



HAL
open science

Coping with Design Complexity: A Conceptual Framework for Design Alternatives and Variants

Judy Bowen, Anke Dittmar

► **To cite this version:**

Judy Bowen, Anke Dittmar. Coping with Design Complexity: A Conceptual Framework for Design Alternatives and Variants. 16th IFIP Conference on Human-Computer Interaction (INTERACT), Sep 2017, Bombay, India. pp.483-502, 10.1007/978-3-319-67744-6_30 . hal-01676163

HAL Id: hal-01676163

<https://inria.hal.science/hal-01676163v1>

Submitted on 5 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Coping with Design Complexity: A Conceptual Framework for Design Alternatives and Variants

Judy Bowen¹ and Anke Dittmar²

¹The University of Waikato, Hamilton, New Zealand

jbowen@waikato.ac.nz

²University of Rostock, Rostock, Germany

anke.dittmar@uni-rostock.de

Abstract. Interaction design processes are characterised by multi-disciplinary teamwork and by an interplay of creative, situated and analytical thinking. Although design in the domain of human-computer interaction has been widely investigated, the focus of research has been mainly on the user's role and several authors refer to the need for a deeper understanding of the increasingly complex interaction design processes. This paper suggests a conceptual framework for interaction design that accommodates and unifies different perspectives from general design research while considering the specificities of the domain. Within the framework, description and analysis is done through the lens of design spaces, design artefacts, and refinement relationships between design artefacts. The framework extends existing concepts of design spaces by introducing complex spaces which acknowledge that design is rarely an individual activity but is more often undertaken by teams of designers. The framework also offers a distinction between design options into alternatives and variants to better describe and guide processes of idea generation and a convergence within, and between different sub-spaces and sub-teams. Different types of refinement between design artefacts are also discussed.

1 Introduction

Digital interactive systems have become more pervasive in work and everyday life. They now play an important role in how people interact with each other and the world. As a consequence, interaction designers are faced with an increasing design complexity [35]. They need to know the needs of people and the possibilities offered by technologies in order to explore and design technological solutions that fit in with users, the activities they want to undertake and the contexts surrounding those activities [3]. Design in the domain of human-computer interaction (HCI) has been widely investigated since the beginning of the field, but research has tended to focus on the user's role in the design process and the effects of designed artefacts on users [39], [43]. This is also reflected in practical design approaches such as user-centred design stating that iterative development and an early focus on the users' tasks and goals drive quality design, or participatory design emphasising the active participation of all stakeholders. While this research is very valuable, and even more so because less

considered in other design fields, the focus is too narrow [39]. HCI research “has not been grounded in and guided by a sufficient understanding and acceptance of the nature of design practice” [35]. (Similarly to [35], the terms HCI research and interaction design research are used interchangeably in the context of this paper.) Stolterman and other authors refer to existing design theories and empirical work (e.g., from cognitive design research) that could be fruitful for HCI research to better inform and improve interaction design practices [35], [39], [43].

In this paper we propose a conceptual framework aiming at supporting a deeper understanding of interaction design processes. It accommodates and unifies different perspectives from general design research while considering the specificities of interaction design. Within the framework, description and analysis is done through the lens of *design spaces*, *design artefacts*, and *refinement relationships* between design artefacts. The concept of design space is widely used in the literature but with different understandings. In the context of this paper, we take a similar stance to Westerlund [40] and consider a design space as a space ‘populated’ by design artefacts. The concept describes design processes as goal-oriented but situated processes of constructing and relating design artefacts. Here, the term ‘goal-oriented’ is meant to be inclusive and can be interpreted from different design perspectives (e.g., from a value-driven perspective). The starting point in our approach is the assumption that there is a ‘contract’ between the designer and the user which basically says that however intangible the design process might be there always emerges at least a minimum set of requirements which must be satisfied by the final design. In our framework, every external design representation that is created for an intended use, or becomes meaningful in the design process, is considered to be a design artefact (e.g., design prompts, sketches, prototypes, scenarios, formal specifications or implemented products). Designers when ‘entering’ a design space are provided with some initial design artefacts which can be interpreted as requirements or design constraints. Their subsequent exploration of emerging design ideas and constraints leaves traces in the design space: new design artefacts are created, some artefacts are modified, others discarded. Even if the designers’ moves within the design space may appear arbitrary their ultimate goal is to fulfil their contracts with users by finally creating design artefacts that implement or refine the initially provided ones. What is therefore equally important to idea generation is the designers’ ability to compare different design artefacts and understand how they are related and whether or not they satisfy some initial or evolving design specifications and constraints. This paper introduces and illustrates different types of refinement that are relevant in interaction design processes. The suggested classification is based on an analysis of existing design paradigms and perspectives from literature. In the simplest case, if design is considered as problem solving, design artefacts describe the solution (i.e., the interactive device to be developed) at different levels of abstraction in a process of stepwise refinement. More complex understandings of design situations produce more diverse design artefacts and refinement relationships including descriptions of the design process itself.

Interaction design typically takes place in multi-disciplinary teams with co-design phases and phases of distributed work [1]. On the one hand, it facilitates the application of multiple design perspectives, on the other hand, it adds to design complexity

due to additional coordination efforts, different working practices, distributed decision making etc. However, in most existing design space concepts, collaborative design activities are insufficiently accommodated. To address this problem, the presented framework extends the common ideas with a more elaborated description of the structure of design spaces. *Complex design spaces* are introduced which are hierarchically decomposed into sub-spaces until the level of simple design spaces (as described above). Participants in complex design spaces are neither exclusively users nor exclusively designers but are rather engaged in a *network of designer-user relationships* by using design artefacts provided by other participants or sub-teams and by designing design artefacts for others.

Design space models such as Laseau's overlapping funnels (see subsection 2.3) illustrate a common view of design as the generation and the convergence of design ideas. Designers need to bring creativity to the creation of distinct design options and the definition of criteria to choose between those options [7]. There is a criticism that decision making, in this context, is mostly understood as a process of selecting one option and that this attitude may impede a diversity of design ideas [40]. A specific characteristic of our proposed framework is the distinction between *alternatives* and *variants* as two different types of design options that 'leave' the sub-spaces of a complex design space. Basically, if designers in a sub-space resolve all discussion points or disagreements an *alternative* is selected among generated options and provided to other sub-teams. However, if designers of a sub-space want to (partly) delay decision making to include viewpoints and expertise from other sub-teams they provide a set of options (i.e. *variants*) as outcomes. The distinction between alternatives and variants may contribute to a more balanced view of design complexity.

