The focus of this appendix is on the methods used to obtain a better understanding of design through empirical studies, as in the DS-I and DS-II stages. As a researcher embarking on empirical studies, it is necessary to understand that methods are often based on particular assumptions about how to execute research. Section A.1 contains a short discussion about paradigms and assumptions. In Section A.2 the checklist for reviewing empirical studies discussed in Section 4.4.2, is described in more detail and closes with an example in Section A.2.21. Because of its relevance, the differences between empirical studies in laboratory and industrial environments are described in Section A.3. An important characteristic of empirical studies is the method by which data is collected. Section A.4 provides short descriptions of some of the main data-collection methods, based on the literature and our own experiences. Some basics about statistics are given in Section A.5.

In all cases, further literature and experts in the field have to be consulted when setting up an empirical study. This appendix is only an introduction, intended to raise awareness of the issues.

A.1 Paradigms and Assumptions

In design research a variety of topics can be studied for a variety of reasons with a variety of methods. The research approach and methods applied should be chosen such that they are suitable for the topic, the aim and the existing understanding, and that they result in valid statements. For that purpose, methodologies for research have been (and are still being) developed in and across disciplines based on what is viewed as valid research. Design research has many facets as shown in the first figure in this book (Figure 1.1), representing various disciplines. Some disciplines have well-established research methodologies, others have several, sometimes conflicting approaches causing heated discussions about which approach is more likely to produce the best, most valid understanding. This is clearly visible in the debates around quantitative versus qualitative research, and theory-driven versus data-driven research (see also Section 4.1).
As a researcher, it is important to be aware of the fact that:

- different schools of thought exist based on underlying paradigms;
- every school has (or should have) a consistent methodology that links the problem to the methods applied and the ways of validation; and
- every methodology has certain premises.

Furthermore, multiple standpoints are found with ‘followers’ of well-established approaches, and taken-for-granted categories and methods of data collection have become problematic (Coffey et al. 1996). As discussed in the main text of this book (Section 4.1), paradigms do change over time and new ones emerge and competing paradigms may exist simultaneously, specifically in immature sciences (Kuhn 1970). For example, the view that it is not possible to be completely objective while observing (which was the traditional scientific pre-requisite) has become more and more accepted in all sciences. The so-called givenness of data, whether numerical or not, is questioned. That is, the view that sources of data can be treated as independent of, and as imposing themselves on, the researcher, is generally rejected (Hammersley and Gomm 1997).

Independent of a particular paradigm, groups of researchers or individuals are likely to have particular views (assumptions) about the topic of investigation – even if this is just the belief that a particular factor is relevant. This influences the interpretation of the findings. These assumptions can hardly be avoided, but should be made explicit, and an attempt should be made to consciously use alternative views to find alternative explanations.

A.1.1 Paradigms

The differences in research approaches and methodologies are due to the underlying paradigms (also called worldviews or belief systems), that are used by a domain or by groups of researchers within a domain, as the accepted perspective at a given time. The paradigms express the basic assumptions upon which the research is built and should therefore be known when applying the related approach or its methods.

To illustrate this, we give two examples of particular views shared by a larger research community and both referring to the role of experience. The philosophical theory of empiricism has been the basis for several paradigms. According to Bullock et al. (1988) empiricism states “(1) that all concepts are derived from experience, i.e., that a linguistic expression can be significant only if it is associated by rule with something that can be experienced, and (2) that all statements claiming to express knowledge depend for their justification on experience”. Empiricists assume that this gives “access to the neutral or unalloyed access to a realm of pure ‘facts’” (Bullock et al. 1988). The focus of empiricism is on objectivity. “Any statement of the empiricist theory, to be consistent with itself, must be empirical or, if not, analytical. An empirical basis for the theory is provided by elementary facts about the way in which the meaning of words is learned.” (Bullock et al. 1988).

A different view, also referring to the importance of experience, but resulting in a very different research approach and methods, is ethnomethodology: personal experience of a situation is considered crucial to developing knowledge about the
situation. The focus is on “how the individual experiences and makes sense of social interaction” (Bullock et al. 1988). To obtain understanding it is important to immerse oneself in a situation; to live it. This refers to a meaning of experience quite different from that from an empiricist view: namely referring to a more subjective view. The focus is on details of everyday life, in particular on details of the practices through which action and interaction are accomplished (Button and Dourish 1996). The determination of what is going on and what is appropriate is considered to be the outcome of local, occasioned and situated action (Anderson 1996).

These examples show that world views are not necessarily applicable to all situations. Ethnomethodology, e.g., would not be suitable to gain understanding about the reliability of products from a technical point of view. However, to understand the reliability of products from the point of user experience, this approach could be very suitable.

Paradigms might also specify the main concepts used. Cook and Campbell (1979), e.g., present some of the different views on the concept of cause.

Even if one is not aware of such paradigms, or does not wish to follow a particular paradigm, one’s view on how to do research, on the suitability of methods and on the topics to be studied, is coloured by beliefs; beliefs about the right, scientific way of obtaining understanding. Despite the best intentions, no researcher is free from ‘coloured glasses’. The least researchers should do is to acknowledge this fact and make deliberate attempts to make their views, assumptions and beliefs explicit. This will improve the quality of one’s own research and help others to decide whether or not to take up a certain view and resulting approaches.

“A world view has a profound interaction with the research questions we choose to investigate. Some questions are interesting or even meaningful only within a particular world view, and some questions limit our horizon to seek alternative world views” (Reich 1994). Cook and Campbell, e.g., state in their book on quasi-experimentation that they adopt “a critical-realist perspective, positing that causal connections are ‘real’ but imperfectly perceived, and particularly stress epistemological theories that restrict the analysis of causation to the analysis of manipulable causes – factors that can be varied ‘at will’” (Cook and Campbell 1979). Denzin states in his book on social methods that he “offers a symbolic-interactionist view of sociological theory and research methodology …… which stresses the self-reflexive nature of everyday and scientific conduct” (Denzin 1978). Patton (1990) describes his particular view on the most suitable approach to research as follows: “the qualitative-naturalistic-formative approach is especially appropriate for programmes that are developing, innovative or changing, where the focus is on programme improvement”. These views not only affect the research questions, but also the choice and intended application of the methods they propose.

---

21 Epistemology is the philosophical theory of knowledge, which seeks to define it, distinguish its principal varieties, identify its sources, and establish its limits (Bullock et al. 1988)
Many researchers consider the combination of different paradigms impossible, as many debates about paradigms show. Research groups tend to hold on strongly to the paradigm of their research field or even their own group. Some researchers advocate the combination of different paradigms (which in itself is a paradigm!). Tashakkori and Teddlie (1998), e.g., propose “a pragmatic view, combining the quantitative positivist paradigm with the constructivist qualitative paradigm” – which they consider compatible – as the basis of a mixed methodology to be applied across all phases of the research process. In their view, methods should be chosen such that they are most appropriate for the research question. This question should predominate over the paradigm. This very much reflects our view.

As a design researcher it is not necessary to join in the debates in the various disciplines about the best methodology, but it is important to read multiple sources before choosing a particular approach to ensure that the data obtained using this approach is valid for the purpose intended and that pitfalls that make data invalid are avoided. Also in the area of design research, certain approaches, methods, types of questions and issues will be considered particularly relevant. Here too, one should be aware that – even if not explicitly stated – relevance strongly depends on the paradigm used by the researcher, the particular facet of design studied, and the research questions and hypotheses addressed.

In design research one can observe at least two views of design that play a role in the research approach taken: a process view and a product view (the first two project descriptions in Appendix C of this book provide an example of each. These views are often considered opposite and mutually exclusive. A process view assumes that the process (the context in which the product is being created) determines the product, and therefore that a better understanding of the process enables process improvement that will then lead to more successful products. The product view can be described as assuming that the product determines the process. Better understanding of the product will make it possible to determine the most suitable process for generating more successful products. We argue that our methodology is applicable in both cases.

A.1.2 Assumptions

To clarify one’s own view and assumptions, one can search for and look into existing theories (i.e., views of reality) that seem appropriate to one’s own research, discuss ideas and approaches with others, and ask the ‘why’ question: ‘Why do I think this is the case?’ ‘Why do I think this is relevant?’ Researchers have to be reflective and self-critical about their approach as a whole, and about each individual step. That is, apart from paradigms, or world views, on the research approach, one also has views (assumptions) on the topic of interest. For example, in interpreting findings, a researcher should consciously try to generate alternative, or rival explanations, in the same way as designers are encouraged to generate alternative concepts for the task they are given. It has to be said that it is unlikely that all assumptions can be made explicit, and many may not become apparent until much later in the research process, for instance while generating these alternative explanations.
A simple example of how certain assumptions might influence our interpretation of findings is the following. Assume that, in our view, designers identify themselves with the product they are developing – it is ‘their child’ – and, as a consequence, always strive to generate the best solution and tend to experience constraints such as time as impeding their process. In a study we collect evidence that those designers who complain about time pressure also produce products of low quality. Based on our assumption, a plausible conclusion would be that shortage of time relates to or even causes low-quality products. Another possible explanation is that the designers were not motivated and used time constraints as an excuse, but this does not fit our assumption. This second explanation would, however, have been plausible had the basic assumption been that employees – and the designers in the study were employees – tend to consider their work as an activity to earn money, *i.e.*, do not have an intrinsic motivation nor identify themselves with their work. A further possible explanation is that another factor played a role, namely inexperience. If designers are too inexperienced for the type of product they have to develop, the quality might be low and they might have problems with time constraints. This would be based on the assumption that designers are not necessarily experienced.

It is therefore crucial to identify and make explicit any assumptions one has made during any of the stages of the research.

### A.2 Reviewing Empirical Studies

In Section 4.4.2, the checklist shown in Table A.1 was proposed for reviewing descriptive studies.

*Table A.1* Checklist for determining the characteristics of empirical studies (Table 4.1) (adapted from Blessing (1994))

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim, research questions, hypotheses</td>
<td>The aim of the research project and of the study, main research questions and hypotheses, Success Criteria and/or Measured Success Criteria and possible constraints.</td>
</tr>
<tr>
<td>Nature of the study</td>
<td>Observational or interventional, comparative or non-comparative, <em>i.e.</em>, whether the study involved intervention in the design process by the researcher.</td>
</tr>
<tr>
<td>Theoretical basis</td>
<td>Paradigms, methodologies, theories, views, assumptions, <em>etc.</em>, that guided the researchers.</td>
</tr>
<tr>
<td>Unit(s) of analysis</td>
<td>The element(s) for which findings are reported and about which to draw conclusions that are intended to be generalised.</td>
</tr>
<tr>
<td>Data-collection method</td>
<td>The method(s) used, such as direct observation using video, participant observation, diary keeping, archival research, questionnaire, interview.</td>
</tr>
</tbody>
</table>
Table A.1 (continued)

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Role of researcher</td>
<td>Involvement of the researcher in the research process.</td>
<td></td>
</tr>
<tr>
<td>Time constraint</td>
<td>Time constraints imposed by the researcher, e.g., available design time, available time to answer a questionnaire, time of the observation (when the phenomena observed lasts longer).</td>
<td></td>
</tr>
<tr>
<td>Continuation</td>
<td>Continuous data collection or sampling</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>Length of the part of the process studied and length of the whole process (note that these can be different).</td>
<td></td>
</tr>
<tr>
<td>Observed process</td>
<td>Starting point and required deliverables of the observed process: e.g., specification as starting point, layout drawing, prototype or product as deliverable.</td>
<td></td>
</tr>
<tr>
<td>Setting</td>
<td>Location of the study, including whether the setting was contrived or natural.</td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Type and complexity of task. Nature of the observed tasks: real, realistic or artificial.</td>
<td></td>
</tr>
<tr>
<td>Number of cases</td>
<td>Number of data sets collected, e.g., the number of experiments, interviews, observed groups.</td>
<td></td>
</tr>
<tr>
<td>Case size</td>
<td>Number of persons, product elements, employees, etc., within each case.</td>
<td></td>
</tr>
<tr>
<td>Participants</td>
<td>Level and type of experience, background, size of organisation, etc.</td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>Description of the design object, company, project or documents analysed</td>
<td></td>
</tr>
<tr>
<td>Coding and analysis method(s)</td>
<td>Methods used to process, code and analyse the data, e.g., use of pre-determined coding schemes or not, and statistics applied.</td>
<td></td>
</tr>
<tr>
<td>Verification method(s)</td>
<td>Methods used to verify the results.</td>
<td></td>
</tr>
<tr>
<td>Findings</td>
<td>Main statements, models, conclusions resulting from the study.</td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td>Anything remarkable or important in the publication that is not covered by the other dimensions. Missing information, relevance for one’s own project, etc.</td>
<td></td>
</tr>
</tbody>
</table>

The dimensions in this table offer various options; those listed in the table are not exhaustive. None of the options can be said to be right or wrong, or providing stronger or weaker statements, as this depends on the research questions and hypotheses addressed with the particular study, and on the combination of options that are chosen. Note that the dimensions are not independent and the specifications can overlap. Furthermore, not all categories may apply to each type of study.
Appendix A: Descriptive Study Methods

If the checklist is used to set up an empirical study, it is important to be creative in specifying the options; the options and their realisation have to be adapted to the aims of the study, without compromising the validity of the study and its results. To illustrate the use of the checklist an example is given in Section A.2.21. For an overview of empirical studies using earlier versions of the checklist, see Blessing (1994) reviewing 66 studies and Dwarakanath et al. (1995) reviewing 90 studies.

The following sections provide a short explanation of each of the dimensions in this checklist.

A.2.1 Aim, Research Questions, Hypotheses

This dimension lists the aim(s) and objectives of the research project and of the empirical study itself, as well as the main research questions and hypotheses. If Success Criteria and/or Measured Success Criteria have been defined, these can be added here, as well as the main constraints relevant for the study.

A.2.2 Nature of the Study

The nature of the study reflects the two types of empirical studies in DRM: DS-I and DS-II. A distinction is made between observational and interventional research, each of which can be comparative. Observational research here does not refer to observation as a research method, but to a hands-off approach by the researcher. Observational research is typical for DS-I. The researcher does not intervene, e.g., by introducing design support. The objective is to identify the influencing factors and their links by studying design ‘as is’. Methods and setting used for empirical studies will often influence what is observed, but that is not considered to be an intervention in the sense used here. The method and setting should be chosen such that the effects are minimal or not relevant, or that the effects can be taken into account in the interpretation of the findings. Interventional research deliberately influences the existing situation in order to investigate the effects of this intervention. This type of research is typical for the evaluation studies in DS-II.

A study is called comparative when the cases are divided into groups with different characteristics, e.g., different tasks, different settings, or different subject backgrounds, before data has been collected. The objective is to find the effect of these differences so as to better identify the influencing factors and their links. In the case of interventional research, a comparative study might divide the cases into those that use the support (the so-called experimental group) and those that do not (the control group) to better identify the effects of the support (see also Section A.4.3). A comparison can also involve the same users, comparing before and after the use of a particular support. The latter, however, is not very common in design research because of the difficulty to compare the situation before and after a design problem has been solved.

A.2.3 Theoretical Basis

Any paradigm, methodology, theory, etc., that guided the researcher, as well as any relevant views and assumptions have to be made explicit. Unless explicitly
mentioned, it may be difficult to determine these from research papers, other than indirectly through the research approach and methods that were used.

### A.2.4 Unit of Analysis

A central concept in research is the *unit of analysis*. It is the main element of the study about which the researcher wants to obtain information, wishes to draw conclusions and make generalisations, *i.e.*, the unit on which the analysis focuses (hence the name). The chosen unit of analysis should allow independent observations. The product can for example be the unit of analysis, if the researcher aims to identify the factors that contribute to its attribute ‘quality’ or its attribute ‘reliability’. The designer would be the unit of analysis, if the aim is to draw conclusions about differences between designers, *e.g.*, how they approach a design problem. The design team would be the unit of analysis, if we aim to draw conclusions about the behaviour of design teams, for example the way in which team members collaborate. Note that in the last example, data is collected at the individual level, *i.e.*, the *unit of measurement* (or *unit of data collection*) is the individual. This data is aggregated and analysed at the team level. Hence, the unit of *analysis* is the team. The unit of measurement can thus differ from the unit of analysis. The unit of measurement relates to data collection, in this case the individual; the unit of analysis relates to data analysis, in this case the team about which we want to draw conclusions.

There are no limits to what can be the unit of analysis, but once chosen, this affects the other dimensions of the study, such as the most suitable data-collection methods, the nature of the study, *etc*. The unit of analysis also determines which conclusions can be drawn and should thus be chosen carefully. For example, the fact that a design team has generated many solutions does not imply that each designer in the team has generated many solutions. That is, the conclusions drawn from an analysis at team level may not apply at the individual level and *vice versa* (doing so is referred to as *ecological fallacy*).

The unit of analysis has to be chosen such that the research questions can be answered and the hypotheses verified with as little interpretation as possible. Units of analysis used in design research are manifold and include: design team, requirements, product module, design process, decision making, human–machine interfaces, information exchange, collaboration, documentation, and organisational strategy.

### A.2.5 Data-collection Method

Literature provides a variety of methods for data collection and recording, each with its specifics and limitations, *e.g.*, direct observation, participant observation, thinking aloud, introspection, diary keeping, document/drawing analysis, questionnaire, interview, product analysis. Section A.4 discusses the most common ones. In most studies a combination of methods is used in parallel or sequence, *e.g.*, a questionnaire about the participants’ background, followed by the observation of a task using video and note taking, and ending with a reflective interview. Drawing
up a matrix showing the research questions and hypotheses against the methods used, as shown in Figure 4.6, can be useful.

A.2.6 Role of Researcher

The way in which the researcher is involved in an empirical study can influence the outcome, even in pure observational studies, and should thus be specified in detail.

For example, in a study in which video cameras are used to observe individual designers designing a product against a particular specification, the role of the researcher is essentially that of an observer. However, experience shows that designers tend to ask questions about the problem, about the background of the assignment, etc. A decision has to be taken as to how to deal with such a situation: not giving an answer may block the process and take away some of the reality (in practice designers do ask others), whereas giving an answer will influence the course of the process, but will be closer to reality. More importantly, answering will provide the particular designer(s) with information that the other designers do not have. One solution is to write down, prior to the first observation, all possible questions one could think of by trying to anticipate what designers would wish to know when designing this product, and to formulate an answer for each question. When a question is asked that appears in the list, the answer is read out without rephrasing, explaining or adding. If the list does not show the question, the researcher tells the participant that he or she does not know the answer. Importantly, a note is made about when which questions were asked, as this may be important in the analysis and interpretation of the data. In this way, all participants will be able to get the same information. Obviously, if the same question is asked over and over again, it can be useful to inform each following participant.

Another method, in which the researcher plays a clear role, is participant observation. Here too, it is important to specify the details of the role of the participant as the following example shows. Hales, in the example used in Section A.2.21, was the main consultant designer in the project he observed. He was continuously involved in most of the issues. This allowed insight he would not have been able to obtain had he only been in a supporting role dealing with a particular design issue, such as was the case in the industrial study undertaken by the first author (Blessing 1994). The specifics of the role thus influence the results and conclusions that can be drawn.

The role of the researcher at the start of the study can also have an influence. It is important to think about and make explicit the way in which participants are contacted and informed, for instance by asking questions such as: Are the participants aware of the aims of the study? Has the information been provided to all participants in an identical manner? Has the contact between the researcher and participants been direct or has, e.g., the company selected the persons? Instructions should be in writing, irrespective of whether the instructions are handed out or read out aloud by the researcher.
A.2.7 Time Constraint

This dimension provides information about any time constraints that were imposed by the researcher. This includes times given to designers to solve a particular problem or to interviewees to answer the questions, as well as the time period during which data about product or process was collected when the phenomena studied covered a longer period.

Any time constraint can influence the process and the results. For example, the last questions on long questionnaires often have no or few answers, or are answered in a hurry, without much thought. As a result, parts of the questionnaire might not be useful. Limiting the time used to analyse the consequences of new features in a product, may only allow detection of failures that occur early on in the product lifecycle.

When the period in which data can be collected is constrained in the sense that it is shorter than the period in which the phenomena observed, it has to be clearly specified what is covered (see also Section A.2.9).

A.2.8 Continuation

This dimension clarifies whether data was collected continuously or whether sampling took place. For example, if data is sampled at certain intervals, the intermediate periods are not covered and it may be difficult to draw conclusions about the whole period (additional methods may be used to fill these gaps). This dimension is also relevant when the process is in principle observed continuously, but involves breaks that are not captured. Breaks, whether in observation or interviewing, allow participants to reflect, to meet others, etc. When observing designers at work, the design process is likely to continue during the breaks (‘I got that idea when I was taking a shower/met my colleague over lunch/saw the vending machine when I got a coffee’). Whether it is problematic or not if this is not captured, depends on the aims and specific research questions of the study. If discontinuity is problematic, observation of shorter processes, a different setup (e.g., lunch in the room), or the addition of interviews after the breaks, are possible solutions.

A.2.9 Duration

The duration is the length of the process or project analysed in terms of time. A difference may exist between the length actually observed and the total length, e.g., due to breaks, in which case both have to be specified.

A.2.10 Observed Process

This dimension mainly applies to observational or archival studies. In order to determine what the results of a study relate to, it is important to know what the starting point (input) and required deliverables (output) of the analysed processes are. In an artificial environment, a requirements list could be the input and a layout drawing requested as output. In a real environment, when only part of an ongoing
Appendix A: Descriptive Study Methods

A process can be studied, a distinction can be made between the input and output of the actual process and those of the observed process.

A.2.11 Setting

This dimension specifies the setting in which the study takes place. This includes the location (the actual physical environment), as well as whether the setting is contrived or natural. In a survey, for example, the answers and their length may differ, depending on whether people are interviewed on the street or at home. The effects of a laboratory or practical setting are manifold, as discussed in Section A.3.

Studies that take place in laboratory environments are often contrived, but not necessarily so. Similarly, a study in which designers are observed within their company is a study in an industrial, but not necessarily natural setting. For example, if designers are observed in their company, but in a room in which they do not normally work and without access to the materials and people in their normal environment, this setting can be considered contrived, if this setting may affect the conclusions that can be drawn from the study. The setting has therefore to be chosen carefully: it should either have no influence, or the influence should be known such that it can be ‘filtered’ out or taken into account when analysing the data and drawing the conclusions.

A.2.12 Task

In many cases it is important to specify the task(s) to which the data refer. In an observation in practice, which will often concentrate on few people, it is important to know their particular tasks within a project; for example, if the persons observed are the analysis experts, one should not be surprised that the findings show an emphasis on analysis activities. The particular task and related materials determine what can be expected and what generalisations can be made. The task also plays a role in archival analysis of product or process documents, as these often only give insight into the results of particular tasks or specific views: it might, e.g., be difficult to find alternatives that were considered.

The complexity of the task can also be relevant. In a contrived setting, indicators might be the number of hours required to complete the task and the knowledge and skills required. In a natural setting, the indicators may be: the number of hours and people involved; the number and types of disciplines involved; the novelty of the product, etc. The task should be compatible with the aims of the study and – in particular when the task is given – with the participants involved.

Furthermore, it is important to determine whether the task is real, realistic or artificial. A real task is one in which the participants are normally involved and is the basis for direct observation in a natural setting, for archival analysis, and for interviews and questionnaires. In our reliability example, studying how designers assess reliability while working in their own project represents a real task. When the researcher provides a task, introduces constraints, or changes the setting – for example to obtain data sets that are more easily comparable or to reduce the length of the process – the task can be called realistic, provided that the task is a derivative
from a real task. In our reliability example, a task could have been devised that is similar but more focused than the one on which the designers normally work. A task would be artificial, when such a task does not occur at all in practice or not in this form. In our reliability example, this would be the case when the designers are given various product drawings and asked to assess reliability. Which type of task is most suitable depends on the aim of the study and the specific research questions and hypotheses to be addressed. Note that the way in which the task is described should be considered carefully, as this can influence the process adopted by the participants (see, e.g. Fricke (1993); Guindon (1990)).

Furthermore, it should be taken into account that a realistic or artificial task may not be taken seriously as the usual motivation and pressure are missing, in particular because there are no consequences if the task is not done properly, and the tasks given are often relatively small and not very challenging. In a real task, the consequences of incorrect decisions can be serious and the tasks can be very challenging. A lack of motivation can, e.g., occur when the participants have been asked to participate by the company. On the other hand, we have also seen an increased level of motivation and pressure caused by the fact that participants are observed and their work and working analysed. It is important to realise that motivation can change during the data collection period, in particular when the task (design task, interview, questionnaire, etc.) takes very long to fulfil. A short interview or questionnaire at the beginning and at the end of the study may reveal some of the motivations and state-of-mind of the participant.

A.2.13 Number of Cases

The number of cases refers to the number of data sets collected, e.g., the number of experiments, interviews, projects, documents, products, individuals, groups or companies. The number of cases is important to determine statistical significance. In the analysis, the number of cases may differ depending on the research question or hypothesis addressed. For example, in a study of design teams, one research question may be about the differences in quality of the work of design teams. The number of cases for the analysis is the number of teams. Another question may relate to the individuals within the teams, such as differences in the contributions of each of the group members. This analysis would involve many more cases, namely the number of individuals involved. It is important that the data allows analysis at different levels (see discussion on unit of analysis in Section A2.4).

A.2.14 Case Size

A collected case (e.g., a team or product) can consist of one or more elements. (e.g., individuals or components). The case size is the number of elements within each case. This dimension can be relevant for the analysis, interpretation and generalisation of the results. For example, groups of two persons will work differently from groups of ten persons and the situation in small companies will differ from that of large companies. Within an empirical study, the sizes of the cases can differ: the number of people in each team or the number of components in each product may not be identical.
A.2.15 Participants

It is very important to know who participated in a study. Their experience, education, current position, motivation, culture, etc., are of interest, as these could help explain the findings. If students have participated in a study, which is very common in design research, generalisation to practising designers might not be possible, and the answers in a questionnaire sent to a company may differ, depending on whether they come from a manager, a designer or a marketing specialist.

The characteristics of the participants that are relevant to know are those that could influence the outcome of a particular study. Experience is a typical influencing factor in design, but needs to be operationalised carefully. The number of years a designer has been working is an oft-used measure of experience. However, if the task used in the study is from a completely different domain, this experience may be of little use. Furthermore, we have found that senior managers can have a long experience, but might not have been designing for some time and thus lost some of the knowledge and skills necessary for a particular task.

A.2.16 Object

Apart from participants, the cases can (also) involve products, projects, companies and documents, as well as design support. Their characteristics have to be specified too, as far as these might influence the outcome.

For products to be designed or analysed, this includes: the name, the number of components, the type (original, variant, redesign), the batch size (mass, large batch, small batch, one-off) and the disciplines involved (mechanical, electro-mechanical, mechatronic, medical, etc.). In a similar way, the relevant details of companies, projects, documents, and other objects have to be specified. These details are necessary to generalise across objects. The required details of a design support to be evaluated in DS-II, can be found in Chapter 5.

In interviews and questionnaires it is important to collect details about the products, projects or companies on which the individual answers are based, in order to correlate the answers to different questions. Most correlations are only possible if the answers relate to the same object (see also Section A.4.9).

A.2.17 Coding and Analysis Method(s)

In most publications in the area of design research, the coding and analysis methods used are not specified, although these are essential for an interpretation of the data. Details of the processing, coding and analysis methods have to be documented. Some of the relevant aspects can be found in Sections 4.7.2 and 4.7.3.

A.2.18 Verification Method(s)

Methods used to verify the results of the study have to be specified (see details in Section 4.7.4). In the area of design research, this is unfortunately not often done.
A.2.19 Findings

To complete the description of an existing empirical study, it is useful to summarise the main statements, models, theories, and conclusions resulting from the study. For the planning of an empirical study, this dimension is used to document the expected (types of) statements, models, theories, and conclusions.

A.2.20 Notes

While reading about a particular study, questions may arise about particular details, information may be missing, inconsistencies may be found, certain methods may be particularly interesting, or differences and similarities with one’s own study may be noticed. This section is intended to make any such notes. If the study seems relevant for one’s own research, contact should be made with the authors to clarify any outstanding issues. Most authors will be pleased to explain their work.

A.2.21 Example

Table A.2 illustrates the use of the checklist to summarise the study of Hales (1987), who analysed an engineering design process in an industrial context.

Table A.2 Example use of the checklist for empirical studies based on the study described in Hales (1987)

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aim, research questions,</td>
<td>Better understanding of an engineering design process in industry in</td>
</tr>
<tr>
<td>hypotheses</td>
<td>which a systematic approach was introduced. Several research questions,</td>
</tr>
<tr>
<td></td>
<td>in particular the identification of the factors that influence</td>
</tr>
<tr>
<td>Nature of the study</td>
<td>the design process.</td>
</tr>
<tr>
<td>Theoretical basis</td>
<td>In particular Grounded Theory (Glaser) and Systematic Design (Pahl and</td>
</tr>
<tr>
<td></td>
<td>Beitz).</td>
</tr>
<tr>
<td>Unit of analysis</td>
<td>The stages in the design process (unit of measurement, a.o. the</td>
</tr>
<tr>
<td></td>
<td>individual participants).</td>
</tr>
<tr>
<td>Data collection and</td>
<td>Participant observation, using diary notes, audio tape recordings,</td>
</tr>
<tr>
<td>recording</td>
<td>weekly reports and design reports.</td>
</tr>
<tr>
<td>Role of researcher</td>
<td>Researcher was the main contract designer employed in the project.</td>
</tr>
<tr>
<td>Time constraint</td>
<td>Time constraints were not set by the researcher, but by the company as</td>
</tr>
<tr>
<td></td>
<td>project deadlines.</td>
</tr>
<tr>
<td>Continuation</td>
<td>The work on the project was restricted to one day a week. Data was</td>
</tr>
<tr>
<td></td>
<td>collected every day on which work on the project took place.</td>
</tr>
</tbody>
</table>
Table A.2 (continued)

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>The whole design project was studied, covering 36 months, and 2368 hours. At that point the company decided to stop the design project.</td>
</tr>
<tr>
<td>Observed process</td>
<td>From initial proposal (planning stage) to near completion of detail design.</td>
</tr>
<tr>
<td>Setting</td>
<td>The project took place in a large company in a natural setting.</td>
</tr>
<tr>
<td>Task</td>
<td>The task was real.</td>
</tr>
<tr>
<td>Number of cases</td>
<td>One project in a large company.</td>
</tr>
<tr>
<td>Case size</td>
<td>The project team consisted of 37 people.</td>
</tr>
<tr>
<td>Participants</td>
<td>Varying backgrounds and positions within the company and a contract engineer (the researcher) who was the main engineering designer.</td>
</tr>
<tr>
<td>Object</td>
<td>The project involved the development of a high-pressure and high-temperature system for evaluation of materials in a simulated slagging coal gasifier environment, to be used within the company. The design was original and one-off.</td>
</tr>
<tr>
<td>Coding and analysis method(s)</td>
<td>Notes were colour coded according to participant. Data was continuously entered in a database for indexing, sorting and categorising (using the structure of systematic design as proposed by Pahl and Beitz, and a list of 103 factors likely to influence the engineering design process drawn from the literature). Data was then transferred to spreadsheets for numerical and graphical analysis. Quantitative and qualitative analysis methods were applied.</td>
</tr>
<tr>
<td>Validation method(s)</td>
<td>Comparison with the literature.</td>
</tr>
<tr>
<td>Findings (main)</td>
<td>Ideal engineering design projects may be characterised by a series of overlapping, bell-shaped ‘phase curves’ each representing the work effort in a particular phase of the project along a time axis. Setting up an ideal phase diagram for a project provides a model against which to measure actual performance. Design work not completed within the envelope of the ideal phase curves for a particular project will have to be completed outside the envelope. This causes diversion of effort and thus increases the cost. The use of methods and aids commonly described in the literature accounted for less than one-quarter of the observed engineering design effort. A further 13 categories of design-related communicating, working, and motivating techniques were identified that accounted for the rest.</td>
</tr>
<tr>
<td>Notes</td>
<td>An extensive list of influencing factors was derived from the literature and this study. The fact that the project only run one day a week allowed time for processing the data in much detail.</td>
</tr>
</tbody>
</table>
A.3 Laboratory Versus Industrial Environment

Analysing real design processes in a practical environment is a type of field research. Field research in general has many advantages. The main advantage is that results are based on reality. The major disadvantage of field research is that the activities of the researcher can hardly be planned and defined when starting the research project, and that there is no guarantee that results will be useful, or the observed situation will continue and without interruption. As the example in Section A.2 showed, a design project might be cancelled. Furthermore, qualitative field research is fundamentally of an improvising nature and depends on the specific situation and on knowledge that has yet to be processed (ten Have 1977).

The difficulties of research in professional fields such as design, is that these fields are relatively closed and have a high degree of organisation, making them difficult to enter (Friedrichs and Luedthe 1975). The literature discusses the many obstacles encountered during the introduction in such a professional field. A major obstacle in design research, not mentioned in the literature on field research, can be the time required from the observed designers in an industrial setting. From the company’s point of view this implies a financial commitment: time is money.

In Tables A.3 and A.4, we have summarised the differences between a laboratory and an industrial setting for design research, as we and our colleagues have experienced these: not to discourage field work, but to raise awareness of the issues that have to be taken into account in the preparation of such study. Which setting is more suitable depends on the aim and the research questions and hypotheses. Several of the differences also apply to other practical settings, such as class rooms or design situations outside industrial contexts.

A.4 Data-collection Methods

Several books exist in the social sciences that provide an overview of available strategies and methods for collecting, analysing, interpreting, and evaluating data. These specialist books greatly facilitate the selection of potentially suitable methods and help clarify the underlying paradigms and assumptions, which will influence the recommended use. We found the following books particularly useful: (Cook and Campbell 1979; Frankfort-Nachmias and Nachmias 1996; Patton 2002). The latest Handbook of Qualitative Research of Denzin and Lincoln (2005) contains interesting but advanced discussions by various authors of different qualitative approaches of inquiry and their methods of data collection, analysis, interpretation, evaluation and presentation.

