineering cycles

2. Theories
   - Scientific inference
   - Research design
Facts versus theories

Facts

- What happens in these cases?
- What average, variance in this sample?

Observed sample of cases

Explanatory theory of the case/sample

- Why?

Generalize

Descriptive theory of the population

- What happens in all cases?
- What average, variance in this population?

Unobserved population

Explanatory theory of the population

- Why?
Three kinds of explanation

Facts

- What happens in these cases?
- What average, variance in this sample?

Explanatory theory of the case/sample

- Explain by
  - Causes
  - Mechanisms
  - Reasons

- Why?

Descriptive theory of the population

- What happens in all cases?
- What average, variance in this population?

Unobserved population

- Why?

Explanatory theory of the population

- Explain by
  - Causes
  - Mechanisms
  - Reasons

Generalize

Observed sample of cases
Example

• **Descriptive question:** Is the light on?
  – Based on observation: Yes.
  – When? Now.
  – Where? Here.

• **Explanatory question:** Why is it on?
  1. **Cause:** because someone turned the light switch, it is on (and not off). Explains difference with off-state.
  2. Why does this cause the light to switch on? **Mechanism:** because the switch and light bulbs are connected by wires to an electricity source, in this architecture ..., and these components have these capabilities ... Explains how on-state is produced.
  3. By why did someone turn the light on? **Reasons:** Because we wanted sufficient light to be able to read, and it was too dark to read. Explains which stakeholder goal is contributed to.
Another example: software

• **Descriptive question:** What is the performance of this program?
  – Execution time for different classes of inputs?
  – Memory usage?
  – Accuracy?
  – Etc. etc.

• **Explanatory question:** Why does this program have this performance (compared to others)?
  1. **Cause:** Variation in execution time is caused by variation in input; etc.
  2. **Mechanism:** Execution time varies this way because it has this architecture with these components
  3. **Reasons:** Observed execution time varies this way because users want to be on-line all the time, and therefore provide these inputs
Another example: method

• **Descriptive question:** What is the performance of this method for developing software?
  – Understandability for practitioners
  – Learnability
  – Quality of the result
  – Perceived utility
  – Etc. etc.

• **Explanatory question:** Why does this method have this performance?
  1. **Cause:** Difference in understanding of methods by software engineers is attributed to differences in the methods.
  2. **Mechanism:** These differences are explained by the structure of the method and/or the structure of cognition.
  3. **Reasons:** Developers use this method because it is currently a hype among developers
Keywords in the three kinds of explanations

- **Descriptive question: What is happening?**

- **Explanatory question: Why did this happen?**
  1. **Cause:** effect attributed to a cause. Explain difference in outcomes by difference in interventions.
  2. **Mechanism:** Outcome produced by interaction among components. Explain capability of system in terms of capabilities of components.
  3. **Reasons:** Outcome contributes to a goal. Explain outcome in terms of rational takeholder choices.
One more example

• **Causal explanation:** effect attributed to a cause. Explain difference in outcomes by difference in interventions. Causation is difference-making.
  – *The coffee made me stay awake late.*

• **Architectural explanation:** Outcome produced by interaction among components. Explain capability of system in terms of capabilities of components
  – *Caffeine has a psychostimulant effect because it antagonizes adenosine, which normally inhibits neurotransmitters such as dopamine.*

• **Rational explanation:** Outcome contributes to a goal. Explain outcome in terms of rational takeholder choices.
  – *I worked late because I wanted to finish the paper before the deadline.*
Internal validity

• Degree of support for an explanation
• Threats that decrease support:
  – Outcome may have many causes
    • Which one is most plausible?
    • Which ones can and cannot be ruled out?
  – Effect of a cause may be produced by various mechanisms
    • Which components played a role, and which did not?
    • How did they interact? How do we know?
  – An action may have many reasons
    • Which ones were operative?
    • What evidence do we have for it?
Two kinds of generalization

- **Observed sample**
  - What happens in these cases?
  - What average, variance in this sample?
  - Explain by
    - Causes
    - Mechanisms
    - Reasons
    - Why?

- **Unobserved population**
  - What happens in all cases?
  - What average, variance in this population?
  - Explain by
    - Causes
    - Mechanisms
    - Reasons
    - Why?

**Explanatory theory of the case/sample**

**Descriptive theory of the population**

- By analogy from cases
- By inferential statistics from sample
Generalization by analogy

• Observation:
  – Artifact: This prototype implementation of the MUSIC algorithm,
  – Context: when used to recognize direction of arrival of plane waves received by an antenna array, in the presence of only white noise, running on a Montium 2 processor,
  – Effect: has execution speed less than 7.2 ms and accuracy of at least 1 degree.

• Generalization by analogy:
  – All similar implementations
  – Running in similar contexts
  – Will show similar performance

**Descriptive generalization.** Implicit assumptions:
1. The mechanisms that explain this performance will be present in all similar artifacts and contexts, and
2. will not be undone by other mechanisms.
Generalization by analogy

• Observations:
  – Artifact: this version of the UML
  – Context: Used in this software project
  – Effect: Produces software with less errors and less effort than in similar projects without the UML,
  – Explanation: UML models are easier to understand for software engineers because they resemble the domain more than other kinds of models, and
  – So the software engineers make less errors and there is less rework.

