Chapter 2
Research Goals and Research Questions

To frame a research project, you have to specify its research goal (Sect. 2.1). Because a design science project iterates over designing and investigating, its research goal can be refined into design goals and knowledge goals. We give a template for design problems in Sect. 2.2 and a classification of different kinds of knowledge goals in Sect. 2.3.

2.1 Research Goals

To understand the goals of a design science research project, it is useful to distinguish the goals of the researcher from the goals of an external stakeholder. The researcher's goals invariably include curiosity and fun: curiosity what the answer to knowledge questions is and fun in the design and test of new or improved artifacts. In this sense, all design science research is *curiosity-driven* and *fun-driven research.*

The researcher may have additional goals, such as the desire to improve society or to promote the well-being of people. This kind of goal is similar to the goals that external stakeholders may have. One of the external stakeholders will be the *sponsor* of the project, which is the person or organization paying for the research. The sponsor allocates a budget to the research project in order to achieve some goals and expects to receive useful designs that serve these goals and useful knowledge about those designs. For most sponsors, design science research projects are *utility driven* and *budget constrained.* Some sponsors however may be willing to sponsor some researchers to do *exploratory research.* The sponsor may still hope that useful results will emerge, but whether this will happen is very uncertain.

Putting all of these motivations together gives us a wide variety of kinds of projects, ranging from market-oriented projects in which an enhancement to a particular product must be designed to exploratory projects where even the sponsor

R.J. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*, DOI 10.1007/978-3-662-43839-8_2

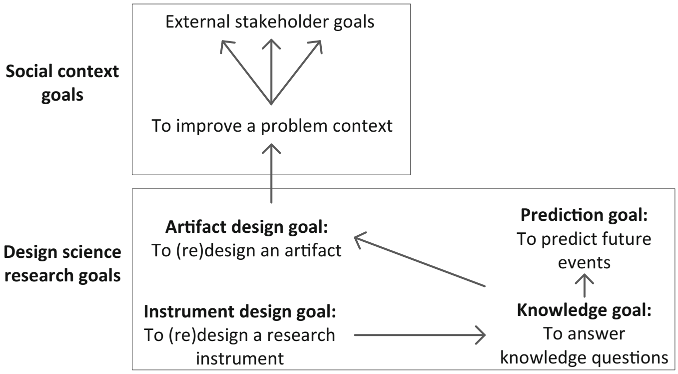

Fig. 2.1 Goal structure of a design science research project. The goals on the left concern improvement of the real world, and those on the right concern our beliefs about the world. In an exploratory project, there may be no higher-level improvement goals

has only a vague idea of the possible utility of the designs or knowledge that will come out of the project. These goals require different capabilities of the researcher and have a different risk profile for the sponsor.

In all these cases, design science research projects have a goal hierarchy with the characteristic contribution structure shown in Fig. 2.1. The goals on the right hand concern our beliefs about past, present, and future phenomena. The goals on the left are design goals or more generally improvement goals. We now discuss the goal structure in more detail, starting at the right-hand side.

Design science researchers often have a **prediction goal**. For example, we may want to predict how an artifact will interact with a problem context or how a problem would evolve if it were not treated. A prediction is a belief about what will happen in the future, which will turn out to be true or false. To make these predictions, we need knowledge.

Possible **knowledge goals** of a design science research project are to describe phenomena and to explain them. For example, a knowledge goal may be to describe what happens when an artifact interacts with a context and to explain this in terms of internal mechanisms of the artifact and context.

In order to answer the knowledge questions, some design science research projects may have to design instruments. For example, the researcher may have to build a simulation of an artifact in context or to construct a questionnaire to collect user opinions. These **instrument design goals** are the lowest-level design goals in Fig. 2.1.

Moving up in the diagram, design science research projects usually have a higher-level design goal such as to improve the performance of some artifact in a context. We call this an **artifact design goal** or, alternatively, a **technical research goal**.

The goal of artifact design is in turn to solve, mitigate, or otherwise improve some problem in the social context of the project, such as the goal to make viewing satellite TV in a car possible or to audit data location compliance in cloud computing.