The paper starts with a detailed review of the different existing perspectives on design. The specificities of interaction design and corresponding notions of design space are discussed. We also review results from formal software design which informed the development of our framework. Based on the given background, section 3 introduces and explains the basic concepts of the suggested framework. Then, section 4 considers its application by discussing an illustrative design situation. Furthermore, some results of an exploratory empirical study are briefly discussed. The paper closes with a discussion along with future work and conclusions (section 5).

2 Background and Related Work

Design activities are unique human activities of inquiry and action [35]. Stolterman additionally emphasises that design deals with the specific: "[i]t is about creating something in the world with a *specific* purpose, for a *specific* situation, for a *specific* client and user... and done within a *limited time* and with *limited resources*" [35]. The intended changes are often characterised as changes that are desired [35] or that improve the current world [11]. Interaction design in particular is "the specification of digital behaviours in response to human or machine stimuli" [16]. In a larger sense, interaction design is the creation of spaces enabling complex webs of interactions between people and multiple interactive devices [41]. It addresses the question of

which actions and experiences should be supported by a particular interaction space and how to achieve it. Jackson [20] points out that the complexity of interactions is a general theme in many design disciplines but especially when it comes to the design of software systems. In [35], a recognition and acceptance of both the complexity of the artefact under design and the complexity of the design situation itself is demanded. This section reviews conceptualisations and perspectives on (interaction) design and design complexity. It discusses the different but overlapping understandings of design activities, relevant design artefacts and design spaces.

2.1 Paradigms in Cognitive Design Research

From a cognitive perspective, design is commonly understood as a satisficing activity aimed at finding “good enough” solutions to “ill-structured” problems [39]. Existing paradigms differ in their assumptions about design problems and their treatment [13].

The Rational Problem Solving Paradigm. In the classical view of design that goes back to Herbert Simon in the late 1960’s, design problems are assumed to be given and design is seen as rational search in a ‘problem space’ [13]. Even if problems are ill-structured they can be transformed into structured ones that can be tackled by decomposition [39]. This view is to be found, for example, in traditional software design methodologies with stepwise refinement. Formal refinement (which we discuss further in section 3) provides structured mechanisms for transforming specifications and models (formal design artefacts) into implementations. Typically this is done via a number of small transformations (or steps) which each move closer to a final solution - hence, stepwise refinement. Wirth [42] recommends for software design “to decompose decisions as much as possible, to untangle aspects which are only seemingly interdependent, and to defer those decisions which concern details of representation as long as possible.” Such simplification of the nature of design problems and corresponding overestimation of systematic problem decomposition have been criticised later. However, it is worth noting that Wirth, in his paper from 1971(!), already mentions ideas of design rationale: “[e]very refinement step implies some design decisions. It is important that these decisions be made explicit, and that the programmer be aware of the underlying criteria and of the existence of alternative solutions... [this] may be particularly helpful in the situation of changing purpose and environment to which a program may sometime have to be adapted” [42].

Design as Argumentation Process. The distinction between wicked (ill-defined) and tamed (well-defined) problems in [32] is one of the first attempts to overcome the limitations of the problem solving paradigm. According to Rittel and Webber, most design problems are wicked problems which cannot be defined independently from their solution. Among other characteristics, wicked problems are unique and every implemented solution has consequences that have to be taken into further consideration. Rittel and Webber [32] state that “part of the art of dealing with wicked problems is the art of not knowing too early what type of solution to apply” and suggest instead a collaborative argumentative process of considering and negotiating emerg-

ing issues and possible solutions. Design rationale approaches which explain and record why an artifact is designed the way it is are based on argumentation and can be classified into two broad categories [23]. Structure-oriented approaches and corresponding notations such as Design Space Analysis with the QOC notation (*Question, Options, Criteria*) [25] help to identify relevant design issues and to explore and assess alternative solutions. They became popular in HCI research in the 1990's [27] and more recently in other areas of software design [36], [37]. Psychological design rationale approaches are more holistic and follow task-artifact cycles. First, tasks are identified that should be supported by the system under design and scenarios are created for a collaborative exploration of possible consequences of using it. After its introduction, the system's actual use is studied and compared with the designers' assumptions. Observed negative effects are addressed in a next iteration of the design.

The Reflective Practice Paradigm goes back to the work of Schön [34] who considers design situations as 'messy' situations in which designers find themselves and which they cannot tackle by applying predefined methods. Instead the designer must be in a reflective conversation with the design material of the specific situation. Design is understood as problem setting or framing and Schön describes design exploration as 'moves' within a problem frame where the designer uses 'reflection-in-action' (move, observe, re-frame) as an intuitive process and 'reflection-on-action' as a tool to develop a repertoire of design experience. In current research, the designers' behaviour is commonly described as solution-led: designers jump to ideas for solutions before they have fully analysed the problem and they transfer the developed partial solution structures back into the problem space to extend the problem description and to consider implications of alternative solutions [8]. Cross points out in this context that "both generating few alternative concepts and generating a large number of alternatives were equally weak strategies, leading to poor design solutions" [8].

Designing as Construction of Representations. The above mentioned approaches are not necessarily contradictory but rather focus on different aspects of design activities. Dorst [13] notes, for example, that the designer's expertise influences their perception of the nature of a design problem. While the rule-following behaviour of novice designers must be described following the problem solving paradigm, the behaviour of competent designers, their involvement in and reflection on design situations need an additional explanation within the reflective practice paradigm [13]. Besides that designers are typically faced with both routine (tamed) and nonroutine (wicked) problems requiring either the application of well-known procedures or more advanced approaches [39]. Jackson [20] argues similarly that, in any design task, there must be a combination of 'normal' (routine) design (with well-known requirements and corresponding design experience) and 'radical' (nonroutine) design (with no presumption of success). Studies about co-design activities of teams aimed at a shared understanding of the design problem and possible solutions often use an argumentative approach for their analysis. For example, the QOC approach [25] has been applied in [30] to analyse the discussions of software designers. Viewpoints have been studied in [10] and it has been shown that, during a multi-disciplinary meeting, the participants ex-

press different viewpoints which further evolve through an argumentation process (including arguments by comparison, analogy, and authority) until integrated viewpoints are constructed and shared by the participants. Viewpoints, in this context, are representations of certain combinations of design constraints.