Care should be taken to consult primary and not secondary sources. Secondary sources, such as this appendix and the literature describing specific empirical studies, can be useful to find out about the experiences of those who applied the methods, but cannot be considered authorised texts on those methods. For similar reasons, care should be taken using the Internet, and the background of the authors checked for their experience and research area.
Table A.3 Differences in process of observing: industrial versus laboratory setting

<table>
<thead>
<tr>
<th>Process of observing</th>
<th>Industry</th>
<th>Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficult to plan.</td>
<td>Time and location can be determined in advance.</td>
<td></td>
</tr>
<tr>
<td>Difficult to control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• interrupts from others;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• usually no explicit starting or end point (when did the first idea come up?);</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• topic cannot necessarily be chosen.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous observation may be difficult. Tasks may be abandoned or stopped for a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>short while or for months.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interference with existing processes can cause problems with obtaining allowance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>for the study (time = money). In particular, recording equipment can interfere or may</td>
<td>No or hardly any interrupts if carefully planned. However, no positive effects of</td>
<td></td>
</tr>
<tr>
<td>not be allowed.</td>
<td></td>
<td>interrupts either (talking to others, sudden insights through other</td>
</tr>
<tr>
<td>Confidentiality of observed processes and results may hinder data collection and</td>
<td></td>
<td>activities, gestation of ideas)</td>
</tr>
<tr>
<td>publication of results. Anonymity of participants to the outside is possible, but</td>
<td></td>
<td></td>
</tr>
<tr>
<td>difficult to achieve internally.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple studies on the same topic impossible or very difficult.</td>
<td></td>
<td>Multiple studies using the same topic but other participants is possible.</td>
</tr>
<tr>
<td>Existing environment not easily optimised for observation.</td>
<td></td>
<td>Environment can be optimised for observation.</td>
</tr>
<tr>
<td>Work on one task can extend over months requiring specific data-collection methods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Very difficult to capture all work on a particular topic, as many people may be</td>
<td></td>
<td></td>
</tr>
<tr>
<td>involved and work may be discontinuous. Duration, i.e., involvement of participants,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>difficult to predict.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time consuming for researcher and potentially for participants.</td>
<td></td>
<td>Observation time often much less than analysis time.</td>
</tr>
</tbody>
</table>
Table A.4 Differences in the observed process: industrial versus laboratory setting

<table>
<thead>
<tr>
<th>Observed process</th>
<th>Industry</th>
<th>Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>The product to be designed has a history and a future within the company, which</td>
<td>The product to be designed has a history and a future within the company, which has to be taken into account. This may require additional data-collection methods.</td>
<td>Task is self-contained, therefore easier to analyse (all data available), although the designers bring in their own history. This may require an additional data-collection method.</td>
</tr>
<tr>
<td>has to be taken into account. This may require additional data-collection methods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level of complexity can be high: the number of components of the design object</td>
<td>Level of complexity can be high: the number of components of the design object can be very high (thousands of components). Not easy to single out a particular assembly or component on which to focus, because other parts of the design may become relevant.</td>
<td>Possible to focus on a low level of complexity. Usually restricted to designs with tens of components.</td>
</tr>
<tr>
<td>can be very high (thousands of components). Not easy to single out a particular assembly or component on which to focus, because other parts of the design may become relevant.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data will show effects of influences from personal to macro-economic level, such</td>
<td>Data will show effects of influences from personal to macro-economic level, such as company goals, costs, availability of components, disagreements, suppliers, etc.</td>
<td>The participants determine the design. The focus of the design process can be very functional.</td>
</tr>
<tr>
<td>as company goals, costs, availability of components, disagreements, suppliers,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observation of individuals cannot take place without considering influences of</td>
<td>Observation of individuals cannot take place without considering influences of others in the project. Only part of a project is captured, as not all the work of all participants can be captured.</td>
<td>Analysis of individual is possible.</td>
</tr>
<tr>
<td>others in the project. Only part of a project is captured, as not all the work of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>all participants can be captured.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem definition and requirements will change, due to duration, relation with</td>
<td>Problem definition and requirements will change, due to duration, relation with the market, etc.</td>
<td>Essentially a frozen assignment unless a change is deliberately introduced by the researcher.</td>
</tr>
<tr>
<td>the market, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Only parts of the process can be covered.</td>
<td>Only parts of the process can be covered.</td>
<td>Whole process can be covered.</td>
</tr>
<tr>
<td>Observed process can be chaotic and complex due to interrupts and the fact that</td>
<td>Observed process can be chaotic and complex due to interrupts and the fact that designers may need to attend to issues that are not related to the design project.</td>
<td>A ‘smooth’ process, determined by the participants.</td>
</tr>
<tr>
<td>designers may need to attend to issues that are not related to the design project.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results show reality.</td>
<td>Results show reality.</td>
<td>Results may not relate to reality.</td>
</tr>
<tr>
<td>Difficult to determine correlations, causes and effects.</td>
<td>Difficult to determine correlations, causes and effects.</td>
<td>Setup can be chosen such that correlations and causalities can be determined.</td>
</tr>
</tbody>
</table>

This section focuses on the main characteristics of a variety of common data-collection methods, their application in design research, our experiences and that of our students with these methods, and some references to the literature. None of the methods is generally more suitable than the others: it all depends on the research questions and hypotheses that were formulated, the context in which the method is to be applied, and how the method is tailored to and applied within this context.
Many books and articles have been written on each of these methods. In particular the books in the Applied Social Research Methods Series of Sage Publishers contain good introductions to many of the methods and provide useful pointers to more detailed literature. Reading this literature is essential to ensure the effective and efficient use of the chosen method and to avoid bias and unexpected problems, when tailoring and applying the method.

**A.4.1 Observation**

Observational methods involve the researcher recording what is actually taking place either by hand or using recording or measuring equipment. Observational methods are real-time methods. Observation, whether in a laboratory or practical setting, is one of the most common ways of data collection. An experiment is a classical observational method (see Section A.4.3).

The quality of observational data is highly dependent on the skill, training and competency of the observer. In the words of Patton “The trained observer is skilled in identifying and accurately describing meaningful human interactions and processes. In addition to training and practice, the fieldworker needs concentration, patience, alertness, sensitivity and physical stamina.” (Patton 1987). He also gives a useful account of the training requirements. Careful preparation is essential as chance favours the prepared mind.

Patton discussed six dimensions of observational studies, which we discuss in the context of design research. Although specific for fieldwork, these dimensions are useful for the planning of the majority of observational studies.

- Role of observer: from full participant, to partial observation, to onlooker observation. The role may vary and evolve over time. In design research, researchers took a variety of roles, although full participation has been limited.
- Insider (emic) versus outsider (etic) perspective: the categories used to classify the data are those used by the participants, those created by the researchers, or a combination of both. In design research both perspectives have been taken, with a preference for the outsider perspective.
- Degree of collaboration of the participants: from individual or teams of researchers, to partial or periodic involvement of participants, to full collaboration and participation. Design research has involved all types of collaboration.

---

22 Patton focuses on studies in a natural setting (fieldwork), not in a laboratory context

23 In earlier publications (such as Patton (1987) five dimensions were used, some of which have now been split or merged. The third dimension is newly included. The changes clearly show a shift in the role of the participants in field research, from passive to active.

24 Note that ‘participatory research’ refers to the role of the participant, where as ‘participant observation’ refers to the role of the researcher (see first bullet point). ‘Action research’ (see Section A4.9) combines both: collaboration exists throughout the research project, with varying levels of involvement from each side depending on the phase of the project.
• Overt versus covert observations: this involves two issues. First, do participants know about the observation? Options are: all involved know that observations are being made and who the observer is (overt observation); the observer role is known by some, not by others; those involved do not know they are being observed (covert observation). Second, in the case of overt or partial overt observations, do the participants know the purpose of the evaluation? Options are: full explanation of real purpose to everyone, partial explanations, no explanations, or false explanations. The tendency in design research is overt observations but without providing a full explanation of the purpose, so as not to influence the design process to be studied. Explaining, e.g., that the purpose is to study the use of decisions making in the design process, might lead to more frequent, explicit decision making than would normally have occurred.

• Duration: single observation with limited duration to long-term, multiple observations. Long-term or repeated observations can be useful in evaluations of design support or design teaching (such as Bender (2004)) to capture learning and motivational effects, but are not often applied.

• Focus of the observation: narrow focus, observing a single element, to broad focus seeking for a holistic view of the entire process and all its elements. Again, design research, often being explorative, has covered the whole range.

Pure Observation

Typical for pure observational methods is that the researcher is not involved in the process and does not interfere with the process while the process is ongoing (although interference caused by the observer’s presence cannot be ruled out). In order to observe particular phenomena, the researcher might or might not have created the context in which the process takes place. This approach is considered the most objective.

Participant Observation

The term participant observation is used in fieldwork when the role of the researcher is not restricted to that of an onlooker: the researcher participates in the process. Participant observation can help gain acceptance and increase familiarity with the field and the problems. As an insider the researcher might also be able to collect more in-depth data and is in a better position to interpret these, such as Hales (1987) discussed in the example in Section A.2.21.

Some authors, such as Denzin (1978) view participant observation as a research strategy that simultaneously combines several data-collection methods, such as document analysis, interviewing, direct participation, observation and introspection. Other authors, such as Yin (1994) view participant observation as a method of data collection based on a special mode of observation – namely one in which the observer participates in the observed process. This data can be used as one source of evidence in, e.g., a case study. In our view, participation is a role
following specific playing rules that better allows collecting certain types of data in a natural setting and often involves a variety of data-collection methods.

Participant observation requires the researcher to have “the commitment to adopt the perspective of those studied by sharing in their day-to-day experiences” (Denzin 1978). The observer not only shares the subject’s world, but also takes on their language, rules and behaviour, and takes actively part in what is happening. Patton gives a useful account of the training requirements (Patton 1987).

An important issue in participant observation is that ‘experiencing what the observed experience’ is considered essential for obtaining insight, but at the same time, brings with it the danger that researchers lose their research perspective and see the world too much from the point of view of those with whom they are identifying themselves. For some schools of thought, this ‘going native’ is not acceptable: they consider the collected data invalid because of bias through subjectivity. However, a general trend towards a more involved perspective can be observed. For others, such as ethnomethodologists, this is the only way to gain true understanding. In our view, the roles of observer and participant can be combined, but this involves an awareness of the dilemma between objectivity and subjectivity, and of observing a field and being part of it. The challenge, according to Patton (2002) is “to combine participation and observation so as to become capable of understanding the setting as an insider, while describing it to and for outsiders”. The different types of notes described later in this section can be used to separate data accordingly.

Combining the roles of researcher and participant also gives rise to a very practical problem: increased participation reduces observation possibilities, observation time and interaction, and vice versa. Participating in a design process requires focusing on a specific design problem, which means working individually for part of the time. Hence, the time to observe the other ongoing processes reduces. A further practical problem is that researchers might not have the necessary (design) qualifications to be a complete participant. In many studies, therefore, researchers have a supporting role only.

In design research, participants are normally aware that they are being observed: to enter and work unnoticed in a professional field is highly unlikely, if at all desired. An interesting alternative is the study described by Eckert (1997). Those observed knew she was working on her thesis, but not the details. Because the researcher’s professional qualification inhibited participant observation, the researcher ‘disguised’ herself as a mixture of a placement student and a “child visiting an aunt”, i.e., watching an expert doing her daily activities and chatting to her. She experienced that she got the best answers when she told the designers that she had difficulties with a particular task and got them to tell her how to do it.

Obviously, in participant observation, the background of the researcher, the role of the researcher and the possible research questions are closely linked.

Observing Participant

In participant observation, the researcher is designing, i.e., temporarily taking on the role of designer. In some studies, designers observe and document their own process. We call this ‘observing participant’: the designer is observing, i.e., temporarily taking on the role of researcher. Researchers might have asked
Designers to do so, or researchers – with the necessary qualifications – take on the role of main designer in their own design process and observe themselves. An example of the latter is a study in which researchers with design qualifications observed themselves designing a product in their academic environment (Waldron and Waldron 1988). Various methods can be used by the observing participants to capture the data. As a real-time method, data collection takes place continuously, or very regularly, to avoid the problems inherent to retrospective methods such as questionnaires and interviews. If the designer is not the researcher it may be difficult to obtain the designers’ commitment if a long period of time is involved.

The terms ‘observer as participant’ and ‘participant as observer’ used by Denzin (1978) look similar to our terms, but have a different meaning. The ‘observer as participant’ role refers to a situation in which typically only one visit or interview is included in which no relationship builds up. The ‘participant as observer’ does form relationships with the other participants. In our terminology, both variants of Denzin belong to participant observation, although we do not consider the first variant, a one-visit participation, likely in the context of design.

Non-occurrences

Apart from recording what has been observed, it can be useful to make explicit what has not been observed, if prior knowledge suggests that certain things ought to have occurred or when in a particular case – in contrast to other cases – something did not occur. “Making informed judgements about the significance of non-occurrences can be among the most important contributions” but “this clearly calls for judgement, common sense, and experience” (Patton 2002). Such non-occurrences indicate, e.g., that the finding cannot be generalised and will raise the question as to what did occur instead and why. Making explicit statements about non-occurrences in one or more cases can emphasise the occurrences of this in the other cases and confirm correlations. An example is the observation that experienced designers, prior to undertaking a particular action, considered whether it was worthwhile pursuing this action (Ahmed 2001). This was reinforced by the observation that this behaviour did not occur in the processes of the novice designers. Here, the statement about the non-occurrence was simply the opposite of the statement of the occurrence. Note that this is not always the case (see also the discussion at the end of Section 2.4.1).

Recording

Technological developments and general availability of affordable recording equipment such as video camera’s, have made it much easier to capture large amounts of rich data, but analysing these recording is still time consuming and not very easy, despite progress made in software packages for quantitative as well as qualitative data analysis.

Despite the advances in technology, taking notes has not lost its importance. Many thoughts and context details occurring during observation cannot be recalled later and might not have been captured by the recording equipment used. It is essential to make notes and go through these as soon as possible after the observation to add where necessary to obtain notes that will remain meaningful.
While observing, it is important to distinguish different types of notes to avoid confusion as to whether the note refers to, e.g., what someone has said, what has been observed, or an interpretation of something that was observed. ten Have (1977) suggested 4 types of notes: observational notes, methodological notes, reflective notes and theoretical notes. In Blessing (1994) two more types of notes were added: interview notes and organisational notes.

- **Observational notes.** These notes describe pure observation of events without interpretation. The data can be biased because the underlying mental processes cannot be observed and because data depend on the view of the observer only. Observational notes can be supported by recording equipment, taking away some of the bias. It is important to be as detailed and concrete as possible and avoid general terms describing actions and conditions such as ‘the sketches were very detailed’. The level of detail has to be specified, e.g., ‘contained comments’, ‘used drawing conventions and views’, ‘showed many of the shape details’, etc. Instead of writing ‘She was not happy with the design’ one should write ‘She shook her head, did more calculations, erased parts of the drawing and corrected several times while regularly saying “oh dear”’. A distinction has to be made between what was said and what was observed, e.g., using a particular coding scheme, such as using quotation marks for what was said, or using different colour pens. The source, date and context have to be recorded too.

- **Interview notes.** These notes do not contain an interpretation either (although some interpretation cannot be avoided as part of the process of interviewing (Ackroyd and Hughes 1981)). Interview notes are more reliable than observational notes because they do not rely on observation or what was observed to be said (see also Section A.4.8 on interviews). The source of the data has to be recorded as well as date and context.

- **Methodological notes:** Methodological notes are descriptions of the research approach, i.e., the planned data-collecting process, adjustments that were made in the data-collection method before or during data collection, the way of analysing the data, and the experiences with the methods used.

- **Reflective notes.** These notes contain the reactions and feelings of the researcher, reactions of the participants and others, and thoughts about the role of observer and researcher. These notes enrich the observations because they can make the observer more conscious of his or her own behaviour and of the changes this causes in the field. These reflections may result in changes to the research approach, which are then described in the methodological notes.

- **Theoretical notes:** These notes contain thoughts about the collected data, such as interpretations, comparisons and characterisations that come up during the observations. These notes require verification, once all data have been collected. They can lead to further research questions and hypotheses.

- **Organisational notes:** These are notes about the organisation, the role of the participants in the organisation, aspects of their background that emerge
during the observations, and events that do not directly involve the observed process, but could have an effect.

The observational and interview notes have to be strictly separated from the other notes to avoid bias in the analysis of the resulting data.

Before planning to use any recording equipment in an industrial context it is essential to ensure that this is allowed. A confidentiality agreement might not cover this type of recording. In addition, each of the participants needs to agree to the use of recording equipment and should be allowed to request the recording to be stopped for a certain period of time, if necessary.

A.4.2 Simultaneous Verbalisation

Simultaneous verbalisation refers to the situation in which participants speak aloud while working. Participants may have been asked to do so, or this may be a natural part of their work. The aim is to provide insight into the cognitive behaviour of participants, which may not be obtained through normal observation. Simultaneous verbalisation has often been used to analyse problem solving behaviour based on the theory of information processing, which makes fundamental assumptions about the nature of cognition. The most important characteristic of simultaneous verbalisation is the real-time aspect: while working on the problem the problem solver is thinking aloud. Under controlled conditions it appeared that individuals (when trained to concurrently verbalise their thoughts) could reveal a remarkably accurate picture of their cognitive processes while engaged in problem solving (Eckersley 1988).

When participants are asked to verbalise their thoughts while working, this is called think aloud. The term talk aloud is also used, in particular if participants are just asked to speak while working, as if no one is observing. When subjects are asked to reflect on their thoughts, this is called introspection. This is more intrusive than thinking aloud and not often used in design research (an exception is Visser (1990). The least intrusive method is capturing the utterances of two or more participants who work together on a problem; less of the individual thought processes might be captured, but more of the reflections when participants explain to each other their thoughts. Note that the definitions of these terms differ depending on the sources used. Some authors consider thinking aloud the same as introspection and different from talking aloud. Patton (2002) views the think-aloud protocol approach as a specific kind of qualitative interviewing, that is, a retrospective rather than a real-time approach. The interviewing should take place “as close to the action as possible” to illuminate “what’s going on in a person’s head during the performance of a task”. He refers to concurrent use (as common in design research) as an interesting exception. Other authors, such as we, consider thinking aloud a real-time approach, distinct from introspection, but similar to talking aloud.

Whatever terminology is chosen, important is the careful consideration of the wording used to ask the participants to speak while working, as this will affect the resulting data and hence the suitability of the data to address the research questions and hypotheses.
Simultaneous verbalisation sessions are usually a few hours and not longer than a day, not only because of the effort required by the participants, but in particular because of the effort required for analysis.

Audio tapes were found to be of limited use as the sole source for a detailed analysis of a process such as design, involving drawings and gestures. Notes taken during observation may prove insufficient to provide the missing details. This is illustrated by the following bite of an audio recording of a designer who pointed at his drawing and explained to his colleague how he imagined that the component should be cast: “and this is sand, sand, sand, sand, sand” (source: own research). Video recordings were found to be far more useful for this type of design research: the context is captured, and the data can be analysed by others and long after the recordings were made.

Protocols are the transcribed recordings of the utterances and activities of the participants. These protocols are the basis for the analysis. The collected data is very rich, but consists of fragments of sentences and seemingly inconsistent lines of thought as shown in Figure A.1. The most extensive book on protocol analysis is that of Ericsson and Simon (1996), but various other books describe qualitative and some quantitative methods for analysing and deriving meaning from such data.

<table>
<thead>
<tr>
<th>design</th>
<th>time</th>
<th>text</th>
<th>researcher</th>
<th>act</th>
<th>focus</th>
<th>write</th>
<th>trans</th>
<th>write</th>
<th>crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:56:14</td>
<td>I am sure you can do this</td>
<td>m</td>
<td>s3</td>
<td>a1w</td>
<td>fun</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:56:21</td>
<td>we are getting 15 degrees movement in there</td>
<td>m</td>
<td>s3</td>
<td>a1w</td>
<td>fun</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:56:33</td>
<td>really it can be quite crude</td>
<td>s</td>
<td>s3</td>
<td>s3</td>
<td>a1w</td>
<td>a1w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:56:37</td>
<td>I mean, how primitive the machine technology is</td>
<td>s</td>
<td>s3</td>
<td>s3</td>
<td>a1w</td>
<td>a1w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:56:41</td>
<td>provided the sockets is got to wear a bit</td>
<td>s</td>
<td>s3</td>
<td>s3</td>
<td>a1w</td>
<td>a1w</td>
<td>env</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:56:57</td>
<td>if you wanted to be really sort of terribly that makes so many clamps doesn't it</td>
<td>m</td>
<td>nn</td>
<td>nn</td>
<td>com</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:56:59</td>
<td>because we have all these clamps off here</td>
<td>m</td>
<td>s2</td>
<td>pwg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:57:02</td>
<td>clamping to that</td>
<td>m</td>
<td>s2</td>
<td>pwg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:57:14</td>
<td>could this thing actually move at all</td>
<td>m</td>
<td>s2</td>
<td>pwg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:57:23</td>
<td>if somebody tried to lift this out</td>
<td>m</td>
<td>s2</td>
<td>pwg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:57:28</td>
<td>there still would quite some, some of the strain would be on that joint there</td>
<td>m</td>
<td>s2</td>
<td>pwg</td>
<td>mech</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:57:36</td>
<td>I think that is the way to do it</td>
<td>m</td>
<td>s2</td>
<td>pwg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00:57:41</td>
<td>how does it actually look like</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A.1 Part of a think-aloud protocol of a designer designing a small mechanical device and categorisations of his utterances and activities used to address the research questions (Blessing 1994)

The effort involved in protocol analysis should not be underestimated. Our colleagues and we have found that detailed transcription of the video recordings of design sessions into protocols for analysis, such as the first two columns in Figure A.1, can easily take 8 hours per hour of video. A detailed analysis of the protocol using several classification schemes, resulting in codes such as those in the last six columns in Figure A.1, can take up as much as 40 hours per hour of video.

The advantage of a detailed transcription, including some information about the context (such as the material that was used or paper on which the participant wrote), is that further analysis can rely largely on the protocol, without the need to consult the original recordings to understand what was going on. Moreover, detailed transcription facilitates the use of the transcripts by other researchers.
When few aspects have to be analysed and the data can be divided into large chunks, it may be more efficient to make a summary protocol, describing rather than transcribing each segment, or not to transcribe but watch the recording to analyse each aspect. Which level of transcription and analysis is required, depends on the research questions and hypotheses that need addressing.

It is worthwhile to look for the possibility to use software packages specifically developed to support the analysis of video recordings, which could remove the need for detailed transcription and reduce the effort required for analysing the data and representing the findings. The website of the American Evaluation Association\textsuperscript{25} lists several such software packages. Selection has to be done carefully, based on the analyses required. The currently available packages focus on particular types of analyses typical for certain domains. We found that several packages have limited possibilities for using multiple classification schemes, and hence do not allow analyses such as that shown in Figure A.1.

Using simultaneous verbalisation in a natural, industrial setting and on a real task can be problematic. We found that designers considered it difficult and even embarrassing to think aloud while designing in the design office amongst their colleagues. Furthermore, companies may not allow such detailed data to be captured.

Specific problems in transcribing and analysing the recordings of teamwork we have encountered are: more words (data) per time unit compared to recordings of individuals; overlapping data ‘streams’ because people interrupt each other and talk at the same time (a specific notation in the transcription is necessary); parallel processes when one or more team members become engaged in another issue than the rest of the team members; and team members ‘doing their own thing’ in silence.

A.4.3 Experiments, Quasi-experiments and Non-experiments

In the context of design the term experiment is often used incorrectly. Our field rarely provides the possibility to do true experiments. This section aims at clarifying the terminology.

*Experiments*

Classical experimental research is comparative research in which:

- the researcher has control over the context in which the phenomena to be studied occur;
- the participants or objects are randomly assigned to the groups involved;
- the participants or objects are representative of the target population.

Note that *randomly assigned* is not the same as *randomly selected*. Randomly assigned refers to how the participants or objects involved in the study are divided into groups to be compared, and thus relates to the setup of the study and the internal validity (see Section 4.7.4). Randomly selected refers to the way in which

\textsuperscript{25} http://www.eval.org/Resources/QDA.htm, accessed 13 December 2008
the participants or objects were chosen to participate in the study, and thus relate to sampling and external validity (see section 4.7.4).

Classical experiments can be repeated under controlled conditions and one or more independent variables can be manipulated to test the underlying hypothesis about the effects on the dependent variable. The dependent variable is the variable the researcher wishes to explain. The variable that is expected to change the dependent variable is the independent variable (see also Section 4.5.2).

These types of experiments are the most rigorous methods available to determine causality, that is, to determine the time order between concepts, covariance, and the exclusion of rival factors (see Section 4.7.3). Experiments are well known in the field of natural and engineering sciences. For a discussion about experiments in social sciences see Denzin (1978) and Cook and Campbell (1979).

In experiments use is made of at least one control and one experimental group. A group is a set of cases or objects of study, each of which is observed separately. A case in design can be as diverse as an artefact, a designer, a design team, a company, or a drawing. The cases in the experimental group are exposed to the independent variable, they are ‘treated’ or ‘trained’ (or treatment/training is withheld, depending on what the ‘normal’ situation is). In design, the treatment could be the training of a design method or the introduction of a new tool. The cases in the control group are not exposed to the independent variable, that is, they represent the normal situation. The groups are equivalent, or at least assumed to be, because the cases are assigned to the groups at random.

Each group is observed at least using the same measurement method(s): once before (the pre-test) and once after the experimental group has been treated (the post-test). The aim of the pre-test is to detect any differences between the groups before the exposure to the independent variables. The aim of the post-test is to detect the effects. Statistics are used to determine the significance of any differences in findings. This depends on the number of cases in each group, for which reason the sample size is an important factor. This type of research is represented schematically in Figure A.2a. The ‘O’ stands for an observational test as a measurement method; the ‘X’ for a treatment or exposure to the independent variable.

Denzin (1978) suggests that if, and only if, pre-tests are not possible, the pre-tests may be left out, because the fact that the cases are assigned at random will often suffice as a control for the pre-test. An advantage is that it removes any problems with changes that might occur between the two tests. This is illustrated in Figure A.2b. The “purest of the experimental models” (Denzin 1978) is the Solomon Four-Group Design in which the groups differ as to whether they are exposed to the independent variable and undergo a pre-test. This is illustrated in Figure A.2c.

Quasi-experiments

In many cases it is very difficult to fulfil all requirements for experimental research (control over setting, random assignment, and representative of target population). In quasi-experiments, the researcher has less control over the experiment than in
classical experiments, but still enough to allow the logic of the experiment to apply. Quasi-experiments are experiments that have treatments, pre-tests, post-tests, and cases, but do not use random assignment. Instead, the comparisons depend on non-equivalent groups. To be able to interpret the data, the effects of the treatment have to be separated from those due to the initial differences between the groups (Cook and Campbell 1979).

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>O₁</th>
<th>X</th>
<th>O₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td>O₁</td>
<td></td>
<td>O₂</td>
</tr>
<tr>
<td>a. Classic experimental research plan: equivalent groups, identical pre- and post-test</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>X</th>
<th>O₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td></td>
<td>O₂</td>
</tr>
<tr>
<td>b. Experimental research plan: equivalent groups, pre-testing not possible</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experimental group 1</th>
<th>O₁</th>
<th>X</th>
<th>O₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group 1</td>
<td>O₁</td>
<td></td>
<td>O₂</td>
</tr>
<tr>
<td>Experimental group 2</td>
<td>X</td>
<td>O₂</td>
<td></td>
</tr>
<tr>
<td>Control group 2</td>
<td></td>
<td>O₂</td>
<td></td>
</tr>
<tr>
<td>c. Experimental research plan: Solomon-four-group design</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A.2 Experimental research plans based on random assignment of cases to groups. ‘O’ denotes observational tests; ‘X’ denotes exposure to the independent variable (or treatment)

Cook and Campbell discuss a large number of different quasi-experiments, based on varying the various elements in Figure A.2 to take into account any deviations from the conditions necessary for a classical experiment. Some of the research plans do permit reasonable causal inference, others do not. Those that do, include research plans that use different methods for pre- or post-testing, involve several treatments, or are observed several times, see Figure A.3. In quasi-experimental designs, statistical techniques substitute for the experimental method of control. In other words, because the data is not ‘as hard’ as in a classical experiment, statistics is used to ‘filter out’ these uncertainties. For example, differences between groups might have to be larger in order to be significant. See Frankfort-Nachmias and Nachmias (1996) for a discussion.

---

The term ‘quasi-experiment’ is not generally accepted. King et al., e.g., argue that the researcher either has control over the observations and values of the key causal variables (experimental research) or not (non-experimental research) (King et al. 1994).
Appendix A: Descriptive Study Methods  267

Figure A.3 Examples of quasi-experimental research plans assuming non-equivalent groups but permitting reasonable causal inferences to be drawn: ‘O’ denotes observational tests; ‘X’ denotes exposure to the independent variable

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>O₁</th>
<th>X</th>
<th>O₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td>O₁</td>
<td></td>
<td>O₂</td>
</tr>
</tbody>
</table>

a. Typical quasi-experimental research plan: non-equivalent groups, identical pre-test O₁ and post-tests O₂

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>Oₐ₁</th>
<th>X</th>
<th>Oₖ₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td>Oₐ₁</td>
<td></td>
<td>Oₖ₂</td>
</tr>
</tbody>
</table>

b. quasi-experimental research plan: non-equivalent groups, different pre-test Oₐ and post-test Oₖ

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>O₁</th>
<th>O₂</th>
<th>X</th>
<th>O₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td>O₁</td>
<td>O₂</td>
<td></td>
<td>O₃</td>
</tr>
</tbody>
</table>

c. quasi-experimental research plan: non-equivalent groups, repeated pre-test

Non-experiments

Those research plans that are “normally not sufficient for permitting strong tests of causal hypotheses because they fail to rule out a number of plausible alternative interpretations”, are useful for suggesting new ideas (Cook and Campbell 1979). These so-called non-experimental designs (Denzin 1978) consist of: pre-test and post-test, but only a single, experimental, group; two (non-equivalent) groups but only a post-test; and – the weakest of all – only one group and one post-test. These research plans are illustrated in Figure A.4.

An example of non-experimental research of type c in Figure A.4 is the analysis (post-test) of companies (experimental group) who have introduced a particular design support (treatment or exposure to the independent variable). Valuable information can be obtained from observing the use of the method and the outcome. However, nothing can be said from such an observation about the differences in process and outcome compared to the situation before the introduction of the support (although often such conclusions are drawn). Interviewing participants or using questionnaires can provide some indications about these differences, in particular if the various companies and participants involved express the same opinion. The non-experimental design type a requires a pre-test (such as the observation of the particular task and outcome before the introduction of the method) and would allow more statements. However, clear statements about causal inference are not possible because rival or alternative explanations cannot be ruled out. Examples of such rival explanations are an increase of experience, differences between the design tasks (the design tasks have to differ, if the same designers are involved).
Three non-experimental research plans: ‘O’ denotes observational tests; ‘X’ denotes exposure to the independent variable

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>O₁</th>
<th>X</th>
<th>O₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Non-experimental research plan:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no control group</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>X</th>
<th>O₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. Non-experimental research plan:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-equivalent groups, only post-test</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experimental group</th>
<th>X</th>
<th>O₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>c. Non-experimental research plan:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No control group, only post-test</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A.4 Three non-experimental research plans: ‘O’ denotes observational tests; ‘X’ denotes exposure to the independent variable

Design research usually involves non-experimental and quasi-experimental research, in particular for DS-I, although some product-focused research may allow experimental research plans. For empirical studies to evaluate design support (DS-II) quasi-experimental or even experimental studies are more common.

We are not certain whether design research can ever be truly experimental other than by approximation; either we conduct experiments in a contrived setting or resort to quasi- or non-experiments in a natural setting. The decision to go for an experimental or non-experimental approach involves a trade-off between realism and precision (Rossi et al. 1999). Currently, in particular because of the lack of understanding in design, much research will be non-experimental, but not less worthwhile or necessarily easier; “both experimental and non-experimental research have their advantages and drawbacks; one is not better in all research situations than the other” (King et al. 1994) and “paradoxically, the ‘softer’ a research strategy, the harder it is to do” (Yin 1994).

A.4.4 Case Study

The term case-study is often used to describe a study that involves data from a real setting (in our case often a setting in practice), and is seen as equivalent to an observational study in which only one or very few cases are involved. For obvious reasons, a one-shot-case study cannot be used for testing causal relationships, but can provide very valuable information. Such case studies are primarily used for exploratory research or for pre-testing some research hypotheses. Note that a one-shot case-study is not the same as the non-experimental research plan c in Figure A.4. The latter involves multiple cases. The earlier mentioned participant observation studies are examples of case studies. A useful introductory book about case studies is Yin (1994).
A.4.5 Collecting Documents

Retrieving documents related to a particular project, topic or product, from a variety of sources can be very useful as an additional data-collection method. Often, a study in industry will start with the collection and analysis of documents to understand the organisation, the background of the project, and the experience of the designers. Documents will be created by designers at all times. Examples are sketches, drawings, notes, calculations, minutes of meetings, emails, etc., and will therefore be part of the data in most observational studies.

Documents can also be the main source of data. Examples in design research are the collection of maintenance and service data to determine the reliability of various products and the effects of improvements (Stephenson 1995) and the collection of documents that were used and produced in order to study the information flow in design projects (Vroom 2001). It is often useful, but for historical data not always possible, to support document collection with other methods such as interviewing. This helps to overcome one of the main limitations of documents, namely the usual lack of data about the context in which they were created, and the rationale behind the contents.

Special methods to analyse textual data are available under the keywords of ‘document analysis’ or ‘content analysis’. For the analysis of the other types of data produced during design, such as sketches, no guidelines are available. We can only refer to the publications of design researchers who worked with such data.

A.4.6 Collecting Products

Physical products and any mock-ups, prototypes, and other physical models can be part of the collected data, e.g., to trace the development of a product. The products could be variants, members of a product family, versions that were developed over time, modules, etc. The focus can be on a particular aspect, or on the product as a whole. An example is the evaluation of the suitability of current product configuration methods for products with large numbers of variants (Hami-Nobari 2007). For traditional engineering research, focusing for example on the analysis of product behaviour, products are the main source of data.

A.4.7 Questionnaires

Questionnaires are used to collect thoughts, beliefs, opinions, reasons, etc., from people about past, present or future facts and events, by asking questions. A particular focus is on data that cannot be captured using observation or simultaneous verbalisation, and on data about the past that was not captured. “They allow us to enter into the other person’s perspective” (Patton 2002).

Disadvantages of these methods are the time required from the participants and the potential bias of the results due to the fact that people are forgetful, see things from their present point of view, or give an answer that is coloured by what they conceive as more desired with respect to the purpose of the interview, social standards (e.g., political correctness), or their own behaviour and that of others. A
design process, e.g., may be represented as more systematically and reflective than it really was.

Questionnaires may seem easier than real-time methods and to ensure data from a larger number of cases. However, the effects of poorly formulated questions on the resulting data, and the effort required for a proper analysis should not be underestimated, nor should the return rate be overestimated. Pilot studies are always necessary, which should include the actual analysis of the data.

In their book on mail and Internet surveys, Dillman et al. (2008) provide a very useful set of guidelines on how to formulate good questions and construct open and closed questions. To conduct a survey, questionnaires are used to address a large number of subjects. They are often sent by mail and increasingly by Internet, but also used in telephone surveys and surveys in public places.

Questions should be unambiguous, interesting, and quick to answer, because there is little incentive for people to spend their time and effort on answering a questionnaire. This is reflected in the notoriously low return rate (often less than 5%) unless specific measures are taken. Companies sometimes receive several survey questionnaires per week from research students, often on topics in which they are not interested. If at all, they will only answer those that are of interest; a low return rate thus also implies that the returned questionnaires often are not representative. Questions should not be biased, that is, should not suggest an answer. A questionnaire should also be self-explanatory, as the researcher is not at hand, and should be answerable by one person. We have regularly seen questionnaires combining questions about different areas of expertise, e.g., on company strategy, methods to generate ideas, methods to assess market needs. Few persons, if any in a company, can answer all these questions. No company will put the effort in involving multiple people in order to answer the survey.

Apart from questions about the topic of interest, the following additional questions are necessary.

