Design Science Methodology
Part A

Prof. Dr. Roel Wieringa
University of Twente,
The Netherlands
www.cs.utwente.nl/~roelw

Revised by Max HE with Thanks

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 (自闭症修补匠)
  - For technical scientists, social scientists are chatterboxes (喋喋不休者, 唠叨的人)
  - For physicists, statistics is stamp collecting
  - Mathematicians think that they provide the foundations of civilization

Our approach

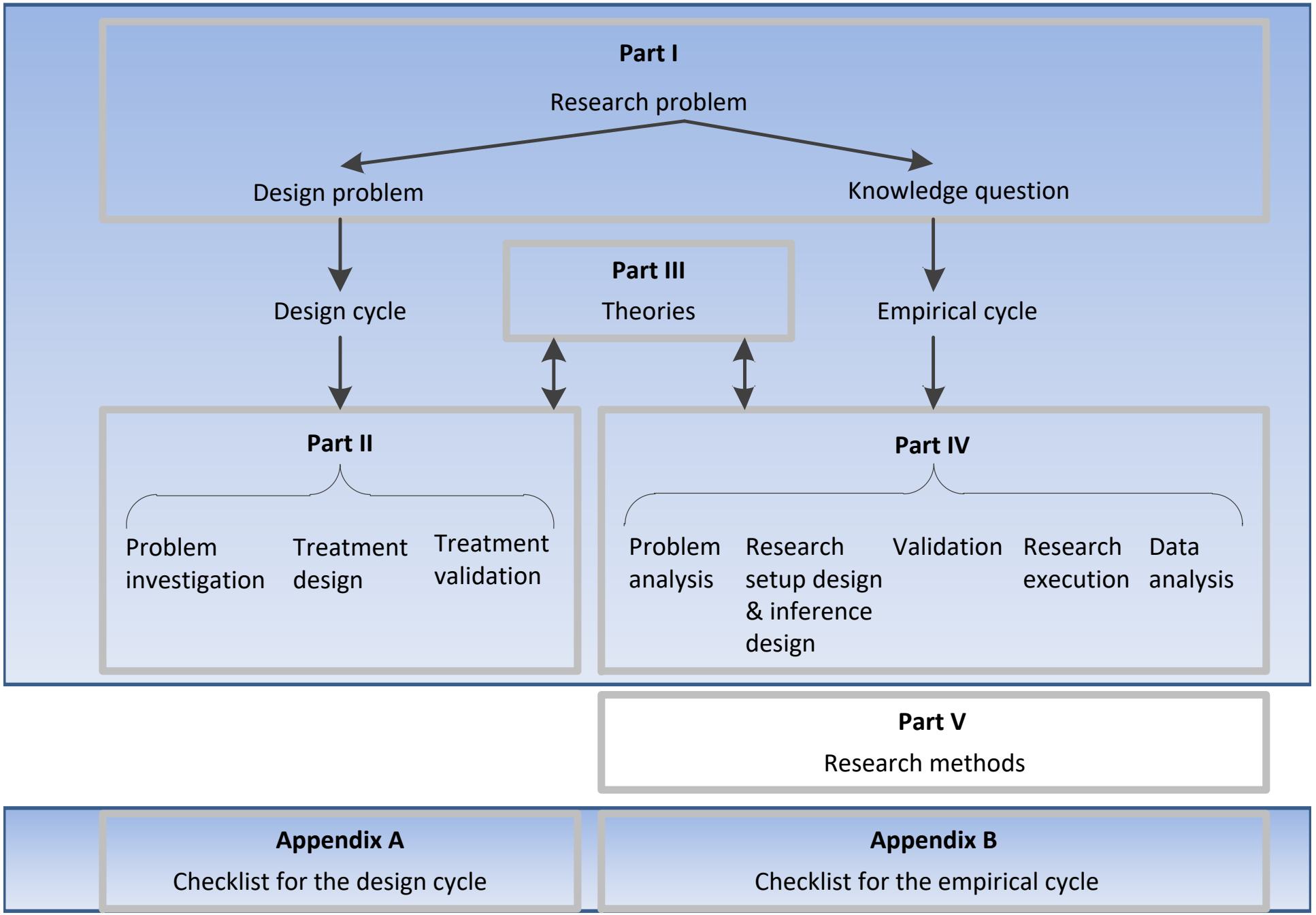
- All research in all disciplines is **problem-solving**
- The problems in design science research are design problems
  - Goal is to design something useful
  - Research method is the design cycle
- The problems in empirical research are knowledge questions
  - Goal is to acquire theoretical knowledge
  - 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