Common to all design approaches is their recognition of the role of design representations although with differences in what should be represented and for what purpose. According to Visser [39], design is about generating, transforming, and evaluating representations “until they are so concrete, detailed, and precise that the resulting representation... specify explicitly and completely the implementation of the artefact [under design]”. The author suggests, therefore, to consider design activities as domain-specific construction of representations and to pay attention to both: the created design representations and the corresponding construction process. In our framework, we follow a similar approach and focus on external representations in interaction design and how designers relate them to arrive at the digital interactive artefact.

2.2 Specificity of Interaction Design

Cross [9] describes how in earlier times the making of artefacts was not separated from the design process. A potter, for instance, used no distinct external design representations and worked directly with the clay to make a pot. In the design of interactive software systems, although we do have a variety of design representations (sketches, scenarios etc.) there is not always a clear distinction between such representations and the end-products, as ideations may be extended into implementations or prototypes may evolve to the final system. Even models of users or context-of-use models may be incorporated into the systems. Therefore, the intermingled character of design and surrounding activities that generally exists [39] is intensified further. Later in our framework, we consider every external representation (including descriptions of initial ideas and requirements up to and including the final implementation) that is created for intended use, or becomes meaningful a *design artefact* (*design*, in short) and do not distinguish between requirements analysis, design and implementation activities.

Role of Users in the Design Process. HCI research was dominated for a long time by considering and rethinking the users’ role in the design process. Three approaches are shortly discussed here: user-centred design, participatory design and meta-design. User-centred design [17] requires from the design team an early focus on the goals, tasks and needs of the users, on the work domain, and on the specific context of use. Participatory design approaches emphasise that the introduction of new interactive artifacts transforms work or everyday life. Conflicts are therefore inherent to interaction design processes and must be resolved by the active participation of all stakeholders [4]. User-centred and participatory approaches have been criticised as being engineering approaches to design with a limited understanding and support of creative design practices (see the above subsection) [14], [43], [35], [16]. However, their contributions to improve the designer-user relationship and to increase the understanding that interaction design has to be embedded in a deliberate transformation of the users’ practices are invaluable. An interesting related approach is the idea of meta-design

introduced in the context of end-user development by Fischer et al. [15]. The authors question that designers should aim at developing complete systems (a goal in conventional design) because user needs and usage situations are never fully predictable. Instead, design is considered to be an open and continuous process with the designers acting as meta-designers who apply a technique called under-design to provide design spaces for the end users (seeding stage) allowing them to act as co-designers by appropriating the system to their specific context of use (stage of evolutionary growth) and by sharing with the designers (re-seeding stage). In our framework, designer and user are not considered to be identities but roles. Participants in the design process typically act in both roles. This view is informed by the above approaches, but especially inspired by Morgan's ideas [28] presented in the following paragraph.

Separation from and Integration with Software Engineering. Winograd [41] predicted the (partial) detachment of the field of interaction design from mainstream computer science because of the foreign methods, skills, and techniques that were required for understanding people and designing spaces for human communication and interaction. In contrast, Diaper [11] suggests that the historical division between HCI and software engineering is unfortunate "because both are engineering disciplines concerned with the same types of systems and their difference is merely one of emphasis, with software engineering focusing more on software and HCI more on people". Above we have seen arguments against a purely engineering approach to interaction design, but nonetheless software engineers and interaction designers collaborate in multi-disciplinary design processes characterised by phases of distributed work where each designer or sub-team has their own sub-task to perform and by co-design phases where participants share goals and contribute to their achievement by applying different perspectives [1]. Co-design is necessary, for example, if usability concerns have to be considered early in the software architecture [21]. Bellotti et al. [2] argue that for an effective collaboration, a revision of each others' assumptions can be necessary. As an example, the authors refer to the conventional notion in the software engineering community that "formal methods are only useful if used within a structured development context from the beginning of a project, through refinement, to implementation". However, a strength of formal approaches may be their suitability for unifying ideas. Robin Milner describes in his Turing award lecture [26] the striving for unifying frameworks at the example of concurrent computation. "I reject the idea that there can be a unique conceptual model, or one preferred formalism, for all aspects of something as large as concurrent computation... we need many *levels of explanation*: many different languages, calculi, and theories for the different specialisms... But there is a complementary *claim* to make, and it is this: Computer scientists, as all scientists, seek a common framework in which to link and to organise many levels of explanation." Our framework is influenced by Morgan's uniform approach to refinement in software design [28]. He suggests banishing the distinction between specifications, sub-specifications, and computer programs and considering all of them as programs. Programs are contracts which have to be negotiated between clients and programmers. They describe what one person wants (the client role) and what another person or computer must do (the programmer role). A hierarchical refinement of pro-

grams is assumed (starting with high-level specifications until programs, executable on the computer) which is closely associated with the problem solving perspective.

External Design Representations According to [39], the ultimate design representation must express three aspects of the artefact under design: the what (the artefact itself), the how (the process of implementation), and the why (the design rationale). Design representations in interaction design can support what is called by Diaper [11] the narrow view of HCI focusing on the user-computer interface or the broad view concerning “with everything to do with people and computers” including real-world consequences. Typical forms of representing the what, according to the narrow view, are sketches, prototypes, and models of the user interface, but also functional models of the digital interactive artefact. QOC diagrams [25], claims as known from scenario-based design [33], and task models as used in [31] reveal some of the why, process models such as the evaluation-centred star life cycle model [19] some of the how. Problem and interaction scenarios [33], current and envisioned task models as recommended in [11] or user models such as personas [18] are representations supporting the deliberate transformation of the users’ (working) practices, and hence, the broader view of HCI design. Flexible design processes need to be supported by a co-evolution of the various design representations. Although there are approaches to relate different types of representations such as user interface sketches and formal specifications [5], task models and QOC-diagrams [22], or prototyping and argumentation [12], the effective coupling of different external representations is still poorly understood in interaction design. What we especially consider in our framework is the designers’ ability to compare representations and understand how they are related and whether or not they satisfy some initial or evolving requirements and constraints.

2.3 Design Spaces

The concept of design space is central to the suggested framework. Before introducing the framework in the next section, we briefly discuss existing conceptions of design spaces to position our view. In engineering contexts, a design space is often understood as being defined along a set of (possibly orthogonal) dimensions. For example, Nigay and Coutaz [29] suggest a design space for multi-modal systems in terms of level of abstraction, use of modalities and fusion. Design spaces, in this sense, support a view of designing as problem solving. They are generic tools providing a common vocabulary for classifying and comparing system designs (determined by certain values for the dimensions) which guide the designers in choosing an optimal solution for their design problems. Some proponents of design rationale understand design spaces both as a conceptual tool guiding argumentation processes and as “an explicit representation of alternative design options, and the reasons for choosing among those options” [24] which emerged in a particular argumentation process. They suggest that the result of a design process should be conceived as a design space rather than a single specification or product. While corresponding representations such as QOC-diagrams [24] depict how design options and criteria are related to each other, design space models such as Laseau’s funnel model with its variants (Fig. 1)

emphasise a balance between the designer’s creation of distinct design options (concept generation in elaboration funnel) and decision making (concept convergence in the reduction funnel). The chosen design solution is represented by the focal point.