- Questions about relevant characteristics (e.g., function or background) of the person who answered the questions, as this can influence the answers.
- Questions related to the characteristics that were used to select the sample, in order to verify whether those that returned the questionnaires are representative of the target group. These can be the characteristics of the person, project, company, etc.
- Questions about relevant characteristics of the context (product, project, situation, or other) to which the answers refer, in order to be able to correlate answers of different questions within one questionnaire. Determining correlations between answers makes sense, only if the answers refer or apply to the same context. A simple but effective solution is to ask the participants to answer the questions for, e.g., a particular project, and then to repeat the questions for another project. Specifying a particular feature (the last, the current, a particularly successful or problematic project) can help the participants in their choice, result in more focused answers (rather than ‘it depends’) and provide useful additional information for analysis. To know how typical a particular answer is, additional questions are necessary.
As a supporting method, questionnaires are often used to obtain information about the participants or organisations involved and their opinion about the study. Questions can be asked prior to the study (e.g., to obtain data about background and expectations of the participants) and afterwards (e.g., to obtain their opinion about the effect of the research environment on their way of working). The return rate is usually near 100%, because the questionnaire is part of a larger study. This also implies that the questionnaires can generally be more extensive and can contain more open-ended questions than survey questionnaires. Nevertheless, to obtain valid, comparative and useful data, the questions should be formulated and the questionnaire constructed in an equally thorough way as survey questionnaires. Although the researcher may be available, help should not be necessary.

**A.4.8 Interviewing**

The purpose of interviewing is similar to that of questionnaires (see previous section) but are done carried out face to face. They are used to collect thoughts, beliefs, opinions, etc., about past, present or future facts and events, with a focus on data that cannot be observed or was not captured in the past. The interview should “provide a framework within which respondents can express their own understandings in their own terms. One of the greatest obstacles to overcome is unlearning the bad habits practiced and reinforced daily in our ordinary conversations, such as lack of depth, miscommunication, lack of clarity in our questions, interrupting the answers given, and lack of direction in the dialogue” (Patton 1987). Not only the questions asked, but also the interviewer has a large influence on the process and outcome of the interview. Patton (2002) provides useful guidelines for formulating interview questions and conducting in-depth interviews. He distinguishes the following types of interview:

- The *informal conversational interview*, in which questions are generated spontaneously. This is often part of participant observations.
- The *interview guide*, which is a list of questions or issues generated prior to the interview. These issues are to be explored during the interview and do not prescribe the precise questions. The list allows the same topics to be covered across interviews, and ensures coverage.
- The *standardised open-ended interview*, which consists of carefully worded questions (a questionnaire) that are asked to all interviewees in the order given in the list. This allows easier comparison, and can be very useful when multiple interviewers are employed. When time is limited, it might also ensure that all topics are covered. The flexibility to pursue topics that emerge unexpectedly is limited.
- The *closed quantitative interview* (multiple-choice questionnaire) in which questions and answers are determined in advance. The interviewee only chooses an answer. Although this eases analysis, it does not necessarily capture the experiences and opinions of the interviewees.

The latter three are often referred to as *unstructured, semi-structured* and *structured*, respectively. The informal conversational interview is usually not considered a real interview. Structured interviews are easier to analyse and
compare. Unstructured interviews are more suitable for an exploratory study. Interviews can be used as the main data-collection method and in conjunction with real-time studies, as discussed in the previous section on questionnaires. Compared to the anonymous questionnaires, interviews are potentially more confrontational and might require more effort.

Focus groups are group interviews that focus on a specific topic. The group dynamics can enhance the overall outcome of the interview, but may have a negative effect on the contribution of some participants, depending on the person, the topic, and the differences in status of the participants.

The questions and the structure of the interview should follow the same careful preparation as questionnaires, and the same additional questions are necessary to obtain information about the context (see previous section).

Tape recording interviews supports analysis because it captures what has actually been said and the intonation used, but also supports the interviewer who can concentrate on the interviewee and the direction of the interview, rather than on taking notes. Note taking will remain important to record potentially relevant observations and – in semi- or unstructured interviews – to have an overview of the direction of the interview, which can act as a reminder of aspects that need to be addressed or clarified. The latter will avoid the need to interrupt the interviewee (‘Before I forget …’).

Interviewing can reveal many of the reasons for behaviour that cannot be observed and only partially be derived from simultaneous verbalisation, team discussions and surveys. Disadvantages are similar to those of questionnaires, but with interviews, the ability to verbalise one’s thoughts also plays an important role. According to Mintzberg (in Bessant (1979)) there is no evidence to suggest that people can effectively translate complex reality into meaningful abstraction. The results rely on the verbalising capacities of the observed, and the researcher’s ability to interpret what has been said. A detailed discussion of this multiple translation problem in interviewing can be found in Ackroyd and Hughes (1981).

There are six types of questions that can be asked to people, in particular in interviews (Patton 2002):

- experience/behaviour;
- opinion/belief;
- feeling;
- knowledge;
- sensory (what you see, etc., used to find out the stimuli the interviewee is subject to);
- background and demographic.

Regarding the order of questions, it is useful to start with some non-controversial, easy questions. One should avoid asking a long list of background questions right at the beginning as these are considered particularly boring. Their number should be minimised (what do we really need to know) and distributed throughout the interview. It is also important to keep interviewees focused on relevant issues, as they may drift off into topics they would like to discuss, e.g., because these are particularly important for the interviewee at the time of the interview. A related problem is that interviewees may tell what they think the interviewer will want to
hear, rather than their own opinion or experience. This is a particular problem when support is evaluated that has been developed by the researcher. A focus on the research questions, a good ear, and practice are needed to overcome these potential disadvantages.

Methods and techniques for the analysis of spoken data can be found under the keywords ‘conversation analysis’ or ‘discourse analysis’.

A.4.9 Action Research

Action Research is an approach to introducing and evaluating change, originally in organisations and programmes, but increasingly in design (e.g., Björk (2003)). Action research has the dual aims of action and research. Through cycles of action and research a better understanding is obtained, while at the same time the organisation or programme under investigation is gradually changed. Action research is usually qualitative, data driven, participatory, and makes use of multiple data sources. “Action researchers help transform inquiry into praxis, or action. Research subjects become coparticipants and stakeholders in the process of inquiry. Research becomes praxis – practical, reflective, pragmatic action – directed to solving problems in the world. … Together, stakeholders and action researcher co-create knowledge that is pragmatically useful and is grounded in local knowledge” (Denzin and Lincoln 2005).

This approach was developed as a reaction to the failure of social sciences to produce results that were useful in solving society’s problems. The close relationship with practice has its effect in the practical relevance of the work and its similarity with consultancy work. The combined responsibility for actual change as well as for research is demanding and the demands on responsiveness and flexibility are high. Furthermore, action research often emphasises local relevance (that is, responsiveness) at the cost of global relevance (that is, generalisation) (Dick 1997). Some specific developments of Action Research are the Critical Action Research Approach of Carr and Kemmis (1981), the Evaluation Methodology of Guba and Lincoln (1989), and in particular the Soft Systems Methodology (SSM) of Checkland (1999).

A.5 Statistical Analysis

The aim of this section is to introduce the terminology needed to select the most suitable methods for analysing quantitative or quantified qualitative data; in particular when this data is nominal or ordinal, and when there are no assumptions about the distribution of the population out of which this data has been taken. A useful book is Vogt (1999) about statistical concepts and methodological terms in

---

27 Dick (1997) gives a very practical overview of how to do action research, summarizes the main methodologies that can be applied and provides an extensive list of the literature with short descriptions.
social and behavioural science providing non-technical definitions of these terms and concepts with examples.

Statistics can be divided into *descriptive statistics* that enable the researcher to describe and analyse data without drawing conclusions or inferences about a larger group, and *inferential or inductive statistics* that enable decisions or inferences to be made about a larger group by interpreting data patterns. Because such inferences cannot be absolutely certain, the language of probability is used in stating conclusions. All statistics can be used for interval and ratio scales, but far fewer can be used for nominal and ordinal scales (see Section 4.7.2).

The analysis methods can be divided into univariate, bivariate and multivariate depending on the number of variables taken into account. Which specific method within these three groups are most suitable, depends on the scale of the variable and on the distribution of the data.

Typically, *univariate analysis* methods look at only one variable and are descriptive: frequency distributions, measures of central tendency such as mean and median, basic measures of tendency, such as variance and standard deviation, and type of distribution.

*Bivariate analysis* methods measure the relationship between two variables. The first step is usually to construct a bivariate table, placing the categories of the variables along the two axes.

*Multivariate analysis* methods measure relationships between multiple variables. Two types of methods can be distinguished; methods to verify relationships between dependent and independent variables, and methods to discover relationships between variables when the variables have not been divided into dependent and independent. This reflects the difference between an hypothesis-driven and a data-driven approach. Table A.5 lists the basic multivariate methods to verify relationships. A premise is that the data is numerically coded.

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>Independent variable</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>interval or ratio</td>
<td>interval or ratio</td>
<td>Regression analysis</td>
</tr>
<tr>
<td>interval or ratio</td>
<td>nominal</td>
<td>Variance analysis</td>
</tr>
<tr>
<td>nominal</td>
<td>interval or ratio</td>
<td>Discrimination analysis</td>
</tr>
<tr>
<td>nominal</td>
<td>nominal</td>
<td>Contingency analysis</td>
</tr>
</tbody>
</table>

Contingency analysis is one of the more frequently used set of methods when verifying hypotheses in design research, since much of the qualitative data can only be coded on a nominal scale. Of these, the $\chi^2$ (Chi-square) method is one of the most widely known. These methods assume that it is possible to observe the variables involved. When use is made of hypothetical constructs that cannot be observed, such as motivation, LISREL (Linear Structural Relationships) is a
suitable method. Conjoint Measurement is often used when the dependent variable is measured on the ordinal scale.

The following methods are used to discover dependencies:

- **Factor analysis**, to reduce a large set of variables into fewer ‘core’ variables. These variables are often constructs; it may not be possible to measure these directly. Examples of such core variables are quality, experience or motivation.
- **Cluster analysis** does not group variables, but cases (units of analysis). All cases in a cluster are more similar to each other than to those in other clusters.
- **Multidimensional scaling** is similar to factor analysis but is based on a similarity or dissimilarity assessment between cases using, for example, pairwise comparison, rather than on evaluating each property of a case. It allows one, e.g., to determine how a participant sees a unit of analysis.

The statistical tests included in software such as spreadsheets usually incorporate assumptions about the underlying distribution of data, that is, the parameters of the population from which the sample is drawn (such as normal distribution) and require that the variables are measured at least at an interval scale. **Non-parametric statistics** are so called because they make few or no assumptions about the distributions, and do not require interval scales. There is at least one non-parametric equivalent for each parametric test, as shown in Table A.6.

Among the most often used is the $\chi^2$ (Chi-square) method based on contingency or cross-tables and its variants. Also commonly used is the Spearman’s rank correlation coefficient, as the alternative to the standard Pearson product-moment correlation coefficient for which the variables have to be metric. In choosing a method, not only the type of comparison is relevant, but also the type of data and, most importantly, the minimum required number of cases.

A few words about the sample size required for a particular study. The relevance of statistics and hence of sample size, is related to the approach taken, i.e., the paradigm chosen. The most suitable sample size depends on many factors, but foremost on the research questions and hypotheses to be addressed. For that reason, a study with a sample size of ‘1’ can be as valuable as a study with a sample size of 1500, but only if the research questions, the research methods, and the conclusions are in line with the sample size.

In particular in quantitative approaches, the calculation of the sample size is essential to be able to draw conclusions. The calculation is mainly based on the expected variability of the data and on the difference or precision one considers relevant. If the testing of an hypothesis requires a certain difference to be measured, e.g., a difference in quality of the products analysed, or an improvement caused by a new support, the study should be designed such that a difference can be detected and that – given a certain level of confidence – this difference is not coincidental and is relevant. The so-called **power** is the probability that the study will successfully detect a difference. When the variability of the data is not known and cannot be estimated, it may be necessary to run a pilot study. In more exploratory studies, the sample size should ensure that the estimates from the study have adequate precision. Many statistical methods demand a minimum sample size
(either overall or per category). In various books and websites, methods to calculate sample size or power can be found.

**Table A.6** Non-parametric alternatives to parametric statistical tests (Burke 1998)

<table>
<thead>
<tr>
<th>Types of comparison</th>
<th>Parametric methods</th>
<th>Non-parametric methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differences between independent groups of data</td>
<td>t-test for independent groups</td>
<td>Wald–Wolfowitz runs test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mann–Whitney U test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kolmogorov–Smirnov two-sample test</td>
</tr>
<tr>
<td></td>
<td>(ANOVA/MANOVA)</td>
<td>KruskalvWallis analysis of ranks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Median test</td>
</tr>
<tr>
<td>Difference between dependent groups of data</td>
<td>t-test for dependent groups</td>
<td>Sign test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wilcoxon’s matched pairs test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>McNemar’s test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\chi^2$ (Chi-square) test</td>
</tr>
<tr>
<td></td>
<td>ANOVA with replication</td>
<td>Friedman’s two-way ANOVA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cochran Q test</td>
</tr>
<tr>
<td>Relationships between continuous variables</td>
<td>Linear regression</td>
<td>Spearman R</td>
</tr>
<tr>
<td></td>
<td>Correlation coefficient</td>
<td>Kendall’s Tau B</td>
</tr>
<tr>
<td>Homogeneity of variance</td>
<td>Bartlett’s test</td>
<td>Levene’s test, Brown and Forsythe</td>
</tr>
<tr>
<td>Relationships between counted variables</td>
<td>Coefficient Gamma</td>
<td>$\chi^2$ (Chi-square) test</td>
</tr>
<tr>
<td></td>
<td>Phi coefficient</td>
<td>Fisher exact probability test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kendall coefficient of concordance</td>
</tr>
</tbody>
</table>
Appendix B

Prescriptive Study Methods

A wide range of methodologies and methods exist to support the development of products. A similarly wide range exists for the development of software. Design support can be seen as a product that primarily deals with information and is increasingly in the form of software. Therefore, both the above methodologies are potentially useful for support development. This appendix outlines some of these methodologies and methods, and provides pointers to more detailed sources. Some are general approaches that can be beneficial throughout the support development process. Others are task specific, or particularly useful for the development of computational design support.

The first two sections of this appendix (Sections B.1 and B.2) focus on product development and software development, respectively. In each section, first an overall development methodology is outlined and then a list of methods for supporting each stage of the methodology is given, with references to further literature. Section B.3 discusses methods specifically for user-interface design. Section B.4 presents a checklist to aid the documentation of the scope and assumptions of the support throughout the development process.

B.1 Product Development Methodologies

Product development methodologies propose that designing should be done in a series of stages, progressively detailing the product under development. An example is the approach of Pahl and Beitz (2007). At each stage a series of steps is proposed to lead the designer from problem understanding to solution. An example is the series of steps suggested by Roozenburg and Eekels (1995): analysis, synthesis, simulation, and evaluation and selection. The following sections provide a list of methods for each of these steps. Some methods are more suitable for the earlier stages of development, while others are for more detailed stages. The listed methods have been selected on the basis of their relevance for design support development. Unless otherwise specified, the methods are taken from the following books: Pahl and Beitz (2007) (abbreviated as PB); Roozenburg and Eekels (1995)
(abbreviated as RE); Jones (1970) (abbreviated as J); Cross (1994) (abbreviated as C).

B.1.1 Methods for Analysing Objectives and Establishing Requirements

The following (Table B.1) is a list of methods that help analyse objectives, clarify the requirements that the support should fulfil, the relationships between the requirements, and their relative importance.


<table>
<thead>
<tr>
<th>Method</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stating objectives [PB, RE, J, C]</td>
<td>To identify the external conditions with which the support must be compatible</td>
</tr>
<tr>
<td>Literature search [J]</td>
<td>To find published information that can be useful</td>
</tr>
<tr>
<td>Interviewing users [J]</td>
<td>To elicit information known only to the users of the intended or existing support</td>
</tr>
<tr>
<td>Questionnaires [J]</td>
<td>To collect information from the members of a large population</td>
</tr>
<tr>
<td>Investigating user behaviour [J]</td>
<td>To explore the behaviour patterns and to predict the performance limits of potential users</td>
</tr>
<tr>
<td>Interaction matrix [J]</td>
<td>To permit a systematic search for relationships between elements within a problem</td>
</tr>
<tr>
<td>Interaction net [J]</td>
<td>To display the pattern of relationships between elements within a problem</td>
</tr>
<tr>
<td>Classification of design information [J]</td>
<td>To split a design problem into manageable parts (this should help solve the problem as well as help modularise the support)</td>
</tr>
<tr>
<td>Objectives Tree [RE, C]</td>
<td>To clarify objectives and sub-objectives of the support, their relationships and their weightings</td>
</tr>
<tr>
<td>Function Analysis [C, PB, RE]</td>
<td>To establish the functions required and the system boundary of the support to be designed</td>
</tr>
<tr>
<td>Performance specification [C]</td>
<td>To make an accurate specification of the performance required of a support</td>
</tr>
<tr>
<td>Quality Function Deployment (QFD) [RE, C]</td>
<td>To set targets for the characteristics of the support so that they satisfy user requirements</td>
</tr>
<tr>
<td>Specification writing [J]</td>
<td>To describe an acceptable outcome of the planned development process (can be useful in writing the future situation expected of a support)</td>
</tr>
<tr>
<td>Design specification procedure [RE]</td>
<td>To specify the requirements by listing, analysing and editing objectives</td>
</tr>
</tbody>
</table>
B.1.2 Methods for Synthesising Support Proposals

The list of methods in Table B.2 could be used at various stages of synthesis in support development. The use of some of these may be straightforward in the context of generating design support; others may require adaptation. The methods could help generate a variety of alternative proposals for fulfilling the individual requirements of the support, and help combine these into overall proposals.


<table>
<thead>
<tr>
<th>Method</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brainstorming [J, RE], Brainwriting [RE], Checklists [RE]</td>
<td>To stimulate a group of people to produce many ideas quickly</td>
</tr>
<tr>
<td>Synectics [J], Random stimulus [RE, C], Intermediate impossible [RE], Concept challenge [RE]</td>
<td>To direct the spontaneous activity of the brain and the nervous system towards the exploration and transformation of development problems</td>
</tr>
<tr>
<td>Removing mental blocks (Adams 1993)</td>
<td>To find new directions of search when the space searched has yielded no acceptable solution</td>
</tr>
<tr>
<td>Function-Means Tree (Hubka and Eder 1988)</td>
<td>To develop functions and means to fulfil the functions together.</td>
</tr>
<tr>
<td>Morphological charts [J, C, PB, RE]</td>
<td>To widen the area of search for solutions</td>
</tr>
<tr>
<td>Value engineering [C]</td>
<td>To increase or maintain the value of a support to its user whilst reducing its cost to its developer (Can be particularly useful when modifying an existing method).</td>
</tr>
<tr>
<td>Functional innovation [J]</td>
<td>To find a radically new type of support capable of creating new patterns of behaviour and demand</td>
</tr>
<tr>
<td>System transformation [J]</td>
<td>To find ways of transforming an unsatisfactory support so as to remove its inherent faults</td>
</tr>
<tr>
<td>Alexander’s method of determining components [J]</td>
<td>To find the right components of a structure such that each component can be altered to suit future changes in the environment</td>
</tr>
</tbody>
</table>

According to Pahl and Beitz (2007), synthesis methods can be classified into intuitive and discursive methods. Cross (1994) calls these creative and systematic methods respectively.

Roozenburg and Eekels (1995) divide the creative methods into association methods and creative confrontation methods. Association methods, such as
brainstorming, encourage spontaneous reactions to ideas expressed earlier. Creative confrontation methods are – like association methods – characterised by connecting initially unconnected ideas, but such connection is now ‘forced’ by a particular step in the method. An example is Synectics.

Systematic methods are based on the systematic analysis and description of a problem, the drawing up of a variety of solutions to sub-problems, and systematic variation and combination of these sub-solutions into solution variants.

In Table B.2, the first four rows list some well-known creative methods, while the rest are a sample of available systematic methods.

**B.1.3 Methods for Simulating Support Behaviour**

The following list of methods (Table B.3) support simulation of designs at various levels. Roozenburg and Eekels (1995) distinguish four kinds of models: structure models (e.g., flow diagrams), iconic models (e.g., dummies, scale models or prototypes), analogue models, and mathematical models.


<table>
<thead>
<tr>
<th>Method</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of interconnected decision areas (AIDA) [RE]</td>
<td>To identify and evaluate all compatible sets of sub-solutions</td>
</tr>
<tr>
<td>Failure modes and effects analysis (FMEA) [RE, PB], fault-tree analysis [RE]</td>
<td>Methods for analysing reliability of a new support by finding possible causes and effects of failure early in the development process</td>
</tr>
<tr>
<td>Simulation of product form [RE]</td>
<td>To simulate the look and feel of the support to provide insight into its semantic and aesthetic functions. Can be particularly useful for user-interface design, the look of a workbook, the layout of a checklist, etc.</td>
</tr>
<tr>
<td>Business and economic simulation [RE]</td>
<td>To analyse the costs and benefits of a new support</td>
</tr>
<tr>
<td>Value analysis [RE, C, PB]</td>
<td>To analyse functions and sub-functions of a support and compare their value to their costs</td>
</tr>
<tr>
<td>Ergonomic simulation [RE]</td>
<td>To use a support model and a human model together to simulate ergonomic aspects of using the support</td>
</tr>
</tbody>
</table>

Rules for development and interpretation of *structure models* are often intuitive. Such models may be useful in determining the ways a proposed support should work, especially at the early stages of its development. Structured design methodologies in software development have many such methods, see Section B.2. In *iconic models* the properties of the design are represented in the model using the same properties. In support development, a scaled-down version may be developed
Appendix B: Prescriptive Study Methods

(say for developing a constraint propagation algorithm for a hundred constraints rather than the required thousands of constraints). This can be used for testing the feasibility of the idea before developing full-scale prototype software. Analogue models use a different property to represent a given property of the original. For instance, a method for finding an optimum may be seen as analogous to finding the best path to move among a tree of paths, which in turn can be seen as hill climbing. The hill-climbing algorithm can then be used as an analogue model of the original task, which, once solved, could give an answer analogous to the original. Mathematical models are symbol models that can be analysed using rules from mathematics. For instance, the exhaustiveness of combinations produced by an algorithm could be calculated mathematically.

B.1.4 Methods for Evaluating and Selecting Support Proposals

The following is a list of methods (Table B.4) commonly used for evaluation and selection of proposals at various stages of the development process, in particular for identifying the right criteria for evaluation and assigning appropriate relative importance, as these are often non-trivial tasks.


<table>
<thead>
<tr>
<th>Method</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checklists [J, PB]</td>
<td>To enable support developers to use knowledge of types of requirements generally found relevant in similar situations</td>
</tr>
<tr>
<td>Estimating weighting factors [PB, RE]</td>
<td>To indirectly assess the importance of criteria by externalising the decision-maker’s preference for hypothetical alternatives</td>
</tr>
<tr>
<td>Identifying and selecting criteria [PB, J]</td>
<td>To decide on how to recognise acceptable support proposals</td>
</tr>
<tr>
<td>Ranking and weighting [PB, RE, J]</td>
<td>To compare a set of alternative support proposals using a common scale of measurement</td>
</tr>
<tr>
<td>Estimating evaluation uncertainties [PB]</td>
<td>To estimate the reliability of evaluation results</td>
</tr>
<tr>
<td>Searching for weak spots [PB]</td>
<td>To identify the weak areas of a support variant</td>
</tr>
<tr>
<td>Ordinal methods (majority rule, Copeland rule, rank-sum rule, lexicographical rule, Pugh Charts) [RE]</td>
<td>To rank alternatives per criterion on an ordinal scale and compare the alternatives against a list of criteria and their importance</td>
</tr>
<tr>
<td>Cardinal methods [RE]</td>
<td>To rank alternatives by quantifying judgement of the effectiveness of the alternatives and the importance of the criteria on an interval scale</td>
</tr>
</tbody>
</table>
B.2 Software Development Approaches

Software development approaches have much in common with design approaches. They too have some main stages such as specification, development and validation. However, software development has its own idiosyncrasies over and above those involved in product development, that are relevant for the development of computational design support. Section B.2.1 discusses CaeDRe, a methodology specifically developed for supporting computational design support development that uses DRM as one of its underlying bases. The two generic software development paradigms, functional and object-oriented are outlined in Section B.2.2, and some generic methodologies in Section B.2.3. One of the methodologies, the waterfall model is discussed in more detail in Section B.2.4. Realisation of design support may require use of generic technologies drawn from areas such as artificial intelligence; some of these are discussed in Section B.2.5. The last section (Section B.2.6) focuses on computer-aided software engineering (CASE) tools that are available for some of the methodologies discussed previously. Unless otherwise stated, the information provided here is taken from the classic book of Sommerville (2006).

B.2.1 A Design Support Software Development Methodology

This section describes CaeD (computer-aided engineering design research methodology) (Bracewell et al. 2001) and its associated computational environment CaeDRe (computer-aided engineering design research environment) (Bracewell and Shea 2001). Figure 5.1.4 in the main part of this book illustrates the methodology. CaeD is a “practical methodology for computer aided engineering design research” that is intended to enable the development of useable computational design support early on in research projects. Its application is “intended to enable rapid, robust implementation of research design methods suitable for empirical evaluation in academic and industrial settings”. The methodology uses DRM as its underlying basis, and makes extensive use of software and social science technologies.

CaeDRe is developed in response to the difficulties in practical software implementation that they see as a major reason why the observed integration and evaluation of computational design support tools resulting from fundamental design research (Culley 1999) is such a problem. Some of the causes behind the difficulties of practical software implementation, according to Bracewell and Shea (2001), are “ignorance of, or failure to apply, fundamental software engineering principles”, while others are “specific to the particular nature of computational tool design research”. The methodology is a variant of the evolutionary or prototyping methodology (Section B.2.3).

The research process supported by the CaeD methodology follows the four stages of DRM. In the first stage, called ‘Criteria’28, Measurable Success Criteria for the tool are identified linking back to overall business objectives. In the second

28 Based on the name we earlier used for the Research Clarification stage.
stage, Description I, the existing design process is analysed to discover relations between Measurable Success Criteria and the actual design process, thus identifying where application of a design support could lead to improvements in this process. In the third stage, Prescription, insights gained in Description I are used to create a storyboard for an improved design process that could result from using the new design support. For computational design support, this storyboard creates a starting point for specifying and implementing a prototype system. Finally, in Description II, the design support is tested to determine whether it works as intended and whether it actually impacts the Measurable Success Criteria.

The authors divide the support development process into five activities:

- task definition;
- choice of representations;
- choice of methods;
- definition of visualisation, interaction and distribution strategies;
- theoretical and experimental validation.

CeaDRe uses the product platform concept as “a set of sub-systems and interfaces that form a common structure, from which a stream of derivative products can be efficiently developed and produced”. Using the product platform concept emphasises that individual software solutions should have interface definitions that allow them to form a flexible but coherent architecture.

CaeDRe “provides an open, flexible environment for the development of computational design support, allowing progressive code-hardening from scripts to robust efficient compiled software for third party applications, using a choice of development tools (high level languages, tools and integration platforms) that are usable by researchers and programmers with a wide range of software expertise without creating barriers to system integration”. Its architecture is a modular system of client, server and extension packages, and allows an iterative, rapid prototyping approach to the solidification and testing of exploratory research ideas.

### B.2.2 Software Development Paradigms

There are two major paradigms in software development: function-oriented and object-oriented.

The *function-oriented paradigm* relies on decomposing the system under development into a set of interacting functions with a centralised system state shared by the functions. Functions may also maintain local state information but only for the duration of their execution. A functional approach is most suitable in systems where the amount of system state information is minimised and information sharing is explicit. Systems whose responses depend on a single stimulus or input and that are not affected by input histories are naturally function-oriented. Functional approaches were practised informally since the early days of
programming, but were only developed into a formal paradigm in the seventies (also called ‘structured’ approaches). A number of analysis, design and programming methods are available within this paradigm that help identify, develop and implement the functions that are necessary. The approach of Yourdon (1989) is a typical example.

The approaches based on the object-oriented paradigm differ from the functional approaches in that they view a software system as a set of interacting objects with their own private state, rather than as a set of functions that share a global state. Objects are abstractions of real-world entities that are responsible for managing their own private state and offering services to other objects. They are independent entities that may be readily changed because state and representation information is held within the objects. System functionality is expressed in terms of operations or services associated with the objects. According to Sommerville (2006) object-oriented systems are easier to maintain and change as objects are independent, and there is a clear mapping between real-world entities and their controlling objects in the system. This improves the understandability and hence maintainability of the system. A number of analysis, design and programming methods are available that help identify and develop objects and their interactions (Booch et al. 1994; Coad and Yourdon 1990; Jacobsen et al. 1993).

Since the late 1980s, object-oriented approaches have become increasingly popular, also with some of the protagonists of the function-oriented approaches, such as Yourdon. He mentions the following three difficulties from which the function-oriented (structured) approaches suffer (Yourdon 1990):

- Function-oriented approaches place an enormous emphasis on the modelling of functions, and little on that of data. Although entity–relationship diagrams have been introduced to alleviate this problem, many project teams ignore this altogether while modelling user requirements. Object-oriented approaches deliberately package both data and functions together into a single container – the object.
- The diagramming notation in function-oriented approaches provides little mechanism for using reusable components. Through the inheritance mechanism, object-oriented methodologies promote reuse of attributes and methods (functions) of existing objects.
- Due to its history of development in an era when graphical user interfaces were unknown, these methodologies offer little to support user interface development.

However, according to Sommerville (2006), function-oriented and object-oriented approaches should not be treated as competing approaches, but chosen or even combined according to their suitability to the application at hand.

### B.2.3 Generic Software Development Methodologies

There are various generic methodologies for software development. The waterfall model is the most commonly used. It has several stages (Royce 1970), each of which produces its own distinct type of deliverables and associated documentation (see also the next section). The model is a cascade from one stage to another.
The stages in the waterfall model are:

- Requirements establishment: The system’s services, goals, and constraints are established and defined in a manner that is understandable for both users and developers.
- System and software design: the requirements are allocated to either hardware or software systems, and an overall system architecture established. Software design involves representing the system functions into a form that may be transformed into executable programs.
- Implementation and unit testing: the software design is realised in terms of a set of program units. Unit testing is used to verify that each unit meets its specification.
- Integration and system testing: the individual program units are integrated and tested as a complete system to ensure that the software requirements are met. After this, the software is delivered to the customer.
- Operation and maintenance: the system is installed and put into practical use. Maintenance involves correcting errors that were not discovered during development.

A strong advantage of the waterfall model is that it provides a very clear-cut method for managerial control. However, practice has shown that it has a number of disadvantages (Schreiber et al. 2000).

- It is sometimes difficult to see progress in the early stages since these are mainly document-oriented; visible and operational results in terms of software can only be tried much later in the software development process.
- Prefixed stages make changes during the project difficult and costly. Changes can arrive from changed external circumstances, new insights gained during the project, or changing needs and requirements.

An alternative methodology, the \textit{V model}, relates each development stage not only to its immediate predecessor and successor, as in the waterfall model, but also to the related testing stage at the same level of detail (Gram and Cockton 1996). The left leg of the \textit{V} represents the development stages – problem analysis, requirements specification, system design, software design, module design – the right leg of the \textit{V} the testing stages – module testing, integration testing, system testing, acceptance testing, and use and maintenance. The coding stage joins the legs. This shows that acceptance tests have to be created as part of the specification, and used during the final installation. Similarly, software design is at the same level as integration testing, and so on. The V-model, however, is still a variant of the waterfall model in that it does not easily allow backtracking to a phase once development has advanced beyond it (Schreiber et al. 2000).

The \textit{evolutionary or prototyping approach} (Smith 1991) can be seen as the opposite of the waterfall model: it aims to produce practical results quickly in a number of iterative improvements based on learning from the previous cycle. This approach is therefore highly adaptable and experimental. However, this is also its weakness: due to the lack of structure it is not really possible to generate sound project goals and plans in advance. In other words, it is too flexible.
A model that attempts to combine the good features of both the waterfall and prototyping approaches is **Boehm’s model** (Boehm 1988). His model is also known as the spiral approach, combining the linear waterfall approach and the cyclic prototyping approach. The aim is to achieve progress by means of subsequent cycles that may be adapted on the basis of experience gained in earlier cycles. Depending on the situation, one may decide to follow analysis and design as in the waterfall model, or prototyping activities if these are more illuminating or useful.

A variant of the waterfall model, but carried out using formal mathematical methods, is called **Formal System Development Model**. In this model, the goal is to produce a formal mathematical system specification first, and then transform this, using mathematical methods, to construct a program. Verification of the system components is carried out by making mathematical arguments about how these component-functions conform to the specification.

Another methodology gaining increasing acceptance is called **Reuse-based Development**. It is based on the assumption that a significant number of reusable software components exist, and the goal of the system development processes is to integrate these components into a system rather than developing them from scratch. While reuse of available software components is informally encouraged in all software development methodologies, it is formally used in this methodology. The typical steps are: requirement specification, component analysis, requirement modification (to reflect reuse of available components), system design with reuse, systems development and integration, and system validation. Note that the stages are similar to those in the waterfall model, but emphasise maximum reuse of existing software components.

Apart from the methodologies discussed in this section, development methodologies are also available for specific types of software systems, such as knowledge-based systems (Buchanan and Duda 1983) (see Section B.2.5).

### B.2.4 The Waterfall Model

The waterfall model is discussed separately in this section, as it provides a highly detailed approach with a logical breakdown of its stages into steps of increasing detail. These steps can be useful to follow, as long as they are followed flexibly, as aimed in Boehm’s spiral model. Unless otherwise indicated, the description is taken from Sommerville (2006).

**Requirements Establishment**

In software engineering two levels of requirements are considered:

- A requirements *definition* is a statement, in natural language and diagrams, of the expected services of the system and the constraints – including potential users – within which it is expected to operate.

- A requirements *specification*, also called a functional specification, is a structured document that sets out the system services in detail and should be precise.

The requirements definition and specification are developed in four steps:
• Feasibility study: An estimate is made whether, or the extent to which, the identified user needs may be satisfied by using the current technology.
• Requirements analysis: The system requirements are derived, e.g., through observation of existing systems and discussions with potential users.
• Requirements definition: The information gathered during the analysis activity is translated into a document that defines a set of requirements, reflecting what the customer wants.
• Requirements specification: A detailed and precise set of requirements is formulated to act as a basis for a contract between client and system developer.

Table B.5 provides a list of methods, some generally applicable, others with a function-oriented or object-oriented flavour, that aid requirements establishment.

