

Models in A

45. Models in Architectural Design

Pieter Pauwels

At one time, architects and construction specialists used to rely mainly on sketches and physical models as representations of their own cognitive design models. Today, they rely increasingly on computer models including parametric models, generative models, as-built models, building information models (BIM), and so forth. Of course, processes of abstraction and the actual architectural model-based reasoning itself remain in the mind of the practitioner who is in control of the design and construction process. However, this whole new array of alternative computer-based representation models has profoundly affected decision-making in architectural design and construction. In this chapter, a brief overview is first given of the state-of-the-art in design thinking research. Following this, an outline is given of how diverse data models, such as BIM and parametric models, are currently used in architectural design and construction. An indication is then given of how these models relate to the in-mind model-based reasoning on which architectural designers and construction experts rely in decision-making and creative thinking. This outline will not only review well-known theories of design thinking and architectural design practice, it will also integrate ongoing theoretical research about analogical reasoning and about abductive, deductive, and inductive reasoning.

45.1 Architectural Design Thinking	976
45.1.1 The Architectural Designer as a Practitioner	976
45.1.2 Where Are the Models in all This?	976
45.1.3 Abstraction, Sense-Making, and Framing into Mental Models	978
45.1.4 Accessing Background Knowledge Through Analogical Reasoning?	979
45.1.5 Abstraction from Representation Model to Mental Model	980
45.2 BIM Models and Parametric Models	981
45.2.1 New Technological Media in Design Thinking	981
45.2.2 BIM Models and Parametric Models	982
45.2.3 Features and Issues in the Usage of the New Modeling Applications	982
45.3 Implementing and Using ICT for Design and Construction	984
45.3.1 Pragmatic Usage of Semantic Modeling Applications	984
45.3.2 The Usage of Design Agents or Assistants	985
References	987

The word *model* is ubiquitous in the current practice of architectural design and construction. Whereas architects and construction specialists used to rely mainly on sketches and physical models as representations of their own cognitive design models, now they rely more and more on computer models (or computer representations of their cognitive design models). Parametric models, generative models, as-built models, BIM, and so forth,

are used day in and day out by any architectural design and construction practitioner. Although processes of abstraction and the actual architectural model-based reasoning itself still occur in the mind of the practitioner, who is in control of the design and construction process, of course, this whole new array of alternative computer-based representation models has its impact on decision-making in architectural design and construction.

45.1 Architectural Design Thinking

Understanding how designers think has been the goal of many research initiatives during previous decades. Several relevant overviews are available that describe the evolution of these research initiatives and their outcomes [45.1–3], therefore we will not elaborate here in extensive detail. With the emerging interpretation in the 1970s of the design process as a process in which *wicked problems* [45.4, 5] or *ill-structured problems* [45.6] are to be *re-solved*, over and over again, design is now more and more considered as a practice or a discipline in its own right, rather than a science that can be addressed using a rigid methodological approach (see for example [45.7, p. 11]).

45.1.1 The Architectural Designer as a Practitioner

The basis of this interpretation relies heavily on the theories by *Cross* [45.8], *Lawson* [45.9], *Schön* [45.10], and *Simon* [45.11]. These theories typically acknowledge the complexity of the design process and the role of design thinking within this process. An architectural design situation is not necessarily considered as a design *problem* that is defined by a well-structured set of constraints, and in which a number of adjustable parameters is available. Instead, a design situation in these theories is typically understood as a snap-shot, in terms of time, in the overall design process, in which a limited number of constraints and parameters are taken into account and adjusted by a designer, in order to *satisfice* the design situation, as interpreted at that moment, into an alternative and new design situation. The term *satisficing* refers to the attitude of architectural designers to *sufficiently*, instead of *entirely*, satisfy constraints (see also [45.6] and [45.12, p. 224]). In terms of optimization approaches to design, it is typically indicated that designers look for suboptimal (but satisfactory) solutions rather than for *the* most optimal solutions.