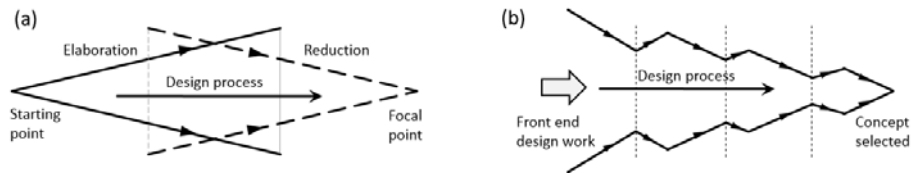


Fig. 1. The designer’s moves in a design space: (a) Laseau’s overlapping funnels, and (b) a refined version assuming some front-end work resulting in a product design specification and an alternation between concept generation and concept convergence step-wise leading to finer levels of granularity in the design (discussed in [7]).

Westerlund [40] considers a design space as the set of “all possible design proposals that would be regarded as meaningful to use by some people in relevant contexts”. He criticises the models in Fig. 1 for their assumption that the initial brief, assignment or problem will be stable during the process and for their focus on one goal and one final solution, which may impede a diversity of design ideas. In his view, proposals that work lie within the design space, proposals that do not work are outside the space [40]. This is in line with Binder et al. [38] who, from a creative design perspective, describe the emergence of a design space out of a collaborative process of creating and manipulating a variety of design representations or artefacts. Transforming representations and shifting between different material highlight different aspects of design and widen the design space. We follow the last mentioned authors and consider the design space concept as a tool for designing and understanding design processes, a tool for the reflective practitioner supporting a less prescriptive approach to design. However, to our knowledge, our approach of complex design spaces and the distinction between alternatives and variants is novel.

3 Basic Concepts of the Framework

3.1 Design Spaces and Design Artefacts

As our starting point we determine the existence of a design space as an essential entity within the design process. Even if it is not explicitly defined or understood by the design teams, the design space is the conceptual gathering together of all, and any, artefacts used within the process. Recall that we consider ‘design artefacts’ as any materialised form of design concepts (or ideas, constraints, discussions etc.) that form part of the design activities. At the most basic level we can imagine the solution to a small and straightforward design problem is found by the designer exploring several ideas in a linear fashion, before finalising and selecting one which satisfies the problem description. Of course, in interactive system design we are typically interested in much larger design problems and so extend this concept to the base case (one design

team) as in Fig. 2(a) which is then part of the recursive definition of the complex design space in Fig. 2(b).

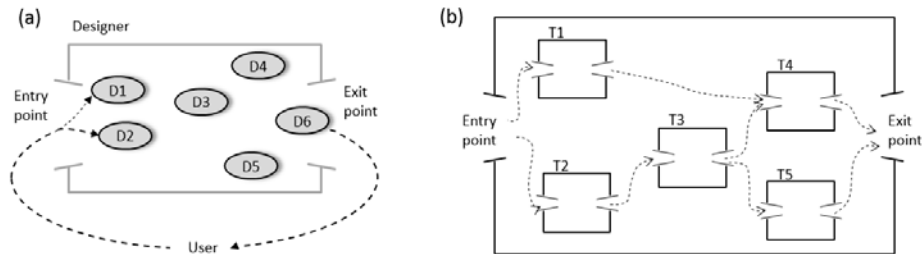


Fig. 2. (a) Simple design space, (b) complex design space with sub-spaces.

Within the simple design space each of the ellipses D1..D6 in Fig. 2(a) represents some design artefact. These may all be of different types and represent all and any considerations currently taking place within this part of the design process by any member of the team (individually or as a group). The entry and exit points shown represent the underlying user-designer relationship (where designers from other parts of the process may also be considered as users). The entry point indicates any initial designs or requirements provided to the designer, who in turn will provide designs at the exit point which satisfy some, or all of those requirements. The various artefacts (D1..D6) within the space may be considered as *alternatives* if one is chosen to leave at the exit point, or *variants* if a choice is not made and more than one leaves at the exit point as part of delayed decision-making. We discuss alternatives and variants in more detail later. We do not suppose that the design happens in a linear fashion from left to right within the space, but there are relationships between the different design artefacts, so designers may bounce around ideas and try out different things that are subsequently discarded, or use more formal techniques to make specific decisions around particular parts of the system design.

Given the multi-disciplinary approach typically taken within design, the actual design space is not simple, but is complex, as depicted in Fig. 2(b). Here each of T1..T5 are sub-spaces, that is they are design spaces (and perhaps contain further sub-spaces), such that ultimately all of the different design processes from each of the groups and individuals involved in the process can be captured inside a single high-level design space. These are, of course, abstract representations (hence the term high-level), we can 'zoom in' on any one of the design artefacts to understand what it represents, and then how the different artefacts are related. Similar to the relationships between design artefacts, we see in the right hand picture of Fig. 2 of the complex design space the relationships between individual sub-spaces. These may be one-one, one-many, many-one etc. and uni- or multi-directional, e.g. some provide inputs only to other design spaces while others involve a 'negotiation'. We discuss these different types of relationship further and some potential underlying causes in section 4.

These design spaces define the basic structure of our framework which then captures the ideas of multi-disciplinary teams working iteratively, both independently and together, towards a solution guided by an evolving understanding of the design prob-

lem and constraints (based upon an emerging set of requirements). Participants in the design process may not be aware of all of the design spaces but rather focus only in the area they are working. Hence the requirement to ensure that there is overall a consistency in the end-goal of all of the design spaces such that there are not additional conflicts introduced by incompatible decisions being made in different spaces.