<table>
<thead>
<tr>
<th>Method</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checklist of critical characteristics of software (S)</td>
<td>To provide a list of requirement characteristics to choose from when developing software</td>
</tr>
<tr>
<td>Viewpoint-oriented analysis (S)</td>
<td>To analyse requirements taking into account the viewpoints of all parties involved</td>
</tr>
<tr>
<td>Method-based analysis (VORD) (S)</td>
<td>To analyse requirements using a structured method to understand the system</td>
</tr>
<tr>
<td>Semantic data models (S)</td>
<td>To analyse requirements by identifying the logical form of the data processed by the system</td>
</tr>
<tr>
<td>Object-oriented models (S)</td>
<td>To analyse requirements by representing both data and its processing</td>
</tr>
<tr>
<td>Data-flow models (S)</td>
<td>To analyse requirements by identifying data flows through a sequence of processing steps</td>
</tr>
<tr>
<td>Standard format approach for requirements definition (S)</td>
<td>To define requirements based on a standard format of the requirement and rationale</td>
</tr>
<tr>
<td>Requirements specification approaches (Davis 1990)</td>
<td>A number of approaches to add structure to the specification to reduce its ambiguity</td>
</tr>
<tr>
<td>Software prototyping techniques (N.N. 1989; Smith 1991)</td>
<td>A number of approaches for rapid prototyping of software in order to understand its requirements</td>
</tr>
<tr>
<td>Checklist for requirements validation (S)</td>
<td>To provide a checklist of characteristics that a requirements specification should have</td>
</tr>
</tbody>
</table>

**Software and System Design**

Software and system design is the stage that leads to the transformation of informal ideas into detailed implementation descriptions. Note that this is not yet the actual implementation into software. The outcome is comparable to technical drawings in product development.
A general model to represent a software design is a directed graph. The target of the development process is the creation of such a graph without inconsistencies, where the nodes represent entities such as processes or functions, and links represent relationships between these such as calls or uses. A software design is iteratively developed through a number of different versions, increasing its formality and detail in each version, making it more consistent and complete. Frequent backtracking is needed to correct earlier, less formal and less detailed designs.

The software and system design stage has six, iteratively related, steps:

- **Architectural design**: The sub-systems making up the system and their relationships are identified and documented.
- **Abstract specification**: For each sub-system, an abstract software specification is produced of the services it provides and the operation constraints.
- **Interface design**: The interface of each sub-system with other sub-systems is designed and documented.
- **Component design**: Services are allocated to different components and the interfaces of these components are designed.
- **Data structure design**: The data structures used in the system implementation are designed in detail and specified.
- **Algorithm design**: The algorithms used to provide services are designed in detail and specified.

The last five steps are repeated for each sub-system until the components identified can be mapped directly into programming language components such as packages, procedures or functions. As in the requirements establishment phase, it is possible to use both function-oriented and object-oriented paradigms in this stage.

The design activities in a function-oriented development process are to:

- model the system using data-flow diagrams showing how data passes through the system and is transformed by each system function;
- model how functions are decomposed into sub-functions using graphical structure charts;
- describe the entities in the design and their interfaces;
- describe the control structure of the design using a program description language that includes conditional statements and looping constructs.

The design activities in the object-oriented development process are to:

- identify the objects in the system along with their attributes and operations;
- organise these objects into an aggregation hierarchy that shows how objects are part of other objects;
- construct dynamic object-use diagrams that show which object services are used by other objects;
- specify object interfaces.
Implementation, Integration and Testing

Software implementation involves the actual realisation of the support into software units, that are integrated into modules, sub-systems, and the overall system. Each unit, module, sub-system and the overall support need to be tested to ensure that they work as intended. This involves the following steps.

- Implementation of units: implementing individual program units and ensuring that they work as intended.
- Implementation of modules: integrating individual units into modules, and ensuring that the modules work as intended and no unanticipated interactions between the units occur.
- Implementation of sub-systems: integrating modules into sub-systems, and ensuring that the sub-systems work as intended and no unanticipated interactions between the modules occur.
- System implementation: integrating sub-systems into the overall system, and ensuring that it works as intended by the requirements and no unanticipated interactions between the sub-systems occur.

As these steps show, implementation and testing must go hand in hand. The commonly used cycle is called debugging, which has the following steps.

- Implement or modify the program: a provisional version of the intended program is implemented.
- Test the program: the program is run and output collected.
- Evaluate the output: the output is analysed for its correctness, possible errors detected, possible causes hypothesised, and possible remedies proposed.
- Go back to the first step: the remedial proposals generated in the previous step are used for modifying the program in step 1. The cycle is continued until the output of the program is satisfactory.

Software should be tested at each stage. There are two types of testing: verification and validation. Verification involves checking that the program conforms to its specification. Validation involves checking that the program as implemented meets customer requirements. Boehm (1979) summarises these differences as:

- ‘Verification: Are we building the product right?’
- ‘Validation: Are we building the right product?’

Verification and validation can be done using static and dynamic techniques. Static techniques are concerned with the analysis and checking of system representations such as the requirements document, design diagrams and the program source code. Static techniques can only check correspondence between various levels of specification (verification); they cannot demonstrate that the program is operationally useful (validation). Dynamic techniques involve exercising an implementation, and can be applied both for verification and validation as long as an executable program is available. Two types of dynamic testing can be distinguished.
• **Statistical testing** may be used to test the program’s performance and reliability.

• **Defect testing** should find areas where a program does not conform to its specification. This testing is most common during program implementation, where each component developed should be checked against its intended functionality, and modified if it does not provide this functionality. Test-debug-test is the usual program implementation cycle.

Except for small programs, systems should not be tested as a single, monolithic program, but stepwise from unit, to module, to sub-system, to the overall system, as discussed earlier. Once the overall system is tested, *acceptance testing* (sometimes called *alpha testing*) can take place. This involves testing of the system with data supplied by the user rather than simulated test data. This may reveal errors and omissions in the requirements definitions or reveal that the system does not really meet the user’s need because it introduced additional influences not anticipated at the development stages. A further level of testing is *beta testing*, which involves delivering a system, prior to its marketing, to a number of potential customers who agree to use the system. The results may lead to further modification and beta testing, or to marketing. A number of available testing strategies are summarised in Sommerville (2006).

**Operation and Maintenance**

Operation and maintenance issues have to be taken into account during support development and testing (see DS-II, Chapter 6), but these are normally not a stage in the development of support as part of academic research.

**B.2.5 Generic Technologies**

Awareness of generic technologies available for performing various software tasks is necessary for efficient and effective software development. Those discussed briefly in this section are artificial intelligence (AI), expert/knowledge-based systems (KBS), knowledge engineering (KE) and computer-supported collaborative work (CSCW), because of their widespread use in developing computational design support. The technology used has ramifications to the whole research approach. For instance, choosing knowledge-based systems to realise a support might mean using knowledge-acquisition techniques during DS-I and earlier stages of PS.

**Artificial Intelligence**

The area of AI has two main goals (Schank 1990): to develop methods that will make computers far more intelligent and therefore more useful than they are at the moment, and to find out about the nature of intelligence. Many issues that AI deals with are relevant for developing design support and AI is a major source for potential computational solutions.

The two major issues in AI are representations and methods. AI proclaims that good representations are a key to good problem solving as they support explicit,
constraint-exposing descriptions. Methods are procedures that use representations to solve specific problems. In AI, generate and test, mean-ends analysis, and problem reduction are three powerful and generic methods. A wide variety of heuristics is available to be used in conjunction with these basic methods. The methods and their combinations are used for solving a variety of problems in a wide range of applications, such as qualitative and quantitative constraint resolution, optimisation, and learning. Many books on AI have been written: for an excellent introduction, see Winston (1993). The AI in Design Webliography of Brown (2008) provides a collection of potentially useful sources of information on AI in Design, as well as on Knowledge Based Design, Intelligent CAD, Computational Approaches to Design, and Design Theory & Methodology.

**Expert Systems, Knowledge-based Systems and Knowledge Engineering**

Expert systems technology is one of the spin-offs of AI that has found many real-world applications. Expert systems are able to execute a task that, if carried out by humans, would require expertise (Schreiber et al. 2000). Expert systems are typically rule-based systems, built using a set of if-then rules. Kline and Dollins (1989) provide a set of guidelines for designing rule-based systems, helping to answer questions such as:

- What knowledge-representation technique should be used?
- What problem-solving strategy should be employed?

Rule-based systems do certain tasks well, but they do not reason on multiple levels, nor do they use constraint-exposing models. They do not look at problems from multiple perspectives, do not know how and when to break their own rules, and they do not have access to the reasoning behind their rules (Winston 1993).

Expert systems evolved into knowledge-based systems, which refer to programs that require extensive use and explicit representation of knowledge. Unlike expert systems, which are often devoted to automating expert tasks, knowledge-based systems use a variety of ways for representing and processing knowledge, and are often used in supporting rather than automating tasks. In a sense, all information processing systems can be called knowledge systems, since they all use knowledge. However, according to Schreiber et al. (2000), the main distinction is that in a knowledge-based system one assumes that there is some explicit representation of knowledge.

Knowledge engineering is the approach taken to acquire and formalise knowledge, and use this for building knowledge-based systems. Knowledge engineers use domain experts to acquire useful knowledge. Knowledge engineering has three benefits (Schreiber et al. 2000).

- It helps one to spot the opportunities and bottlenecks in how organisations develop, distribute and apply their knowledge resources, and so gives tools for corporate knowledge management.
- It provides methods for obtaining a thorough understanding of the structures and processes used by knowledge workers, leading to a better integration of information technology in support of knowledge work.
It helps, as a result, to build better knowledge systems: systems that are easier to use, have a well-structured architecture, and are simpler to maintain.

The classical model of knowledge-based system development was given by Buchanan and Duda (1983). It contains six steps.

- Identification: Together with the domain experts the knowledge engineer identifies the important aspects of the problem, including participants, problem characteristics, resources and goals.
- Conceptualisation: The key concepts and relations resulting from the previous steps are made explicit.
- Formalisation: The concepts identified are represented in a formal language.
- Implementation: Knowledge from the last step is represented in a system shell.
- Testing: Together with domain experts, the completed system is tested on sample cases and weaknesses are identified.
- Revision: The results from testing, may require redesign and re-implementation, which may involve domain experts.

There are two cases in which domain knowledge is particularly important:

- One is where domain knowledge exists with experts, but for the sake of cost or scarcity of experts, it has to be extracted, organised and even optimised, so as to develop a knowledge-based system. For example, knowledge of best practices of developing a product may have to be acquired, reordered and made available to novice designers. In order to do this, expert designers could be used as a resource to draw upon knowledge, and as a means of validating the resulting knowledge-based system.
- The other case is where the knowledge required does not exist, but in order to generate and evaluate this knowledge, it is necessary to have an understanding of the domain. In our reliability example, for instance, the goal was to create a method for estimating, at early design stages, the potential reliability of a product, but the knowledge required for such estimation was not available. In order to develop and evaluate the necessary knowledge, sufficient understanding of the domain of the technical product considered and its designs had to be developed.

In design research, there are two major reasons for acquiring domain knowledge. One is for developing the support, and the other for its evaluation. The knowledge acquired can take various forms and can have been acquired during the DS-I or PS stages.

- Product descriptions: These can be extracted from real products, detailed drawings of products, product descriptions, sketches, verbal or written descriptions (functional, behavioural or structural) in design catalogues or designers’ documentations. In our synthesis example, for instance, mechanical designs in a number of these forms were collected and analysed.
to extract building blocks for synthesis. In the reliability example, detailed drawings of the sub-assemblies having reliability problems and their failure data were collected in order to identify elements in these designs with inherent problems of reliability, and also to test the validity of the model of reliability so developed.

- **Process descriptions**: These are descriptions of processes used to develop products. In the reliability example, for instance, knowledge was collected about the ways in which reliability is currently calculated in order to contrast these with the desired situation and to include these, as far as suitable, into the intended support.

Various techniques for knowledge acquisition are available, especially from the discipline of knowledge engineering. According to Grosso *et al.* (1999), knowledge engineers are required to:

- become familiar with the problem domain;
- characterise the reasoning tasks necessary to solve the problem;
- identify the major domain concepts;
- categorise the type of knowledge necessary to solve the problem;
- identify the reasoning strategies used by experts;
- define an inference structure for the resulting application;
- formalise all these in a generic and reusable way.

Depending on the type of support developed, several of these steps might be useful to follow.

A number of generic knowledge engineering methodologies exist, the most notable of which is the Common KADS methodology (Schreiber *et al.* 2000). This methodology provides a suite of knowledge models and modelling capabilities so as to answer three types of questions:

- **Why?** These questions help develop the context of the knowledge-based system to be developed: Why is the system a potential help or solution? For which problems? Which benefits, costs, and organisational impacts does it have? These questions are answered by developing models of the organisation, tasks and agents involved.

- **What?** These questions help develop the conceptual description of the knowledge applied in the tasks to be supported, such as: What is the nature and structure of the knowledge and communication involved? The questions are answered by developing models of the knowledge and communication involved, based on the information available in the models resulting from the ‘Why’ questions.

- **How?** These questions help focus on the technical aspects of realising the system: How must the knowledge be implemented in a computer system? How do the software architecture and computational mechanisms look? These questions are answered by developing a design model that provides a technical system specification in terms of architecture, implementation platform, software modules, representational constructs and computational
methods needed to implement the functions laid down in the knowledge and communication models resulting from the ‘What’ questions.

Computer-supported Collaborative Work

Computer-supported collaborative work (CSCW) or groupware may be defined as “hardware, software and processes designed to aid in group related tasks such as basic communication, information sharing, decision making, scheduling/control, and analysis/design” (Saunders 2008). There may be several individuals or groups involved in a collaboration, located in the same or different location and time, and involved in different kinds of processes or activities. Groupware is a term that encompasses many technologies and business process areas. Specific technologies include electronic mail, digital voice mail systems, text conferencing, video tele-conferencing, collaborative databases, workflow, group decision support systems, and living worlds.

Three major points should be considered as the individual groupware technologies are discussed. First, the greatest power from groupware is exhibited when the various technologies can be combined with each other, and be integrated within the business processes of the organisation. Second, success in implementing groupware requires a critical application and a critical mass. Groupware requires group work – the work of all the groups, within an application in which they all participate. And third, the major challenges in the groupware discipline are not technical or economic, but social. This section is based primarily on Saunders (2008). Several books have been written on this topic. An overview of resources can be found in Foraker Design (2005).

B.2.6 CASE Tools

CASE (computer-aided software engineering) systems offer computational support for various aspects of the software development process. They can be classified into (Fuggetta 1993).

- Tools: These support individual tasks such as checking the consistency of a design or compiling a program. They may be stand-alone or grouped into workbenches.
- Workbenches: These support process phases or activities such as specification or design, and normally consist of a set of tools with some degree of integration.
- Environments: These support a substantial part of the software development process, and normally include several workbenches that are integrated in some way, often based on a specific software development methodology.

The most widely used types of workbench are the following (Fuggetta 1993).

- Analysis and design workbenches: These support the creation of models of the system (such as a data flow diagram) in the analysis and design stages of the software development process. These are sometimes called upper-CASE tools. The tools vary from very method specific to general.
• Programming workbenches: These support program development. The main components are assemblers and compilers that translate high-level programming languages to machine code. Other tools are editors, debuggers and printers. These are often referred to as lower-CASE tools.

• Testing workbenches: Testing tends to be application and organisation specific. Consequently, there are not many ready-made testing workbenches. Those that exist tend to be open systems that evolve to suit the system being tested. Tools common in these workbenches include test managers (manages the running and reporting of tests), test-data generators (generates test data for the program to be tested) and simulators (simulates the machines on which the program is to be executed).

The downsides of these CASE supports are the relatively long learning curve and high capital investment. Several books have been written about CASE tools. A list of CASE tools can be found in Wikipedia.

B.3 User-interface Design

This section outlines essential aspects of user-interface design, or human–computer interaction (HCI) which can be a very important part of support development.

Interactive computer systems are built in order to help people achieve some goals as efficiently as possible (Gram and Cockton 1996). The user interface is often the yardstick by which a system is judged, causing at best a high level of user errors to be incurred, at worst, not using the software irrespective of its functionality. Therefore, user interfaces need to be developed sufficiently well so that hypotheses about the functionality of the system that are relevant to the research aims can be evaluated. The more interactive the software, the more important is the role of its user interface.

This section concentrates on the development process for interactive systems, the roles of the users in this process, and the available tool environments.

B.3.1 User-interface Development Issues

A user-interface needs to resolve two key issues:

• How can information given by the user be presented to the support?
• How can the information provided by the support be presented to the user?

Quality of a user interface is measured by two types of properties of the interface and the support. Gram and Cockton (1996) distinguish these two property types as follows. Although they focus on computational support, most of the properties are also relevant to other types of support.

• From the user’s perspective: The interface should be pleasant, reliable, easily understandable, and have sufficient functionality, so that all tasks can be performed with ease. These external properties fall into five categories:
  - goal and task completeness: you can do what you thought of doing;
- flexibility: you can do things in several ways;
- robustness: you can avoid doing things you wish you had not done;
- learnability: the ease with which novice users can acquire competent performance;
- user satisfaction: how a system makes users feel in terms of sense of achievement or excitement.

- From the software engineer’s perspective: The interface, as any other part of the system, should have a number of properties that are defined through software and hardware properties of the system. Particularly relevant are:
  - modifiability: how easy it is to modify the system when facilities have to be extended;
  - portability: how easy it is to change its hardware and software environments;
  - evaluability: how easy it is to evaluate the system against quality goals;
  - maintainability: once installed in an environment, how easy it is to maintain the system;
  - run-time efficiency: whether the system consumes an acceptably low fraction of computer resources;
  - user-interface integratability: how easy it is to integrate the interactive system with existing or new software applications;
  - functional completeness: whether the system has sufficient functionality to support users to do their tasks;
  - development efficiency: whether the most effective use of resources is being made during system development.

Some of these properties are ‘soft’ and can only be defined and measured by taking the user’s cognition and understanding into account; others are ‘hard’ and can be measured by standard software engineering methods.

According to ISO 9241 (ISO 1998), the main standard for working with computers, defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” Effectiveness can be defined as the degree of accuracy and completeness with which the user goals are satisfied. Efficiency is taken as the effectiveness of system usage in relation to its costs in terms of effort or time. Satisfaction relates to user’s comfort or acceptance of the system. Ergonomics of human–system interaction enabled by software should satisfy the following (ISO 1998):

- software should enable solving of the specified tasks;
- software must speak the language of users;
- users should be in control of the software;
- the software should present familiar things in a familiar manner;
- users have a right to err;
- users are different from each other;
- software should qualify.
An important issue in user-interface development is to determine the level of interaction required: how should the user (designer) interact with the support? In the case of computational support development, there can be different possible types of interaction, as discussed in Section 5.5.3 and shown in Figure 5.10. The level of interaction required is important for two reasons. It clarifies and constrains the kind of support that needs to be developed, and it indicates the level of implementation necessary for evaluation of the support.

Other useful standards are ISO/IEC 9126 (ISO 2001) on software product evaluation, and in particular ISO 13407 (ISO 1999) on human-centred design processes for interactive systems.

B.3.2 Levels of Abstraction

Gram and Cockton (1996) distinguish four levels of abstraction for interchanges with an interactive system:

- **Functional level**: This is the highest level of abstraction. At this level the operations or abstract commands and objects provided by the system are described. It is the first level below the ‘task’ level that is considered outside the interactive system. For example, a functional level command maybe ‘start draw program’, or ‘set date and time’.
- **Dialog level**: This level is concerned with the temporal behaviour and the interdependencies among the operations and objects. For instance, the above two functional commands are expanded further into: ‘open DrawImage window’, or ‘select month; advance month; select date;…’.
- **Logical interaction level**: This level expands on how to do the interaction with reference to presentation entities rather than raw device values, and with some generalisation over low-level events. For example, the above two operations are described as: ‘move mouse to DrawImage icon; click mouse’ or ‘move mouse to menu; move mouse to ‘month’ item; click mouse;…’.
- **Physical interaction level**: This is the lowest level of abstraction that describes what really happens during interaction. This level may be unnecessary when the underlying system automatically takes care of its details.

B.3.3 User-interface Development Processes and Methods

The software development models discussed in B.2 must be modified to take into account human–computer interaction aspects in interactive systems development. According to Gram and Cockton (1996) HCI approaches:

- model new aspects for system design by introducing task, performance and conceptual models (the latter describe systems at the functional level);
- introduce new detailed design concerns related to output formatting, interaction techniques, and the use of colour and sound as well as other media and modalities in information coding;
- add new software components especially for the dialog, such as help, history, undoing, macros, tailoring, and tutoring;
- produce new development models with different orderings of development phases, \textit{e.g.}, designing the user interface first;
- create new forms of testing, \textit{e.g.}, formative and summative usability testing;
- give rise to new forms of installation plans, \textit{e.g.}, special training plans for dialog-intensive systems;
- introduce new problems of maintenance, \textit{e.g.}, for self-adaptive systems that change the dialog by exploiting the user’s emerging pattern of usage.

Schematically, following the abstraction levels of Gram and Cockton discussed in the previous section, the development process starts with a task analysis identifying the tasks to be supported. At the functional level, the task steps are conceptualised as abstract commands applied to objects. These are then refined through the remaining levels into specific sequences of renderings and communication devices (such as speech input or output, graphic displays, mice, tablets, \textit{etc.}) at the physical interaction level.

An important issue in designing interactive systems is keeping the software components for user-interface functions separate from those of the rest of the system, which may be termed the functional core (Gram and Cockton 1996). The functional core provides the computational realisation of the problem domain functionality for an interactive system, while the user-interface components represent this functionality to end-users and support them in the use of these representations. The term UIMS (user-interface management system) was coined in an attempt to promote this concept.

According to Sommerville (2006), an exploratory development is the most effective approach to interface design. This, according to him, must initially lead to creation of paper-based mock-ups before developing into screen-based designs that simulate user interaction. A user-centred approach (Norman and Draper 1986) should be used, with end-users of the system playing an active part in the design process – as evaluators or as co-developers. Sommerville suggests the process for user-interface development shown in Figure B.1.

![Figure B.1 Process for user-interface development (after Sommerville (2006))](image-url)
This process contains three iterative, interrelated activities: analyse and understand user activities, produce prototypes and evaluate prototypes, before implementing and evaluating the final user interface. The prototypes may be at various levels of abstraction.

Various authors have listed design principles for user-interface development. Elaborate guidelines are available in Shneiderman (1998). The following is an example from Sommerville (2006).

- **User familiarity**: Terms and concepts used in the user interface should be drawn from the experience of potential users.
- **Consistency**: Terms, concepts and operations should be consistent; comparable operations should be activated in the same way.
- **Mental surprise**: Users should never be surprised by the behaviour of the system.
- **Recoverability**: Users must be allowed to recover from errors.
- **User guidance**: Meaningful feedback about errors and context sensitive user help must be provided.
- **User diversity**: The interface should enable appropriate interaction for different types of users.

Some methods for design of user interfaces are listed in Table B.6. For a more elaborate overview, see Sommerville (2006).

**Table B.6 Methods for aiding design of user interfaces (Sommerville 2006)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct manipulation <em>(e.g., graphical user interface)</em></td>
<td>To present users with a model of their information space and allow them to interact via direct actions</td>
</tr>
<tr>
<td>Desktop metaphor</td>
<td>To provide consistency and user familiarity by making the interface model analogous to some real-world model that users understand</td>
</tr>
<tr>
<td>Menu systems</td>
<td>To provide user navigation of a large information space while remaining aware of its current position</td>
</tr>
<tr>
<td>Checklist of issues relevant for information presentation</td>
<td>To provide information such that its purpose is fulfilled</td>
</tr>
<tr>
<td>Methods for user guidance</td>
<td>To provide user guidance at various levels</td>
</tr>
<tr>
<td>Checklist of issues in error message design</td>
<td>To provide useful error messages</td>
</tr>
<tr>
<td>Multimedia</td>
<td>To provide multiple media in documents</td>
</tr>
<tr>
<td>Issues in help system design</td>
<td>To provide help of various kinds</td>
</tr>
</tbody>
</table>
B.3.4 Interactive Software Development Environments

The CASE environments specifically aimed at supporting development of interactive software are called Interactive Software Development Environments (ISDE). Many attempts have been made to develop practical tools that assist with the development of user interfaces and with management of the interaction between the user interface and the functional core. Typically the support provided by ISDE varies with the hardware and operating system used, the preferred look and feel, the assumptions about the types of interaction required, and assumptions about how interactive system designers work.

B.3.5 User-interface Evaluation

Since user interface development can have far-reaching (indirect) effects on the user evaluation of the software functionality, it is important that care is taken in evaluating the user interface. As systematic evaluation of a user interface can often be an expensive process, a series of simpler, less expensive evaluations, especially during the software development phase should be used. Some of these are the use of questionnaires collecting information about user’s thoughts about the interface, observation of users at work with the system, or inclusion in the software of code which collects information about the most-used facilities and most-common errors (see also Chapter 6).

Broadly, there are two classes of methods for evaluating user interfaces:
- Predictive methods that can be used very early in the design process, as soon as a specification or even a low-technology prototype is available.
- Experimental methods that can be used when a running prototype or some mock up of the system is available.

Sommerville (2006) provides list of specific methods under these two classes, see Tables B.7 and B.8.

Table B.7 Some predictive methods (Sommerville 2006)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCI-based design heuristics</td>
<td>These allow inspection by specialists for certain technology aspects (principle-based inspection) or for conformance with published style guides (Style conformance inspection).</td>
</tr>
<tr>
<td>Formal methods</td>
<td>These aid in assessing properties, such as using a formal specification of a dialog, to prove that it has some specified properties.</td>
</tr>
<tr>
<td>Cognitive-theory-based methods</td>
<td>These allow inspection by specialists for learning problems (Cognitive walkthrough, see Polson et al. (1992), or use of a cognitive model using Goals, Operators, Methods and Selection rules for a system to evaluate learnability or efficiency of a dialog (GOMS method).</td>
</tr>
</tbody>
</table>
Table B.8 Some experimental methods (Sommerville 2006)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participative design</td>
<td>In these, the user interface and the functionality of the developing system is presented to user representatives.</td>
</tr>
<tr>
<td>Summative evaluation</td>
<td>These involve structured and planned evaluation of the completed software by usability specialists, with measurement against required targets.</td>
</tr>
<tr>
<td>Heuristic evaluation</td>
<td>This involves the informal but planned examination of whether the system fulfils a pre-identified set of heuristic usability criteria (Nielsen 1992).</td>
</tr>
<tr>
<td>Usage observation</td>
<td>This involves semi-structured monitoring and observation of real user’s interaction with the system.</td>
</tr>
</tbody>
</table>

The purpose of experimental methods, even if based on a prototype user interface, is to obtain objective measures of user difficulties and subjective impressions from the user of ease of use, good and bad features, ease of learning, etc. Such user tests need very careful planning and preparation. The inevitable weaknesses of a prototype (e.g., missing functionality) may lead to user frustration if the users are not suitably instructed. Subsequent design must take into account these subjective impressions, as they can be more important than objective measurements.

In contrast to a prototype system, a system functional walkthrough need not be conducted with a computer-based system. It could just as easily be based on low-technology prototyping such as flip charts, recorders or other presentation mechanisms. Developments in participative design have greatly expanded approaches to low technology prototyping (Muller et al. 1993).

Whichever mechanism is chosen to derive user impressions and study the usability of the design, it is important that the process is not merely a single step in the development process. Iterations will be necessary until both software engineers and users are content with the proposed user interface.

B.4 Support Outline: Summarising Scope and Assumptions

The use of design support will always change the working situation, and the effects can be intended or unintended. This has to be taken into account from very early on in the development of design support. Several questions need to be answered: What is the scope of the support? Which tasks are supported? What are the desired effects? How does the work environment change? What is the relation to existing support?

It can be argued that most of the design support developed in academia consists of concepts or prototypes to illustrate new ways of working rather than commercially robust products, and that therefore considerations of implementation, use and maintenance do not need to be considered. Potential exploitation, however,
is a criterion for success of a research project and often a criterion for funding. The earlier all these aspects are considered, the more likely it is that the ideas will find their way into practice.

Unfortunately, it often shows that, although the researcher has quite a good mental picture of the intended support, this is not made explicit (Blessing 1997).

- The information provided in proposals or reports is insufficient to understand the support, let alone to assess it.
- The overall scope and aim are not expressed.
- The assumptions on which the support is based have gradually been forgotten.
- Only the positive effects of the use of the support and the core functionality are mentioned, not the potential side-effects.
- As a consequence, many support proposals are interesting but unrealistic.

The checklist shown in Table B.9 (Blessing 1997) is intended to help summarizing and illustrating the envisaged design support by identifying its scope and the underlying assumptions, as early as during the planning stage. The resulting description clarifies the problem that is addressed, the approach and the possible implications, and can thus allow the intended support and its concept to be more easily understood and assessed. For the researcher the checklist helps to reveal how realistic the envisaged support and whether its scope has to be narrowed. The checklist can be used for drawing up profiles of existing support and hence allows comparison between different supports.

It is recommended to start using the checklist right at the beginning of the PS stage and to continue updating the resulting description during the support-development process. Note that not all aspects mentioned in the checklist are equally applicable to each kind of support.

Table B.9 Checklist for identifying scope and assumptions of design support

<table>
<thead>
<tr>
<th>Area of use</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Aims</td>
<td>What are the underlying aims and objectives of the support? What is the ultimate goal? (general, specific, ...)(scientific, social or both).</td>
</tr>
<tr>
<td>Product type or domain</td>
<td>What type of product or what domain is being served with the support? (general, mechanical, electrical, ...) (aerospace, rehabilitation, ...) (mass, made-to-order,...).</td>
</tr>
<tr>
<td>Process type</td>
<td>What type of design process should be supported? (original, redesign, variant).</td>
</tr>
<tr>
<td>Users and tasks</td>
<td></td>
</tr>
<tr>
<td>Tasks or process to be supported</td>
<td>What are the tasks or processes the support is intended to support? The task to be supported is related to the current way of working, not the future situation. Tasks of direct users and of indirect users (those who maintain, install or use the results) should be considered.</td>
</tr>
<tr>
<td>Functions to be fulfilled</td>
<td>What are the specific functions the support has to fulfil in order to support the task? (An outline of its intended behaviour in terms of input and output and the action on the input).</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Number of users working in parallel</td>
<td>Is the support intended to be used by a single user, a number of individually working users or a group of interacting users? Include direct and indirect users, human as well as computer software.</td>
</tr>
<tr>
<td>User description</td>
<td>Who are the users, including both direct and indirect users. Are the users experts or novices? Are the users familiar with the task supported by the support, or has the support been introduced to provide the knowledge they are lacking or are less familiar with, such as guidelines for manufacturing, or do they only use the data resulting from the support? (users in all life-cycle phases: introduction, customisation, use, maintenance).</td>
</tr>
<tr>
<td>Interface</td>
<td></td>
</tr>
<tr>
<td>User's main role, human computer interaction</td>
<td>What are the roles of the various users in applying the support? How and how often does interaction take place? Is the interaction continuous? Who is sender, who is receiver, who takes main initiative for the various tasks? How active is the tool? How much knowledge and support does the support provide?</td>
</tr>
<tr>
<td>Input characteristics</td>
<td>What type of input is required? (numerical, verbatim, graphical, symbolic), (complete or incomplete), (specification, drawing, models..). Is a template provided?</td>
</tr>
<tr>
<td>Output characteristics</td>
<td>What is the type of output of the support? (Numerical, verbatim, graphical, symbolic, schematic..) (complete, incomplete, ..) (exact, range, fuzzy, ..) (specification, drawing, sketch or other model of the product). What does the output represent? Is it a restructuring of the input, or does it have additional data? Does it provide intermediate answers or a trace of the process? Does the support always provide an output? Is it clear when an output is incorrect?</td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>Customisation</td>
<td>How much is the support tailored to a particular product, process, discipline or company? What has to be customised? What is the expected effort? Who is customising the support?</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Does data stored in the system need to be updated? What type of maintenance is involved? What is the expected effort? Who is maintaining the support?</td>
</tr>
<tr>
<td>Links with other systems or methods</td>
<td>Links to which other systems or methods are required? What links are possible? Which data is needed as input or output for the link to be effective?</td>
</tr>
</tbody>
</table>
Table B.9 (continued)

<table>
<thead>
<tr>
<th>Effects</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needs</td>
<td>What needs from practice are addressed? Why would a company or institution want to use (and purchase) the support? Why would a user want to use the support?</td>
</tr>
<tr>
<td>Problems</td>
<td>What are the particular problems that are expected to be solved?</td>
</tr>
<tr>
<td>Problem-solving method, approach</td>
<td>How is the support expected to solve the problem? What are the procedures involved? What is the rationale behind it? What are the limitations of this approach?</td>
</tr>
<tr>
<td>Expected effect on the work situation (assumptions)</td>
<td>What are the expected effects of the use of the support on the work situation? How does the support intend to change the situation?</td>
</tr>
<tr>
<td>New work situation</td>
<td>What is the new work situation?</td>
</tr>
<tr>
<td>Potential side-effects</td>
<td>What are the potential side-effects of the support?</td>
</tr>
<tr>
<td>Validation</td>
<td>What methods were used or can be used to validate the support and what test data is being or can be used?</td>
</tr>
</tbody>
</table>
Appendix C

Example Research Projects

In this appendix we discuss seven design research projects, which are presented by the supervisors or PhD students involved, and place these projects in the context of DRM.

C.1 Overview of the Examples

The seven design research projects presented in this appendix took place before the methodology was fully developed and are thus not examples of our methodology. The aim of this appendix is to provide a glimpse of the variety of issues tackled in design research, and the variety of research methods and techniques used in tackling these issues. Most importantly, the projects are used as examples to illustrate how different research projects can be placed within the framework of DRM. An overview is given in Table C.1; detailed discussions can be found in the Reflection sections at the end of each project description.

Together, the projects cover many important areas of research: synthesis, design for quality, design for reliability, teamwork, process support, developing metrics for design process performance, and generating methodologies. The projects range from the highly theoretical work of Mørup and Andreasen to the largely computational work by Shakeri et al. Some have a distinct process-oriented flavour: Blessing et al., Mabogunje et al., Frankenberger and Birkhofer, and Shakeri et al. fall in this category. Others carry a substantially product-oriented flavour: Chakrabarti and Bligh, Mørup and Andreasen, and Stephenson and Wallace fall in this category. The aims of the projects range from identifying influences for a particular issue (teamwork (Frankenberger and Birkhofer), quality (Mørup and Andreasen)), through process support (Mabogunje et al., Blessing et al.), to support for specific tasks and activities (Shakeri et al., Chakrabarti and Bligh, Stephenson and Wallace). Methods used include, amongst others, protocol analysis, questionnaires, observation, interviewing, agent-based simulation, historical case studies, noun phrase analysis, and product analyses.
The process of classifying these projects as well as others, using the DRM framework, helped us to improve the descriptions of the stages such that the variety of research projects in design can be represented.