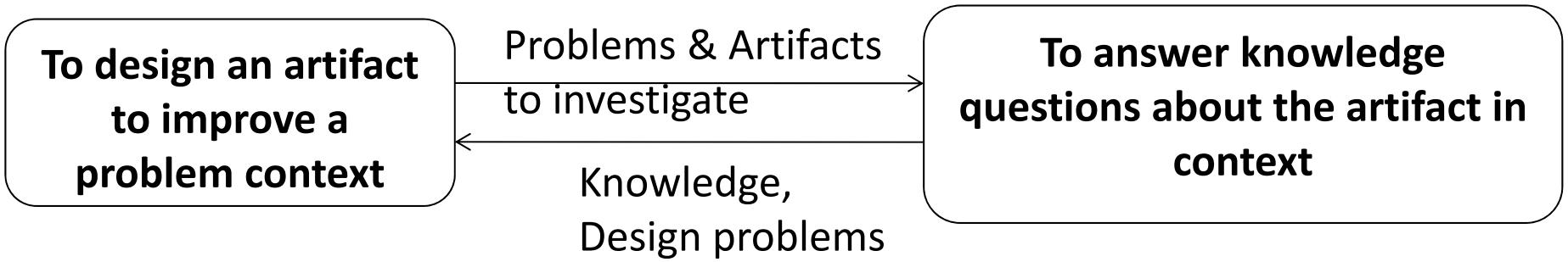
Outline


Part I
Research problem
Design problem
Knowledge question
Part III
Theories
Design cycle
Empirical cycle
Part II
Problem investigation
Treatment design
Treatment validation
Part IV
Problem analysis
Research setup design & inference design
Validation
Research execution
Data analysis
Part V
Research methods
Appendix A
Checklist for the design cycle
Appendix B
Checklist for the empirical cycle

What is design science?

- Design science is the **design** and **investigation** of artifacts in context

Two kinds of research problems in design science



- *Design software to estimate Direction of Arrival of plane waves, to be used in satelite 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*

Is the artifact **useful**?