No goal exists in a normative vacuum, and the problem improvement goal in turn often supports some higher-level stakeholder goals. There may be a range of different external stakeholder goals all served by the project improvement goal. For example, the parent's goal may be to keep children in the backseat of a car quiet, the children's goal is to watch TV in a car, and the car manufacturer's goal is to increase sales.

Market-driven projects have a very clear goal hierarchy. Exploratory projects may have a more fuzzy goal hierarchy where the higher-level goals are speculative or may even be absent:

☐ The DOA project is market driven. Starting from the bottom up in Fig. 2.1, the lowest level goal was to build simulations and prototypes of DOA algorithms and of an antenna array. This is an *instrument design goal*. These instruments were used to answer *knowledge questions* about the performance of different DOA algorithms—a knowledge goal. This knowledge was generalizable and could be used to *predict* the performance of all implementations of the algorithm—another knowledge goal. Answering these questions also contributed to the *artifact design goal* of designing a DOA estimation component. This in turn contributes the goal of *problem context improvement*. The DOA estimation component will be part of a directional antenna for satellite TV signal reception, which is to be used in a car to allow passengers on the backseat to watch TV. The *sponsor's goal* is to develop and sell components of the IT infrastructure needed for this.

☐ As an example of an exploratory project with only knowledge goals, a project that we will call ARE (for Agile Requirements Engineering) studied how requirements were prioritized in agile software engineering projects [1]. This is a knowledge goal that was achieved by answering *knowledge questions* about a sample of projects. Achieving this goal enabled another knowledge goal, namely, to *predict* how requirements were prioritized in similar projects. There was no artifact design goal, although the results would be potentially useful to improve requirements engineering in agile projects.

2.2 Design Problems

Goals define problems. How do we get from here to the goal? A **design problem** is a problem to (re)design an artifact so that it better contributes to the achievement of some goal. Fixing the goal for which we work puts us at some level in the goal hierarchy discussed in the previous section. An instrument design goal is the problem to design an instrument that will help us answer a knowledge question, and an artifact design goal is the problem to design an artifact that will improve a problem context.

Design problems assume a context and stakeholder goals and call for an artifact such that the interactions of (artifact $\times$ context) help stakeholders to achieve their goals. We specify requirements for the artifact that are motivated by the stakeholder goals. This gives us the schema for expressing design problems shown in Table 2.1.

Table 2.1 Template for design problems (aka technical research questions). Not all parts to be filled in may be clear at the start of the project

- Improve <a problem context>
- by <(re)designing an artifact>
- that satisfies <some requirements>
- in order to <help stakeholders achieve some goals>.

We discuss the role of stakeholder goals, requirements, and the problem context in more detail later on. Here, I give some illustrations only:

☐ The DOA design problem has this format:

  – Improve satellite TV reception in cars
  – by designing a DOA estimation algorithm
  – that satisfies accuracy and speed requirements
  – so that passengers can watch TV in the car.

  At the start of the project, the requirements on the algorithms were not known yet.

☐ In a project that we will call MARP (multi-agent route planning), Ter Mors [2] designed and investigated multi-agent route planning algorithms for aircraft taxiing on airports. The design problem was to:

  – Improve taxi route planning of aircraft on airports
  – by designing multi-agent route planning algorithms
  – that reduces taxiing delays
  – in order to increase passenger comfort and further reduce airplane turnaround time.

  This was an exploratory project where the interest of the researcher was to explore the possibility of multi-agent route planning. The aircraft taxiing was a hypothetical application scenario used to motivate the research and used as an example in simulations.

Not all elements of the design problem template may be known at the start of the project, and some may be invented as part of a hypothetical application scenario. Stating your design problem according to the template is useful because it helps you to identify missing pieces of information that are needed to bound your research problem. Table 2.2 lists some heuristics by which the elements of a design problem can be found.

We can now see what is the problem with masquerading a design problem as a knowledge question. Take the following knowledge question:

- "What is an accurate algorithm for recognizing direction of arrival?"

This is really a design problem. Using the template, we see what is missing:

- Improve <a problem context>
- by designing a DOA estimation algorithm
- that satisfies accuracy requirement
- so that <stakeholder goals>.