• Generalization
  – In similar projects, UML will have similar effects
  – Assumptions: The mechanisms that produced these effects will be present in all similar projects, i.e. UML is used in the same way, and any relevant social and cognitive mechanisms are present in similar projects too, and
  – The effects will not be undone by other mechanisms
Generalization by analogy

• Must be based on architectural similarity
  – Similar components, with similar capabilities
  – Similar mechanisms involving these components

• Analogy based in similarity of superficial features, without knowledge of underlying mechanisms, is too weak a basis for generalization.
Example of an unsound analogic generalization

- Wallnuts look like brains.
- Brains can think.
- Therefore .... Wallnuts can think

- This is only superficial similarity
- There is no mechanism that produces thinking in brains and wallnuts!
External validity

- Degree of support for generalization by analogy
- Threats that decrease support:
  - Cases that look superficially similar may not be architecturally similar.
  - Analogic generalization is not universal: it may be falsified by interfering mechanisms.
- Mitigate this by analytic induction: Cases are studied one by one, theory updated in between
  - Start with an initial theory about how mechanisms produce phenomena
  - Update the theory after each case
  - Look for confirming as well as falsifying cases
Three kinds of explanation

Facts

- What happens in these cases?
- What average, variance in this sample?

Explain by
- Causes
- Mechanisms
- Reasons

Explanatory theory of the case/sample

Descriptive theory of the population

- What happens in all cases?
- What average, variance in this population?

Explain by
- Causes
- Mechanisms
- Reasons

Explanatory theory of the population

Observed sample

- By analogy from cases
- By inferential statistics from sample

Unobserved population

© R.J. Wieringa

ICWE 8th June 2016
Generalization by statistical inference


• Sample of 20 open source web applications from the population of all OS web applications. Count the number of security vulnerability caused by coding errors (rather than by design flaws or configuration errors).

• Observation: The average percentage of vulnerabilities caused by coding errors per OS web application in the sample is 73%.

• Generalization by statistical inference:
  – Assuming a random sample, and
  – assuming that the proportion of coding errors is constant and independent across web applications,
  – the average percentage of vulnerabilities caused by coding errors in any OS web application in the population is roughly 73% ± 4% with roughly 95% confidence: 95% of the times we conclude this, the conclusion is correct
Example continued:
Further generalization by analogy

• *We may want to generalize by analogy to similar populations, e.g. the population of all OS software, or of all Web Applications.*
  
  – *This would be justified if the mechanisms and reasons that produced the phenomenon in the study population are the same in these other populations.*
  
  – *But we do not really know what these mechanisms are....*
Generalization by statistical inference

• Hypothetical example:
  • Four groups of 9 to 26 students made UML domain model from Use case model for two systems, with or without using System Sequence Diagrams (SSDs) and System operations contracts (SOCs). Four-group crossover design.

• Observation:
  – In the observed samples, when SSDs and SOCs were used, average correctness of models was higher, and effort to produce them was lower.

• Generalization by statistical inference:
  – Pairwise t-test, simple repeated measures ANOVA and mixed repeated measures ANOVA support the generalization that average correctness of models and effort to produce them is better when SSDs and SOCs are used in the population of all software engineering students. This conclusion is plausible but not always correct.

• Explanation:
  – By listing all possible causes, and assessing them on their plausibility, the use of SSDs and SOCs is the most plausible cause of these effects (and not the competence of the students or the positive expectation of the experiments, or ...)
Example continued

• We may want to **generalize by analogy** to similar populations, e.g. the population of professional software engineers.
  – Need to discuss if the social or cognitive mechanisms that produce the results in the student population, are the same as those in the population of professional software engineers.

• NB the setup of the experiment resembles the classical Randomized Controlled Trial used to validate the effect new drugs
An aside


• They did this ..... but unfortunately found hardly any support for a statistically significant difference.
Statistical conclusion validity

• Degree of support for a statistical inference
• Threats:
  – The study population may be undefined
  – Sampling may not be random
  – Assumptions of statistical techniques may not be satisfied
If the sample is almost the size of the population, and the unobserved part of the population is similar to the observed part, statistical inference may be omitted.

- descriptive statistics of the sample may allow us to generalize to the population based on similarity.
- This is often called statistical learning (e.g. regression or classification)

Based on an analysis of data about 95% of the Dutch male population, you compute an average height of 1m75.

However, your sample excluded all males taller than 2m.