In the theories following the above understanding, a key role is typically taken by the designer as a decision-maker. Designers are considered to be reflective practitioners [45.10]. *Schön* hereby refers to architectural designers, baseball pitchers, and musicians as example practitioners [45.10, pp. 54–55]. These practitioners continuously decide which constraints they wish or do not wish to adhere to, and which parameters they wish to use in what way. In contrast to the earlier belief of designers having a *problem-focused* strategy, they are now believed to have a more *solution-focused* or *goal-oriented* strategy instead [45.11, 12]. They proceed forward through the design process, continuously facing new design situations and addressing them as

they see fit in order to obtain the goal they have in mind at that specific moment in time. After addressing these design situations, the goal and the architectural knowledge (that were used to rely on) are typically adjusted based on the “back-talk of the design situation” or “situational feedback” [45.10]. Continuously taking action on design situations results in a co-evolution of problem space and solution space (see Fig. 45.1 and [45.13, 14]). In other words, architectural designers learn while doing, not only about the design situation at hand, but also about architecture and design in general.

Two key features of the prevalent current understanding of the design process can be distinguished in the above paragraphs:

1. An intensive interaction exists between designer and design context, thereby resulting in a stepwise proceeding through the design process.
2. Design thinking has an important reflective, *learning-while-doing* character [45.10], enabling designers to learn from experience.

We provide a simple schematic image of this interpretation of the (architectural) design process in Fig. 45.2. It indicates how a designer forms a mental model from an observation of the external world or design situation, and uses this mental model to devise an appropriate action for altering the design situation into a new one that can ideally be considered more optimal than the previous one.

45.1.2 Where Are the Models in all This?

The title of this chapter suggests that models are key to the above outlined features of the interaction between designers and design situations. Although the above paragraphs did not contain the word *model* too abundantly, it is present in all stages of the design process, in each step taken by designers in the direction of a fi-

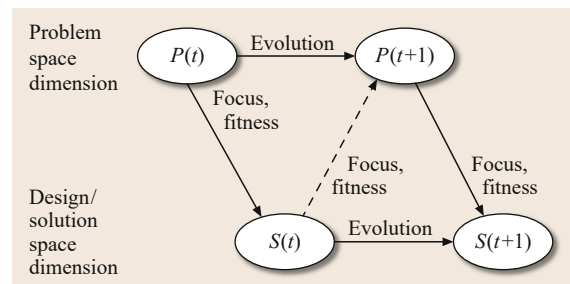


Fig. 45.1 Maher and Poons problem-design exploration model

nal (satisficing) design situation (not solution). In each interaction, namely, architectural designers rely on their background knowledge in making the appropriate decisions. This background memory is central to the above design process. After each decision, feedback or situational backtalk is returned to the designers by the design situations with which they interacted [45.10] (see also the oil painting example given by *Simon* [45.11, p. 163]). As this situational backtalk is interpreted by the designers, it also reshapes the background knowledge of these designers.

In learning-while-doing, designers build up knowledge in direct reference to concrete experiences. This knowledge might be related to “a designerly way of knowing” [45.8, 15], which was originally put forward by *Archer* in 1979 [45.16, p. 348]. On the basis of this kind of knowledge, designers make design decisions in newly encountered design contexts. Through their ongoing interaction with new design contexts, designers continuously modify or adjust their designerly way of knowing. Obviously, these adjustments have a significant effect on future design decisions.

It is important here to consider the stepwise evolution of the design process. As each step in the design process (or each consecutive design situation) can be considered a snapshot, the evolutions in design knowledge of the designer can also be considered in a stepwise manner. Each step in the evolution of someone’s design knowledge can hereby be considered a design model. This idea follows the theory by *Schön* about the architectural designer as a reflective practitioner, continuously interacting with the surrounding context and being affected by the backtalk of that surrounding context [45.10]. As explained here, Schön also indicates that design thinking depends on the repertoire or knowledge and experience of the designer. So, the context and background of designers play a key role in all steps of the decision-making processes of those designers. This idea also explains the notion of co-evolution in design problem and design solution [45.13, 14], assuming that the design *problem* can be considered the same as the

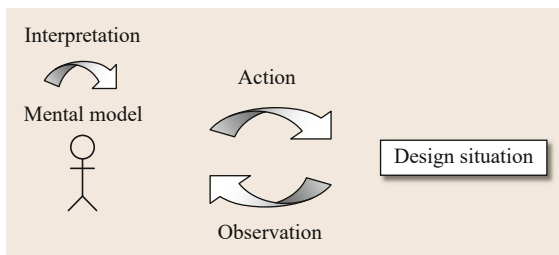


Fig. 45.2 Schematic outline of the steps (observation – interpretation – action) that are taken by designers during the interaction with an external design situation

current design situation, and that the design *solution* can be considered the same as the internal interpretation or model of the design situation in the designer’s mind (Fig. 45.3).