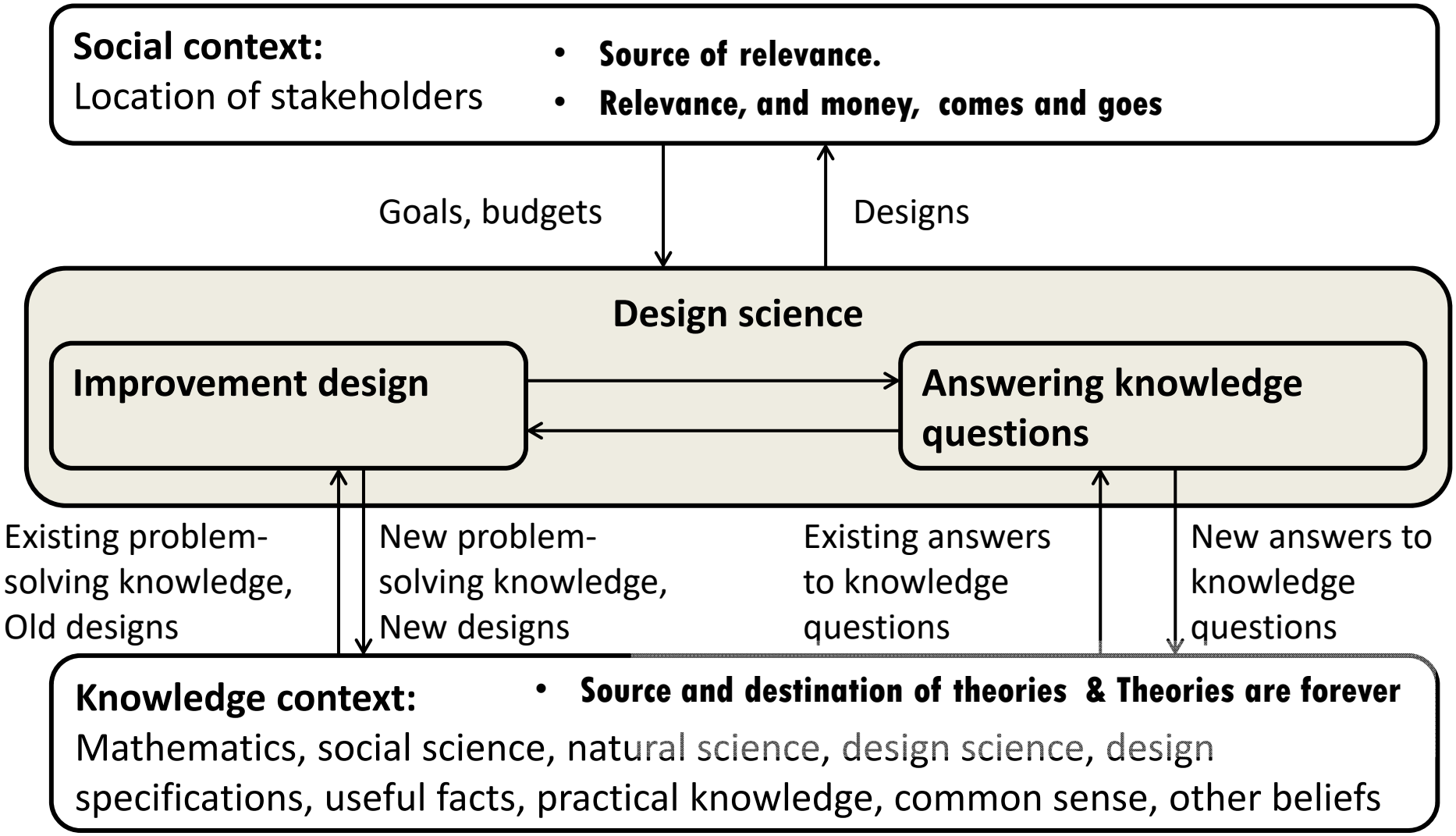
- *Is the DoA estimation accurate enough`in this context?*
- *Is it fast enough?*
- *Is this routing algorithm deadlock-free on airports?*
- *How much delay does it produce?*
- *Is the method usable and useful for consultants?*

Is the answer **true**?