3.2 Refinement

There is an understanding that design artefacts leaving a design space are, in some sense, more refined than those at the entry point. This implies that some progress has been made (design being a goal-directed activity) in at least a part of the design. Refinement is a central concept in more formal software development processes where it represents a structured transformation from a formal model towards an implementation in a way which guarantees certain properties of the formal model are preserved. This understanding of refinement supports the problem solving perspective on design (see subsection 2.1) by assuming that the formal specification describes the right system to be built (the design problem is fully understood) and the refinement relationship ensures that the system is built in the right way. As we have shown above, interaction designers are mostly faced with ‘wicked’ design problems requiring a co-evolution of problem and solution. Hence, when we consider refinement in the context of interaction design we do not have the same concept of using a transformation calculus on a formal model of the interactive artefact to be developed. Rather we have to additionally consider intermediate design artefacts describing design rationales to support designers in understanding situations of use of that artefact as well as design artefacts describing the design situation itself to support designers in creating the right (complex) design space. Accordingly, we suggest four types of refinement.

1. Refinements based on formal methods to ensure to build the system right.
2. Refinements based on lightweight notions to ensure a transition between informal and formal designs.
3. Refinements based on validation techniques to ensure to build the right system.
4. Refinements based on reflection to ensure that the design process is right.

Our refinement approaches are framed in the idea of ‘contractual utility’ as in [5]. At its simplest, contractual utility implies that if our customer is satisfied with a system, S , we can replace it with system S' if it meets all of the criteria agreed upon (i.e. satisfies the contract) for S . This is often simply stated as “We can replace S with S' provided the customer can do all of the things they could do before (and perhaps more)”. Note that in this notion of refinement the requirement to preserve properties (which may be the satisfaction of requirements or adherence to design decisions already made) remains. As such the entry and exit points in each of the design spaces represent a refinement relationship where artefacts at the exit point retain properties from those at the entry point but may also have additional properties (based on new decisions made) or the removal of variants which have come from other sub-spaces. Formally we consider that we can weaken pre-conditions and strengthen post-conditions as a legitimate refinement process, and that this results in a larger range of application

situations of the artefact under design (weakening pre-conditions) and in a strengthening of expected desired effects and/or mitigating of expected undesired effects of using the artefact (strengthening post-conditions). We will further discuss the different forms of refinement in the next section by using an illustrative example.

3.3 Alternatives and Variants

We discussed briefly above the difference between alternatives and variants. These terms are frequently used interchangeably or without clear definition in the literature. One of our contributions here is to give such a definition which can then be used unambiguously in both our framework and in subsequent discussions. Both terms represent design artefacts of a design space which are related, in that they refine the artefacts in the entry point of that space, but which contain some differing options. We call *Alternatives* those artefacts where a choice is made which determines that one is selected over the other, so within a design space if there are several alternatives only one will be selected to leave the exit point. We call *Variants* those artefacts where decision-making is delayed or postponed for subsequent members of the design team to make. In other words, a single solution at the exit point (alternative) represents a closed process of generating possible solutions and choosing a good one (reduction of design complexity) while variants stand for a somewhat open decision process (keeping or increasing design complexity). In the latter case, the designer provides options to the user which share some common elements, but not all, to satisfy the requirements. However, the designer is aware that it is beyond their current competency and knowledge to make a selection or that a selection would unnecessarily limit the user's activities, including their creativity. This awareness is especially important in multi-disciplinary work. So design variants may proceed through the entire set of design spaces and may even end up as choices in the final system that the end-user can decide upon (as a form of personalisation or customisation).

4 Application of Framework

In this section, we first illustrate the application of the framework by discussing a small example design situation. Then, some results of an exploratory empirical study are shortly presented that support the subsequent discussion of the framework.

4.1 Illustrative Example

The example design situation is completely fictive but loosely based on the classic Bomberman game, a strategic, maze-based computer game, in which the players have to place bombs to kill enemies and destroy walls. The original game was published in 1983 and new games have been published ever since (Wikipedia). We identify some design artefacts and refinements of the example along with a discussion of the different types of the refinement relationships.

<p>Current scenario S1: It is a Thursday evening and Jack is playing bomberman when his younger brother Thomas enters his room and immediately starts begging to play himself. As so often, Jack gives in. He interrupts his actual game and switches to the two-player mode. He would have preferred to leave Thomas alone but after the recent discussions with their mother he hesitates to do so.</p>	<p>Envisioned scenario S2: It is a Thursday evening and Jack is playing his newest bomberman version when his younger brother Thomas enters his room and immediately starts begging to play himself. As so often, Jack gives in. He interrupts his actual game, but before starting a new instance, he switches to the two-player mode and to the diamonds collecting variant...</p>
--	---

Fig. 3. Scenarios in the example.

Design Spaces and Design Artefacts in the Example. Let us imagine that the example design process was initiated by parents expressing their concerns about seeing their young children playing Bomberman. They asked a professional design team to create a less aggressive version of the game (initial design goal). The professional team decided to start their work by analysing gaming practices of children to get a better understanding of the design problem (sub-team T1) and developing in parallel conceptual design ideas (sub-team T2). Based on interviews and observations, sub-team T1 developed a set of current scenarios (see scenario S1 in Fig. 3 as an example) that supported a revised description of the design goal. They handed over their results to sub-team T3 who had to create a first prototype. Meanwhile sub-team T2 came up with some ideas and assessments but made no final commitment. Instead, they provided a QOC-diagram to sub-team T3 (black text of Fig. 4). T3 realised that the ideas captured in the QOC-diagram do not satisfy the revised design goal and they asked T2 to rethink their ideas. T2 added a new option (O22) to question Q2 which lead to a consequent design question (Q3). They also added a new dimension concerning the design of the field maps (grey text of Fig. 4). Based on this modified diagram sub-team T3 developed a family of prototypes representing all combinations of suggested options (with tiles and objects represented in an abstract way by coloured squares and circles, see left part of Fig. 5). It was then decided to organise a workshop together with parents and children (sub-team T4). The participants (working in sub-groups) reflected on the prototypes by discussing the current scenarios and developing envisioned usage scenarios of the new design (e.g., scenario S2 in Fig. 3). They made suggestions for concrete tiles and objects (e.g., grass and water tiles, pump and life vest). They further required a ‘softer’ way to defeat an opponent than in the original game. Finally, they decided to restrict the set of all possible game variants to a smaller set of predefined game configurations which the end-user can choose between. The advanced prototype shown on the right of Fig. 5 and the supporting envisioned scenarios left the exit point of the design space of sub-team T4. Table 1 gives an overview of the design artefacts provided to and by the sub-teams.