The changing background knowledge of the designer has been discussed at length by *Lawson* [45.9, p. 159]. He uses the term *guiding principles*. These guiding principles can be understood as the personal background knowledge or the knowledge by experience of a designer. They consist of familiar design patterns that a designer relies upon throughout the design thinking process. A designer thus never starts a design from an empty page, never from scratch or a blank mind. Instead, a designer always relies on a lifetime of knowledge built up by experiences. It is documented in [45.9, p. 179] how these guiding principles, in combination with a mental model of the situation at hand, essentially guide practitioners (including designers) through their thinking process. They play an important role not only in framing the design situation, but also in generating solutions for a problem, devising experiments, and in learning from experiences.

According to *Lawson* [45.9, p. 159], these guiding principles include not just objective, factual information, but include much more information, involving, for instance, motivations, beliefs, values, and attitudes. Guiding principles may be very intense and clearly structured or, on the other hand, vague and unclear, but they always influence design decisions one way or another. These characteristics of guiding principles can be related to the tacit dimension suggested by

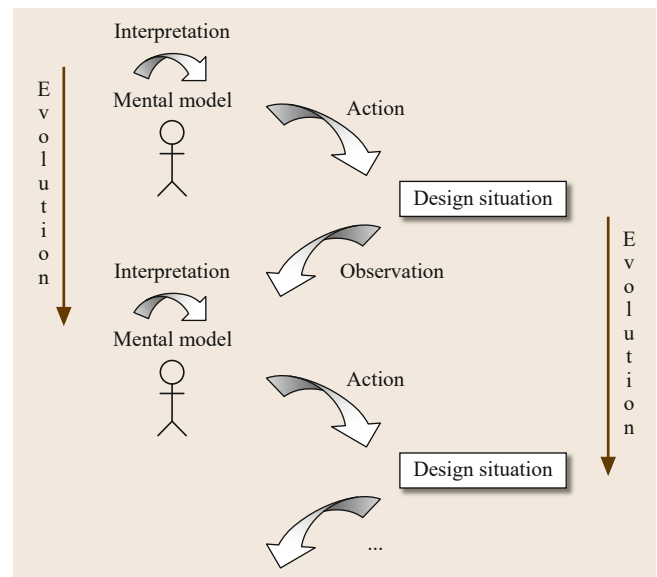


Fig. 45.3 Schematic outline of the co-evolution of the mental model of a design situation and the external design situation

Polanyi [45.17, 18], who states that some knowledge cannot be formalized and is essentially experience-based, vague, and thus tacit. In some research initiatives, guiding principles are almost considered part of a *personal religion*, thereby implicitly redefining design as “a very complicated act of faith” [45.19, p. 3]. This refers to the sometimes profound intensity of the designer’s belief in personal guiding principles, making it *morally right* to follow them. It is similarly indicated by Ward [45.20] how imagination relies almost entirely on known concepts, and, although modifications are made, they are typically only constituted by different combinations of known elements. It is hard to entirely step outside one’s own categories and beliefs, also in imagining [45.20].

It is very unclear in what form guiding principles are stored in the mind of a designer. What is clear though, is that this background information serves as a kind of repertoire of reference models for the designer to continuously and actively reorganize and restructure new design situations in memory into new abstract mental models or understandings of those design situations. In this context, references can be made to the work on case-based reasoning (CBR) [45.21–23], in the sense that the concept of CBR captures the idea of matching new cases with previously encountered cases in order to appropriately act upon them (Fig. 45.4).

45.1.3 Abstraction, Sense-Making, and Framing into Mental Models

From the previous sections, we now see that there should be some mechanism or phenomenon that allows architectural designers to *link* incoming design situations to their available repertoire of reference models (their experiential background information) so that they can obtain an abstract in-mind interpretation of those design situations. Basically, this is a moment of *interpretation* or *abstraction*. It occurs when a designer is sketching and all of a sudden gains an insight in the form of recognizing a part of the sketch as something he has seen before in an alternative context. It occurs when an architectural designer is visiting architectural building sites in order to find inspiration for the issues he is struggling with in the design he is working on. It occurs when the architectural designer communicates his latest design to a client or to any related or unrelated third party and gets insight from the feedback he receives through simple conversation. While doing each of these things, the architectural designer appears to interpret or make abstraction of the incoming information, after which he relates it in his mind to what he has experienced before. Previously used names for this phenomenon are *retrieval* (in the context of