The real average is 1m83
How to use all of these kinds of arguments in your research?
Case-based inference

1. Descriptive inference
   - Data from cases
   - Descriptions, summaries

2. Abductive inference
   - Explanations in terms of mechanisms, causes, reasons

3. Analogic inference
   - Generalizations over a population

- Analogic inference to similar cases must be based on architectural explanations (in terms of mechanisms or reasons)
- Even if the sample is very big!
Sample-based inference

- Statistical inference yields descriptive generalization over a study population.
- Differences in outcome may be explainable by causes.
- Analogic generalization to similar populations must be based on architectural explanation of those causes.
Validity of inferences

- a) Descriptive validity: no information added in the descriptions
- b) Internal validity: degree of support for explanations
- c) External validity: degree of support for analogic generalizations
- d) Statistical conclusion validity: degree of support for statistical inference

Data from samples → Descriptions, sample statistics

- a) Descriptive inference
- b) Abductive inference
- c) Analogic inference
- d) Statistical inference

Explanations in terms of mechanisms, causes, reasons → Generalizations over a population
Outline

1. What is design science
   – Research goals and problems
   – The design and engineering cycles

2. Theories
   – Scientific inference
   – Research design
How can you set up our empirical research so that you can support the desired inferences
Design decisions for research setup

- Which treatment (if any?)
- Which measurements?
- Which objects of study?
- Which population?
- How to sample?

Researchers

Object of Study

Sample

Representation

- Treatment data
- Measurement data

- Treatment instrument
- Measurement instrument
# Research designs

<table>
<thead>
<tr>
<th></th>
<th>Observational study (no treatment)</th>
<th>Experimental study (treatment)</th>
</tr>
</thead>
</table>
| **Case-based:**          | Observational case study           | • **Expert opinion** (mental simulation by experts),  
                          |                                    | • **Mechanism experiments** (simulations, prototyping),  
                          |                                    | • **Technical action research** (experimental use of the artifact in the real world) |
| investigate single cases, look at architecture and mechanisms | Survey                            | • **Statistical difference-making experiment** (treatment group – control group experiments) |

**Sample-based:** investigate samples drawn from a population, look at averages and variation
Checklist for the empirical cycle: context

1. Improvement goal?
2. Knowledge goal?
3. Current knowledge?

Design cycle

Empirical cycle

4. ...
   ...
16. ...

17. Contribution to knowledge goal?
18. Contribution to improvement goal?

Designing something useful

Answering a knowledge question
This is a checklist for
• research design,
• research reporting,
• reading a report.
App. B in my book & my web site

Research problem analysis
4. Conceptual framework?
5. Knowledge questions?
6. Population?

Data analysis
12. Data?
13. Observations?
14. Explanations?
15. Generalizations?
16. Answers?

Research execution
11. What happened?

Research & inference design
7. Object of study?
8. Treatment specification?
9. Measurement specification?
10. Inference?

Design validation
7. Object of study validity?
8. Treatment specification validity?
9. Measurement specification validity?
10. Inference validity?

Empirical cycle
## Validity of research design: Does research setup match inference?

<table>
<thead>
<tr>
<th>Case-based research</th>
<th>Observational study (no treatment)</th>
<th>Experimental study (treatment)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Observational study</strong></td>
<td>Investigate single cases, look at architecture and mechanisms</td>
<td>Investigate samples drawn from a population, look at averages and variation</td>
</tr>
<tr>
<td><strong>Experimental study</strong></td>
<td>Support or falsify causal explanations of the differences between treatment and non-treatment</td>
<td>Support or falsify causal explanations of the differences between treatment and non-treatment</td>
</tr>
</tbody>
</table>

- **Case-based**: investigate single cases, look at architecture and mechanisms
  - **Observational case study**:
    - Expert opinion (mental simulation by experts)
    - Mechanism experiments (simulations, prototyping)
    - Technical action research (experimental use of the artifact in the real world)

- **Sample-based**: investigate samples drawn from a population, look at averages and variation
  - Survey
  - Statistical difference-making experiment (treatment group – control group experiments)

---

ICWE 8th June 2016

© R.J. Wieringa
Summary

Design research problems
- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>.

Design cycle
- Problem investigation
- Treatment design
- Treatment validation

Artifacts $\rightarrow$ Design cycle $\rightarrow$ Artefacts

Empirical knowledge questions
- Descriptive: what, how, when, where, who, etc. $\rightarrow$ Facts
- Explanatory: Why $\rightarrow$ explanations

Empirical cycle
- Research problem analysis
- Research design & validation
- Research execution
- Data analysis

Theories $\rightarrow$ Empirical cycle $\rightarrow$ Theories

Analytical knowledge questions
When to use these methods in design science research?
More robust generalizations

Population

Large samples

Small samples

Idealized

Practical

Scaling up to conditions of practice

Street credibility (works in practice)

More realistic conditions of practice

Laboratory credibility (works in theory)

- Just like New Drug Research
• Scaling up:
  – Single-case mechanism experiment (laboratory simulation)
  – Expert opinion
  – Single-case mechanism experiment (field simulation)
  – TAR (apply technique in a real-world project)
Take-home

• Design science
  – Design problems with the design cycle
  – Empirical knowledge questions with the empirical cycle

• Design theories
  – Effects of artifact in context

• Validation research methods
  – Case-based and sample-based research
  – Architectural inference from cases (mechanisms, reasons)
  – Statistical inference from samples (averages, variance)
  – Causal reasoning from experiments

• Scaling up from lab to practice

Wieringa, R.J. (2014) *Design science methodology for information systems and software engineering*. Springer Verlag