Table 2.2 Guidelines for filling in missing parts of the design problem statement template

• What must be designed by the researcher?	⟶ The artifact
• What is given to the researcher? • With what will the artifact interact?	⟶ The problem context
• What is the interaction? • What desired properties must it have?	⟶ The requirements
• To whom should this interaction be useful? • To achieve which of their goals?	⟶ The stakeholder goals

The problem context and stakeholder goals are missing, so that we have no clue about the required accuracy and miss one important requirement, namely, execution speed. We also miss the information needed to set up a test environment.

Many researchers do not want to be perceived as solving "mere" design problems and insist on stating their research problem as a question, with a question mark. The following template does that:

- How to <(re)design an artifact>
- that satisfies <requirements>
- so that <stakeholder goals can be achieved>
- in <problem context>?

It contains exactly the same information as our design problem template. Instead of calling it a design problem, we may now call it a "technical research question." However, I have reserved the word "question" for knowledge questions. If you want to give design problems a more dignified status, I propose to use the term **technical research problem**.

2.3 Knowledge Questions

The knowledge goals of a project should be refined into knowledge questions. A knowledge question asks for knowledge about the world, without calling for an improvement of the world. All knowledge questions in this book are **empirical knowledge questions**, which require data about the world to answer them. This stands in contrast to **analytical knowledge questions**, which can be answered by conceptual analysis, such as mathematics or logic, without collecting data about the world. Analytical knowledge questions are questions about the conceptual frameworks that we can use to structure our descriptions of the world. To answer an analytical knowledge question, we analyze concepts. But to answer an empirical knowledge question, we need to collect and analyze data. There are several ways to classify empirical knowledge questions, discussed next.

2.3.1 Descriptive and Explanatory Questions

One important classification of knowledge questions is by their knowledge goal: description or explanation. **Descriptive questions** ask for what happened without asking for explanations. They are journalistic questions, asking *what* events were observed, *when* and *where* they happened, *who* was involved, *which* devices were affected, etc. Imagine yourself a journalist at the scene of the happening. Your goal is not to judge nor to explain, but to just observe without prejudice.

Explanatory questions ask *why* something happened. We will distinguish three sorts of why questions:

- “What event *caused* this event?” Here we ask which earlier event made a difference to a current event:

  ☐ For example, if a program crashes, we may ask which input caused this crash. This means that we ask which input *made a difference* to the behavior of the program. It also means that we assume that with another input, the program might not have crashed.

- “What mechanism *produced* the event?” A mechanism is an interaction between system components, and here we ask what system components interacted to produce the event:

  ☐ For example, if we have identified the input that caused a program to crash, we can trace this input through the program to find the component (procedure, function, statement, etc.) that failed to respond properly to its input. We may be able to eliminate the failure mechanism by repairing the defective component or by replacing it with another one.

- “What are the *reasons* these people or organizations did that?” Biological and legal persons have goals and desires that motivate their actions, and we can explain their behavior by indicating these motivations.
  Reasons contain an element of choice, and we hold people and organizations responsible for actions that they performed for a reason. This is not the case for causes:

  ☐ For example, someone may push you in a swimming pool. That push is the cause of your being in the pool, but it is not your reason for being in the pool. You had no choice. You had no reason to jump in, and you are not responsible for being in the pool.
  ☐ If consultants refuse to use a method because it requires them to change their way of working, then we hold them responsible for this, because they could have chosen otherwise.
  ☐ A consultant may use a method incorrectly because he or she does not understand the method. Misunderstanding is the cause of incorrect use, not the reason. Given the misunderstanding, the consultant had no choice, desire, or goal to use the method incorrectly.

2.3.2 An Aside: Prediction Problems

Descriptive and explanatory questions ask what has happened and how this came to be. But what about the future? Can we have a knowledge question that asks what will happen in the future? We are asking questions like this all the time. For example,

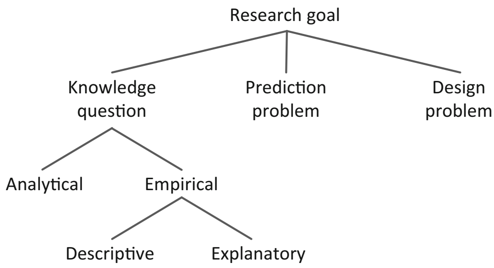

Fig. 2.2 A classification of research goals

what will be the average satisfaction of users of this system? How accurate will this algorithm be when used in a car?