Refinement in the Example. We return now to the considerations of refinement. As mentioned above, we follow an approach framed in the idea of ‘contractual utility’ between the designer and the user: a design (or set of designs) D' refines D (and thus

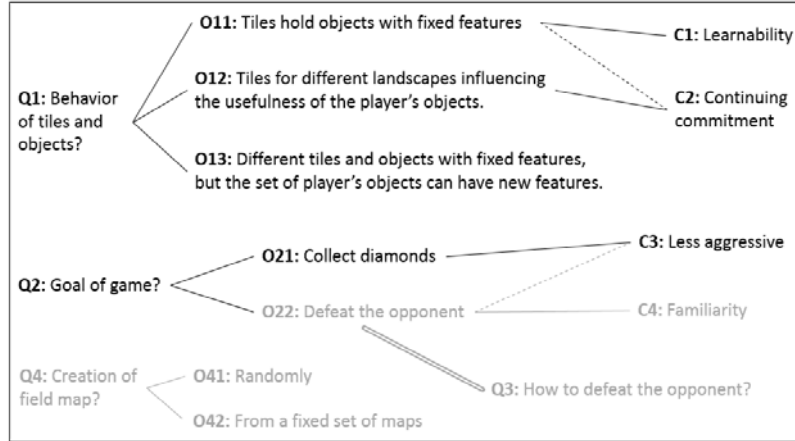


Fig. 4. A QOC-diagram in the example.

Table 1. Design sub-spaces and design artefacts D1..D8 in the example.

Space	Designs in entry point	Designs in exit point
T1	D1: less aggressive Bomberman (initial design goal)	D3: less aggressive Bomberman but still with familiar game concepts, D4: current scenarios supporting D3
T2	D1 (in the second iteration: D3), D2: original game concepts	D5: QOC-diagram (modified in the second generation)
T3	D3, D4, D5	D6: family of prototypes
T4	D4, D5, D6	D7: advanced prototype, D8: envisioned scenarios supporting D7

can replace D) if the user gets at least what they had before (with D) or better. We can break these ideas down further and show how the designs from the example fit within the refinement conditions described in section 3.2. Given two designs, D and D' , D' refines D

- if it preserves all properties of D : Properties here refer to all and any design requirements and criteria that emerge during the design process.
Example: For every game variant suggested in $D5$ (specified by (O11,O21,O41), (O12,O21,O42) etc.) there is a corresponding prototype in $D6$.
- if it preserves all properties of D and has additional properties.
Example: $D3$ preserves the properties of $D1$ (less aggressive game) but additionally the new game is required to have game concepts familiar from the original one.
- if it removes non-determinism present in D : Non-determinism can be represented by an abstract description or by a set of variants provided by D . Consequently, it can be decreased by more concrete designs (preserving properties of the abstract design but adding design decisions) or by a reduction of variants.
Example: $D7$ removes non-determinism existing in $D6$ in at least two ways: first, by deciding about concrete tiles and objects in the game, and second, by reducing the set of variants provided by $D6$ to a smaller set.

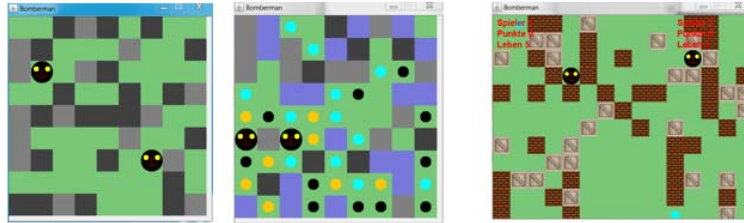


Fig. 5. Prototypes in the example.

Types of Refinement in the Example. Of course without giving formal definitions for the refinements described it may appear that anything can be considered a refinement provided we frame it correctly, but this is not the case. It is, however, also not the case that a direct refinement between the initial artefact and the final design choice exists. We frame the refinement to encompass all of the emerging requirements in addition to the starting point. In fact this is also true of ‘classical’ refinement in formal software development, where requirements (which are assumed to exist prior to the creation of the specification) are all contained within that specification and as such form part of the refinement. One of the differences here is that the evolution of the design artefacts can represent a co-evolution of problem understanding and solution. In other words, requirements may co-evolve and so need to be added to the refinement considerations.

Let us discuss a more concrete example of this in terms of our example which starts with a request to change the Bomberman game so that it is suitable for a younger age-group, and it may be that this leads to decisions that restrict or remove behaviours that were present in the original game. What the refinement relationship does then is enable us to keep track of the effects of design decisions and understand them in the context of the design process. So, in the example the addition of the constraints that will make the game less aggressive mean that the new version will not be a ‘classical’ refinement of the original (e.g., we cannot do everything we could before) but it satisfies the requirements and constraints such that the design process leads to a satisfactory refinement. Using the labels given in Table 1, we have:

- D7 does *not* refine D2
- (D7 and D8) refine (D2 and D4 and D3)

The envisioned scenarios in addition to the description of the redesigned Bomberman game can be considered to be a refinement of the current scenarios, the existing Bomberman game, and the initial and later refined design goal because we preserve something in the envisioned scenario as older and younger children still seem to like to play the game, and we get something additional (desired): that the younger children can play a less aggressive variant with less concerns of some parents. This then is a refinement based on validation (building the right system). Such a refinement also implies that we discard some parts of the existing implementation, in the example some restriction or removal of behaviours of the original game. This reflects the fact that to build the right system must always include negotiations between different viewpoints and compromises, and here explicitly the new scenario and requirements

contradict some behaviours of the original game. We do not assume that classical refinement is abandoned in the development process, but rather that we embed it into our more loose definition of refinement based on validation during the design phases. So while in classical refinement concepts, the emphasis is on “building the right system in the right way”, here our emphasis is on the first part only: “building the right system” (although of course we assume that this will also ultimately be achieved in the right way once the design is complete). Similarly, refinement based on validation has to be embedded into refinement based on reflection upon the actual design activities. For reasons of brevity, this is slightly indicated rather than fully described in the example. But it may be easy to imagine, e.g., that sub-team T4, if only consisting of professional designers and children (or professional designers and parents), could have come to different decisions based on a different ‘building’ of arguments.

Alternatives and Variants in the Example. Alternatives, even if they come with some supporting arguments, hide most of the complexity of design sub-spaces. In the example, team T1 may have discussed various refinements of D1 but only D3 (supported by scenarios D4, see Table 1) leaves the exit point, nothing is known outside T1’s sub-space about those alternative refinements of D1. In contrast, variants keep some of the partly emerging complexity outside a sub-space. Game variants are created at a conceptual level by sub-team T2 (QOC-diagram D5), ‘passed over’ from sub-team T3 to T4 via the family of prototypes (D6), and in a restricted form finally to the end-users such as Jack in the scenario S2 (Fig. 3). Jack not only plays the Bomberman game but also ‘designs’ an appropriate setting for his brother Thomas to play.