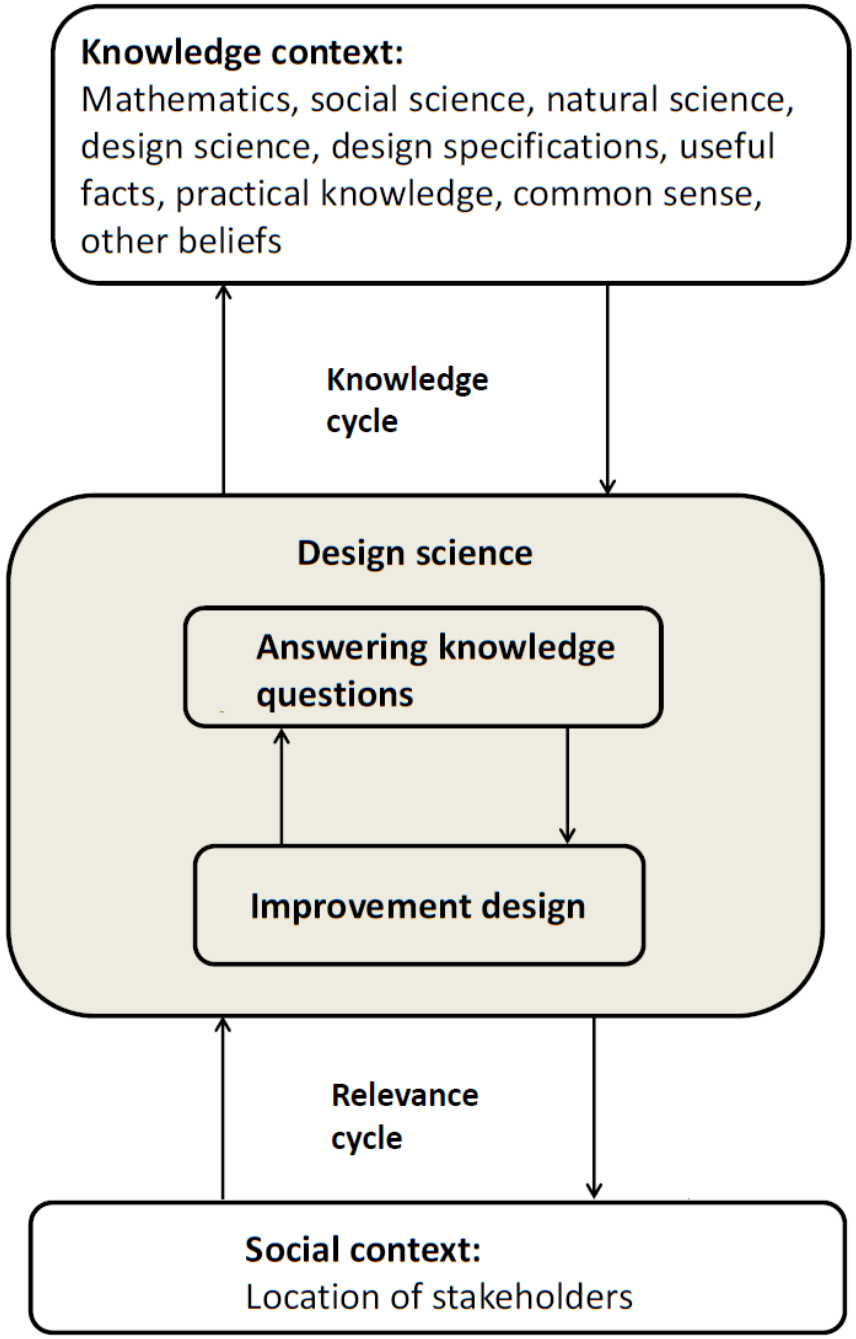
Reality check

- What research problem(s) are you investigating?
  - Artifact and context
- NB
  - The title of your project/thesis is the shortest summary of your research project.
  - Often, it mentions the artifact and the context.