However, these are not knowledge questions but **prediction problems**. A prediction is a belief about what will happen, and this belief does not constitute knowledge. We cannot *know* the future. There is no time travel: we cannot peek at a future event and then return to the present to answer our question. Instead, we must wait and see what happens.

But we can try to predict the future by using our knowledge of what has happened so far and generalizing from this:

□ If system X is going to be implemented in organization A next month, we may ask what percentage of the users will be satisfied. This is a prediction problem. We have no knowledge about this percentage yet.
However, we can ask another question, namely, what percentage of users of X are satisfied with the system in organizations where X has been implemented. This is a descriptive knowledge question.
After empirical research, we find that in a sample of 30 organizations where implementation of X has been attempted, on the average, 80 % of the users are satisfied and give or take 5 %. This describes a fact.
Next, we can generalize: In organizations where implementation of X is attempted, on the average, 80 % of the users are satisfied and give or take 5 %. If sampling has been done in a statistically sound way, then this generalization has statistical support in the above fact. If we can explain it in terms of properties of X and of the users, then it has additional support. In the absence of these kinds of support, it is an informed guess based on the fact reported above.
Whatever the degree of support is, we can use the generalization to make a prediction: In the next organization where X will be implemented, on the average, 80 % of the users will be satisfied and give or take 5 %. The degree of support for this prediction depends on the degree of support for the generalization, and on the similarity of the next organization to the past organizations. In any case, we do not know whether the prediction is correct. In the future, we will know whether it is true.

Knowledge is created by answering knowledge questions, and scientific theories are created by generalizing from this. These generalizations can be used to solve prediction problems. We discuss ways to generalize from empirical research in Parts IV and V.

This gives us the classification of research goals shown in Fig. 2.2. Knowledge questions ask about the past and present, prediction problems ask about the future, and design problems ask for a change of the future. This book is about answering empirical knowledge questions and treating design problems. There are additional classifications of knowledge questions, treated next.

Table 2.3 Examples of empirical knowledge questions

	Descriptive questions	Explanatory questions
Open questions	• What is the execution time in this kind of context?	• What input causes the dip in the graph of recall against database size? • Is there a mechanism in the algorithm that is responsible for this?
	• What do the consultants think of the usability of this method for advising their clients?	• Why do these consultants have these opinions about usability? What reasons do they have? • How is this related to context of use? Can we find a social or psychological mechanism for this?
Closed questions	• Is the execution time of one iteration less than 7.7 ms?	• Why is the execution time of the method in these test data more than 7.7 ms? • Is this loop responsible for the high execution time?
	• Do the consultants think method A is more usable than method B in this context?	• Do consultants prefer method A over method B because method A resembles their current way of working more than method B does?

2.3.3 Open and Closed Questions

A second way to classify knowledge questions is by the range of possible answers that is prespecified. An **open question** contains no specification of its possible answers. It is exploratory. A **closed question** contains hypotheses about its possible answers.

This gives us in total four kinds of empirical knowledge questions. Table 2.3 lists some examples. Note that in research that uses statistical inference, closed descriptive questions are often stated as positive hypotheses, to be confirmed or falsified by empirical observations:

☐ Instead of the closed descriptive question "Do consultants prefer method A over method B?," we may state the following hypothesis about a population of consultants:

- Consultants prefer method A over method B.

This hypothesis is then tested on a sample of consultants. The data may provide support for or against this hypothesis.

Table 2.4 Four important kinds of knowledge questions about designs, with variations

Effect questions: (artifact $\times$ context) produce effects?

- What effects are produced by the interaction between the artifact and context?
- How does the artifact respond to stimuli?
- What performance does it have in this context? (Different variables)

Trade-off questions: (alternative artifact $\times$ context) produce effects?

- What effects do similar artifacts have in this context?
- How does the artifact perform in this context compared to similar artifacts?
- How do different versions of the same artifact perform in this context?

Sensitivity questions: (artifact $\times$ alternative context) produce effects?

- What effects are produced by the artifact in different contexts?
- What happens if the context becomes bigger/smaller?
- What assumptions does the design of the artifact make about its context?