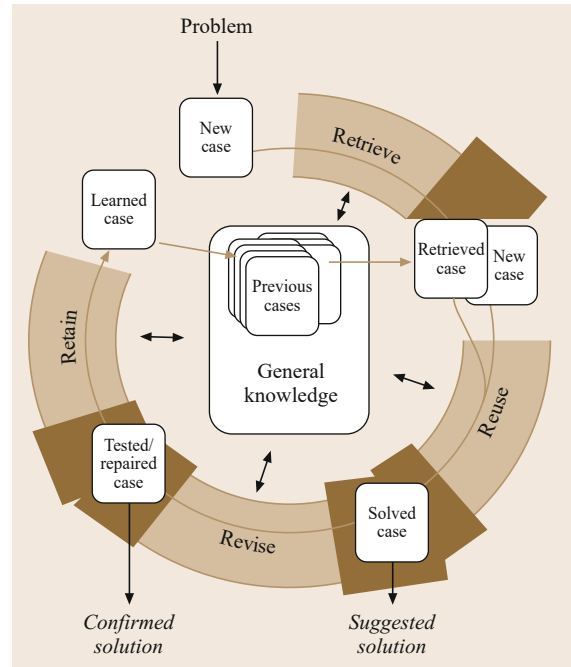


Fig. 45.4 Schematic outline of the case-based reasoning (CBR) process (after [45.21–23])

CBR – Fig. 45.4), *sense-making* and *problem framing/setting* [45.10, 24–27].

In [45.10, 26], for instance, design thinking is characterized as a specific kind of problem solving, in which the designer “must make sense of an uncertain situation that initially makes no sense”. Making sense of the situation then happens by switching back and forth between problem and solution, while continuously reframing both. Schön [45.10, pp. 39–40] refers to problem setting as:

“the process by which we define the decision to be made, the ends to be achieved, the means which may be chosen. In real-world practice, problems do not present themselves to the practitioner as givens. They must be constructed from the materials of problematic situations which are puzzling, troubling, and uncertain.”

In architecture, this often happens in the interaction between the client and the architect. This interaction usually starts with a client having an impression of what he wants to achieve, and an architect who does not have a clue of the client’s desires and needs. Often, these desires and needs are conflicting or do not seem to make sense. But, through the continuous feedback that the architect receives from the client, this design brief gets an increasingly clear structure in the designer’s mind. The architect thus sets the problem (how many floors

are requested in the building, who will be accessing and using the building, and so forth) through interaction with his surroundings. Initially, these surroundings are constituted by the feedback of the client, but later on, they will be formed by the sketch on his paper, visits to the building site, conversations with third parties, and so forth. One might say that the problem and solution are continuously reframed, resulting in a co-evolution of problem and solution. One might also say that the design situation is evolving step-by-step by the impact of an architect who decides based on his background knowledge and the context he is working in.

It is made clear by Schön [45.10] just how important the element of reframing the design situation is in his documentation of the differences between problem solving in a rational world and problem solving in the real world [45.10, p. 40]:

“When we set the problem, we select what we will treat as the *things* of the situation, we set the boundaries of our attention to it, and we impose upon it a coherence which allows us to say what is wrong and in what directions the situation needs to be changed. Problem setting is a process in which, interactively, we name the things to which we will attend and frame the context in which we will attend to them.”

In an architectural design situation, this occurs, for example, in the form of an architectural designer deciding at some point to look at the structural design in that particular design session, and leave out considerations in terms of energy or user comfort. In the following sessions, he might focus on material use, or user access, or something entirely else. But in each session, only one frame of the entire design situation is considered.

What is probably the most interesting moment in this reframing process, is the point where a solution is considered satisfactory enough and ready to be put into practice. This moment resembles the moment in which the well-known flash of insight occurs. This moment is described by [45.27] as the moment in which the two oscillating points, *problem* and *solution*, are still and close enough to be bridged by an *apposite concept* [45.27, pp. 439–440]:

“The crucial factor [...] is the bridging of these two partial models by the articulation of an apposite concept [...] which enables the models to be mapped onto each other. The *creative leap* is not so much a leap across the chasm between analysis and synthesis, as the throwing of a bridge across the chasm between problem and solution. Such an apposite *bridge* concept recognizably embodies satisfactory relationships between problem and so-

lution. It is the recognition of a satisfactory bridging concept that provides the *illumination* of the creative *flash of insight*.”