Framework for design science



(Dis)similarity to Hevner et al. framework

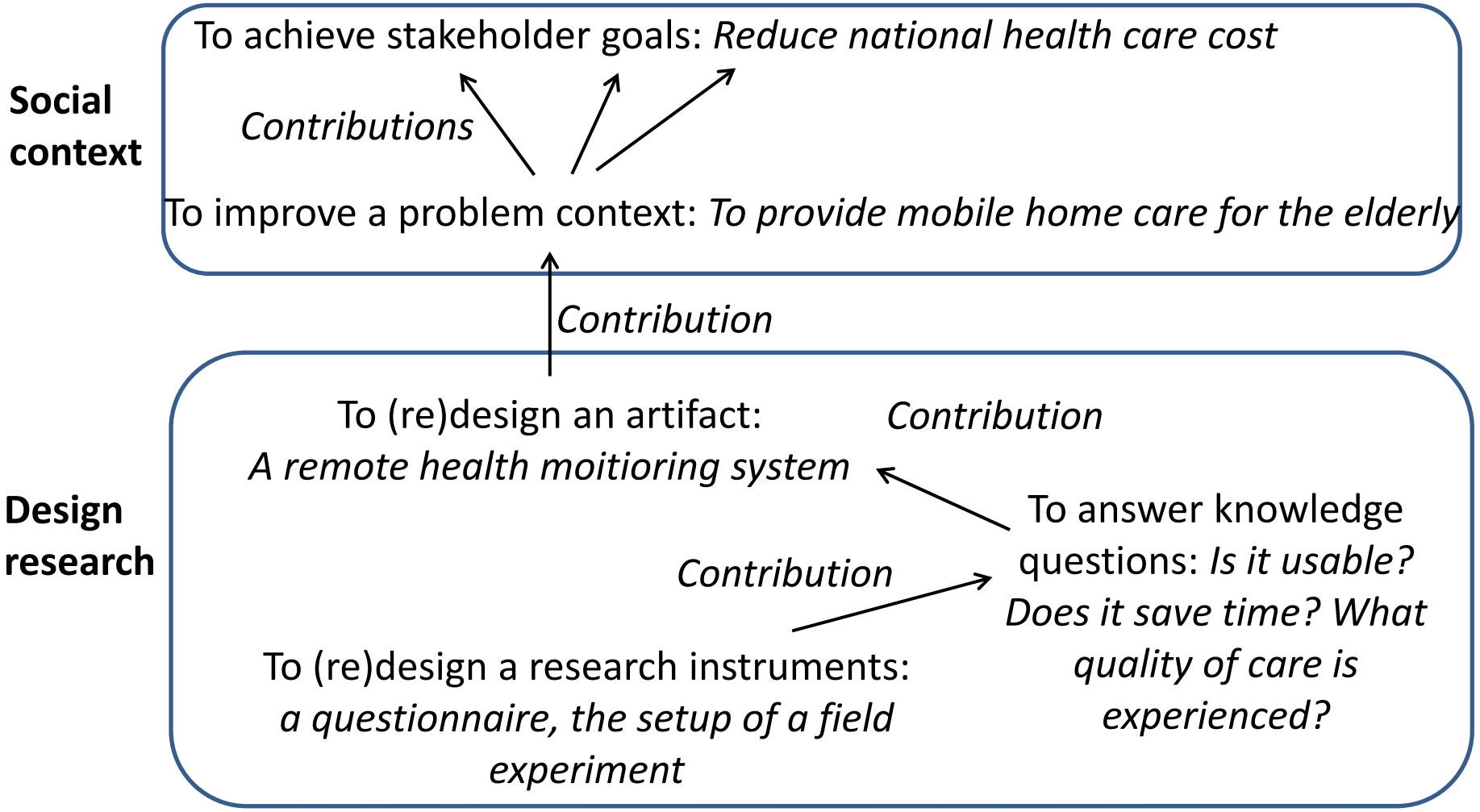


- Hevner et al. want to identify these two activities
- But the methodology of these two activities is totally different