**Table C.1 Classification of the projects in this appendix in the DRM framework**

<table>
<thead>
<tr>
<th>Example</th>
<th>Research Clarification</th>
<th>Descriptive Study I</th>
<th>Prescriptive Study</th>
<th>Descriptive Study II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blessing</td>
<td>Review</td>
<td>Comp</td>
<td>Comp (Initial)</td>
<td>Comp</td>
</tr>
<tr>
<td>Chakrabarti</td>
<td>Review</td>
<td>Review</td>
<td>Comp</td>
<td>Initial (Initial)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Comp</td>
<td>Comp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Comp</td>
<td>Comp</td>
</tr>
<tr>
<td>Frankenberger</td>
<td>Review</td>
<td>Comp</td>
<td>Comp</td>
<td>Comp</td>
</tr>
<tr>
<td>Mabogunje</td>
<td>Review</td>
<td>Comp</td>
<td>Comp</td>
<td>Comp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Comp</td>
<td>Comp</td>
</tr>
<tr>
<td>Mørup</td>
<td>Review</td>
<td>Comp</td>
<td>Initial</td>
<td>Initial</td>
</tr>
<tr>
<td>Shakeri</td>
<td>Review</td>
<td>Review</td>
<td>Comp</td>
<td></td>
</tr>
<tr>
<td>Stephenson</td>
<td>Review</td>
<td>Comp</td>
<td>Comp</td>
<td>Initial</td>
</tr>
</tbody>
</table>
C.2 A Process-based Approach to Computer-supported Engineering Design

Student: Lucienne Blessing (author)
PhD Dissertation, University of Twente, the Netherlands, 1994 (Blessing 1994a)
Supervisors: Harry van den Kroonenberg, Koos Mars, Cees Terlouw

C.2.1 Introduction and Aim of Research

In this century, in the field of mechanical engineering, both products and the process of their creation have undergone major changes. In order to remain competitive, new approaches to product development are needed to cope with the new characteristics of, and increasing pressure on, product and process. This need was addressed in the social aim of the research project:

\[ \text{to improve the mechanical engineering design process, i.e., to realise a more conscious, effective and efficient process.} \]

The following definitions were used. A conscious process is a process that is executed by people who are aware of its structure and use this knowledge in planning and reviewing. An effective design process is a process that results in a product satisfying the actual need. An efficient design process is a process that is effective and in which the applied resources do not exceed the planned resources. Efficiency includes product- and process-related factors; it searches for the best results given the available resources.

The social aim led to an initial investigation into the various means to improve the design process, such as design methodologies and computer tools, in order to determine the scientific aim of the project. The first conclusion was that these means had not had the expected impact on the effectiveness and efficiency of the design process: existing prescriptive models are inadequately exploited; existing knowledge and methods are poorly accessible or unknown; and the range of application and focus of computer-based tools is limited. The second conclusion was that a focus on the design activity (the design process) rather than the more common focus on the design deliverable (the product), along with a focus on computer support of the entire process rather than automation of a specific activity, would be the most promising approach to improve mechanical engineering design. A focus on the design process was expected to increase the impact of existing means; enable the process to be monitored; increase the accessibility of knowledge, methods and tools; and open the way to extend computer support to the entire process.

This led to the decision to address the need to improve design by developing a computer-based support tool based on a model of the design process, combining the advantages of computer processing, the knowledge and capabilities of human designers and the potentials of a methodical approach. The scientific aim of the study then became:
the development and evaluation of a model of mechanical engineering design upon which to build a computer-based support tool, in which the design process is the core.

The hypothesis that the scientific aim helps achieve the social aim was based on the following assumptions, for which the literature was found to provide some support:

- Awareness of a process is considered the first step toward improvement.
- Focusing on the design process (i.e., its activities) influences its effectiveness.
- Focusing on the design process influences its efficiency.
- Computer systems can enhance the effectiveness and efficiency of the design process.

C.2.2 Research Approach

The scientific aim focused on realising a model of mechanical engineering design upon which to base a design support tool. This resulted in the following central research question:

*What model of design can be used to develop a process-based computer support system for mechanical engineering design that improves the design process?*

This research question was then divided into a number of more detailed research questions:

- What are the characteristics of effective and efficient design processes according to prescriptive design literature, *i.e.*, what processes are suggested by design researchers to improve design?
- What are the characteristics of the design processes as they actually take place according to descriptive design literature, *i.e.*, what is the context in which the support system should function and, in particular, what are the characteristics of successful design?
- What are the characteristics of the envisaged type of computer support for design?
- What are the requirements for this type of computer support, *i.e.*, at what combination of characteristics from theory and practice should the system aim?
- What are the elements of a model of design that can be used as the core of this system?

The approach applied to answer the research questions is illustrated in Figure C.1.

The first stage focused on the first two research questions. An extensive study of prescriptive and descriptive literature was undertaken to reveal the characteristics of mechanical design that could be of importance for the development and use of a system to support the improvement of the design process.
Prescriptive literature, such as Pahl and Beitz (1996); French (1971); Asimov (1962), suggests ways of improving the design process by means of models and methodologies. These offer a broader, albeit less detailed, view of the process than descriptive studies. Nearly 30 models were compared. Descriptive literature offers characteristics of design based on observation, revealing the environment in which the developed system would have to function. Over 70 different studies were compared with each other and with prescriptive literature. Because of a lack of detailed studies in industry, a design process in industry was observed for 16 months to find additional characteristics. Both sources were consulted to combine the characteristics of effective and efficient processes suggested by prescriptive literature, with the reality of design practice found in descriptive studies. In total, several hundred characteristics were identified and classified (Blessing 1994b).

At the same time, the general characteristics of computer-based support for design were specified, based on the characteristics of existing design tools (Blessing 1991) and on a personal view of the role of computers in design. A design support system was envisaged to involve the designer as an important reasoning component in solving the design problem; to be subordinate to the designer; to permit designers to apply different approaches; and to allow for multiple users and tasks. Furthermore, the system should support the designer...
throughout the entire process by advising on relevant knowledge, methods, tools and strategies, and it should provide a structure for documenting product and process data. This answered the third research question.

The third stage focused on the fourth research question: the identification of the requirements and the main functions of the support system. The development of the requirements involved a stepwise approach.

First, based on the literature and case study, the identified characteristics of design and support were classified into:

- characteristics to support or stimulate (e.g., the development of a comprehensive requirements list);
- characteristics to prevent or discourage (e.g., the fact that requirements are neglected during evaluation);
- characteristics that cannot be prevented but have to be taken into account (e.g., the fact that requirements change throughout the process).

This provided an initial set of requirements, which was translated into more specific requirements by indicating how the system could support, prevent or take these characteristics into account. The compilation of these requirements extended with the requirements that could be derived from them, resulted in the final set of requirements. The requirements focus on the interaction between system and user. They take into account the design process as it actually takes place and how it could be improved. The final set of requirements was used to define the main functions of the envisaged design support system, which were:

- supporting methodical design activity;
- supporting structured documentation of design data;
- supporting retrieval of design data for reuse;
- structuring and supporting retrieval of knowledge, methods, tools and data of past designs;
- providing two types of context-sensitive advice: assistance (suggesting knowledge, methods, tools and past designs) and guidance (suggesting steps in the process);
- supporting communication and teamwork.

Design data includes:

- **product data**: data describing the product as it evolved, covering versions, all product development stages, the whole product life-cycle, and the alternatives that were considered at any one stage.
- **process data**: the rationale behind the product data (arguments, decisions, design intent)
- **process administration data**: planned and applied resources (who, what, when and how).

The fourth stage of the project involved the development of the model of design (the Design Matrix) and a description of the architecture of the support system built around this Design Matrix. This system was called PROSUS, PROCess-based SUpport System. The aim of PROSUS is to improve the design process by using a
process model to support the capture of the data resulting from design activities and to support the creation of these data throughout the process. Several alternative high-level architectures were generated and evaluated using the main requirements (that encapsulate the way designers design) and the functions that had been formulated. The project did not include the implementation of the system. This stage answered the fifth research question and the first part of the scientific aim.

In the final stage of the project a first evaluation of the concept against the formulated functions and aims took place to determine the applicability and the usefulness of a process-based approach to computer-supported engineering design. This stage contributed to the second part of the scientific aim. The evaluation was based on a comparative analysis of the design processes of designers working with and without the Design Matrices.

C.2.3 Results

The envisaged system PROSUS is shown in Figure C.2.
PROSUS has three levels:

- the primary level, containing the basic building block of PROSUS, the Design Matrix, which represents the design process and is the main working area for the design team;
- a control level to aid in finding the current most promising strategy, that is, the sequence of steps (cells in the Matrix);
- a support level to determine the best resources (people, time, means) for each step.

The chosen model of design, the Design Matrix, is a compilation of models of design proposed in the literature. The granularity of the model was chosen such that it was fine enough to provide the context for the system to be able to act on the input of the designers; and coarse enough not to be a burden to the designers when documenting their process. The model constitutes the system’s generic knowledge of the design process. The Design Matrix represents the design process as a structured set of issues and activities. Figure C.3 shows a simplified design matrix.

![Figure C.3 A simplified design matrix](image)

Each cell can be viewed as a window in which data can be entered, either using the default notebook type interface, or using existing design tools such as requirements capture, drawing or simulation tools. Data can be entered in any order; the sequence of addressing the cells is not prescribed, nor do all cells have to be visited.

The first column contains a generic list of issues for which to generate a proposal. For reasons of clarity, the various issues following ‘Concept’, including material selection, assembly, installation, use, etc. have been grouped in this figure under the heading ‘Detail design’. Each issue can be solved in three steps: Generate, Evaluate and Select. A Generate step results in proposals that address a given issue. Evaluation is the comparison of each proposal with the requirements and results in one or more candidate solutions (decisions) and related arguments. In the case of more than one candidate solution, a Selection has to be made to decide upon the solution to pursue. This activity results in one and sometimes more
solutions (the decision) together with the arguments for this selection. A distinction has been made between Evaluation and Selection because of the nature of these activities. Evaluation focuses to a large extent on the demands in the requirement list, selection focuses more on the wishes (or soft requirements), and will involve making trade-offs. As a consequence, different methods and tools will be used for each of these activities.

Any one design project consists of a set of matrices, all linked together. New matrices are introduced throughout a project, one for each assembly or component that has been considered. This is based on the fact that the process model repeats itself, in a slightly modified form, for every assembly and component that is being generated (the two types of matrices vary slightly). A design matrix thus contains a description of the design process for a specific product element at a certain point in time. The description is structured around the rationale applied and may not be chronological (automatic date/time/person-stamping the entries allows retrieval of the chronological order).

Each of the formulated functions is realised as follows:

- supporting methodical design activity;
  The matrix does not describe how designers do design, nor does it prescribe how they should design. It suggests how designers could design by providing a framework for their collective activities in a project. The matrix and the use of multiple matrices encourages a more systematic approach on a project as well as on the assembly or component level.

- supporting structured documentation of design data;
  The matrix provides a structure for the design team to document and retrieve design data in all stages of the design process, which is enhanced by the use of multiple matrices. In each cell in a Design Matrix users can document the results, both final and intermediate, of executing that particular step, using a notebook-type interface.

- supporting retrieval of design data for reuse;
  The cells in which the data is entered and the matrix provide the context for the system to ‘understand’ what the data is about. That is, the process model (the Matrix) enables the system to determine the context in which the data was generated and to use this to index the data for storage and retrieval.

- structuring and supporting retrieval of knowledge, methods, tools and data of past designs;
  Knowledge, methods, tools and data of past designs (matrices used in past projects) can be linked to specific activities in a design process, i.e., linked to one or more cells of a Design Matrix. This supports integration and is expected to encourage application.

- providing context-sensitive advice:
  - assistance: the Design Matrix provides the context for the system to know in which activity a designer is involved by identifying the cell in which he or she works. This enables the system to advise on relevant knowledge, methods, tools and past designs, because these can be linked to specific activities, that is, to one or more cells.
- guidance: The model as such suggests an overall approach, while allowing different approaches. The Design Matrix in combination with the strategy level of PROSUS will enable the system to suggest which step to take next in a more advanced way.

- supporting communication and teamwork.

As a workbench for the design team all data becomes available as it is being created so that designers can act upon the latest data (apart from the data in the temporary personal notebook area). The comment facility allows suggestions and remarks to be added to entries of others in the team.

C.2.4 Evaluation of the Results

The PROSUS concept was evaluated by evaluating the Design Matrix concept against the formulated functions and aims, prior to computer implementation. The Design Matrix was chosen because it is the core of the system. The applicability and usefulness of the Matrix is crucial to the applicability and usefulness of the system. Evaluation prior to implementation was considered because implementing a system that would be suitable for evaluation would put high requirements on the quality of the user interface (the interface had to allow writing and drawing as easy as in a notebook). Given the available time and computing skills the quality of the interface was likely to have affected the evaluation in such a way that conclusions about the concept could not have been drawn. More importantly, it was considered possible to evaluate the matrix as a concept before actual implementation, even though not all system functions could be evaluated. The solution was to draw the matrices on large sheets of paper on which designers could work, thus avoiding the interface problem. Given the experimental setup, absolute statements about the applicability and usefulness of the matrix concept are difficult, if not impossible, to obtain. Instead, it was decided to focus on obtaining relative statements by comparing designers using the matrices with those who do not use the matrices.

To focus the evaluation, the system functions and the overall aims were translated into evaluation questions. For each of these questions measurable hypotheses were formulated that described the expected effects of the system on the process and on the resulting design. For example, ‘Compared to designers working without design matrices, designers using design matrices document more arguments and decisions when evaluating and selecting their designs’, or ‘Designers using design matrices do not experience difficulties in using the matrices’.

An experiment and a questionnaire were set up to evaluate these hypotheses. The experiment involved eight designers doing the same design task in a laboratory environment. Four designers applied the design matrices (drawn on paper), four designers worked in the usual way without matrices (on blank sheets of paper). The designers were asked to design a wall-mounted swivel mechanism and produce a layout drawing. They were given a one-page problem statement, data about an imaginary workshop and some general handbooks. They worked individually and were asked to think aloud while designing. All processes were recorded on videotape. On average the designers spent 4 hours. The researcher had a list of
possible questions and answers in case designers had any questions about the problem or its context, to avoid different answers being given to different designers. Questionnaires before and after the experiments were used to obtain information about the designers’ backgrounds and their opinions about the experiment. The designers had between 5 and 40 years of experience. The task and set up were a modification of the experiment described in Dylla (1990); Fricke (1993).

The videotapes were transcribed and broken down into events (usually meaningful parts of sentences). Events lasted on average 8.6 seconds. Using a spreadsheet, each event was categorised for each of the issues that needed analysing in order to evaluate the hypotheses. One column, for instance, was used to categorise the events according to whether the event contained an argument and the nature of this argument, another column was used to categorise events according to the cell in which the designer was working. Once categorised, suitable graphs and tables were generated. Given the type of data and the low number of cases (eight) Fisher’s exact probability measure was used to determine significance of the findings.

With respect to the functions of the system, the analysis showed that designers using the design matrices used a variety of approaches. The questionnaires revealed that they did not have the feeling that their ability to work had been strongly restricted because of the design matrices and they felt the design matrices provided a reasonable structure for their design processes. The designers who used the matrices generated more concepts; addressed more issues; documented more arguments and decisions; evaluated more often; and evaluated more continuously throughout the design process. These findings are promising: many typify successful design processes.

With respect to the aims, the designers who used the matrices had slightly more confidence in their solution. Product quality was used as the measure of effectiveness in this experimental setup, which was determined by two independent experienced designers who compared the designs with a set of demands and wishes that were partly derived from the task description and partly generic, such as low wear. No difference could be measured. This result was expected because the subjects were experienced designers and the task chosen was relatively simple because of the time-consuming nature of this type of experiment. Design time was used as the measure of process efficiency in this experiment. Designers who used the matrices took longer than the other designers; but documented more; were on average less experienced (strong correlation between design time and experience was measured); assessed their solutions significantly more often; and addressed more issues, i.e., they covered a larger part of the design process.

The evaluation suggested that the proposed process-based approach to design is applicable. With respect to the functions of PROSUS that could be evaluated in the experiment, the approach was considered useful. The aim of improving the effectiveness and efficiency of the design process could not be proven: possible reasons related to the experimental context were identified. However, the use of design matrices was found to modify the design processes in a way considered likely to contribute to improvement. Based on the evaluation, it seemed worthwhile to implement the concept.
C.2.5 Conclusions About the Research Approach

The research approach developed as the project progressed since no overall approach to design research existed. Research approaches from various non-engineering disciplines had to be consulted and several iterations took place. The use of descriptive studies and the evaluation involving designers were considered extremely worthwhile methods for the development of design tools. The overall approach was found to be sufficiently rigorous to act as the starting point for the design research methodology described in this book.

C.2.6 Continuation

After the project finished, a first implementation of PROSUS was developed. Based on the experience with this implementation a completely revised version is now being implemented to allow a more thorough evaluation of the system.

C.2.7 References

Blessing LTM (1994b) From design characteristics to system requirements. Cambridge University Engineering Department. Technical Report CUED/C-EDC/TR16

C.2.8 Reflections from the DRM Perspective

The work of Blessing et al. has the social aim of improving the mechanical engineering design process by making it more conscious, effective and efficient. With the assumptions that focusing on the design process influences its effectiveness and efficiency, that awareness is the first step towards improvement, and that computers can enhance effectiveness and efficiency of the design process, the scientific aim of this work is to develop and evaluate a model of mechanical engineering design upon which to build a computer based support tool in which the design process is the core. The central question asked is: What model of design can be used to develop a process-based computer support system for mechanical engineering design that improves the design process?

The work is carried out in four stages. The first stage focused on an extensive study of the prescriptive and descriptive literature that could be important for
support development (e.g., what the characteristics are of effective and efficient design processes, successful designs and actual design processes). This was supplemented by detailed observation of a design process in industry (comprehensive DS-I) to identify additional, relevant characteristics of the design process. The chosen Success Criteria are a more effective and efficient design process, the Measurable Success Criteria are increased quantity and quality of documentation (consciousness), high product quality (effectiveness), and reduced design time (efficiency). On the basis of this, a comprehensive PS was undertaken. The Intended Support had the functions of supporting a methodical process, structured documentation, retrieval of data, knowledge methods and tools for reuse, providing context-sensitive advice, and supporting teamwork. A paper-based Actual Support was developed, which was evaluated using a comparative analysis in a laboratory setting of design processes of experienced designers working with and without the Actual Support (comprehensive DS-II). Based on the evaluation, suggestions for improvement of the support were made. The work continued beyond the thesis to develop an initial implementation of the Intended Support (Initial PS).

The work of Blessing et al. is an example of a PhD thesis that carried out comprehensive studies in the stages of DSI, PS and DSII. Using the DRM framework, the project can thus be classified as a Type 7 for the PhD work, as well as for the whole project:

Type 7: RC (Rev) → DS-I (Comp) → PS (Comp) → DS-II (Comp) → PS (Init)
C.3 A Program for Computational Synthesis and Conceptual Design Support

Student: Amaresh Chakrabarti (author)
PhD dissertation University of Cambridge, UK, 1992 (Chakrabarti 1991)
Supervisor: Thomas P Bligh

C.3.1 Introduction and Aim of Research

Conceptual design is an early stage of the design process where design concepts are created, evaluated and explored before being selected for further development. The process is characterised by information that is often imprecise and incomplete. However, it is in this early stage, which is the least resource-intensive among design stages, where most of the product cost is committed. This is why improving conceptual design can have a far-reaching and effective impact on the design process. The broad, long-term aim of this project was therefore to develop means for supporting designers at the conceptual stage in order to develop better products.

An initial study was undertaken into existing literature in order to understand the state of work in this area, and to focus the research aim. The literature studied include characteristics of designers, design process and design aids at the conceptual stage. Design research prescribes that in order to develop better products, designers must generate a wide range of concepts and evaluate these appropriately. Evaluation can only be as good as the criteria used, and many criteria are generated from the positive and negative features of proposed concepts. This means that considering a wide range of concepts should not only help generate more innovative ideas, but also help evaluate these using a wider set of relevant criteria.

However, designers seldom generate more than a few concepts. There are several reasons for this: lack of awareness of (partial) solutions in other designs or domains; bias towards or against specific ideas found useful or difficult respectively in the past; and a limited information-handling ability to detail more than a few ideas, especially since information tends to grow rapidly as ideas are further detailed. Computers are potentially useful for enhancing information handling, and they can be unbiased in terms of the ideas presented. Large databases are easy to create, maintain and use in computers. However, computers are hardly used in this early stage of design. One way to fulfil the aim of this project was to computationally support designers to develop a wide range of concepts, and to help them evaluate these using the widest possible range of relevant criteria.

Existing aids to concept generation were either systematic manual approaches, such as in Pahl and Beitz (1996); catalogues of existing components or designs (Roth 1970); or product development principles (French 1985). These rely heavily on designers as to how effectively they are used, and little domain expertise is reused from the past, except in the case of design catalogues. However, much valuable experience is stored in the designs catalogued, and it was hypothesised that the building blocks that these designs share can be more effectively used if they can be combined freely, rather than remaining part of the designs where they were originally used. Each aid has its merits however. Systematic approaches
prescribing combination of partial solutions into concepts hold the promise that all combinations can be considered without bias. Catalogues of existing solutions, if seen as combinations of common building blocks, promise their reuse, and therefore consideration of a wide range of concepts. Design principles could be used in guiding evaluation of concepts. The specific aim of this research therefore was:

_to develop computer methods for supporting conceptual design by supporting (i) formulation of intended design problem, (ii) unbiased generation of a wide range of concepts to solve the problem by combining a set of building blocks, and (iii) evaluation of these concepts._

However, due to limitations of time, the immediate aim of this research was to focus on the first two of the above sub-aims, with the understanding that evaluation, the third sub-aim, would be left entirely to designers at this stage, without any active support from the computer. The assumption was that fulfilment of the first two aims would support evaluation.

C.3.2 Research Approach

There were four, variously coupled stages in the development of the computer support. The first stage was to sufficiently clarify the required characteristics of the support and its components. Stage II involved developing these components so that these could be used to start developing a flow-chart, and consequently a pseudo-code (stage III) of the eventual program, which could be hand tested by taking example cases. Once this was satisfactory, the pseudo-code could be turned into a real program and tested for these cases (stage IV), before gaining confidence for a more formal evaluation.

In stage I, several alternatives were considered as to how an unbiased generation of a wide range of concepts could be supported on computers. Given that the broad approach was to generate concepts by combining building blocks, generation would require a set of building blocks, and some means for combining these into possible concepts. Several alternative approaches were considered and evaluated. It was decided that a database of a wide range of building blocks, separately developed, would be used for synthesis. In order to make unbiased generation of a wide range of concepts possible, an exhaustive synthesis of these building blocks would be performed by the computer. These solutions would then be offered to designers for consideration, further exploration and evaluation.

This required developing (stage II) a representation of design problems and conceptual solutions, and a program (i.e., a knowledge base of building blocks, and a synthesis procedure) capable of generating an exhaustive set of conceptual solutions to given design problems, both expressed using the representation developed. The research method for developing these was to use as ‘data’ known design solutions and the design problems they solved. The knowledge base of building blocks was to be such that it could be used to adequately describe at least these solutions. The reasoning procedures should be able to generate back at least these solutions.
The method is shown in Figure C.4. First, a set of design problems, in terms of their functions, and their known design solutions was collected. By analysing these for commonality across their functions and means, a representation for these designs and their functions was developed, and a knowledge base of common building blocks to describe the designs created. A reasoning procedure was then developed for combining these building blocks into concepts for solutions to a given design problem. These concepts were then compared with the known designs. It was important that the concepts generated by the computer include the existing designs as well as new concepts. To be generally useful, it must also generate concepts for new designs for design problems not used in the development of the program.

Designs were collected from the literature, particularly design catalogues, for a wide variety of mechanical design problems to determine a common representation. Analysis of these designs revealed that mechanical devices had temporal as well as spatial (topological and configurational) features: they function through time, and at each instant in time they can assume potentially different spatial configurations. However, it was also noticed that the designs analysed have, or can be approximated to simple, orthogonal configurations at some instants of time during their operation. In other words, if the design problems were considered as a combination of a set of instantaneous functions, then a design solution could be synthesised to satisfy one of these instantaneous functions, this could then be checked for satisfaction at the other instants for complete satisfaction of the temporal function. The synthesis process then became two-stepped, one of solving for an instantaneous function, and then for checking for the overall temporal function. This simplified generation of solutions from the temporal perspective, although they had to be synthesised at both topological and configurational levels, as many existing designs used spatial variations of the same principle. Developing representation and the knowledge base of building blocks required, therefore, describing concepts and their building blocks at these levels.

The central issue in developing the database of building blocks was that their exhaustive combination had to provide a wide range of concepts. Several alternatives were considered. The process was to iterate through analysing the designs and consulting the relevant literature, especially design catalogues, trying
to identify the minimum number of building blocks with which these designs could be represented, without any of these building blocks being rigid combinations of the other building blocks. The building blocks developed were named basic elements.

The first step in developing the synthesis procedure was to clarify the relationship between the functions of the building blocks and the overall function of the designs to which they belonged. The approach then was to consult relevant literature and adapt available combinatorial algorithms. Although AI Search literature provided general guidance, there was nothing that could be used directly. There were two difficulties. One was that potentially an infinite number of topological and spatial combinations were possible among a set of building blocks. This meant that a rational approach had to be provided to limit these possibilities. Investigation revealed that the reason the number of topological combinations was infinite was due to not restricting the number of building blocks used in a single solution. It was tentatively decided to generate an exhaustive set of building block combinations within a pre-specified restriction on the number of building blocks to be used in a single combination. In this was verified by discussions with a number of designers within the department, who all felt that this was justified, as designers would like to explore simple concepts (with a minimum number of elements) first. Regarding the possibility of an infinite number of spatial configurations for a concept, it was a compromise decision to generate only orthogonal spatial configurations for the concepts. This way at least one configuration, and often many, for each concept could be generated, and with a facility for their configuration modification this can be extended by the designers. The second problem was that while usual search procedures had single initial and goal states, a mechanical design problem could have several inputs and outputs, which required substantial modification of the search procedures. The approach was to do this in steps. First, the single input–output procedure was generated, implemented and verified. This was inverted to use as multiple input–single output procedure, and these two were used as building blocks for the development of multiple input–output procedure.

Stage II of the development of the synthesis procedure was the development of its flow chart and pseudo-code. This was tested, prior to computer implementation, by the researcher simply following the procedure by hand, and evaluating the outcomes. The testing was tedious and error prone, and needed doing several times before gaining confidence. However, once this was done, the representation was much closer to stage III, implementation. Implementation was a major step, and required a language in which it was easy to describe the representation and that provided the symbolic manipulation the reasoning procedure demanded. LISP was chosen, because it was available and satisfied the above criteria. The approach was to implement the simplest, lowest-level functionalities first, and bind them using the high-level functions only when confident that the low-level code worked. Little effort was spent on user-interface development, as the aim was to see if concepts were generated exhaustively. The comparison with existing design problems and solutions was to be done by the researcher himself.
C.3.3 Results

The research led to a database of building blocks, a method for representation along with a reasoning procedure implemented as a computer program that can automatically synthesise these blocks into concepts to solve ‘instantaneous’ versions of a required temporal function (the problem). These are illustrated with three devices: a door latch, a paper punch and a jig-saw mechanism (Figure C.5).

In terms of their functions, each device requires some input and produces some output, sometimes more than one. Each input (I) or output (O) has characteristics that may change with time. However, if the functions of the devices are observed at a single instant of time, their functions can be adequately described in terms of the following I/O characteristics: (i) their kind such as force, torque, linear or angular motion; (ii) their direction of action (e.g., up, down or sideways); and their (iii) magnitude and (iv) position. For details, see Chakrabarti (1992) and Chakrabarti and Bligh (1994, 1996a, 1996b). This is shown in Figure C.6.

A design problem is represented by its intended function, represented by a number of inputs and outputs, each having a kind, direction, magnitude and position. Using this representation, for example, one could present the door latch function as a vertical downwards force input (I) to be transformed into a horizontal leftwards translational output (O), see Figure C.7.
How are these overall, intended functions realised by the devices? Each device is composed of a number of elements (building blocks) connected in a certain way. Some of these elements function in a similar way, although their embodiments are different. For instance, when the door latch handle is pushed on the input end it rotates at the other. Although visually different, functionally this element is similar to the top part of the paper punch that, when pushed down on its input edge produces a rotation at its pivot end. The crank of the jigsaw is similar, where the kinds of input and output are simply reversed. The spatial configurations of the input and output of these elements have a definite relationship: they are orthogonal and non-intersecting. The direction of their inputs and outputs depend on the direction in which the element is laid out in space. Solution concepts, therefore, are described as combinations of functional elements such as a lever, of which the door latch handle is an example embodiment. Each element is defined as one of five basic types or their combinations, see Figure C.8, and is distinguished by the spatial relationships between its inputs, outputs and distance between their positions (i.e., the length of the element).

The five basic elements were arrived at by finding that an input and output of a mechanical element, both of which are (pseudo-) vectors, can have only five spatial relationships: both input and output can be parallel to each other and coaxial to the length of the element (L); both can be parallel to each other but perpendicular to L;
the input can be perpendicular to L but output parallel; the input is parallel to L but output perpendicular; or both input and output are perpendicular to L.

Figure C.9 Representation of a door-latch solution

C.3.4 Evaluation of Results

The evaluation was to test whether the program generated an unbiased and wide range of concepts. Two evaluations were carried out. The first included a test of exhaustiveness so as to test unbiased generation of solutions, i.e., whether all valid combinations of given building blocks were produced. This was tested by comparing the number of solutions generated with those calculated mathematically. They matched in all cases (over fifty) tested. As exhaustive combinatorial synthesis is computationally expensive, it was critical to measure the computational efficiency of the program and how this related to the size of the database of elements and the specified maximum number of elements allowed in a single solution. As the time to generate a set of solutions was proportional to the number of solutions generated, computational performance was measured by measuring the number of solutions generated by the computer with various maximum numbers of elements allowed per solution for databases of various sizes. It was found to be exponential with respect to both database size and the maximum allowed number of elements per solution. This was further verified by developing independent, theoretical models. This confirmed the earlier hypotheses: limiting the number of elements in the database by using basic elements to ensure efficient generation without compromising range, and using the maximum allowed number of elements per solution as a means to control the number of solutions generated.

External evaluation was to test whether a wide range of design concepts were generated, which was taken as generating existing as well as new concepts. The program was developed and tested using several devices, including door latches, toilet door locks, bicycle transmissions, lawn-mower drives, nutcrackers, paper punches, power-saws, holding devices and window-opening mechanisms. Some of these were used in the development of the representation and the database. In all
cases, the program generated solutions that included the existing designs along with ‘new’ solution proposals. This indicates the potential of the program for generating a wide variety of solutions in a wide variety of cases. However, all evaluations were done by the researcher, and no studies were undertaken to test the usability and usefulness of the program to practising designers, or to identify side-effects produced by the theory or implementation.

C.3.5 Conclusions About the Research Approach

The program was evaluated by the researcher by comparing its outcome with existing designs for a variety of problems. However, the set of existing designs used for comparison could not be guaranteed to be exhaustive. Comparison can be extended by considering more problems, and more existing designs for each problem. Comparison can also be made with concepts generated during the design process. As the next step, the potential of such a program can be tested by a systematic evaluation by designers using it, and by comparing designers’ concept generation performance with and without the use of the program, preferably in real design tasks. The generality of the approach needs to be tested in other domains.

Two issues are crucial for evaluating computer programs that synthesise using building blocks. The variety of solutions produced by such programs depends on the variety of building blocks in the database. Therefore, the more basic and wide-ranging the collection of building blocks in its database, the more likely it is that the concepts synthesised using these will compare well with existing designs. However, a database will always be limited, so the question is how to decide when a program works? It was felt that if the concept of an existing design cannot be generated using the database–procedure combination developed, the test should still be taken as satisfactory if the reason lies in missing building blocks in the database rather than in problems with representation or procedure. The second issue is that of abstraction. Comparison of existing designs with conceptual solutions generated by the program requires abstracting these designs to the level of the program, which requires elimination of detail and can be subjective. One way to resolve this is to get the abstraction done by more than one person and compare these.

C.3.6 Continuation

Since the completion of this thesis, several evaluations have been carried out. The potential of the program for generating a wider range of concepts than designers during the design process has been evaluated by comparing its concepts with those generated by designers during a real design project (Chakrabarti and Bligh 1996c). Hands-on experiments by experienced designers, where they tried to solve a common problem, indicated that the program generated all solutions expected by the designers and more. The experiments, however, highlighted two side-effects that had to be solved before the program could be used by designers. One was the number of concepts generated that was too large to be considered manually in-depth. The other was the abstract nature of the representation that was difficult for designers to understand. Research was continued to eliminate these effects.
(Chakrabarti and Tang 1996). The issue of large number has been tackled by developing clustering methods and heuristics for grouping similar solutions (Langdon and Chakrabarti 1999), and for eliminating infeasible solutions (Liu et al. 1999a). Visualisation has been improved by developing methods for generating possible physical embodiments of the abstract solutions and for qualitatively animating their behaviour (Liu et al. 1999b). The generality of the approach has been further evaluated by extending it to synthesis of solution principles in the physical systems domain (Chakrabarti et al. 1997).

C.3.7 References


C.3.8 Reflections from the DRM Perspective

The work of Chakrabarti and Bligh had the aim of supporting designers at the conceptual stage in order to help develop better products.

A study of the literature on the characteristics of designers, design processes, and design aids at the conceptual stage was used to clarify the state of progress in this area, and to decide on the characteristics of the support (review-based DS-I). From the literature, it was found that, in order to develop better products, designers must generate a wide range of concepts and evaluate them appropriately. The literature indicated that evaluation is enriched by exploring the features of the concepts considered; thus it was argued that considering a wide range of ideas
should help both generation and evaluation of concepts. It was also found that designers often consider only a few ideas, and reasons include lack of awareness, bias, and limited information processing ability.

The literature study was used to formulate the specific aim of the support – to support the conceptual stage of design by supporting formulation of the intended design problem and unbiased generation of a wide range of concepts to solve this problem, for consideration, inspiration and evaluation by the designer.

The overall Success Criterion was product quality (better products), and Measurable Success Criteria were adequacy of support for formulation of intended problem, and wide range of ideas generated. The support development stage (comprehensive PS), consisted of four related steps. The first was to clarify the required characteristics of the computer support and its components; the second to specify the components using flowcharts; the third to develop these into pseudo-code and hand-evaluate with example cases (Support Evaluation), and the fourth was to turn the pseudo-code into a computer program and evaluate using example cases and mathematical models (Support Evaluation). A further Support Evaluations was carried out once the support was realised to check for internal consistency and efficiency (whether the synthesis approach was exhaustive in a reasonable time). This was followed by an Application Evaluation, with the researcher as user, to evaluate the adequacy of problem formulation, and to evaluate the range of concepts generated by the support compared to known concepts for a list of test problems (initial DS-II).

The work of Chakrabarti and Bligh is an example of research with a primary focus on computer-based support development (comprehensive PS), along with a review-based DS-I and an initial DS-II carried out by the researcher. DS-I used study of the literature along with argumentation in order to ascertain the needs for, and requirements of the support system. The result of PS was an Actual Support that was close to the Intended Support except for its user interface, which was still inadequate for use by potential users. After the PhD, work was continued to carry out in the following way. A user interface was developed to create the first Intended Support (Comprehensive PS), the system was evaluated with designers and additional retrospective real-design process cases (comprehensive DS-II). This led to identification of areas of improvement and further development using new PhD and post-doctoral researchers (Comprehensive PS). The latter illustrates the efficacy of DRM for explaining and planning larger research projects.

Using the DRM framework, the project can thus be classified as a Type 5 for the PhD work, and Type 7 for the whole project:

\[ Type \, 5/7: \, RC \, (Review) \rightarrow DS-I \, (Review) \rightarrow PS \, (Comp) \rightarrow DS-II \, (Init) \rightarrow PS \, (Comp) \rightarrow DS-II(Comp) \rightarrow PS \, (Comp) \]
C.4 Teamwork in Engineering Design

Student: Eckart Frankenberger (author)
PhD dissertation Technical University of Darmstadt, Germany, 1997
(Frankenberger 1997)
Supervisor: Herbert Birkhofer

C.4.1 Introduction and Aim of Research

In the research project described in this section, engineers and psychologists investigated engineering design processes of teams in industry. The overall aim was to identify the main factors influencing design work and their interdependencies, and to build up a model of collaborative design work in practice. This model should describe the interaction of the different influencing factors on the design process and provide the basis for further development of systematic design with special emphasis on teamwork. Figure C.10 shows the general model of influences from the individual, group, external and organisational conditions on the design process, from which the project started.