The term *apposite* makes an intended reference to the notion of appositional reasoning, which was originally coined by Bogen [45.28] and which is considered similar or the same as *abductive reasoning* by Cross [45.29].

Suppose that our architectural designer is still focusing on the structural load-bearing capacities of a building (cfr. framing of the situation). This architect might successfully relate the current design situation with a situation he encountered before and consequently decide to apply a similar structural design (for example, steel columns for load-bearing instead of concrete columns or brick walls), because the features of this structural design choice not only address issues in the overall building structure, they also appear to geniously address many other issues in terms of light penetration in the building, accessibility, fire safety, and so forth. This train of thought involves a bridge between an unsure problem and an unsure solution by an apposite concept.

45.1.4 Accessing Background Knowledge Through Analogical Reasoning?

The above distinguished *apposite bridge* between current design situation and the designer’s background knowledge is often addressed and investigated as a kind of analogical reasoning or CBR (see for instance [45.30–33]). Analogical reasoning is hereby explained as the cognitive ability to think about relational patterns [45.34–37], which allows one to find a structural alignment or mapping between a base and a target pattern residing in (partially) different domains [45.34, 37–40]. During design practice, architectural designers thus continuously make alignments between the current design situation (the base pattern) and previously experienced design situations (the target pattern). Relying on such alignments, designers infer which action to take for specific design situations and hence move forward.

This understanding was formed earlier by the investigations of Douglas and Isherwood, and Cross. Douglas and Isherwood [45.41], for example, indicate that [45.41, p. viii]:

“there is a prior and pervasive kind of reasoning that scans a scene and sizes it up, packing into one instant’s survey a process of matching, classifying and comparing. [...] Metaphoric appreciation, as all the words we have used suggest, is a work of

approximate measurement, scaling and comparison between like and unlike elements in a pattern.”

Later on, *Cross* [45.29] refers to several other research initiatives that distinguish a very similar kind of reasoning as fundamental for design thinking, thereby mentioning the terms abductive reasoning, productive reasoning and appositional reasoning as called by their respective inventors *Peirce* [45.42], *March* [45.43] and *Bogen* [45.28].

This kind of reasoning is obviously relied upon in the interpretation step which is considered in the previous sections of this chapter. This kind of reasoning is very poorly understood in general. The only thing we appear to know, is that it happens. As we are confined to behavioral studies of human design activity, and we cannot simply access the human mind during design activity, there is no real trustworthy indication of *how* it happens. When turning to the interpretation of the “work of approximate measurement” or “metaphoric appreciation” [45.41, p. viii] as a kind of analogical reasoning, we can find out that analogical reasoning often occurs between a new design-related experience (building, sketch, three-dimensional (3-D) model, conversation, and so forth) and a previous design experience as it is stored in the human mind [45.30]. But also in the very act of sketching, analogical reasoning is crucial, because it allows reinterpreting or *seeing as*, as *Goldschmidt* puts it [45.32, 33]. In *seeing as*, the designer reinterprets the sketch and, as such, adds new and origi-

nal meaning to it, thereby generating new ideas [45.32, 33]. For many student designers, who have little experience in architectural examples, *seeing as* often occurs in a more superficial way. They tend to find similarities between their sketches and other, often unrelated concepts and things based on geometrical features and shape (*Look, this looks like a ship. Maybe we can... or We are near the sea, why don't we make the building in the shape of a wave?*). Experienced architectural designers often make more abstract, meaningful and/or direct analogies, because they have a much richer set of background experiences on which they can rely.

Because analogical reasoning is guided by encountered target patterns [45.34, 37–39], the designer appears to proceed *in an unstructured and perhaps aimless way*. Therefore, the earlier mentioned definition of imagining [45.9, 44] is also closely related to analogical reasoning. A similar conclusion is given by *Boden's* research on the creative mind [45.45]. She stresses the importance of the incubation phase in creative thinking. In this phase, the conscious mind focuses on other domains, problems, or projects, thus enabling the creative mind to make diverse alternative and previously unconsidered analogies with the situation at hand [45.45, pp. 33–35].