4.2 Exploratory Study

The actual types of artefacts included in the design spaces will be particular to a given design situation (characterised by the design problem, the design team and sub-teams etc.) but we expect in our framework that there will be a common set of attributes seen across all design sub-spaces. That is, sub-teams have similar behaviours in discussing the problems, options, decision making and explicitly create and forward design artefacts to other sub-teams as we describe it in the framework. So where some people may use diagrams to explicitly elaborate decisions to be made along with accompanying sketches, others may have design meetings where such decision making takes place as discussion and they create representations of their results afterwards to provide them to other sub-teams. To start investigating the applicability of the suggested framework as a descriptive tool for design processes we conducted exploratory studies of two small design teams. While the focus of our work here is to present our general framework and its uses rather than explicitly discuss the case studies, the knowledge gained from these has enhanced our understanding of the use of such a framework in real-world design processes. Hence, we present a brief overview of one of the studies and discuss some of the results.

The study took place within a locally based web-design company who were tasked with re-designing the web site of a large medical company. The design process took place over a period of 6 months and there were 6 members from the design company

involved, 5 of whom were co-located. The other team member, who was the project manager, was located in the company head office 150km away, close to the client. Communications between co-located team members occurred in face-to-face meetings as well as via email and the use of specific design tools and online meetings were used to communicate with the project manager. The project manager and clients had face-to-face meetings. Although the primary focus of the study was to identify the design artefacts, decision-making processes and ecosystem of the design resulting from this, what also emerged was that there were implicit constraints which were not articulated or recorded but which had a clear effect on the process. For example, all of the design team knew, from previous experiences, that they should only recommend solutions that could be handled by the existing technologies used by their company. This was explicit knowledge within the design team but hidden from the client, as such solutions were only ever suggested that met this criteria and so it had the effect of constraining the choices offered and made. Secondly the organisational culture meant that none of the design team ever disagreed with the project manager, and as such some of the rational decision-making processes were abandoned when it became clear that the preferred solution (of the project manager) would not result from such a process. While we emphasise in our framework the designers' creation of external design artefacts and their refinement, we must be aware of such implicit constraints which may affect idea generation and (distributed) decision-making. This brings us to the discussion of the proposed framework of complex design spaces.

5 Discussion and Conclusions

The framework can help to understand the role of design representations that are used during multi-team, multi-disciplinary design processes. We have discussed the comparison of such design artefacts within the framework and propose that they can be used to keep track of the history of decisions throughout the spaces. We might consider a trace of the movements between such design spaces as a pathway through the design process where ideas which have been discarded, amended or selected can be viewed at the relevant point. Not only does this history allow such an overview, but also means we can ensure that critical decision points have been made in accordance with the requirements and that we have not lost valuable elements of the design.

Alternatives and variants (in combination with refinement) in complex design spaces provide a more relaxed view on the designer's goal-directed activities than approaches such as those depicted in Fig. 1 by allowing the convergence of design ideas within and across design sub-spaces (thus preventing an unnecessary or even undesired reduction of design complexity). Therefore, these concepts support the awareness of expertise and fruitful contributions of different designers or sub-teams in multi-disciplinary design work. The complex design spaces emphasise the importance of understanding local design goals and values within a global context. Again this supports a more cohesive view of the overall design path than individual design spaces and enables a better understanding of why particular decisions have been made (global design rationale) where they seem to contradict previous decisions (local de-

sign rationale). To our knowledge, such a comprehensive, explanatory framework of interaction design activities is novel and brings together fundamental understandings from design research with practical applications in software design activities. In keeping with Stolterman's [35] call for "high-level theoretical... ideas and approaches that expand [interaction] design thinking but do not prescribe design action (reflective practice, human-centred design, experience design, design rationale, etc.)" our framework does not prescribe a new process model for design, but rather exposes a unifying view on the diverse design representations and their refinement and the abolishing of the dualism between designers and users. This allows us to accommodate the different design perspectives we reviewed in section 2.1.

Limitations and Future Work. The refinement concepts we have presented here are grounded in traditional refinement theory, but we have presented them as light-weight concepts without any practical techniques for supporting their identification. In [6] we gave some formal definitions for such refinements and in future work we should consider the application of these within the framework in a suitable lightweight manner - by which we mean lightweight practical techniques that can be used by interaction designers rather than formal methods specialists.

Our primary focus is on external design representations. However it is understood that the interplay between internal and external representations generally needs to be investigated more deeply in design research [39] and we make no further contribution to that here. Also, while we have undertaken some exploratory empirical work more is needed to explore the applicability of the framework as a tool that can guide analysis, description and design of interaction design processes. We should explore how the framework can serve in real-world design as a way of preventing implicit constraints from dominating the design process (or at least identify it is happening).

We have shown in section 2.2 that multi-disciplinary design requires a revision of each others' assumptions and concepts. This paper suggests the 'transfer' of revised concepts from formal software engineering, such as refinement and contractual utility, to other design practices. Of course, at this stage it is not clear whether this transfer will be 'accepted' and how it can contribute to a model that is positioned equally between engineering design and creative design.

Conclusions. The paper presented a framework for considering multi-team, multi-disciplinary design of interactive systems. Our contribution is given by the proposed framework, along with definitions for alternatives and variants and a high-level view of refinement within the framework. The intention being to give a more concrete method of viewing and understanding interaction design (a complex and 'messy' process) in a structured way.

Acknowledgement. We thank Wanying Yang and the participants of course 23149 on interactive systems. We are grateful to the anonymous reviewers for their insightful comments and suggestions.