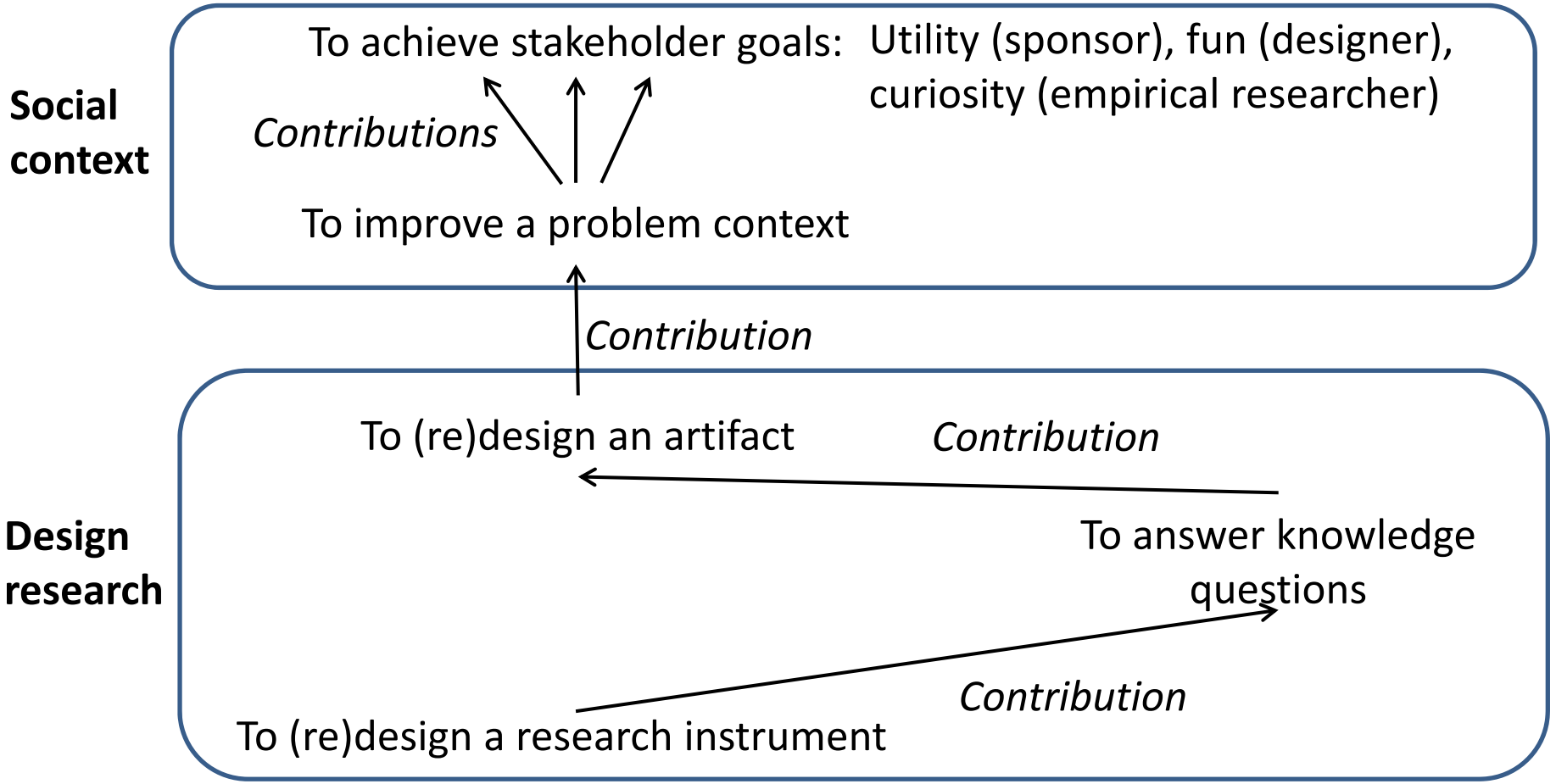
Outline

1. What is design science
   - **Research goals and problems**
   - The design and engineering cycles
2. Theories
   - Scientific inference
   - Research design