When turning back to our initial schema of the design process (Figs. 45.2 and 45.3), we can locate the position of analogical or appositional reasoning somewhere between the design situation that is observed and the mental model resulting in the designer's mind (Fig. 45.5).

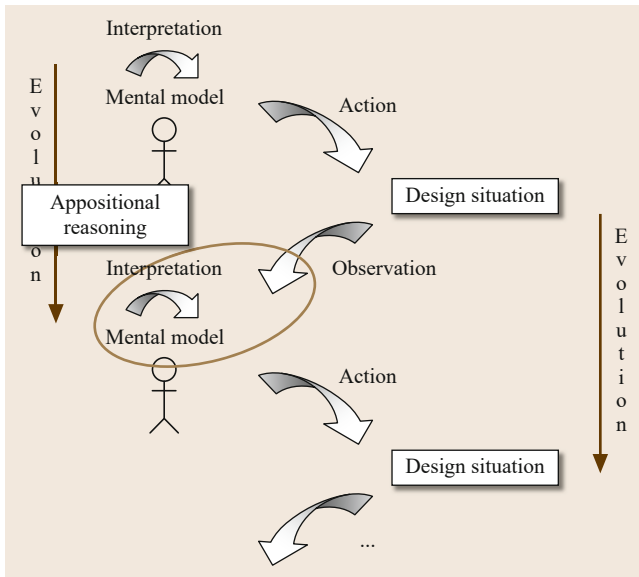


Fig. 45.5 The approximate location of analogical or appositional reasoning in our earlier schema of the design process (Figs. 45.2 and 45.3)

45.1.5 Abstraction from Representation Model to Mental Model

If the bridging between the current design situation and the background knowledge of the designer occurs through analogical reasoning, the *base pattern* [45.34, 37–39] is tremendously important. Namely, this implies that what designers experience from design situations are the sole *seeds* from which they are able to make interpretations and act creatively upon.

Of course, it is not realistic to assume that designers interact with design situations as a *whole*, something that might be concluded from the schemas in Figs. 45.2, 45.3 and 45.5. Instead, designers interact with a *medium* that provides a bounded interface (a frame) to the current design situation (Fig. 45.6). These media of interaction are of various kinds, including people, objects, sketches, 3-D models and so forth.

The most obvious experiences of design situations in architecture (elements of interaction in Fig. 45.6) are sketches [45.46]. As *Goldschmidt* indicates [45.32, 33],

sketches are not only visual expressions of what one wants to express, they are also elements for reinterpretation and thus for generating all kinds of new knowledge. *Cross* similarly refers to the importance of sketching as it enables a designer to explore several solutions and problems to a certain design situation at once, thereby considering several levels of detail at once [45.8, 47]. *Schön* [45.10], in turn, refers to the habit of many designers to continuously make representations of the design situation at hand in documents, plans, sketches, and so forth, thereby allowing a designer both to answer a previous design situation, and frame the design situation anew into an alternative perspective. To say it in *Schön's* words, the designer “shapes the situation in accordance with his initial appreciation of it, the situation *talks back*, and he responds to the situation’s back-talk” [45.10, p. 79].

We will not dive into all the characteristic features of sketching, but we will instead generalize among very diverse possibly available base patterns used to initiate analogical reasoning. Sketches, namely, are but one of the many possible representation media that can be used by designers to reflect on the design situation. Besides sketches, designers can use conversations with colleague architects, physical scale models, walking around on construction sites or inspiring related pieces of architecture, and so forth. Rather recently, this array of interaction media has notably enlarged through the development of all kinds of information technologies. New media are now available to the designer, among which there are parametric design models, two-dimensional (2-D) computer-aided design (CAD) models, 3-D BIM models, databases, websites on the Internet, teleconference applications, virtual game engine environments, and so forth. So, design representations can