![Figure C.10 Initial model of possible factors influencing the design process and its result](image-url)
C.4.2 Research Approach

Although a variety of empirical studies has been undertaken in engineering design, interdisciplinary investigation of individual- and group-related factors in an industrial environment involves a new field of research, requiring a further development of existing research approaches to deal with the specific conditions of engineering design work in industry (Frankenberger and Badke-Schaub 1997).

The complexity of dealing with the large number of variables in an industrial situation makes it impossible to investigate so-called ‘comparable groups’ in several companies. The investigation therefore focused on a detailed study of two cases over an extended period of time. The researchers did not participate in the process to minimise the effect of the observer on the observed process and to allow enough time to concentrate on relevant aspects. The first case involved a company producing agricultural machinery. Over the course of four weeks, the design process of a group of four designers redesigning a fruit press was observed. The observed interrelations between the different influencing factors were integrated into the initial model. This model was then validated by means of a second case study, in which the important variables were reviewed. The second case study involved a company making capital goods. Three projects were observed of a design team developing and redesigning several components of a particle board production plant over a period of eight weeks.

The large number of influencing factors requires a variety of investigation methods to be used. The following sections describe the methods used to capture the ‘external conditions’, the ‘design process’, the ‘individual pre-requisites’ and the ‘pre-requisites of the group’.

**External Conditions and Design Process**

Several external conditions were captured, such as ‘branch’, ‘economic situation of the company’, its ‘culture’ and ‘organisation’, the ‘flow of information’ and the ‘communication’ within the organisation, and last but not least, the ‘direct working environment with its restrictions’.

To capture the dynamic course of the design process a detailed description at short time intervals is needed. The duration of the interval was determined by the process characteristics (the categories) used to describe the process. For this a standardised approach for investigating co-operative design in industry was developed, combining direct and indirect methods.

The primary direct method was continuous non-participant observation, involving two observers – a mechanical engineer and a psychologist - sitting in the same room as the designers. The mechanical engineer observed the activities of the designers in terms of, e.g., working steps in accordance with those used in systematic design approaches such as Pahl and Beitz (1996), and the development of the technical solution in terms of sub-functions/components, ideas and solution variants. The psychologist focused on the social aspects such as decision making and group interactions. A standardised laptop-based ‘online’ protocol was used to document the observations real-time. This provided a first description of the design work as a problem-solving process. Video recordings of all team work and relevant
phases of individual design work were used to review and obtain a detailed account of specific interesting phases (Frankenberger and Auer 1997).

The final protocol consisted of a word-by-word transcription of the entire process with an average duration of 30 seconds per protocol line. These protocols formed the material for a qualitative and quantitative analysis of the process, using special software, that allows easy analysis by presenting graphical representations of each process characteristic against time. These graphs, e.g., represent the development of the solution by showing the moves between the subproblems and solution variants.

In addition, indirect methods were used, such as diary sheets on which designers could write down the sub-problems on which they worked, how they solved problems and when they contacted their colleagues. The sheets were designed to be completed with a minimum of effort in order to avoid a loss of motivation. Moreover, the documents produced by the designers were collected and they were asked about their process and the results. These interviews, based on the diary-sheets and the documents, provided important information about the design process and helped to understand the development of the solution and the technical decisions. Figure C.11 shows the procedure of compiling data of the design process and an excerpt of a revised online protocol.

![Figure C.11 Compiling the design process using direct and indirect investigation methods](image)

**Figure C.11** Compiling the design process using direct and indirect investigation methods

**Individual Pre-requisites**

Individual behaviour is influenced by several factors. A reduction of the complex cognitive, emotional and behavioural processes to one or two ‘important’ characteristics seems almost impossible. People usually behave according to the...
situation at hand: no paradigms can be considered universally valid for any situation or any person’s behaviour, with the exception of a few psychological theories (e.g., Müller–Lyer Deception (see Dorsch (1982)). For example, a person confronted by a novel, complex problem will take longer for analysis if there is enough time, if the problem is important, or if there seems to be a good chance of solving the problem. To obtain the individual pre-requisites, the following methods were chosen (Table C.2). Biographical data and personal opinions of the working environment were mainly collected by means of semi-structured interviews.

Table C.2 Variables and methods for compiling individual pre-requisites

<table>
<thead>
<tr>
<th>Field of data</th>
<th>Variables</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>biographical data</td>
<td>- age</td>
<td>- semi-structured interview</td>
</tr>
<tr>
<td></td>
<td>- professional education, career</td>
<td>- questionnaire</td>
</tr>
<tr>
<td></td>
<td>- qualification and experience</td>
<td></td>
</tr>
<tr>
<td>work environment</td>
<td>- motivation; job satisfaction</td>
<td>- semi-structured interview</td>
</tr>
<tr>
<td></td>
<td>- evaluation of the organisation</td>
<td>- questionnaire</td>
</tr>
<tr>
<td></td>
<td>- evaluation of the actual project</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- relationship to colleagues and to superiors</td>
<td></td>
</tr>
<tr>
<td>ability to deal with complex problems</td>
<td>- analysis and information-gathering</td>
<td>computer-simulated micro-worlds</td>
</tr>
<tr>
<td></td>
<td>- action planning</td>
<td>- fire (individual)</td>
</tr>
<tr>
<td></td>
<td>- dealing with time pressure</td>
<td>- machine (individual)</td>
</tr>
<tr>
<td></td>
<td>- dealing with stress</td>
<td>- Manutex (group)</td>
</tr>
<tr>
<td>special competencies</td>
<td>- heuristic competence</td>
<td>questionnaire (Stäudel 1987)</td>
</tr>
<tr>
<td></td>
<td>- social competence</td>
<td>- observing and analysing the interactions of the group</td>
</tr>
<tr>
<td>abilities concerning the design process</td>
<td>- clarification of the task</td>
<td>- diary sheets/marks-on-paper</td>
</tr>
<tr>
<td></td>
<td>- search for conceptual solutions</td>
<td>- online protocol of the design process (video and tapes)</td>
</tr>
<tr>
<td></td>
<td>- selection and control</td>
<td></td>
</tr>
</tbody>
</table>

Assuming that design processes are fairly realistic examples of complex problem-solving processes, it is important to look at the strategies of engineers in complex and novel situations. The ability of the designers to deal with complex problems was assessed by analysing the thinking and action-regulation behaviour of each designer while solving computer-simulated problems (see Dörner and Wearing 1995). Each designer had to solve two problems, which were both novel, complex and dynamic. These simulations were selected because they require different manners and strategies of action regulation. Contrary to design tasks, the computer-simulated problems can be solved without any specific previous knowledge. The focus is on the action-regulation styles of individuals when being confronted with the specific requirements of different complex situations. Thus, in using these standardised problems individual heuristics and strategies can be investigated (Badke-Schaub and Tisdale 1995).

The behaviour of the subject is not measured as a single, numerical variable (e.g., the ‘quality’ of problem solving). The planning and actual processes of the
subject were investigated as strategies containing sequences of different variables such as evaluations of questions, decisions, etc. Other studies showed that the strategic behaviour in these simulated problems is similar to that in design work. These similarities can be interpreted as individual action styles (Eisentraut 1997).

The assessment of the special competencies (heuristic and social competence) of the designers was based on their design process (captured in the final protocols and the diary sheets) and on a self-assessment questionnaire developed by Stäudel (1987). Several studies on heuristic competence indicate that a positive self-assessment of problem-solving abilities supports successful problem solving in complex situations (see Stäudel (1987)). The social competence of the individual designers was assessed using the observations of group activities, both in the case studies and in the simulations. The aim was to capture the individuals’ abilities to guide a group or to integrate into the group.

*Group Pre-requisites*

The aim of collecting group pre-requisites was to investigate the structure and the organisation of the group during the problem solving process and to investigate how the group approaches the problem in terms of behavioural patterns that may be responsible for producing the observed results. It was decided to focus on group interaction processes and describe these in terms of individual and group behaviour patterns. Consequently, we chose group interaction processes during the design processes and described them in terms of individual and group behaviour patterns. Another important diagnostic situation was a third computer-simulated problem that was given to the designers as a group.

Whereas the problem-solving activities demand a high extent of goal-analysis and emphasising priorities, the group situation causes the necessity for each individual to express their own ideas and strategies of proceeding. Getting his or her own suggestions accepted is linked to different characteristics of the individual, mainly the concept of social competence, which includes several abilities of acting in groups (e.g., the ability to co-operate, the ability to communicate, etc).

The results of the computer simulation were compared with the results of specific periods in the observed design process. The same encoding system was used in both cases, based on the phases of action regulation developed by Dörner (1996). Additionally, socio-emotional behaviour and organisational aspects were categorised.

*Methods and Initial Model*

A summary of the data of the different elements of the initial model – the domains of influencing factors – is given in Table C.3.

*Evaluation and Modelling*

Distinguishing between critical situations and routine work

The preparation and evaluation of the extensive data called for a new approach that connected the data from the different fields (design process, external conditions, the individual and the group) and allowed for both, the proof of the
relations and the generalisation of the findings. The basic idea of our method is the reduction of the documented design process to phases of routine work on the one hand and critical situations on the other hand where the design process takes a new direction on a conceptual or embodiment design level.

Table C.3 Methods for compiling data on the elements of the initial model

<table>
<thead>
<tr>
<th>methods</th>
<th>domains of influencing factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>design process and result</td>
</tr>
<tr>
<td>interviews</td>
<td>•</td>
</tr>
<tr>
<td>online protocols</td>
<td>•</td>
</tr>
<tr>
<td>diary sheets</td>
<td>•</td>
</tr>
<tr>
<td>marks-on-paper</td>
<td>•</td>
</tr>
<tr>
<td>questionnaires</td>
<td>•</td>
</tr>
<tr>
<td>computer-simulated problems</td>
<td>•</td>
</tr>
</tbody>
</table>

Types of critical situations

This method of critical situations sounds familiar with reference to the ‘critical incidents’ by Flanagan (1954) or the ‘critical moves’ by Goldschmidt (1996) but the identification of critical situations follows defined rules fitting the action-requirements of general problem solving processes (see Dörner 1996, EhrLenspiel 1995) and of the social context. There are five types of different critical situations that can be classified into situations of goal analysis, solution analysis, solution search and additionally disturbing or conflict management as is shown in Figure C.12.

Establishing the model

Critical situations, although relatively short, determine the course of the process and are therefore of special interest in isolating the main influences on the design process. In order to extract these influences and to explain the effect of a critical situation, a sub-model was developed of the interdependencies between the influencing factors and the process characteristics for each critical situation (see Figure C.13). Evidence for each identified relation was gathered separately. Special interviews with the designers combined with video-feedback of selected ‘critical situations’, helped to revise the sub-models.

The sum of the different interrelations in the individual sub-models led to a model of relations between influencing factors and process characteristics in all critical situations of the design process. Altogether, 265 critical situations were identified in the four analysed projects of the two case studies. These explained the course of work by more than 2200 single interrelations between factors, process characteristics and the result. The reduction to 34 different influencing factors illustrates the suitability of the model.
C.4.3 Results

As a first step, the importance of the influencing factors and their interrelations has been determined by the frequency of occurrence in all critical situations in the four design projects. However, each type of critical situation has a specific role in the design process. In order to make more specific statements on the core mechanisms leading to success or failure in the design processes, the models of each type of critical situation have been analysed separately. On the basis of this analysis we can
answer questions such as, ‘which are the main factors responsible for a deficient analysis of goals?’ or ‘which are the mechanisms leading to low quality’?

In the following example the factors and relations that are important for successful or deficient decisions of solutions are analysed. Figure C.14 describes the main mechanisms responsible for deficient decisions of solutions found in the four projects. The thickness of the arrows depicts the frequency (in per cent) of the relations occurring in this type of critical situation. The thickness of the frames depicts the frequency (in per cent) of the factors identified in all critical situations of ‘deficient decision’.

![Diagram](image.png)

**Figure C.14** Factors and relations responsible for deficient decision of solutions (in % of all 36 critical situations of this type; - - less X less Y, + - more X less Y)

From Figure C.14 we can conclude that non-availability of information is the crucial factor responsible for deficient analysis of solutions, which is the major reason for deficient decisions. The main reasons for lack of information are loss of individual motivation (e.g., for transferring knowledge to inexperienced colleagues), lack of experience, insufficient goal analysis (caused by the routine of highly experienced engineers) and limitations in group organisation. Based on detailed knowledge of the situations, measures can now be proposed to avoid the identified negative mechanisms. For instance, putting emphasis on goal analysis especially if designers are highly experienced, or clearly organising the responsibilities for the clarification of requirements.

In the same way the mechanisms supporting and hindering good solution decisions and any of the other types of critical situations have been analysed. Moreover, analysis of the situations that directly influence the result reveal the
interrelations leading to a high or low ‘design productivity’ in terms of the quality, time and cost.

C.4.4 Evaluation of Results

The central mechanisms for successful or deficient critical situations provided insights into the conditions and consequences of factors in the different decisive situations of the design process. After evaluating the results an important question remains: In spite of the high number of analysed single cases (265 critical situations with more than 2200 single detected effects of factors), is it possible to quantify generally accepted determinants and relations on the basis of the data of two companies and four projects?

Figure C.15 depicts the number of influencing factors that had to be added as each project was analysed in order to explain the critical situations in that particular project. The percentage of newly added factors as well as the relationships decreased from project to project. Furthermore, the most important relations occurred in all four projects and did so very often. This result leads us to the assumption that by analysing the four projects we captured the most important influencing factors and relations.

![Figure C.15 Number of influencing factors added in the four projects and number of relations occurring in n processes](image)

Many of the influencing factors related to the individual and group pre-requisites. The importance of the human factor on co-operative design work becomes evident by the fact that in spite of the fact that designers worked individually 85% of their time, 88% of the critical situations occurred in group situations! Structured interviews with engineering designers and department leaders in 10 companies supported the results of the observations.

C.4.5 Conclusions About the Research Approach

The detailed compilation of data by various methods was necessary to detect critical situations in design and to explain the course of design work by means of influencing factors. The new concept of ‘critical situations’ allows quantification of the importance of central mechanisms determining design in industry. The knowledge gained on important positive and negative mechanisms acting in
particular design situations can be used to develop suitable measures in industry and to make design education at universities more practically relevant.

However, to compile data with the methods used requires enormous effort. Consequently, one aim of further research is to develop investigation methods that require less effort but are still accurate enough to allow more focused research questions.

C.4.6 Continuation

In order to generalise the model based on a larger variety of design situations, additional investigations in the field of teamwork in engineering design practice are planned. After detailed interviews in 10 companies, a training course to enable engineering designers to detect and analyse their own critical situations was developed. So far, two investigations with a combined approach of observation and self-assessment of critical situations were carried out. Essentially the same central mechanisms of success and failure in product development were detected.

C.4.7 References

C.4.8 Reflections from the DRM Perspective

The work of Frankenberger and Birkhofer is primarily an investigation into the engineering design process in industry to identify the main factors influencing design work and their interdependencies, so that a model of collaborative work in practice can be developed (comprehensive DS-I). This would form the basis for further directions in supporting systematic design with emphasis on teamwork in an industrial environment.

The study led to a model of group design processes consisting of two kinds of situations – critical and routine situations – and the factors that influence these. Furthermore, the model described how these factors, especially the critical ones, influenced these situations. Critical situations are those where the design process takes a new direction on a conceptual or embodiment level; routine situations are those where it does not. Four domains of influence were identified: individual influences, group influences, external conditions and organisational conditions.

Measurable Success Criteria were taken to be product quality, cost and development time. Since the complexity of dealing with the large number of variables in an industrial situation makes it impossible to investigate the so-called ‘comparable groups’, the investigation focused on a detailed study of two cases over an extended period of time, one to formulate the model, and the other to validate this model. This illustrates the importance of validation of the Reference Model in an empirical study and the inherent iterations involved in the DS-I stage.

The investigation combined a variety of research methods: continuous non-participant observation by an engineer and psychologist analysing the technical and social aspects of the design respectively, online protocols, diary sheets, questionnaires, document analysis, and interviews of participants.

This project is an example of research where the primary focus is on a comprehensive understanding of the current situation (Comprehensive DS-I), with the eventual aim of support based on this understanding. It is an example of DS-I carried out in industry, where successive industry observations are used for formulating and validating the understanding. It is also an example of the results being described in what we define as ‘Reference Model’.

This project continued beyond the PhD, and led to the development of a support in the form of a training programme for industry (Comprehensive PS), which has been introduced and evaluated (Comprehensive DS-II). This also illustrates a different type of support, namely training, and the possibility of using DRM to explain research programmes.

Using the DRM framework, the project can thus be classified as a Type 1 for the PhD work, and Type 7 for the whole project.

Type 1/7: RC (Review) → DS-I (Comp) → PS (Comp) → DS-II (Comp)
C.5 Measuring Conceptual Design Process Performance in Mechanical Engineering: A Question-based Approach

Student: Ade Mabogunje (author)  
PhD dissertation, Stanford University, Palo Alto, USA, 1997 (Mabogunje, 1997)  
Supervisors: Larry Leifer, Rolf Faste, Sheri Sheppard, Ilan Kroo

C.5.1 Introduction and Aim of Research

The investigation reported in this dissertation began with a casual observation of the pervasiveness of pronominal questions (i.e., what, why, how, how much) in several of the structured design methodologies (e.g., value analysis, quality function deployment, design for assembly) used by engineers in several Japanese companies, particularly those in the automobile industry. Knowing the important gains in the market share by these companies towards the end of the last decade and the widespread adoption of these methodologies by other companies (Barkan 1992), it seemed possible that there could be a link between the product development performance of companies and the question-asking behaviour of their engineers. This was the basic premise of the dissertation.

An important motivation was that establishing such a link could enable the development of other means (tools and methods) to augment the question-asking behaviour of engineers and possibly lead to further improvements in product development performance. For example, as the use of computers becomes more prevalent in the engineering profession, engineers conceivably could be given design process feedback based on their question-asking behaviour, see Figure C.16.

![Figure C.16 Schematic of a process monitoring system for designers during product development where feedback is based on the questions being posed](image_url)
If a link could be established between product development performance and the question-asking behaviour of designers, it is conceivable that questions could be used as a means of monitoring the design process. Given the increased use of computers in engineering this sort of monitoring could be used as a basis for giving process feedback to designers. A designer predisposed to asking questions of a particular nature could be encouraged to explore asking other types of questions or to pair up with another designer with a different question-asking style. It may even be possible to distinguish between a variety of question-asking styles that have led to creative solutions and those that have not.

C.5.2 Research Approach

As I thought more about the possible link between product development performance and question-asking behaviour, I realised that even if the link existed, it would be difficult to establish because the product development process as a whole is very complex. This difficulty of establishing the link greatly influenced the research methodology and cannot be overstated.

Engineering design today takes place in a complex internal (company) and external (market) environment. There are several components involved and these are related to each other in multiple ways. This sort of complexity makes the process nonlinear, highly iterative, and difficult to study. This difficulty manifests itself in two distinct research environments today – those conducted in a laboratory and those conducted in industry (Stomph-Blessing 1991). Table C.4 gives an overview of the difficulties associated with either environment. It points to the fact that the dynamic nature of real-world design and the complexity of the organisational context tend to make laboratory results irrelevant to industry, and industry results difficult to generalise.

<table>
<thead>
<tr>
<th>Table C.4 Effect of the environment on types of tasks studied and on the research process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Industry Environment</strong></td>
</tr>
<tr>
<td><strong>Design Task</strong></td>
</tr>
<tr>
<td>Task has a history and future</td>
</tr>
<tr>
<td>Problem definition and requirements may change</td>
</tr>
<tr>
<td>Internal and external factors emerge from data</td>
</tr>
<tr>
<td><strong>Research Process</strong></td>
</tr>
<tr>
<td>Difficult to plan</td>
</tr>
<tr>
<td>Difficult to control</td>
</tr>
<tr>
<td>Observation interferes with project</td>
</tr>
<tr>
<td>Not repeatable</td>
</tr>
<tr>
<td>Existing environment</td>
</tr>
<tr>
<td>Design time in months</td>
</tr>
</tbody>
</table>
It was therefore obvious very early in this investigation that an important issue to be addressed was the level of complexity of the environment in which a product was designed and developed. As I delved into the literature, I found strong evidence supporting the idea that the practice of design was in reality a set of strategies for dealing with complex situations (Vincenti 1990; Lawson 1990). This in turn suggested to me that design was a way to do design research. In other words, design could be turned on itself. I will quote at some length from two of the sources I came across because they provide concrete illustration of the research methodology adopted in this thesis. The first source is historical and is from a book titled “What Engineers Know and How they Know it”, (Vincenti 1990). It reviews the history of development of the “control-volume analysis”, an important method of analysis for problems in the field of thermodynamics that is used by engineers and considered inadequate by physicists.

“The undoubted value of theorems (i.e., of control-volume analysis) lies in the fact that their application enables one to obtain results in physical problems from just a knowledge of the boundary conditions. .. engineers frequently must deal with flow problems so complex that the underlying physics is not completely understood or the differential equations that describe the phenomenon point by point cannot practically be solved throughout the flow. In such situations control-volume analysis, by working with information only on boundaries and ignoring the interior physics, can often supply limited but highly useful results of an overall nature”

An important point to note in the above excerpt is the interplay of problem and results in the engineer’s work. The results desired help to determine the appropriate formulation of the problem. The second source is empirical. It was based on a study of final-year architecture students and final-year physics students solving a contrived laboratory problem (Lawson 1990).

“The scientists tended to use a strategy of systematically exploring the problem, in order to look for underlying rules which would enable them to generate the correct, or optimum, solution. In contrast, the designers tended to suggest a variety of possible solutions until they found one that was good or satisfactory. .. The problem-solving strategies used by designers probably reflect the nature of the problems that they normally tackle. These problems cannot be stated sufficiently explicitly that solutions can be derived directly from them. The designer has to take the initiative in finding a starting point and suggesting tentative solution areas. ‘Solution’ and ‘problem’ are then both developed in parallel.”

As in the previous excerpt, the interplay between problem and solution should be noted. The excerpt also highlights the manner in which designers solved problems through an iterative generation of solutions. From the foregoing, four design strategies that have been used to deal with complex problems can be enumerated. These are: (1) the reduction in the size of the problem, (2) the focus on results, (3) the generation of multiple solutions and (4) the parallel elaboration of problems and solutions. As a whole, these strategies provide the key constituents of the research methodology used in this investigation.
Strategy 1: Reduction in the Size of the Problem

The necessity of carrying out design research in the real world and the difficulty associated with such studies, led to a need to reduce the size of the problem. There was the opportunity to use a design class in the university as a test bed. This class, ME210, uses industry-sponsored engineering projects as a framework for teaching graduate students the product development process. Every year, during the fall quarter, the students are supported in forming three-person teams that then bid on 12 to 18 projects submitted by a variety of companies. Over the next seven months the students propose alternative designs, investigate these alternatives, build, and test a functional prototype of the preferred design.

Figure C.17 Carrying out design research in less complex environments than large organisations provides a way of overcoming some of the difficulties usually associated with research in large organisations

The change in environment from industry to the classroom represents an important reduction in the complexity associated with the environment. A further reduction is possible if specific projects in the class are further isolated and studied, so to speak, in a laboratory. This strategy is illustrated in Figure C.17, where design projects and resources are taken from the larger organisation to the smaller one and lessons learned in terms of tools and method are transferred from the smaller organisations to the larger one.

Strategy 2: Focus on Results

The strategy calls for an early identification of desired outcomes as an aid to determining the aspects of the phenomenon that should be given attention. It was therefore important to have a tentative criterion for measuring performance. Earlier, I alluded to the increase in the market share of Japanese automobile companies as a measure of product development performance. In the context of this investigation I chose to begin by looking at development time. As will be seen later, as different solutions were elaborated, the criteria for performance changed. Nevertheless, a distinctive feature of this investigation was that there was always a measure of performance.
This feature marked a departure in approach from some earlier empirical studies that were particularly informative of more salient features of the engineering design process (Tang 1989; Minneman 1991). According to their approach, design study begins with observation. There is then an attempt to understand design activity and to provide design support on the basis of this understanding. If the support proves ineffective, what is proposed is to go back to better understand through observation before attempting further design support.

In the approach used in this dissertation, the research activities revolve around design performance rather than the more general notion of design activity. Starting with an assumed model of the design process, the proposition is that understanding can be developed through trial-and-error attempts to improve design performance. The error in such attempts is defined as the difference between the assumed model and the real model. Each trial inevitably seeks to minimise the error by changing the assumed model. Perfect understanding ideally occurs when the assumed model mimics the real model. From this perspective the problem involves finding appropriate parameters of performance and building models of the process that can be used to predict the performance. The two approaches are shown in Figure C.18. If an analogy is made with the case of the control-volume analysis cited earlier, it should be clear that while this approach may miss some of the detail and scope of the design process, it should make it possible to obtain operationally useful results.

**Strategy 3: Generation of Multiple Solutions**

Having reduced the complexity of the environment from industry to the classroom, three areas for improving performance were explored. The first, named 210-X, explored the area of information retrieval in design. The second, named Virtual 210, explored the area of computer-based organisational simulation. The third, named 210-NP, explored the area of noun-phrase analysis of the quarterly reports submitted by project teams in ME210. These three solutions were connected by two underlying themes: (1) all were based on the premise of a link between design performance and the question-asking behaviour of designers and (2) all the explorations were clustered or localised around the ME210 class.

**Strategy 4: Parallel Elaboration of Problems and Solutions**

In the course of elaborating the problems and solutions (i.e., 210-X, Virtual 210, and 210-NP), several hypothesis were formulated and tested. This process led to the final hypothesis of this research. To better understand the premise of the final hypothesis, it is important to clarify the relationship between the three solutions, 210-X, Virtual 210, and 210-NP. To do this I will summarise each area explored under the following four sub-headings: problem, solution, results, and need. Where it is appropriate, a brief background will be included.
Figure C.18 A basic difference exists in the approach of this study when compared to the work of Tang and Minneman. While both approaches go through the cycle of observation, understanding, and support, the focus is different, as shown by the way that the problem at the centre is formulated. In the previous approach, there is an activity to be understood, In the proposed approach, there is a performance to be improved and as such there is a greater emphasis on the relationship between the support and the performance.

C.5.3 Results and Evaluation of Results

Solution 1: 210-X

a) Problem
Given the focus on development time as a parameter of performance, there was a need to think of ways in which questions could have an effect on development time. A scenario was imagined in which the questions asked by a designer during the development of an $n$th generation design would be made available to another designer during the development of an $(n+1)$th generation design in such a way as to facilitate the access to answers.

Hypothesis: The hypothesis was that facilitating the access to answers would reduce the development time.
b) Solution
To meet the objective, two designers were separately asked to redesign a rotary friction damper to meet a new set of specifications. They were provided with a comprehensive report of the previous design and also had access to a member of the previous design team. They were given roughly six hours to come up with a design that met the new specifications. In addition, they were instructed to verbalise their questions so that these could be recorded for later analysis.

Based on the analysis of the questions, a system for indexing and retrieving answers from the comprehensive report was developed. The indices were composed of a fixed list of descriptors and a variable list of subject words that were taken from the report of Baya et al. (1992). The subject words were in turn used in building hierarchical question-based models of the previous requirements and the previous artifact. Based on a set of heuristics about different relationships between the models and the indices, the system was able to reformulate questions and retrieve possible answers (Baudin et al. 1992; 1993).

c) Results
Preliminary empirical tests using the system developed, DEDAL, showed that it reduced the development time in the \((n+1)\)th generation design. However, this gain was largely offset by the additional effort and time needed to index the report and build the question-based models that increased the development time of the \(n\)th generation design and hence the overall development time.

d) Need
Based on these initial tests, it was clear that the key to improving the overall performance was to minimise the effort in indexing the report and building the question-based models.

**Solution 2: Virtual 210**

a) Background
While 210-X concentrated on a possible symbiosis between the laboratory and the classroom, Virtual 210 focused on a possible symbiosis between industry and the classroom. In 1991 a program named Virtual Design Team (VDT) had been developed in the civil engineering department at Stanford University to simulate the impact of new communication technologies on the performance of construction projects in industry (Cohen 1992; Christensen 1993). The independent variables of the model consisted of a set of communication tools and the organisation structure, and the dependent variables were the project duration and quality. The intent was to model DEDAL as a communication tool and explore the impact of its use on project duration and project quality in ME210. In other words could we predict the effect of a question-asking support program like DEDAL on the project performance of designers in ME210?

b) Problem
The reality was that the version of VDT at the time could not adequately represent information technologies as complex as DEDAL.
c) Solution
I went ahead by using a simpler and easier to model information technology than DEDAL. While this deviated from the original intent, it allowed me to explore the limits and opportunities of VDT and to gain a better sense of how it could be used to develop an organisational or multi-team perspective on the link between question-asking and performance.

Hypothesis: Since a project-based engineering class like ME210 is similar to a construction organisation in terms of the team organisation and relationship to the client, the duration, and the end product, the hypothesis here was that project-based engineering classes could be simulated with at least the same degree of realism as current computer simulations of engineering organisations.

d) Results
In VDT, there is a procedure for calculating the complexity of both the requirements and the solutions of a project. I soon realised that the question-based models developed for DEDAL could be reused in VDT for calculating these complexities. This was an unexpected synergy between design information retrieval and organisational simulation. In addition, the results of the simulation showed that much more than having the same degree of realism as current computer simulations of engineering organisations, programs like VDT could be used as a curriculum planning tool for project-based engineering classes like ME210. This was not the sort of outcome I anticipated but it clarified the need and provided additional motivation to continue the research along the lines of 210-X as will now be explained.

e) Need
The most time-consuming aspect in the VDT modelling procedure was calculating the requirements and solution complexities. This was because, compared to other inputs, they required a deeper knowledge of the requirements and the artifact. Given the benefits of the simulation, it was obvious that finding an easier way to calculate these complexities will increase the usability of VDT. This situation was similar to that encountered in 210-X where the key to improving the overall performance of DEDAL was to minimise the effort in indexing the report and building the question-based models. Therefore, at the very least, it seemed that the effort to model ME210 in VDT helped to reinforce the need and motivate the search for ways to minimise the effort involved in model building.

Solution 3: 210-NP

a) Problem
Revealed by experience from 210-X and Virtual 210, a need became apparent for a quick way of building the hierarchical question-based models. Satisfying this need would benefit designers, instructors of project-based classes, and design researchers. In the case of designers, indexing and model building will be easier to do in real time, and this will make their reports more reusable in future. In the case of instructors, simulation studies could become an additional tool for planning the curriculum and designing the organisation. In the case of researchers, the reduced
overhead would mean that a larger number and variety of projects could be used in experimental studies like 210-X. This increase in sample size in turn would improve the reliability of the research results. Thus, finding a quick way of building the hierarchical question-based models had a potentially high payoff in the sense that a single solution would be beneficial to three groups of potential "customers" simultaneously.

b) Solution
Since the hierarchical question-based models, and the requirements and solutions complexity matrices were built from the subject words in the design reports, and since the subject-words often turned out to be the noun phrases, a quick way was needed to extract noun phrases from the report.

Definition: A noun phrase is a word or group of words with a noun as its head and functioning as the subject, object or complement of a sentence. An example of a noun phrase is: “the steel bolt”.

c) Need
From a different perspective, the hierarchical question-based models and complexity matrices functioned as substitutes or surrogates for the questions whose answers were documented in the report. Since the reports were graded at the end of the quarter, it was felt that if indeed a link existed between the question-asking behaviour of designers and the development performance, a correspondence must exist between some properties of these surrogates and the grade assigned to the report. This led to the third and final hypothesis of this research.

Hypothesis: It is hypothesised that certain properties of the noun phrases extracted from the report will be positively related to the project grade in ME210.

d) Results
A quick way for extracting the noun phrases from design reports was developed using a parts-of-speech tagger and a suite of Microsoft Excel macros. The tagger takes as its input, a string such as:

The inner hub holds the steel friction disks and causes them to rotate.

and produces a tagged output consisting of two strings.

The inner hub holds the steel friction disks and causes them to rotate
at jj nn vbz/2 at nn nn nns cc vbz/2 ppo/2 to/2 vb

The lower string consists of parts of speech tags corresponding to words in the upper string. E.g., the following sequences were used to identify the noun phrases in the tagged output: “jj nn” – “inner hub”; “nn nn nns” – “steel friction disks”

Thus, it was possible to extract noun phrases from all the reports for each quarter of an entire academic year and test the hypothesis. Thirty reports (averaging 40 pages each) and representing ten projects were analysed. The results showed that the number of distinct noun phrases in a report was strongly associated with the project grade (for the winter quarter gamma = 1; for the spring quarter, gamma = 0.7). That is, projects with grades A+ or A, had a higher number of distinct noun phrases than projects with grades A– or B+, which in turn had a higher number of
distinct noun phrases that projects with a B. The degree of association was not as strong as this when the project grade was compared with measures of the readability of the documents, the number of pages, or the number of words in the document. In Figure C.19, the project grades have been superimposed on a graph of the quarterly variation of the number of distinct noun phrases.

Figure C.19 The amount and change in number of distinct noun phrases per quarter versus the academic grade

On closer observation, it was also seen that between the start of the project in the fall quarter and the end in the spring quarter, several new noun phrases were formed by the students to express their ideas. In other words, new knowledge was literally being created. This phenomenon had not been observed during the experiments in 210-X, where the process lasted less that six hours. These results were quite unexpected and had several important implications. One of these being that there is a part of the process, distinct from question asking, that was involved in the creation of new knowledge. The fact that it can be readily observed and that its effect on design performance can be traced opened up a new and exciting line of research that unfortunately is beyond the scope of the dissertation.

C.5.4 Conclusions About the Research Approach

An important trade-off in this research was one between a synthetic approach and an analytic approach to research. Alternating between these two approaches was critical to obtaining operationally useful results. At the same time it had the disadvantage of introducing discontinuities in the logical lines of thought. I have
had to be explicit about these discontinuities when they occur for they provide fertile ground for further research. Needless to say, it is obvious that as advances are made in our knowledge about the design process, the synthetic approach to research will become better codified and easier to explain.

C.5.5 Continuation

Since the end of this dissertation, two lines of work have been pursued. The first line has to deal with the capture of tacit knowledge during design. Yen (1999) has extracted noun phrases from several other media besides the requirements documents. Using presentation documents created by designers, email records, and video transcripts of design meetings, he found that the incidence of noun phrases is in fact much higher in informal than in formal documents. He has designed a computer-based tool, RECALL, to improve access to this informal information and is currently investigating the impact of such access on design performance.

The second line of work is aimed at understanding how the learning that occurs during the design process impacts product development performance. In traditional learning situations, the concepts to be taught and the new vocabulary are known in advance by at least one party in the transaction, the instructor. By contrast, in design-based learning situations, the concepts and in particular the new vocabulary are not known in advance. Given the increase in incidence of new noun phrases during design it is obvious that learning is occurring. What is not so obvious is how it can be improved to have a positive impact on development time or product quality.