References

1. Baker, M., Détienne, F., Burkhardt, J.M.: Quality of collaboration in design: Articulating multiple dimensions and viewpoints. In: 1st Interdisciplinary Innovation Conference (2013)
2. Bellotti, V., Shum, S., MacLean, A., Hammond, N.: Multidisciplinary Modelling in HCI Design in Theory and in Practice. In: SIGCHI Conference on Human Factors in Computing Systems. pp. 146–153. CHI'95, ACM (1995)
3. Benyon, D., Turner, P., Turner, S.: Designing interactive systems: people, activities, contexts, technologies. Addison-Wesley (2005)
4. Bødker, S., Grønbaek, K., Kyng, M.: In: Schuler, D., Namioka, A. (eds.) Participatory design: Principles and practices, chap. Cooperative design: Techniques and experiences from the Scandinavian scene, pp. 157–176. Lawrence Erlbaum Associates, Hillsdale, NJ (1997)
5. Bowen, J., Reeves, S.: Formal Refinement of Informal GUI Design Artefacts. In: Australian Software Engineering Conference (ASWEC'06). pp. 221–230. IEEE (2006)
6. Bowen, J., Dittmar, A.: A semi-formal framework for describing interaction design spaces. In: 8th ACM SIGCHI symposium on Engineering interactive computing systems. pp. 229–238. EICS'16, ACM (2016)
7. Buxton, B.: Sketching User Experiences: Getting the design right and the right design. M. Kaufmann (2007)
8. Cross, N.: Design Cognition: Results From Protocol And Other Empirical Studies Of Design Activity. In: Eastman, C., McCracken, M., Newstetter, W. (eds.) Design Knowing and Learning: Cognition in Design Education, pp. 79–103. Elsevier (2001)
9. Cross, N.: Design Thinking: Understanding How Designers Think and Work. Bloomsbury Publ. (2011), <https://books.google.co.nz/books?id=F4SUVT1XCCwC>
10. Détienne, F., Martin, G., Lavigne, E.: Viewpoints in co-design: a field study in concurrent engineering. Design Studies 26(3), 215–241 (2005)
11. Diaper, D.: The handbook of task analysis for human- computer interaction. In: Diaper, D., Stanton, N. (eds.) Understanding Task Analysis for Human-Computer Interaction. L. Erlbaum Associates Inc. (2004)
12. Dittmar, A., Piehler, S.: A Constructive Approach for Design Space Exploration. In: 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems. pp. 49–58. EICS'13, ACM (2013)
13. Dorst, K.: On the Problem of Design Problems - problem solving and design expertise. Journal of Design Research 4(2) (2004)
14. Fallman, D.: Design-oriented Human-computer Interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 225–232. CHI'03, ACM (2003)
15. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N.: Meta-design: A Manifesto for End-user Development. Commun. ACM 47(9), 33–37 (2004)
16. Goodman, E., Stolterman, E., Wakkary, R.: Understanding Interaction Design Practices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1061–1070. CHI'11, ACM (2011)
17. Gould, J.D., Lewis, C.: Designing for Usability: Key Principles and What Designers Think. Commun. ACM 28(3) (1985)
18. Grudin, J., Pruitt, J.: Personas, participatory design and product development: An infrastructure for engagement. In: Proc. of PDC'02. pp. 144–161 (2002)
19. Hix, D., Hartson, H.: Developing User Interfaces: Ensuring Usability through Product and Process. Wiley, New York (1993)
20. Jackson, M.: Representing structure in a software system design. Design Studies 31(6), 545–566 (2010)

21. John, B.E., Bass, L., Sanchez-Segura, M.I., Adams, R.J.: Bringing Usability Concerns to the Design of Software Architecture, pp. 1–19. Springer (2005)
22. Lacaze, X., Palanque, P., Barboni, E., Bastide, R., Navarre, D.: From DREAM to Reality: Specificities of Interactive Systems Development With Respect To Rationale Management. In: Dutoit, A., McCall, R., Mistrik, I., Paech, B. (eds.) *Rationale Management in Software Engineering*, pp. 155–172. Springer (2006)
23. Lee, J., Lai, K.Y.: What's in Design Rationale? *Human-Computer Interaction* 6(3), 251–280 (1991)
24. MacLean, A., Bellotti, V., Shum, S.: Developing the design space with design space analysis. In: Byerley, P.F., Barnard, P.J., May, J. (eds.) *Computers, Communication and Usability: Design issues, research and methods for integrated services*, pp. 197–219. Elsevier (1993)
25. MacLean, A., Young, R., Bellotti, V., Moran, T.: Questions, Options, and Criteria: Elements of design space analysis. *Human-Computer Interaction* 6(3), 201–250 (1991)
26. Milner, R.: Elements of Interaction: Turing Award Lecture. *Commun. ACM* 36(1), 78–89 (1993)
27. Moran, T.P., Carroll, J.M. (eds.): *Design Rationale: Concepts, Techniques, and Use*. L. Erlbaum Associates Inc. (1996)
28. Morgan, C.: *Programming from Specifications* (2nd ed.). Prentice Hall International (UK) Ltd. (1998)
29. Nigay, L., Coutaz, J.: A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In: *INTERACT'93 and CHI'93*. pp. 172–178. ACM (1993)
30. Olson, G.M., Olson, J.S., Carter, M.R., Storrøsten, M.: Small Group Design Meetings: An Analysis of Collaboration. *Human-Computer Interaction* 7(4), 347–374 (1992)
31. Paterno, F.: *Model-Based Design and Evaluation of Interactive Applications*. Springer (2000)
32. Rittel, H., Webber, M.: Dilemmas in a General Theory of Planning. *Policy Sciences* 4, 155–169 (1973)
33. Rosson, M.B., Carroll, J.M.: *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann Publishers Inc. (2002)
34. Schön, D.: *The Reflective Practitioner: How Professionals Think in Action*. Basic Books (1983)
35. Stolterman, E.: The Nature of Design Practice and Implications for Interaction Design Research. *International Journal of Design* 2(1), 55–65 (2008)
36. Tang, A., Aleti, A., Burge, J., van Vliet, H.: What makes software design effective? *Design Studies* 31, 614–640 (2010)
37. Tang, A., Han, J., Vasa, R.: Software Architecture Design Reasoning: A Case for Improved Methodology Support. *IEEE Software* 26(2), 43–49 (2009)
38. Telier, A., Binder, T., De Michelis, G., Ehn, P., Jacucci, G., Wagner, I.: *Design Things*. MIT Press (2011)
39. Visser, W.: Designing as Construction of Representations: A Dynamic Viewpoint in Cognitive Design Research. *Human-Computer Interaction* 21(1), 103–152 (2006)
40. Westerlund, B.: *Design Space Exploration - Co-operative creation of proposals for desired interactions with future artefacts*. Dissertation, Kungliga Tekniska högskolan, Stockholm (2009)
41. Winograd, T.: Beyond Calculation. In: Denning, P.J., Metcalfe, R.M. (eds.) *The Design of Interaction*, pp. 149–161. Copernicus, New York, NY, USA (1997)
42. Wirth, N.: Program Development by Stepwise Refinement. *Commun. ACM* 14(4), 221–227 (1971)

43. Wolf, T.V., Rode, J.A., Sussman, J.B., Kellogg, W.A.: Dispelling "design" as the black art of CHI. In: Proceedings of the SIGCHI conference on Human Factors in computing systems. pp. 521–530. CHI'06, ACM (2006)