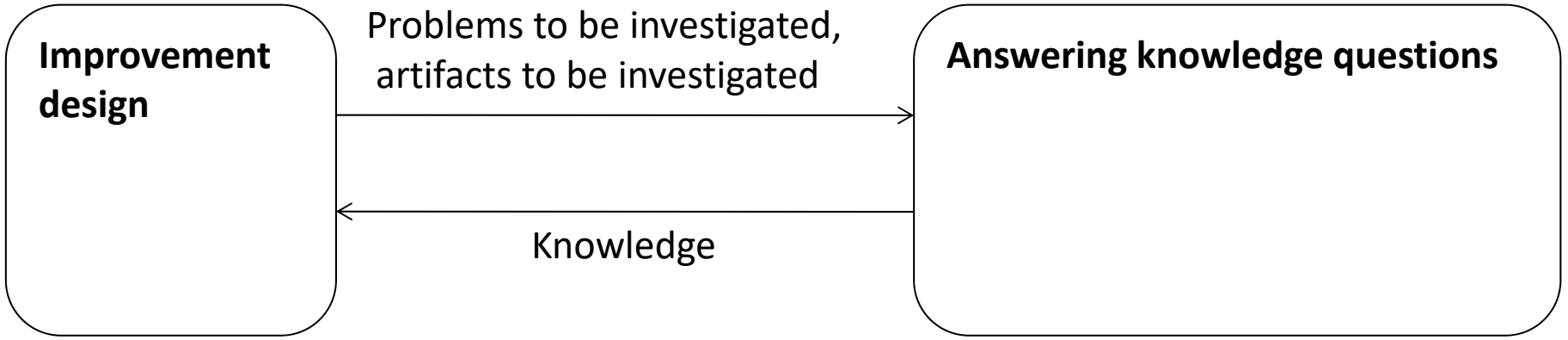
Goal structure: example



Goal structure



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

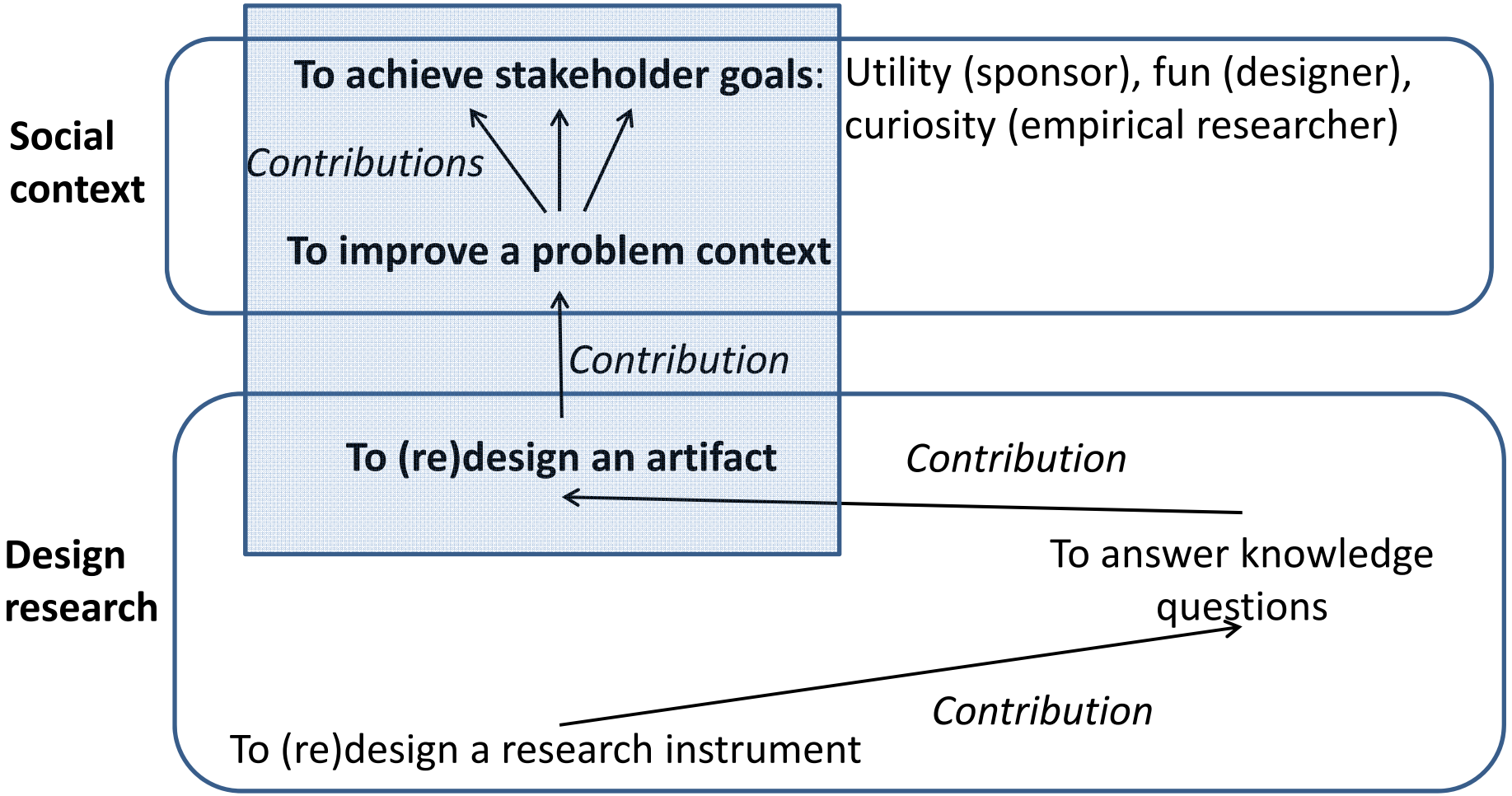
- Improve <problem context>
- by <treating it with a (re)designed artifact>
- such that <artifact requirements>
- in order to <stakeholder goals>