C.5.6 References

Cohen GP (1992) The virtual design team: an information processing model of design team management. Department of Civil Engineering, Stanford University, Stanford, California, PhD thesis
Christiansen TR (1993) Modeling efficiency and effectiveness of coordination in engineering design teams. Department of Civil Engineering, Stanford University, Stanford, California, PhD thesis
Minneman SL (1991) The social construction of a technical reality: empirical studies of group engineering design practice. Department of Mechanical Engineering, Stanford University, Stanford, California, PhD thesis
Tang JC (1989) Listing, drawing, and gesturing in design: A study of the use of shared workspaces by design teams. Department of Mechanical Engineering, Stanford University, Stanford, California, PhD thesis

C.5.7 Reflections from the DRM Perspective

The work of Mabogunje et al. is on measuring conceptual design performance using a question-based approach. The investigation started with the observation of the pervasiveness of pronomial questions in several structured design methods used by a number of Japanese companies, which led to an initial hypothesis that question-asking behaviour of designers may have a strong influence on the product development performance of companies, and therefore could be developed into an indicator of its performance. Several criteria for success were used, product development time was one of these. A central assumption was that student design cases could be taken as a first indication of professional performance, and hence the study could involve students rather than professional designers.

Three linked student case studies were used to progressively develop the understanding of the relationship between question-asking behaviour and performance (comprehensive DS-I), and develop and validate computer programs to enhance this behaviour (comprehensive PS and comprehensive DS-II). The evaluation showed various side-effects that offset the principal, anticipated effect, which led to the modification of the programs (revisiting PS and DS-II). An interesting feature of this project was that the solution developed for offsetting the side-effect turned out to correlate very well to product development performance and shifted the focus from question-asking behaviour to processes leading to noun-phrase creation as a principal indicator of product performance. This is an example of the opportunistic nature of scientific inquiry.

The work of Mabogunje et al. is an example of research where a comprehensive understanding of the current situation is developed (Comprehensive DS-I) in order to feed into a Comprehensive PS, developing a support that is then evaluated comprehensively (Comprehensive DS-II). The results led to further Comprehensive PS and Comprehensive DS-II.

Using the DRM framework, the project can thus be classified as:

Type 7: RC (Review) \(\rightarrow\) DS-I (Comp) \(\rightarrow\) PS (Comp) \(\rightarrow\) DS-II (Comp) \(\rightarrow\) PS (Comp) \(\rightarrow\) DS-II(Comp)
C.6 Design for Quality

Student: Mikkel Mørup
PhD dissertation, Technical University of Denmark, Denmark, 1993 (Mørup 1993)
Supervisor: Mogens Myrup Andreasen (author)

C.6.1 Introduction and Aim of Research

In the modern, technological society products have become an inextricable part of our private and professional lives. In turn, the ability of individuals, companies and even society to operate and prosper depends heavily on the quality built into products. Without such quality we have at the least annoyance; at worst, poor quality can cause accidents and even disasters. The idea of quality improvement has received enormous interest from industry during the last two decades. The reason is that quality, in terms of both high quality on the market and low cost of quality within the company, is a major competitive factor on the international market. Therefore many companies spend large resources on implementing and using various approaches to the management of quality; some companies do it because ‘it pays’, to others it has become a matter of survival.

The subject of this thesis is the creation of quality, as perceived by the customer and the manufacturing company, during product design and development, in short, Design for Quality.

Through the literature and industrial studies, the research shows that despite considerable efforts in Total Quality Management and advances in technology, many companies still have serious problems in developing, manufacturing and marketing profitable products that live up to customer expectations of quality.

Quality is believed to be a strategic competitive factor, and the mastering of the quality and costs are seen as crucial in the competition with Japanese quality results. Quality is built in during mainly conceptual design, but there are many dispositional effects related to quality (Olesen 1992). It is our basic belief that quality cannot be created by control, but needs a sound synthesis approach.

At the Department of Control and Engineering Design we have worked for a decade with different Design for X-tools. It was naturally to assume that DFQ should follow the same pattern of qualitative and semi-databased principles, governed by some kind of procedure. We see product development as being the single most important contributor to product quality. Ideally seen DFQ should be based on ‘best practice’ organisation and support of the product development function, including

- strong links to the company’s strategies;
- a supporting organisational structure;
- technology implementation linked to business objectives;
- a system for measuring quality performance.

This led to the following objectives for the research project:
To define the concept(s) of product quality in relation to the customer, the manufacturing system and other life-phase systems, such as distribution, sales, and service.

To establish a theoretical basis for Design for Quality including terminology and descriptive models. Major elements in this theoretical basis must be:

- a description of where quality is placed in the technical system, \( i.e. \), an identification of how different product characteristics influence product quality;
- an identification of how these characteristics can be manipulated during design.

To formulate a general framework for Design for Quality, containing the major means for supporting and performing Design for Quality. Obvious means already known could be: teamwork, quality tools and techniques, design procedures, organisation, management, etc. The DFQ framework should be exemplified and presented in a pedagogic and operational form.

The work is aimed at the following company types:

- belonging to the mechatronic industry, where the product is a mixture of mechanical, electronic and software components;
- having medium to high volume production;
- covering all activities necessary to manufacture products, including: stakeholder analysis, product development, marketing, production, distribution and sales.

The research project hereby has an interesting problem in identifying the research object. On the one hand, the object is the product developer’s skill and knowledge in building in quality. On the other hand, the object is the proper understanding of the relations of his work to the business and operations of the company and to the product life phases and the quality results here. This proper understanding should be ‘implemented as a mind-set’ at the product developer, ensuring integrating efforts. The four central objects of research are: the product development project, the core product, production processes, and the customer/user.

In our efforts to establish a scientific basis for the research, the following eight separate and distinct theory areas or established research areas were drawn upon: Theory of Technical Systems, Design Theory, Manufacturing process theories, Statistical theory, Theory of consumer behaviour, Man Machine Interface theory, Organisational (management) theory and Cognition theory. It is interesting, that Total Quality Management did not fit into our mapping, being based upon fragments hold together by an ideal model and ‘guruficated’ believes. In spite of this, the TQM approach became central in the research.
C.6.2 Research Approach

The first hypothesis of this research is that the opportunities for quality improvement can only be exploited if we acknowledge the central role that product development has in creating quality in all product life phases. If Design for Quality is to be developed accordingly, a first step of this research would therefore not be to look at high-level organisational or managerial issues, but to start with investigating the core of the design process – synthesis – and then expand from that. The design process is ultimately where the genesis of product quality takes place.

A second hypothesis is that it is possible to identify those product characteristics that ‘carry’ quality; both quality with respect to the customer (e.g., ease of use, reliability, robustness) and with respect to the company (e.g., robustness towards variability, low quality control costs).

The third hypothesis is that the successful pursuit of the two first hypotheses will create an insight into core processes of designing for quality, which is instrumental in analysing and enhancing already existing approaches and methods for DFQ and for developing new ones.

Finally, this research has operated with hypotheses regarding the choice of scientific viewpoint. It is argued that the Theory of Technical Systems and Design Theory should constitute the theoretical outlook of this research.

The hypothesis and scientific questions were a natural consequence of the identified industrial needs and problems, but also showing where we believe to find insight in the phenomena of quality creation. Finally, they mirror the Department’s tradition as ‘tool makers’.

The sequence of activities of this research is not so easy to identify. Through a final-year master thesis the area of DFQ was scrutinised, so a certain pattern of necessary insight element was established. The scientific procedure (see Figure C.20) is based on the ideas behind critical rationalism and fallibilism, in which existing models and methods are improved to provide a better description of the empirical reality. This is done through the literature studies, logical structuring, empirical observations, experiments, etc. The research has its starting point in a practical problem base where real phenomena in industry and in the literature are analysed. In parallel the discovered problem areas are analysed in the context of the theoretical basis, in which new hypothetical statements on the nature of quality and Design for Quality are also formulated. In order to check their validity and their applicability the solutions and hypotheses are applied to product examples and design cases and presented to designers and researchers.

The main sources of information, inspiration, and experience in this research will briefly be commented upon.

Literature

Because of the large interest in quality in industry and research many publications on this topic exist. It is characteristic that the publications are mainly about

---

29 Today we would call these basic assumptions
management aspects, case studies, or isolated methods (e.g., reliability techniques or design of experiments). Disagreement between authors on how quality issues are handled during design still prevails, and coherent theoretical work on the concepts of product quality and Design for Quality is missing. The literature on Design Theory has had a greater degree of clarification to offer, but in the discussion of quality issues focus is still on technical functionality and manufacturability and not so much on quality in relation to the customer. Design for Manufacture and Assembly literature has provided inspiring models for this research, especially because of their successful application in practice.

![Diagram of DRM method](image)

**Figure C.20** A method for applied research in which attention is focused on the interplay between theory and practice, after Jørgensen (1992)

*Interviews and Case Studies in Industry*

In order to obtain an understanding of practical design problems and problem-solving patterns, the research comprised a three-month stay in a Danish manufacturing company as active project participant. Together with shorter case studies in Danish and American industry this stay has been used as a means to detect problems of methodological nature in a designer’s working practice. In addition, interviews with quality and product development managers in six American companies have been made. The interviews concentrated on general procedures for product development and specific methods and techniques, especially the Quality Function Deployment method.
Discussions with Colleagues and Professors

Education and research in so-called quality engineering is still limited to single topics such as reliability techniques or problem solving techniques in quality management. However, the project has benefited greatly from discussions with colleagues and professors in a variety of countries on the scope and future perspectives of Design for Quality.

Research in the United States of America

The project included a 10-month stay at the University of Wisconsin-Madison as a doctorate student and guest researcher at the CQPI (Center for Quality and Productivity Improvement). The stay at the CQPI led to valuable insights into the statistical aspects of quality and the methods in this field.

The research process was, on the one hand, arranging the above mentioned activities opportunistically based upon expected insight, on the other hand the ‘gradual filling up’ of files of results, which became the chapters of the thesis.

C.6.3 Results

The discussion of the inadequacy of current approaches to the management of quality led to the conclusion that incremental improvements of these approaches are not enough; a change of paradigm is needed. This should be based on the fact that:

- Quality is created in product development, not only product quality during use but to a great extent also quality in other product life phases.
- Product quality is actually \textit{built in} during the synthesis process, whereas activities where quality is specified, analysed or verified are ‘only’ means of indirectly controlling synthesis.

Thus, product design and development hold the key to solving many of the quality-related problems and, perhaps more importantly, to exploit the opportunities for achieving better quality in company activities and on the market.

The main result is a framework (see Figure C.21) that describes the aspects of product development that should be emphasised in order to conduct DFQ. In the process of defining this DFQ framework several other results were obtained. Each of these and the framework will be described in the following sections.

Defining Quality

The discussion of traditional definitions of product quality led to the conclusion that the quality terminology belonging to product development is confusing and poor, which can have serious consequences for how quality is handled during design. Consequently, a new set of quality concepts was introduced, which make a clear distinction between customer perceived quality, termed ‘Q’-quality, and quality perceived by stakeholders within the company, termed ‘q’-quality. The concepts also distinguish ‘true’ quality from the quantifications of quality that can be expressed in, say, product properties or manufacturing variability.
Defining the Objects of Quality

After clarifying the quality concept, those objects within the product that carry quality were identified and classified. A particular role is played by the organs (also called function carriers), because they carry the product functionality and quality properties, which in turn are perceived by customers as quality. During design, parameter models are the primary means for modelling the relationships between quality properties and the organ solutions.

Clarifying the Relationships Between Quality and the Realisation of the Product

Due to the fact that manufacturing variation influences quality properties, manufacturing processes are often subject to meticulous quality control. However, it was stated that variability evolves in the meeting between the process and the product, and that by making the product robust to variation the need for quality control can be lessened.

Apart from influencing variability in manufacture, the product, and therefore DFQ, also has significant influence on the Q quality and q quality in all other product life phases. Hence, it was suggested that DFQ should be conducted in close contact with important stakeholders, and that their immediate statements about quality (Q and q) should be acknowledged.
Defining the Quality Mind-set of Product Developers

The new quality concepts (Q- and q quality, quality properties, robustness, etc.) were incorporated in the context of a ‘quality mind-set’, which encompasses those parts of the product developers’ understanding of quality-related aspects and objectives, which cannot be formulated in traditional specifications.

Developing a DFQ Framework

A new framework for DFQ was presented, which is derived from the analyses of industry’s needs, and that is based on the theoretical findings of the research.

The framework has eight ‘elements’, all of which should be incorporated in a comprehensive implementation of DFQ in practice. The elements cover major activities, from strategic work to the detailed design process, in product development, where quality is specified, built into the product (synthesised), and verified.

The originality of this research lies in:

- The division of the quality concept into Q quality and q quality reflecting the two main categories of stakeholders, external and internal.
- A description of how quality is perceived throughout the product life phases, and how the product itself, due to dispositional mechanisms, can play a prominent role in what is normally regarded as ‘quality of sales’, ‘quality of service’, etc.
- The establishment of a complete chain of quality concepts, namely Q and q and their relation to product characteristics, their variability and subsequent breakdown of quality specifications.
- A comprehensive framework for Design for Quality which covers the major elements of DFQ, including quality-related activities at all levels of product development from the formulation of product strategies to the everyday activities in design work. The framework makes it possible to map existing elements of DFQ, including procedures, tools and techniques, and to identify voids and inconsistencies in a company’s total approach to Design for Quality.
- The identification and description of the quality mind-set and the role it plays in product development. Among other things, the mind-set enables the product developer to constantly evaluate the means he chooses with regard to quality. Hence, the mind-set is very dynamic and flexible in its use.
- The characterisation of robustness and robust design in relation to the new theoretical insight. For instance, robustness can be analysed by means of parameter models derived from knowledge about the product’s organs.
- The explanation of the basic significance of technology on product quality, and how technologies can be utilised in DFQ. For instance, mechanical technologies can be used for creating Q quality in the product, and process technologies can improve q quality in manufacture.
C.6.4 Evaluation of Results

Direct verification of design tools and methods is only achieved through successful application to practical design problem. Buur (1990) argues that this method is unrealistic due to the stochastic nature of the design process and the large number of influencing factors that make repetition of experiments virtually impossible. Buur suggests two methods for verifying the validity of design theory:

Logical verification:

- Consistency: there are no internal conflicts between individual elements of the theory.
- Completeness: all relevant phenomena observed previously can be explained or rejected by the theory.
- Well-established and successful methods are in agreement with the theory.
- Cases and specific design problems can be explained by means of the theory.

Verification by acceptance:

- The theory is accepted by a relevant scientific community.
- Models and methods derived from the theory are acceptable to experienced designers.

Logical verification has the draw-back that design theory is basically confirmed by analysis of cases and observations. This does not automatically provide a guarantee for the validity of the synthesis activities. Verification by acceptance implies a pedagogic problem: the willingness of a designer to accept a statement or method (his need, knowledge, experience), the complexity of the information (how much training is required?), and the pedagogic presentation (Buur 1990). Both types of verification method were applied in this research.

The research marks the first attempt to describe the fundamentals of Design for Quality in relation to all aspects of product development, in particular by way of new concepts and descriptive models. Thus, the primary objectives of the research are fulfilled. The approach to evaluate the results is mirrored in the selection of industrial activities and confrontations, see above, and the described verification concerns in this section.

The verification of the theoretical results constitutes a major obstacle to this work. Verification has been limited mostly to logical reasoning, in that the thesis represents a line of argumentation to show that the proposed concepts and models conform to the theoretical basis (theory of technical systems and properties theory) and are internally consistent.

As for verification by acceptance, the concepts of Q quality, q quality, and positioning and obligatory properties have been implemented by at least two major Danish manufacturing companies in their quality manuals/specifications, quality training programmes, etc. This has led to changes in mindset, language and written routines (ISO 9000). The pedagogic formulation of the other scientific results has not yet been pursued to a level where product developers in industry would feel confident about ‘experimenting’ with their use.
C.6.5 Conclusions About the Research Approach

The research has sought a delicate balance between practical studies in industry and pure academic research, where the latter has been predominant due to the novelty of the research topic, and in order to maintain an ‘unbiased mind’ regarding what is supposed to be so-called ‘best practice’.

It has been the goal to give a comprehensive review of the literature on Design for Quality, to ensure the best scientific basis for the research. However, due to the large number of publications on Total Quality Management, specific quality tools and techniques, and other issues related to DFQ, only a limited selection of the quality tools and techniques has been covered.

Part of the research that did not reach a satisfactory level of results was the discussion of the three last elements of the DFQ framework (tools and techniques, methodical design, and quality mind-set). This may lead to the incorrect assumption that these elements have a low priority in the framework. On the contrary, methodical design and the quality mind-set should play a crucial role in DFQ practice and be subject to further research.

The thesis core research approach problems were

- How can we add to a theory? The expansion of the Theory of Technical Systems and the Theory of Properties is made based upon the perceived need for better concepts and the ‘invention’ of new explaining models. From this point of view it does not matter how formal the empirical studies were.
- What type of phenomenological understanding or insight does the designer need for coping with DFQ? This mind-set and its proper implementation is an open question beyond proof or making probable in this thesis.
- What is formally regarded to be a framework? The DFQ framework seems powerful for communication and similar frameworks have been worked out based upon the idea in this thesis. But actually we do not know the formal identity of such a framework.

C.6.6 Continuation

The proposed quality concepts have been taken up by many important Danish companies as an important step forwards in their DFQ operations and dialogue between design and production. The thesis has led to an articulation of quality as an explicitly treated dimension of design in our teaching at the university and in industrial courses. The research has continued with a focus on ‘soft quality elements’ and the handling of Man–Machine Interfaces and Industrial-Design-related qualities.

C.6.7 References

Jørgensen KA (1992) Paradigms for research work. Instituttet for Produktion, AUC Denmark (lecture notes in Danish)

C.6.8 Reflections from the DRM Perspective

The work of Mørup and Andreasen is on identifying the influencing factors in the design process on product quality, with the intention of building these into a framework for companies to benchmark their design process for quality. They establish, from the literature, product quality as a strategic competitive factor, and the Success Criterion for this project. The central assumptions in their work are that product development plays a central role in creating quality, that it is possible to identify those product characteristics that ‘carry’ quality, and that successful pursuit of the above two should lead to an insight into the core process for designing in product quality.

The objectives are (i) to define the key concepts in quality in relation to the customer, the manufacturing and other life-cycle systems, (ii) to establish what aspects of a technical system influence quality and how these can be manipulated, (iii) to formulate a general framework for supporting and performing design for quality. The focus is on a Comprehensive DS-I using a literature survey, interviews and case studies in industry. The results are a new definition of quality that differentiates between quality perceived by stakeholders within the company (q) and that perceived by customers (Q), a list of product characteristics that influence these, and an initial framework (Initial PS) that can be used for benchmarking companies’ design processes for quality. Logical verification (DS-I) and (after completion of the PhD) verification by acceptance in industry (initial DS-II) are used to evaluate the results.

Using the DRM framework, the project can thus be classified as a Type 2 for the PhD work and Type 5 for the whole project.

Type 2/5: RC (Review) → DS-I (Comp) → PS (Initial) → DS-II (Initial)
C.7 Multi-disciplinary Design Problems

Student: Cirrus Shakeri,
PhD dissertation, Worcester Polytechnic Institute, USA, 1998 (Shakeri 1998)
Supervisors: David C. Brown (author) and Mohammad N. Noori

C.7.1 Introduction and Aims of the Research

Introduction
The scope of this research is the multi-disciplinary design of engineered systems. The research aims to develop and explore a new approach to discovering design methodologies for multi-disciplinary design problems. The objective is to demonstrate the generation of design methodologies using that approach. A design methodology is a scheme for organising reasoning steps and domain knowledge to construct a solution. It provides both a conceptual framework for organising design knowledge and a strategy for applying that knowledge (Sobolewski 1996). A design methodology can provide the knowledge for decomposing the problem into sub-problems, synthesising partial designs, evaluating and then combining them into more complete partial designs, ordering design tasks by considering proposals from all participants, and discovering and resolving conflicts.

Current multi-disciplinary design methodologies are based on ad-hoc strategies for handling the complexities that multiple points-of-view bring to the design process. Ad-hoc techniques reduce the complexity but give up the potential advantages of diversity. The common methodologies for multi-disciplinary design are based on compromising between different disciplines rather than collaborating between them. These methodologies do not use a systematic approach and, consequently, are not as efficient and effective as they could be.

In particular, the most common methodologies use sequential design to overcome the complexities of multi-disciplinary design. In sequential design, different disciplines take part in the design process sequentially. Hence, information sharing between different disciplines is limited to the interfaces between disciplines (Levitt et al. 1991). As a result, conflicts between disciplines are not discovered until they are very expensive to resolve, because their resolution may need to destroy the partial designs generated by the previous discipline.

Often in sequential design there is a lead discipline, perhaps a designer that makes some of the key decisions and tries to anticipate and remove some of the conflicts. The other disciplines conform to those decisions. However, that may prevent them from producing their best solutions. In a lead-discipline approach a single point-of-view dominates, and therefore constraints from that discipline are favored. This produces a lower quality design product and increases the number of iterations required to reach an answer. Hence, a key goal for better methodologies that was followed in this research, is to ensure well-integrated reasoning that provides equal opportunity to all the disciplines involved.

The approach to generating better design methodologies for multi-disciplinary problems was computer simulation of the design process. Analysing the behaviour
of physical systems in engineering applications by computer simulation using mathematical models has been a powerful tool in engineering. It is particularly useful in situations where real experiments with physical prototypes are not viable. This work extends the idea of using simulation-driven analysis to the area of engineering design research, by applying the concept to the design process instead of the design product. In this situation too, ‘real experiments’ would be extremely costly.

A computational model was developed and implemented that simulates the multi-disciplinary design process. By running that simulation system under many different conditions, and analysing the performance, detailed understanding of the design process is gained (Shakeri et al. 1998). As for simulations of physical systems, the computational model used is a simplified one in which the design activities that are usually carried out by humans are performed by software agents in a slightly simplified manner. We have developed these ideas using the multi-disciplinary domain of robot-arm design (Rivin 1988).

Importance

Using system-developed methodologies allows effective and efficient practices to be used from the start of a project instead of being learned from experience. These new methodologies are radically different from the sequential, discipline-based ones. Integration reduces the number of failures and backtracking by facilitating information sharing, thus saving resources and reducing design time. Integration also provides collaboration between different participants that, as a result, enhances the quality of the design.

The agent-based approach that we have adopted for building the computational model allows the incorporation of new technologies systematically and quickly through the addition or deletion of agents (Brown et al. 1996; Wooldridge 1997). Thus, new knowledge can be added, and old knowledge removed rapidly. Running a modified system will result in new designing behaviours being simulated, allowing production of new methodologies in response to a change in knowledge. In addition, design processes can be biased toward more environmentally friendly products, by altering the preferences for the alternative design methods that are built into each agent.

In industry, the number of specialists is increasing, while the number of generalists, capable of doing system integration, is decreasing. An increasingly specialised technological environment tends to force designers to concentrate on some disciplines more than others. Also, the knowledge burden on the designer keeps increasing due to more materials and more options (National Science Foundation 1996). Thus it is becoming harder to develop methodologies for the integration of multiple disciplines in design. This research directly attacks this problem.

Computers have mostly been used to support the manipulation and analysis of design product information. This work focuses on the design process, an aspect that has not benefited from computers very much. Simulation of design processes based on a multi-agent paradigm is a new area of research that has a high potential for practical as well as theoretical impact on the design of products. The use of multi-
agent systems technology is growing rapidly with the development of Java-based systems and agent access across the world-wide web.

The research is also important because it recognises the importance of incorporating knowledge, judgement and experience. “System integration, many consider, is an ill-structured problem. No specific rules have to be followed when doing integration. Experienced designers deal with system integration using judgement and experience. Knowledge-based programming technology offers a methodology to tackle these ill-structured integration and design problems” (Sobolewski 1996).

According to NSF’s report on Research Opportunities in Engineering Design (National Science Foundation 1996), “research areas that will have greatest impact on engineering design over the next 10 years are: Collaborative Design Tools and Techniques, Prescriptive Models/Methods, System Integration Infrastructure/Tools, and Design Information Support Systems”. This work covers all of these areas of research and hence is expected to have a strong impact.

C.7.2 Research Approach

What Was Developed

Part of the goal of the research was to produce an ‘approach’ (i.e., to producing methodologies). First we will describe in more detail the approach that was developed during the research. Then we will describe the way the research progressed: i.e., the approach to the approach!

This work proposed a new approach to the problem of producing better design methodologies for multi-disciplinary design based on the tight integration of different disciplines. The discipline-sequential approach, while poor, is quite simple. Its flaws are well known and have been part of the motivation for concurrent engineering (Brown et al. 1996).

However, integration tends to make the design process more complicated. To overcome this complexity, a computer system was developed based on a multi-agent systems paradigm in order to automate the simulation of the design process. The system also allows multiple design problems to be simulated in a small amount of time.

The system simulates examples of multi-disciplinary design processes while applying integration principles to the problem. The principles were developed from an examination of the literature. These principles include common design knowledge representation schemes and common communication mechanisms; design knowledge sharing among participants; cooperative problem-solving strategies among participants; simultaneous design process where possible; and mechanisms for conflict discovery and resolution. The principles are embodied in the system both in its architecture and at run-time.

The large chunks of discipline-specific knowledge are broken into small pieces, each being represented in the design system by an agent. Agent activation is triggered in an opportunistic manner and is unaffected by discipline boundaries. Agents might participate sequentially or in parallel. This leads to well-mixed use of
knowledge from different disciplines, and the possibility of parallel design activity for tighter integration and better efficiency.

The multi-agent design system is run with a very large number of different design problems. This is done by systematically varying the individual design requirements across their ranges in order to cover the space of requirements. Hundreds of design problems are presented to the system. Some problems do not lead to a successful design.

For each problem the traces of the agent activations (i.e., knowledge use) during the course of the design process are recorded. The many recorded traces consist of orderly patterns of different design actions that have led to a design solution.

Candidate design methodologies are extracted by generalising the patterns in the recorded design traces using clustering techniques. This both groups and identifies common aspects of related traces. The best clusters are the most ‘convincing’ methodologies. For each cluster identified, the commonalities in the Requirements are identified. This allows combinations of Requirements to be recognised as being most appropriately handled by a particular methodology.

Research Questions and Hypotheses

The main hypothesis for this research was that methodologies could be generated by using a computer to build up ‘experience’ by simulating design activity.

The question was ‘how?’. Clearly doing it with real people was impractical. This led to the idea of simulation. The need for integration led to the notion that any knowledge should have the potential to be applied opportunistically at any time, and that the knowledge should be split into pieces. These ideas, and the analogy with the design teams used to support Concurrent Engineering, made us decide to use a multi-agent design system.

The multi-agent paradigm not only matched the problem, but also provided some Software Engineering advantages.

We started by investigating ‘methods’, ‘methodologies’ and ‘integration’, as well as studying the literature on multi-disciplinary design. The latter confirmed the belief that there was a need for a systematic way of building good methodologies, and that many of those currently in place were ad hoc.

Early in the research we picked a domain in which to work: one that was well known to the student, had a clear multi-disciplinary flavour, was of a manageable size without being trivial, and appeared to have no strong, existing methodology of the type we were seeking. Robot-arm design seemed to be perfect. It demonstrated well the tendency for researchers from each discipline to write about the design problem as if their discipline’s contribution were dominant.
We started by implementing a ‘base-level’ working system that used a non-integrated approach to robot-arm design. This tested our understanding of the relevant knowledge and methods used. Early prototype implementation is an important research technique that enhances domain understanding, acts as a catalyst for learning programming and system development techniques, and also forces precise definitions of concepts.

The base-level system also provided some feedback about where errors occurred during designing and where the knowledge might be decomposed into pieces. The choice of Java for the implementation allowed portability between systems and provided the ability to effectively handle agents in parallel.

Next the framework for a multi-agent design system, to be called Robot Designer (RD), was developed. The question of how to split the knowledge from each discipline into pieces was addressed, and the resulting pieces were encoded as agents, and added to the framework. Decisions were made as to what needed to be stored as a record of every agent’s action, such that these traces might be able to form suitable methodologies. The traces were accumulated, but not the designs.

A great deal of time was spent on developing a failure-handling system for RD so that the constraint failures could be recovered from, while allowing parallel paths through the agents to be recorded correctly.

An important issue at this time was the relationship between design quality and methodology quality. Clearly one would like the methodologies produced to lead to high-quality designs. The traces that actually lead to designs must include no failing constraints, and hence at least possess a certain level of quality. But they may not all be of equal quality. In addition, our simulation of the design doesn’t include all of the design knowledge available, and hence may not lead to the best possible designs – in fact, it would be foolish to pretend that one could make a perfect simulation of the design activity in a complex multidisciplinary situation. However, as we expected these methodologies to be followed principally by people, they only need to act as guidance, and the human designer should be able to ensure the quality of the result.

The compound hypothesis was that less precise knowledge might lead to adequate designs, and that adequate designs were associated with traces that might form a methodology capable of guiding the production of high-quality designs. This hypothesis was never fully explored.

A key question to be addressed at this point was how to exercise RD such that the whole design space was explored. This was important because we wanted to generate methodologies for all types of designs in the space. The approach taken was to drive the system with as many different sets of requirements as possible, such that the whole design space was explored. The hypothesis was that all reachable regions of the design space could be found by systematically varying the requirements in order to adequately scan the requirements space. By experimenting with the degree of change between requirements we were able to convince ourselves that this approach was successful.

Another issue that this raised, and that we spent a lot of time investigating, was the relationship between the requirements space, the design space and the trace space. These relationships were explored and revealed using graphical representations.
The next question to be addressed was how to form traces into methodologies. Several techniques were considered until a clustering algorithm was tried that gave appropriate results. While other methods might be appropriate we focused on just one. Once clusters were identified they were re-expressed as rules.

A final issue that was examined was how to best express the requirements that correspond to a particular methodology.

The main hypothesis, that methodologies could be generated using the computer by simulating enough design experience, was demonstrated by this research.

C.7.3 Results and their Evaluation

A knowledge-based model of design was adopted in order to implement the proposed strategies for integration. To implement the proposed model a knowledge-based multi-agent design system, RD, was developed that simulates the design process.

Both the general multi-agent design system architecture, and the RD system developed are results of the research. The approach to breaking knowledge up into pieces such that they can be incorporated into agents is also a result of this research.

The Java-based computer program called RD (Robot Designer) was implemented for parametric design of two degrees of freedom (2-DOF) planar robot arm. We used RD to solve a set of 960 design projects. Figure C.22 shows how many projects followed a specific trace. The promising result is that many projects followed similar traces. The total number of possible traces is the product of the number of design approaches of all the designer agents. For the experiments shown in Figure C.22 (Shakeri and Brown 2004), the total number of possible traces is 2304. However, despite all those possible traces only 84 were followed to generate successful designs, i.e., less than 4%.

The low percentage of successful, relative to ‘possible’, traces indicates that for each group of projects that followed a particular trace there is a unique combination of approaches leading to successful designs. Hence, there is a high chance that if similar projects follow the same trace they will succeed in generating a successful design. As a result, the path followed by those projects can lead us to formulating a design methodology for the projects that followed that trace as well as projects that are similar.

The traces in the set of successful traces that are close enough can be clustered together to form a generalised trace. A generalised trace covers all the projects that followed each of the traces incorporated in the generalised trace. Design methodologies are formulated by extracting the correlation between a generalised trace and the design projects (i.e., the sets of requirements) that produced that trace. The sample design methodology that is shown below is the English translation of the correlation between design projects and the corresponding traces.
Methodology:

- choose the location of the base of the robot: ‘left or below midway of the workspace length’
- choose the material: ‘steel stainless AISI 302 annealed’
- select the shape of the cross section of the link: ‘hollow round’
- choose the structural safety factor: ‘3’
  - do the design and proceed to the next step
- choose the link 2 to link 1 length ratio: ‘0.5’
  - do the design and proceed to the next step
- pick the configuration of the arm: ‘left-handed’
- select the ratio of the cross section dimension of the link to minimum required by stress analysis: ‘4’—if it fails select ‘3’
  - do the design and proceed to the next step
- find the accessible region: use Equation 2-4
- find the deflection of the tip: use Equation 2-14
- choose the type of controller: ‘PD’
  - do the design and finish the process.

Frequency of Traces

![Plot showing frequency of traces](image)

Figure C.22 The plot shows how many projects followed a specific trace (This figure appeared in Shakeri and Brown 2004)
C.7.4 Conclusions About the Research Approach

The main characteristic of this research approach was the use of a computational model to simulate design activity, that, due to its size and complexity, would otherwise be impossible due to the time and cooperation required. The main positive feature of this approach is therefore that it allows the normally ‘impossible’ to be handled. Other positive features should be self-evident from what has been presented so far.

As with all research, there are a variety of trade-offs and assumptions incorporated. Some of the assumptions need to be verified more rigorously.

The largest issue is that for every domain the approach requires collection of a very large body of knowledge – recognised as a very difficult task – and the construction and testing of a large and complex software system. Once that’s done the methodologies should follow fairly easily, assuming that other domains behave in a similar way to our test domain.

However, there is a very large investment in time and effort required in order to start getting results. This approach will only be viable if the methodologies generated are extensively used and if they provide large gains in quality or cost/time savings. Of course, we have limited evidence that such gains are in fact provided.

Also, a big assumption is that both the simplified model of designing (e.g., with respect to failures) and the simplified knowledge that are embodied in the system, are not so simplified that the traces generated are atypical. Also assumed is that all traces that lead to successful designs should be included in the process that leads to methodologies. This is without regard to the quality of the designs that these traces produced: all we know is that they are ‘correct’. This may affect the quality of designs that can be achieved by following the methodologies.

Clearly this leads to a trade-off between the effort required to build the system (more complexity and authenticity leads to more system building effort) and the quality of designs that can be achieved by following the methodologies. This trade-off needs to be explored.

The interesting issue of the relationship between the clustering of designs, the clustering of requirements, and the clustering of traces (to form methodologies) still needs more exploration. The approach of systematically scanning across the requirements space needs to be investigated further to test for sensitivity to different domains and problem areas. Perhaps an adaptive scanning approach could be tried?

The final weakness is that it is very hard to test the methodologies that are generated under realistic situations, as this would require extensive use by multi-disciplinary design teams with many design problems. As a consequence, it is hard to establish whether the designs produced by using the methodologies are indeed of high quality.

C.7.5 Continuation of Project

The potential applications of this research are in multi-disciplinary design situations, such as those that occur throughout the automotive industries, where
large gains can be achieved with integrated methodologies. In addition, current methodologies can be analysed for flaws and bottlenecks, and necessary refinements made. New methodologies can be customised so that they are biased toward specific objectives such as manufacturability or being environmentally friendly. By applying this approach the response time for the incorporation of new technologies in design processes should be reduced. Methodologies can be refined as soon as a change occurs in the market or in the organisation of the company.