Requirements satisfaction questions: Do effects satisfy requirements?

- Does the stimulus-response behavior satisfy functional requirements?
- Does the performance satisfy nonfunctional requirements?

2.3.4 Effect, Trade-Off, and Sensitivity Questions

The above two classifications of empirical knowledge questions are not restricted to design science research and are usable in any kind of empirical research. But the following classification is specific to design science research, because it classifies empirical knowledge questions according to subject matter. What is the question about?

The subject of design science is an artifact in context, and hence design science research questions can be about artifacts, their properties, their context, stakeholders and their goals, etc. Among all these possible questions, we single out four that are asked in virtually every design science research project. They are listed, with variations, in Table 2.4.

Effect questions ask what effect an artifact in a context has. The generic effect question is:

- What effects are produced by the interaction between artifact and context?

Trade-off questions ask what is the difference between effects of different artifacts in the same context, and **sensitivity questions** ask what is the difference between effects of the same artifact in different contexts. **Requirements satisfaction questions**, finally, ask whether the effects satisfy requirements. Requirements satisfaction is a matter of degree, and different requirements may be satisfied to a different degree or may even be violated to some degree where others are satisfied to some degree:

☐ The DOA project has the following knowledge questions:

Q1 (Effect) What is the execution time of the DOA algorithm?
Q2 (Requirements satisfaction) Is the accuracy better than 1°?
Q3 (Trade-off) How do the MUSIC and ESPRIT algorithms compare on the above two questions?
Q4 (Sensitivity) How do the answers to the above questions vary with car speed? With noise level?

☐ Here are three knowledge questions from the DLC project:

Q1 (Effect) What is the usability (effort to learn, effort to use) of the data compliance checking method? Why?
Q2 (Trade-off) Which parts of the proposed method can be omitted with the remaining part still being useful?
Q3 (Sensitivity) What assumptions does the method make about consultants, e.g., experience, required knowledge, and competence?

2.4 Summary

- Different stakeholders in a design science research project may have different kinds of goals. Researchers are usually at least driven by curiosity and fun and may be driven by utility too. Sponsors are usually driven by utility and constrained by budgets but may occasionally allow researchers to do exploratory research.
- Each design science research project has a goal tree containing design goals and knowledge goals. There is always a knowledge goal, and usually there are design goals too.
- A knowledge goal can be related to other research goals and questions in several ways:
  - A knowledge goal can be refined into knowledge questions. These express the same goal but in a more detailed way. Knowledge questions are descriptive or explanatory, they can be open or closed, and they may be effect, trade-off, sensitivity, and requirements satisfaction questions.
  - A knowledge goal may contribute to the ability to solve prediction problems.
  - A knowledge goal may be decomposed into lower-level instrument design goals. These are lower-level design goals that help you to achieve your knowledge goal.
  - A knowledge goal may contribute to an artifact design goals (aka technical research goals), which in turn may contribute to some improvement goal in the context, which in turn may contribute to some stakeholder goals. In exploratory research, some of these goals may be absent.
- A prediction problem is a problem to predict what phenomena will occur in the future. It is answered by applying a theoretical generalization. For example, we may use a design theory to predict what would happen if a treatment would

be implemented or to predict what would happen if a problem would remain untreated.

- A design problem is a problem to (re)design an artifact so that it better contributes to the achievement of some goal. The template for design problems relates the artifact and its requirements to the stakeholders and their goals. Some of this information may be missing at the start of a project or may be speculative.

References

1. Z. Racheva, M. Daneva, A. Herrmann, K. Sikkel, R.J. Wieringa, Do we know enough about requirements prioritization in agile projects: insights from a case study, in *18th International IEEE Requirements Engineering Conference, Sydney* (IEEE Computer Society, Los Alamitos, October 2010), pp. 147–156
2. A.W. ter Mors, The World according to MARP. Ph.D. thesis, Delft University of Technology, March 2010. http://www.st.ewi.tudelft.nl/~adriaan/pubs/terMorsPhDthesis.pdf

http://www.springer.com/978-3-662-43838-1

Design Science Methodology for Information Systems
and Software Engineering
Wieringa, R.J.
2014, XV, 332 p. 43 illus., Hardcover
ISBN: 978-3-662-43838-1