- *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



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 focussed about the artifact while remaining aware of the intended problem context

- *BPMN Plus : a modelling language for unstructured business processes.* ← Artifact / ← Context
- 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?

Context

Artifact

- 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.

Journalistic questions.
Yield facts.

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

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?

- Explanatory questions?
- Analytical questions?

Descriptive knowledge questions; (outcome of interviews)

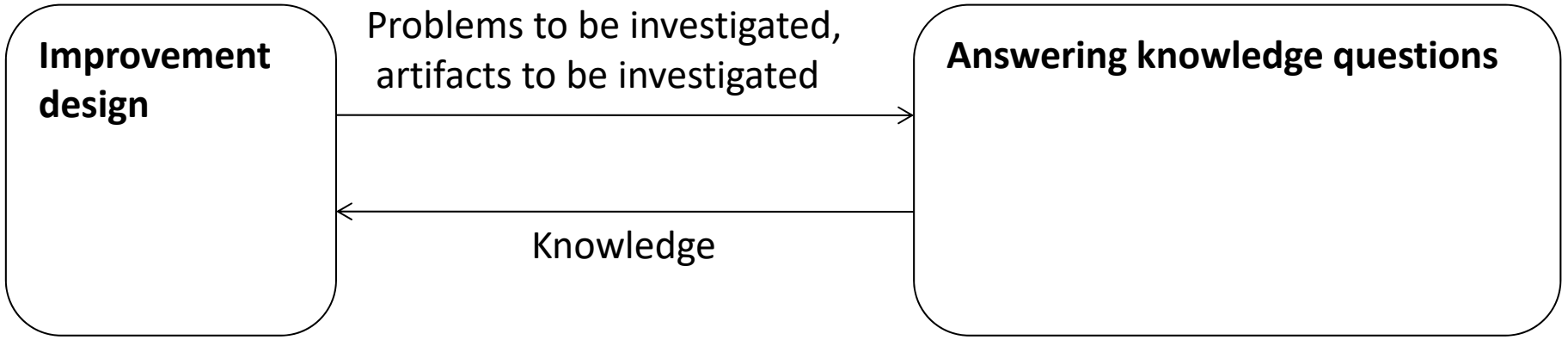
Design problem

- *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?

Design problems

- Descriptive questions?
- Explanatory questions?
- Analytical questions?

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.

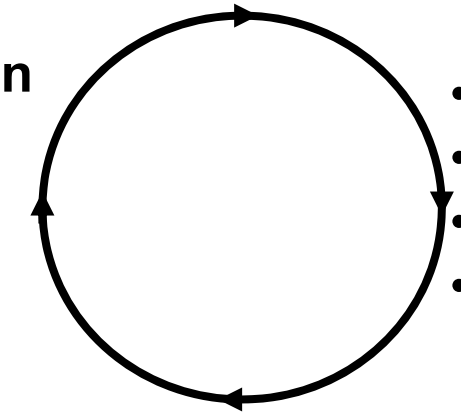
Outline

1. What is design science
   - Research goals and problems
   - **The design and engineering cycles**
2. Theories
   - Scientific inference
   - Research design

Engineering cycle

This is a checklist. See appendix A in the book & on my web site

! = Action
? = Knowledge question



Implementation evaluation = Problem investigation

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

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!

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

To be continued …