At this time no additional work has been carried out on this project since the Ph.D. (Shakeri 1998) was completed. As with any large, student-driven software project, transferability is an issue. There would be a very steep and significant learning curve associated with taking over the complex Java code that was designed. Another serious issue is that finding a student with the right combination of CS and ME skills is very rare, and would take some time to develop. We would very much like to experiment with confirming the quality and utility of the methodologies generated, especially for new multi-disciplinary problems. Automobile, Aircraft, Computer and Mechatronics design should be fruitful areas to investigate. The approach that we have proposed has been developed based on parametric design problems. Applicability of the approach to other types of problems needs to be investigated. This research has proven that the following hypothesis is true: Computers can provide us with better ways of doing design by discovering design methodologies that integrate multiple disciplines into the design process. It has been shown that it is possible to use computers to simulate the design process, and then analyse the results of the simulation to synthesise design methodologies that have superior features. This forms the basis of a new approach to the study of multi-disciplinary design processes.

C.7.6 References

C.7.7 Reflections from the DRM Perspective

The work of Shakeri et al. is to develop a computational approach to the generation of successful design methodologies for multi-disciplinary design, with the intention of reusing these system-developed methodologies in appropriate situations, i.e., for situations in which the methodologies were found to have been particularly successful. The assumptions are that using systematic approaches will lead to effective and efficient design processes, and that common methodologies for multi-disciplinary design do not use a systematic approach and are not as effective and efficient as they could be. The most common methodology used today is sequential design that, the authors argue, is simple, but does little information sharing, which leads to a large amount of backtracking in the design process, a large number of failures and little collaboration. These lead to long time to design, over-use of resources and poor design quality, respectively. Their overall Success Criteria, we interpret therefore as high design quality, low resource use and low design time, and the Measurable Success Criteria as high level of information sharing, low number of failures, low number of backtrackings. Furthermore, the methodology should be simple.

The authors start with a review-based DS-I using the literature search and analysis. They then developed a computer simulation of design processes to identify sets of activities that lead to successful designs (that satisfy all constraints), i.e., successful design methodologies. The software uses software agents as ‘designers’ who have specialist domain knowledge and work together using integration principles taken from the literature. The activities of the agents are traced, and patterns identified using clustering techniques in order to extract successful methodologies (Comprehensive PS). A central assumption in this work was that the simulated processes represent actual design processes. From an AI point of view, an interesting feature of this work was to enable the computer to ‘discover’ design methodologies from test cases.

The work therefore involved a review-based DS-I, which was followed by a comprehensive support development. Using the DRM framework, the project can thus be classified as a Type 3.

*Type 3: RC (Review) → DS-I (Review) → PS (Comp)*
C.8 Design for Reliability in Mechanical Systems

Student: John Stephenson
PhD dissertation, Cambridge University, UK, 1995 (Stephenson 1995)
Supervisor: Ken Wallace (author)

C.8.1 Introduction and Aim of Research

In order to maximise their competitive positions and ensure their future profitability, industrial organisations aim to develop the best products in the shortest time and at the lowest cost. Defining the ‘best’ product is not easy as a complex blend of many characteristics must be achieved and delivered at the right time. Every new product must fulfil its intended functions safely, continue to do so over its intended life, and at a cost that customers are prepared to pay. These external characteristics can be grouped under the broad headings of performance, reliability and economy.

Human processes are controlled by the decisions taken, and all decisions depend on forecasts and the evaluation criteria used. The decisions made during the design stage influence all the subsequent stages in the product development process. During the design process numerous forecasts are needed to answer questions such as: How easy will the proposed product be to manufacture and assemble? How well will it perform in operation? How reliable will it be? What will be the selling price? What will happen to it at the end of its useful life? A great deal of research effort goes into developing methods to improve our ability to forecast.

When starting a design research project one of the challenges is to isolate one aspect on which to focus the study. This is more difficult than in the natural sciences as the influences are so complex and interlinked. It is important to define an area of interest that is sufficiently self-contained that it can be studied in a reasonable period of time and useful conclusions drawn.

The research project described here was located in the product area and addressed the main question: For a mechanical product, how does one ‘design in reliability’ as early as possible in the design process? It did not address performance or economy, though, of course, these are linked. It concentrated on the technological aspects of the product with the intention of developing an understanding of the issues and hence a theory on which a specific design for reliability (DFR) method could be based and tested.

The research was based on the theory of the properties of technical systems developed by Aguirre during his PhD research project at Cambridge in the 1980s (Aguirre 1990). He identified basic properties after creating and analysing a database of around 3500 guidelines extracted from the literature. The properties can be classed as being either ‘external’ or ‘internal’. The external properties, which have already been mentioned, are what the customer or user sees: performance, reliability and economy. The internal properties were identified as simplicity, clarity and unity. Design engineers do not manipulate these properties directly. Aguirre argued that designers should aim to maximise the internal properties of simplicity, clarity and unity. This raises a number of questions: What are the links...
between the internal and external properties? How does one measure these internal properties? What are the guidelines for achieving high levels of these properties?

The objectives of this research were to study specific cases of good and bad reliability, develop a theory of mechanical reliability, test this theory, and produce a method to support mechanical designers. The crux of the research was to establish the relationship between the internal properties and the external property reliability.

How can these research objectives be achieved and tested in a rigorous way in a PhD project? This was achieved by undertaking this research in collaboration with a manufacturer of earth-moving equipment. Extensive empirical data were gathered from case studies of mechanical systems on the company’s backhoe loader (Figure C.23). The company operates in a very competitive market and had identified that increasing the reliability of its products would give them a stronger marketing position.

The company operates a warranty scheme whereby faults are rectified at the company’s expense within the warranty period. If a particular piece of equipment is returned regularly for a particular fault, the company responds by modifying the design to overcome the fault. The company held excellent records of all its warranty claims and all its design changes — but no attempt had been made to relate the two, that is, identify in a systematic manner the effect that specific design changes had on reliability and, more importantly, whether there were design guidelines that could be extracted to help designers ‘design in reliability’ early in the design process and prevent the necessity for corrective action.

The need to improve reliability, the availability of a large quantity of data and a clearly defined problem area provided a clear specification for this research project.

C.8.2 Research Approach

One of the challenges in design research is validating a new method or technique, particularly within the time frame of a PhD research project. The true test of a theory or method is its ability to predict, i.e., make accurate forecasts of future
Appendix C: Example Research Projects 373

results. This is referred to as *predictive testing*. The difficulty arises because of the long timescales involved in designing, manufacturing and testing new mechanical engineering products. By the time that a method has been developed, there is seldom time to undertake predictive testing on a real design project.

To overcome this problem the idea of a *historical testing* was introduced. A series of four case studies, described in more detail later, was planned. The first was to investigated the practicability of gathering, storing and analysing the data with a view to identifying the issues involved. The second investigated whether the results and ideas obtained from the first case study could be repeated and was used to formulate the preliminary DFR method. The third and fourth were used to test and develop the DFR method using historical testing. By chance, two opportunities arose during the project to test the method in a predictive mode.

The idea behind historical testing was to go back to the time to when a particular system was being designed, to pick up the *design data* at that point and then apply the new DFR method (which was not available when the product was being designed) to the data to predict areas where reliability problems would be expected to occur. For this to be a “true” experiment, it was essential not to look at the reliability records of the system being studied until after the predictions had been made using the method.

The research followed three main stages:
- Stage 1: A literature review and survey of design principles
- Stage 2: Developing the theory and method (Case histories 1 and 2)
- Stage 3: Testing the method (Case histories 3 and 4)

*Stage 1: Literature Review and Survey of Design Principles*

A study of the literature was undertaken to determine the extent of the existing work on DFR and to identify relevant theories, principles, guidelines and methods.

The literature review showed that reliability is determined by three phases of the product life: design, production and exploitation. Design has the potential for the greatest effect on reliability. The following reliability approaches were identified:

- reliability apportionment;
- design layout decisions;
- load strength theory;
- robust design;
- reliability prediction;
- failure analysis;
- reliability testing.

Within each of these approaches a number of specific methods are available. For example, nine methods are available for failure analysis including: Failure Modes and Effects and Criticality Analysis (FMECA), see BS 5760; and Hazard and Operability (HAZOP) analysis. Despite the many techniques available, it was concluded that there was no coherent reliability approach available that can support decisions made during conceptual and embodiment design. Although some design layout principles are available, in their current form they do not aid the
development of component shapes to improve reliability at the embodiment design stage. It was also concluded that the most relevant concepts for conceptual and embodiment design are generally applicable design principles. Design principles encapsulate ‘best practice’ and so a theory of reliability based on them suggests that a ‘good’ design will be a reliable design. Considerable effort therefore went into identifying design principles, developing a theory for reliability based on them, and producing a method based on this theory.

The most important contributions in this area were identified as Aguirre (1990); Pahl and Beitz (1996); French (1994); Suh (1990); Taguchi (1993); Hubka and Eder (1988) and Dimarogonas, Redtenbacher, and Reuleaux (Dimarogonas 1993). Of these, only Aguirre and Suh adopt approaches that are truly based on principles or axioms and thus provide a sound basis for a theory of reliability. French, along with others, emphasises the importance of interfaces stating: “It is important, and often a great help to pay full attention to the design of interfaces, particularly in the early design stages, and also because the constraints associated with them point the way to go” (French 1992).

The principles identified by Aguirre, based on internal and external properties, and the importance of interfaces underpinned the theoretical aspects of this research project. Empirical data was gathered from four case studies and in each case study, around 2000 machine records were consulted.

Stage 2: Developing the Theory and Method (Case Histories 1 and 2)

Case study 1 — Return-to-Dig Mechanism (See Figure C.23)

The return-to-dig mechanism is part of the sub-system that controls the positioning of the front bucket. Under the control of the positioning system, the mechanism performs two functions: bucket levelling and return-to-dig. The bucket-levelling function maintains a constant angle of the bucket at the loader arms rise and fall; the return-to-dig function uses some of the same mechanism as the bucket levelling function and speeds the digging cycle by automating the positioning of the bucket between dropping off a load and picking up the next.

Five separate configurations were studied and these provided examples of the differences in reliability between configurations with shared and separate function carriers. Part of a typical configuration history is shown in Figure C.24.

This first case study was used to help identify those design factors that influence reliability. The evolution of the configuration showed that clarity was particularly important, with simplicity having a secondary influence.

Failure probabilities were calculated from the warranty data and presented in a number of forms. A typical representation is shown for three return-to-dig configurations in Figure C.25.

Case study 2 — Boom Lock Mechanism (See Figure C.23)

The boom lock mechanism locks the boom and prevents it swinging around while the backhoe is, for example, being driven on the road. Eleven boom lock configurations were studied.

This case history was used to confirm the relevance and relative importance of clarity and simplicity in reliable configuration. It was during this case study that a preliminary DFR method for assessing clarity and simplicity was formulated based
on a component/interface model. This model will be described later. During the research project a new boom lock configuration was released and this provided a valuable opportunity to test the DFR method as a predictive tool.

**Figure C.24** Part of a typical configuration history

**Figure C.25** Level of failure of return-to-dig configurations A, C and D
Stage 3: Testing the Method (Case Histories 3 and 4)

Case Study 3 — Seat Mechanism

This mechanism provides forwards and backwards adjustment of the driver’s seat as well as allowing the seat to rotate through 180 degrees between the driving position and the backhoe operating position. These functions are performed by a complex slide and turn mechanism that, for obvious reasons, must be extremely robust and reliable. Seven seat configurations were analysed.

This case history was used to test the validity of the DFR method as well as deepening the understanding of the influence the internal properties had on reliability. Special attention was given to the third internal property, unity.

For each configuration, a component/interface model was produced and each assessed for potential areas of unreliability using the DFR method. Only after these analyses had been completed, were the data gathered on the actual reliability of the different seat configurations. This represents a way of validating the DFR method using historical testing. Again, during this investigation a new seat mechanism configuration went into production, again allowing the DFR method to be used as a predictive tool.

The different seat mechanism configurations show different levels of function sharing, particularly within the slide rails — which were the components to fail most frequently. The case history therefore provided excellent examples of successful and unsuccessful function sharing. In addition, the importance of unity was established in the sense of providing sufficient strength for the components.

Case Study 4 — Accelerator Linkage Mechanism

The accelerator linkage mechanism has two inputs: one is the accelerator pedal for determining the speed of the backhoe when being driven; the other is the lever used when operating the loader arm. Four configurations were studied.

As with the third case study, each configuration was analysed using the latest version of the DFR method and potential failure areas identified before the actual failure data was obtained. Again this provided a way of using historical data to validate the DFR method.

C.8.3 Results

One of the key results is the DFR method developed. It will be described by applying it to part of a cable and lever mechanism from a boom lock configuration shown in Figure C.26. To begin with the component/interface diagram is created as shown in Figure C.27. Then, four steps are undertaken as described below.

Step 1: Simplicity

Simplicity is assessed by counting the number of components and interfaces in the model. This approach is not complete as simplicity relates to other concepts such as ease of manufacture and maintenance. However, this assessment is easy to perform and provides a rough comparison of configurations. In most cases only significant differences in the overall numbers of components and interfaces between configurations impact on reliability, provided the levels of clarity and unity are
maintained. For the model shown in Figure C.27, there are 6 components and 7 interfaces. As the cable housing and the handle base are both fixed to the frame, these are considered to be one 'component' and are thus joined together in the model.

**Figure C.26** Simplified lever and cable from one of the boom lock mechanisms

**Figure C.27** Component/interface model
**Step 2: Clarity**

Clarity is assessed by using a ranking system to indicate interfaces with different levels of clarity. The ranking system is equivalent to a ‘penalty’ point system, where a ‘1’ is the clearest type of interface and a ‘3’ is the least clear. The assessment of clarity is based on how the forces performing each function are transferred at an ‘active’ interface. This is a crucial and difficult aspect of the method and is described in detail in Stephenson (1995). Of the 7 interfaces shown in the model in Figure C.27, 1 interface is ranked as Clarity 1, 3 as Clarity 2, and 3 as Clarity 3.

**Step 3: Unity**

Unity in this case is assessed by estimating the peak loads and checking to see if the components are strong enough to fulfil their functions. If there are components that might fail under load, then the appropriate component box in the model is shaded. In the lever and cable mechanism shown in Figure C.26, quick calculations show that each component is strong enough.

**Step 4: Use of Results**

Clarity is the main issue in this example as simplicity is only relevant when several design solutions are being compared. Unity is not an issue as no weak components were identified. From the assessment of clarity, three of the interfaces were ranked Clarity 3 and these have the greatest potential to fail and need to be reviewed carefully. After reviewing the design configuration carefully, the clarity at these interfaces can be improved by:

- Increasing the force of the positioning function (cable tension) by providing a suitable combination of a longer lever, a stiffer spring and a pre-loaded spring.
- Reducing the resistance force by pulling the cable coaxially with its housing and changing the cable forces to reduce friction.

**C.8.4 Evaluation of the Results**

The new DFR method described above requires further research and development. During the project, a number of instances were identified where the method did not work and tentative reasons were suggested for these instances. However, overall the results are encouraging.

The DFR method has been validated in two ways; historical and predictive testing. A total of 27 configurations was analysed. Out of these, 6 were used to test the method historically and 2 predictively. For the historical cases, 14 failure-mode predictions were made and 9 turned out to be correct; and for the predictive cases, 4 predictions were made and 2 were correct. So out of a total of 18 predictions, 11 (61%) were correct suggesting the value of the method. It is important to note that in each of the 8 configurations analysed, the major failure mode in service was predicted.
Failure can occur at an interface or in a component in a mechanism. Simplicity thus relates to the number of potential failure areas (components and interfaces). However, simplicity does not relate to how likely the potential failures are to occur. This is because it does not explain how the failure mode will occur. Therefore simplicity has an indirect relationship with reliability.

Clarity relates to how functions are performed, and so how they can fail. Thus clarity has a direct, causal relationship with reliability and can be most easily assessed at active interfaces.

Unity relates to the strength of components that includes the effects of stress and materials, e.g., rupture, wear, fatigue and corrosion. Like clarity, unity has a direct relationship with failure by being able to explain failures.

It is always interesting when research produces a counter-intuitive conclusion. This occurred in this research. When designers were asked what characteristic they thought had the greatest influence on mechanical reliability, most said it was simplicity. In the case histories analysed, the majority of failure modes occurred due to a lack of clarity. This has resulted in the importance of simplicity in achieving good reliability being challenged. Simplicity, as explained above, is seen as having an indirect relationship with reliability, whereas clarity is seen as having a direct relationship. Complex configurations that have clarity will be more reliable than simple configurations that lack clarity.

The clear and important conclusion from this research is that whilst reliability can be achieved by a product that is simple, it cannot be reliable without being clear.

C.8.5 Conclusions About the Research Approach

A research project sets out to answer a number of questions. It is therefore reasonable to ask at the end of a project whether or not the starting questions have been answered. In the case of this project the main starting question was: For a mechanical product, how does one ‘design in reliability’ as early as possible in the design process? This depended on answering many subsidiary questions such as: What are the links between the internal properties and reliability? How does one measure these internal properties? From what has been described above, it can be seen that substantial progress has been made in answering these questions, though many questions remain unanswered.

It is also important to ask if the research method was rigorous. In this case it was based on a valid, if tentative, theory of the properties of mechanical systems. The research was underpinned by the gathering of large quantities of empirical data from four case studies - each case study involving reference to around 2000 machine records. During these case studies, a total of 27 different configurations were studied in detail. A DFR method was developed and tested both historically and predictively. Of particular importance was the development and use of historical testing as a valuable design research method.

Research always raises more questions than it answers. That was the case in this research project and there is much work to be done. The method is in a preliminary form. A number of instances were found where the method did not work. These
provided valuable insights and highlight the need for further testing and development.

C.8.6 Continuation

There are five areas where further research is needed:

- Developing further the understanding of clarity.
- Broadening the DFR theory beyond its current area of mechanisms.
- Developing a complete theory of the properties of mechanical systems to include the links between the internal properties of simplicity, clarity and unity and the external properties of performance, reliability and economy.
- Extracting guidelines that will help designers achieve high levels of simplicity, clarity and unity.
- Integrating the DFR method into a CAD system.

C.8.7 References


C.8.8 Reflections from the DRM Perspective

The work of Stephenson and Wallace is on design for reliability in mechanical systems, with the specific objective of supporting analysis of reliability potential of solutions during the early embodiment stages of design. Analysis of work from the literature as well as data from industry and its immediate problems with product reliability led to the decision of taking product reliability as the Success Criterion, and reduction of frequency of failure of products, as its operationalisation. A central assumption in this work was that being able to predict potential reliability of a design would support development of more reliable products.

The primary focus of this work is a Comprehensive DS-I using historical testing, which uses data gathered earlier in the company for developing and testing hypotheses. The research led to a method for estimating potential reliability by
analysing the product characteristics (comprehensive PS). While part of the historical data was used in developing the understanding in DS-I, another part of the data was used to test the method developed (initial DS-II).

The work therefore involved a Comprehensive DS-I, which was followed by a comprehensive support development, and subsequent Initial DS-II. Using the DRM framework, the project can thus be classified as Type 5:

Type 5: RC (Review) $\rightarrow$ DS-I (Comp) $\rightarrow$ PS (Comp) $\rightarrow$ DS-II (Init)
References

Bessant JR (1979) Preparing for design studies: ways of watching. Design Studies 1:77-83
Bessant JR, McMahon BJ (1979) Participant observation of a major design decision in industry. Design Studies 1:21–26
Pressverbindungen mit gerändelter Welle. Fachgebiet Konstruktionstechnik und
Entwicklungsmethodik, TU Berlin, Berlin, Research proposal, unpublished
University of Twente, PhD thesis
Blessing LTM (1995) Comparison of design models proposed in prescriptive literature. In:
Perrin J (ed.) The role of design in the shaping of technology (Workshop COST A3/A4),
Lyon
Blessing LTM (1997) A method for identifying scope and assumptions in developing and
selecting methods and tools. Cambridge University Engineering Department, Technical
Report CUED/C-EDC/TR37
International CIRP Design Seminar. CIRP, Hong Kong
behaviour in design: Individuals, Teams, Tools. Springer Verlag, Heidelberg, pp 298–
303
of Ken Wallace. Engineering Design Centre, University of Cambridge, Cambridge, pp
32–41
Sciences de la Conception: l’enjeu scientifique du 21e siècle en hommage à Herbert
Simon, Ed. J. Perrin, INSA, Lyon, France
V (ed.) International Conference on Engineering Design (ICED’95). Heurista, Zürich,
Prague, pp 502–507
Blessing LTM, Chakrabarti A, Wallace KM (1992) Some issues in engineering design
research. In: Cross N (ed.) OU/SERC Design Methods Workshop, The Open University,
Milton Keynes
relation to a general design research methodology. In: Frankenberger E, Badke-Schaub
P (eds) Designers: The Key to Successful Product Development. Springer-Verlag, pp
56–70
Bligh TP (1990) Personal communication
Boehm BW (1979) Software Engineering: R & D trends and defense needs. MIT Press,
Cambridge MA
Compuer 61–72
Booch G, Bryan D, Peterson CG (1994) Software engineering with Ada, 3rd edn. Addison-
Wesley
Bracewell RH, Shea K (2001) CaeDre: A product platform to support creation and
evaluation of advanced computer aided engineering tools. In: Culley S, et al. (eds)
International Conference on Engineering Design (ICED’01). IMechE, London,
Glasgow, pp 539–546
computational design tool research. In: Culley S, et al. (eds) 13th International
188
Braha D, Maimon OZ (1998) A mathematical theory of design: foundations, algorithms and
applications. Kluwer Academic Publishers Norwell, MA, USA
Catterall M, Maclaran P (1997) Focus group data and qualitative analysis programs: coding the moving picture as well as the snapshots. Social Research Online 2:12
Cook TD, Campbell DT (1979) Quasi-experimentation: design and analysis issues for field settings. Houghton Mifflin Company, Boston
Cross N (1994) Engineering Design Methods: strategies for product design, 2th edn. John Wiley and Sons Ltd, Chichester
Fricke G (1993a) Konstruieren als flexibler Problemlöseprozeß: empirische Untersuchung über erfolgreiche Strategien und methodische Vorgehensweisen beim Konstruieren. VDI-Verlag, Düsseldorf
IEEE (1998) IEEE Std 1012-1998: Standard for software verification and validation. Institute of Electrical and Electronics Engineers
ISO (1998) ISO 9241-Ergonomic requirements for office work with visual display terminals (VDTs) – Part II: Guidance on usability. International Organization for Standardization
Kelle U (1997) Theory building in qualitative research and computer programs for the management of textual data. Social Research Online 2:17
Miles MB, Huberman AM (1984) Qualitative data analysis: a sourcebook of new methods. Sage, Beverly Hills
Monod J (1969) From biology to ethics. Salk Institute, San Diego
Preiss M (2006) Ein Reifegradmodell für die Optimierung der Produktentwicklung kundenorientierter Zulieferbetriebe. ETH, Zürich, Switzerland, PhD thesis
Shneiderman B (1998) Designing the user interface, 3 edn. Addison-Wesley Publishing Company
References


ten Have P (1977) Sociologisch veldonderzoek. Boom Meppel, Amsterdam


Yates I (1998) Yes, Prime Minister, but it is also investment, investment, investment,. In: 14th Lord Nelson of Stafford Lecture. Institute of Electrical Engineers, London
action research, 40–1, 193–194, 257, 273
actual Support, 34–35, 142, 145, 147, 167–172, 177, 181, 184, 197–199, 201, 211
actual Support Description, 35, 142, 172
actual Support Documentation, 35, 142, 172, 198
alternative explanation, 22, 71, 92, 94, 123–124, 203, 205, 211, 240, 267
answer Analysis, 99–101
ARC-diagram, 64–66, 217, 234
artificial intelligence, 290–291
attribute, 20–24
value, 21–24, 26–27, 184–185
axioms, 78
bias, 79, 115–116, 123–124, 192, 205, 261, 270
elite, 123
experimenter, 115
measurement artefacts, 116
bibliography, 218, 226
CaeDRe methodology, 174–175, 282–283
case size, 244, 250
case study, 268–269
CASE tools, 175, 294–295
causality, 91, 123–126, 190, 265
checklist
for reviewing empirical studies, 84–85, 243ff
for summarising scope and assumptions, 149
citations, 83, 223, 226
comprehensive study, 18
Computer Supported Collaborative Work, 294
conceptual framework, 77–78
construct, 96, 112, 125–128, 275
content analysis, 269
test group, 206–207, 245, 265
correlation, 93, 121–122, 270, 275
cost-benefit calculation, 204
cost-effectiveness calculation, 204
criteria, 23–24, 26–28, 57–59, 60–63, 185, 281
Measurable Success Criteria, ii, 26–29, 57–59, 185–186, 203, 206, 234
Success Criteria, 26–28, 57–59, 185
data analysis, 102, 120–124, 246
data validity, 115–116
givenness of data, 240
indexing, 111, 116, 120
interpreting, 121–123, 210
processing, 116–121
data coding, 116–121, 251, 261
deductive, 117
double, 120–121
dual, 119–120
e emergent, 117
inductive, 117
inter-encoder reliability, 120–121
post-defined categories, 117
pre-defined categories, 117
pre-set, 117
process, 120
schemes, 116–120
data-collection methods, 98, 103–109, 255ff
data-driven, 78–79, 117, 137, 191–192, 274
debugging, 289
definition
conceptual, 95–96
operational, 20, 91, 95–96, 108
demand characteristic, 111, 115, 127
Descriptive Study I, see DS-I stage
Descriptive Study II, see DS-II stage
Descriptive Study, 76
design, 1–2, 10
  guidelines, 143
  method, 142, 277ff
  methodology, 107, 142, 220, 277
  science, 4
  support, 4–5, 33–34, 141ff
  theory, 3–4, 31
tools, 143
  understanding, 5, 9, 15–17
design research, 2–8, 10, 14, 18
  method, 6, 8–9
  methodology, 9–10, 13ff
  objectives, 5
  status of design research, 6–8, 29–37
  types of design research, 18–19, 60–63
design-test-debug cycle, 176
desired situation, 9–10, 19ff
Digital Object Identifier, 226
distal outcomes, 185, 206
document analysis, 269
documenting research, 135, 215ff
drawing conclusions, 128, 211–212, 274
DRM, 9, 11, 13ff
  entry points, 45
  experiences, 231–236
  framework, 10–11, 14–15, 38
  iterations, 17
  objectives, 14, 231
  stages, 14, 17, 29–38
DS-I stage, 15–16, 31–33, 75ff, 189
  Comprehensive DS-I, 18–19, 60–63, 80–81
  deliverables, 33, 76
  review-based DS-I, 18–19, 60–63, 80–81
  objectives, 32, 75
  process, 80–81
  research plan, 43, 52, 80–81, 94, 102ff, 124
DS-II stage, 16, 35–38, 181ff
  Comprehensive DS-II, 18–19, 60–63, 195–196
deliverables, 38, 181–182
  Initial DS-II, 18–19, 60–63, 195
  input, 198–199
  objectives, 37–38, 181
  process, 196–197
ecological fallacy, 246
element, 20–21
emperical study, 76, 83, 89, 102, 114ff
  checklist, 243–253
empiricism, 240
endogenous change, 210
error
  random, 115
  systematic, 115
  type-1, 211
  type-2, 211
ethnomethodology, 240–241
evaluation, 19, 26–27, 29, 35–37, 60–63, 145, 176–177, 181ff
  applicability, 37, 184, 195, 208
  application evaluation, 37–38, 167, 184–186
  evaluation Plan, 204, 210
  focus, 199–201
  formative, 191
  input, 198
  methods, 258, 281
  program evaluation, 192–193
  questions, 169, 193, 205
  research, 190–195, 208
  software evaluation, 194
  Success Evaluation, 37, 167, 184–186
  summative evaluation, 191
  types of, 184
  usability, 37, 184, 195, 296, 298
  usefulness, 37, 185, 195
  evolutionary approach, 282, 285
  example projects, 305, 307ff, 318ff, 328ff, 339ff, 351ff, 361ff, 371ff
examples
  reliability, 11, 45, 47, 53, 55, 63, 82, 86, 89, 92, 113, 129, 130, 131, 132, 136, 150, 153, 156, 161, 164, 166, 171, 172, 177, 185, 201, 249–250, 292
  synthesis, 12, 157, 163, 166, 169, 170, 177, 186, 202, 292
  existing situation, 9–10, 15–16, 19–21
experimental group, 126, 207, 245, 265–268
experiment, 128, 194, 207, 264–265
explanatory study, 76
exploratory study, 76, 269, 272, 276

facets of design, 2, 5, 90, 143, 183
factor, 19–21, 23
key factor, 21, 25, 37, 149, 184–186
measurable success factor, 27–29, 58, 90, 185–186
success factor, 26–28, 47, 57–59, 90, 185–186
focus groups, 272
formal system development model, 285
generalisation, 36, 83, 126–128, 193–194, 224, 246, 250, 251
going native, 123, 259
grounded Theory, 78–79
Hawthorne effect, 111, 127
holistic fallacy, 123
hypothesis, 60, 71, 92–94, 97–101, 107, 110–111, 206
Impact Model, 20, 24–26, 28–29, 35, 37, 141, 184
Actual Impact Model, 35, 147, 149, 171–172, 185–186, 199–200
Initial Impact Model, 20, 30, 43, 50, 86, 132
Intended Impact Model, 35, 147
impact theory, 192, 206
indicators, 27, 205
industrial setting, 105, 254–256
inferences, 121–123, 274
causal, 123, 126, 266–267
descriptive, 125
influencing factor, 19–21
initial study, 18, 60
Intended Support Description, 35, 142, 166
Intended Support Documentation, 166, 198
interaction diagram, 162–163
inter-encoder reliability, 120–121
interviewing, 271–273
Introduction Plan, 159–161, 212

Actual Introduction Plan, 35, 171–172, 177, 198–199, 203
Intended Introduction Plan, 35, 147, 160, 166, 198–199
introspection, 262

knowledge engineering, 291–293
knowledge issues, 165–166
laboratory setting, 105, 128, 254–256
links between factors, 22–26
literature review, 18, 52, 81–84, 86, 148–149, 217, 222

main functions, 153–155
Measurable Success Criteria, ii, 26–29, 57–59, 185–186, 203, 206, 234
Measurable Success Factors, 27–29, 57–58, 203
method selection, 102ff, 206–209
models, 9, 10, 20, 24, 280–281
analogical model, 281
Boehm’s model, 285
iconic model, 280
mathematical model, 281
structure model, 280
V-model, 285
waterfall model, 284–285, 286–290

network of influencing factors, 19ff
non-equivalent groups, 266, 267
non-experiments, 267–268, 269
notes, types of, 261–262
objectives
book, 10–11, 231–232
design research, 5, 9
DRM, 14, 231–232
DS-I stage, 32, 75
DS-II stage, 37–38, 181
PS stage, 35, 141
RC stage, 30, 43
observation, 77, 103–105, 115–116, 257–262
observational research, 245
observing participant, 259–260
participant observation, 258–259
pure observation, 258
operationalisation, 58, 112, 224

paradigms, 77, 122, 192, 208, 239–242
development paradigms, 175
performance measures, 28, 191
pilot study, 102, 112, 114, 176
plagiarism, 223
post-defined categorisation, 117
post-test, 265–267
power, 276
pre-defined categorisation, 117
prescriptive Study, see PS stage
presentation approach, 219
pre-test, 126, 265–267
product development, see Design
proof-of-concept, 145, 167, 170, 194
protocol analysis, 123, 263
prototyping approach, 145, 176, 193, 285–286
proxy, 15, 27, 57, 96, 185
proximal outcomes, 185, 206
PS stage, 16, 33–35, 141ff
Comprehensive PS, 18–19, 60–63, 143–144, 146–147
deliverables, 35, 141–142, 166, 172
Initial PS, 18, 60–63, 143–144, 146–147
objectives, 35, 141
process, 144–148
Review-based PS, 19, 143–144
quantitative, 21, 26, 79–80, 103, 117, 120–123, 191, 242, 275
quasi-experiments, 241, 265–266
question and answer approach, 220–221
question and hypothesis analysis, 97–99
question-method matrix analysis, 101, 106–109
questionnaire, 269–273
quick read, 52, 82
RC stage, 15, 18, 20, 26, 29–31, 43ff, 217
deliverables, 30–31, 43
objectives, 30, 43
process, 44–45
real-time methods, 104–105
Reference Model, 20–24, 32, 75, 89–90, 124, 132, 149, 183, 199
Initial Reference Model, 20, 30–31, 34, 52, 129
reference list, 226, 383
reflection questions, 69–70, 167, 209
reliability, 103, 124
requirements, 148–150, 152, 175, 277
definition, 286–287
specification, 286–287
research
choosing area of, 51
comparative, 84, 98, 187, 190, 207, 245
focus, 67, 76, 80–81, 89ff, 217
interventional, 84, 245
Overall Research Plan, 31, 44, 67–70, 75
Research Clarification, see RC stage
types of, iii, 18–19, 60–63, 305–306
retrospective methods, 104–105
reuse-based development, 286
review-based study, 18, 60–62
reviewing empirical studies, 83–86
checklist, 84–85, 243–253
rival explanations, 123–124, 242
sample size, 32, 265, 275–276
schools of thought, 76–80, 240
scales of measurement, 118–119
interval, 119
nominal, 118
ordinal, 118
ratio, 119
simulation methods, 280–281
simultaneous verbalisation, 104, 262–264
Soft Systems Methodology, 40–41, 193–194
software development
approaches, 281ff
function-oriented paradigms, 283
generic methodologies, 284
generic technologies, 290–294
object-oriented paradigms, 284
source of statement, 23
spurious relationship, 92, 122–123
statement representation, 22–24
statistical analysis, 79, 93, 115, 121–122, 125, 128, 135, 225, 265, 274–276
descriptive, 121, 274
inductive, 121, 274
inferential, 274
non-parametric, 275
significance, 129, 250
success, 5, 13
Success Criteria, 26–28, 57–59, 185
Success Factors, 26–29, 31, 47, 57–60, 82, 92
support, 4, 5, 9–10, 16, 24–25, 33–38, 141ff, 181ff
Actual Support, see Actual Support
conceptualisation, 147, 153ff, 200
development, 61–62, 75, 141ff
elaboration, 147, 164ff
functionality, 153–154, 167–168
Intended Support, see Intended Support
realisation, 147, 167ff, 200
Support Documentation, 166, 172, 198, 225
task clarification, 146–147, 148ff, 277
survey, 52, 227, 270
synthesis methods, 279

Table of Content approach, 219
task, 37, 85, 249–250
artificial, 249
real, 249
realistic, 249
task clarification methods, 278
taxonomy, 77
testing, 289
acceptance, 290
alpha, 290
beta, 290
theoretical systems, 78
theory, 10, 20, 77–79, 95, 240
design, 3–4
impact, 192, 206
process, 192–193
program, 192–193
theory versus observation, 77
tool development, 162, 168, 173–176, 282ff
triangulation, 103, 108, 116

validation, 5, 78, 124, 175, 182, 289
validity, 8, 9, 103, 123–124, 125, 128, 183
construct, 125, 126–127
data, 115–116
external, 125, 127–128
internal, 125–126
statistical conclusion, 125
variables, 91–94, 96, 113, 125, 265–267, 274–276
control, 92
criterion, 91
dependent, 91–92, 265–267, 274–276
explanatory, 91
independent, 91–92, 265–267, 274–276
predictor, 91
spurious, 122–123
verifiation, 78, 124, 176, 211, 286, 289
visualisation exercise, 68–69