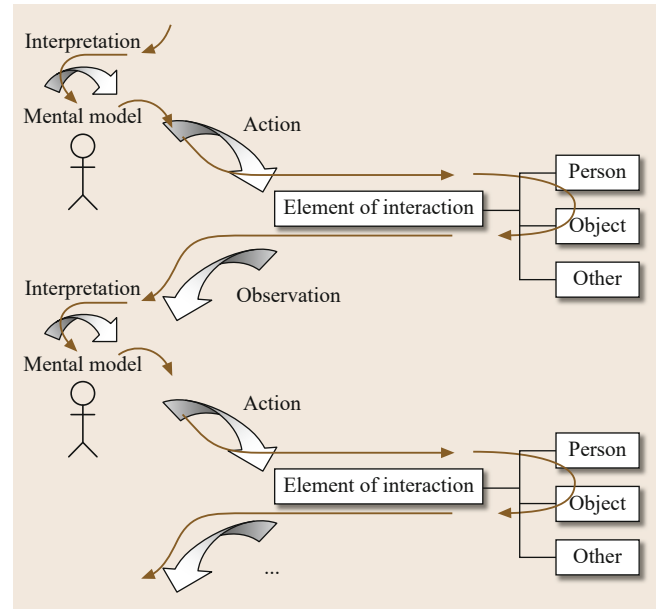


Fig. 45.6 Indication of the diverse elements with which a designer can interact: people, objects, other

take on many forms, including a sketch [45.32, 46, 48], a simple discussion [45.49], a CAD model, and so forth.

The main idea here is that by making alternative representations, designers aim at confirming the abstract model they have of a particular design situation, which is always to some extent unclear, wicked or unknown. By experiencing the resulting design representation, a new understanding or abstract in-mind model of the design situation thus emerges, which reframes the previous design situation and thus alters the design process.

45.2 BIM Models and Parametric Models

The apposite bridging, interpretation, or abductive reasoning step is a capacity that is not available in a computer. As we do not know the way in which our background information is stored in our neurological brain, we are obviously unable to replicate this. As a result, no information system exists that is able to store the target patterns that are required for analogical reasoning, let alone one that is able to match these target patterns with incoming sensory information and thus make analogies in a creative and autonomous manner, as we do as human beings. So, no information system is able to take over such a typically human capacity.

45.2.1 New Technological Media in Design Thinking

Nonetheless, architectural designers can still take advantage of information systems as an additional medium that allows them to make alternative representations with which they can interact in their sense-making or interpretation process. Any computer-based representation thus functions similar to how a sketch functions. Each such representation hereby represents only a limited semantic domain, only a partial reflection of the complete design situation. They are representations of the designer’s mental model, and by no means do they

come close to the original mental model which is always in the mind of the architectural designer and which is inherently ungraspable. Instead, the representations in these media are to be considered as representations that result from this mental model and that form, as such, initiators for further reflection on this mental model. In the following section, we will briefly look into the consequences in the context of BIM modeling and parametric modeling software, as reference examples.

45.2.2 BIM Models and Parametric Models

There have been many developments in information and communication technologies (ICT) for the domain of architecture, engineering and construction (AEC). Most ICT applications in this domain allow to build a certain representation or model of an architectural design (element). A considerable number of the developments in ICT for the AEC domain have focused on enlarging the amount of semantics that can be included within the resulting representations or models. In other words, instead of allowing designers to model a design using lines, points, boxes or surfaces, they now typically allow to model typed objects, such as walls, doors, structural columns, and so forth. Each instance of one of those object types can then automatically be represented using the properties that were predefined for these object types. These properties typically include basic features, such as height, width, and location, but they also include far more complex properties, such as relations with other objects (aggregation, decomposition, neighborhood, etc.), representational properties (texture, geometry, etc.), and so forth.

These developments have resulted in a number of modeling applications with capacities that make them stand out from the traditional CAD or computer-aided drafting applications. One can distinguish the following modeling application types.

Building Information Modeling (BIM) Applications

BIM applications allow to represent buildings using a hierarchical structure of typed objects, including building objects, materials, people, and so forth [45.50]. References can be made to the concept of feature-based modeling (FBM) [45.51]. The workflow in such applications results in a single 3-D BIM model, which is supposed to include all the information needed to build the building (Fig. 45.7).

Parametric and Generative Modeling Applications

Parametric and generative modeling applications allow to represent a design using a number of typically

geometric parameters. By moving sliders, parameter values are changed, and a design model can be regenerated from these modified parameter values. The design model is hereby formed by a network of parameter values and control functions that generate geometry using the associated parameter values (Fig. 45.8).

Database Applications

Many more basic applications in the architectural design and construction industry still rely on rather basic relational database systems. This includes, for instance, four-dimensional (4-D) (time scheduling) and five-dimensional (5-D) (cost scheduling) applications, facility management (FM) applications, energy performance calculation software, and so forth. Of course, such applications also use a semantic model of the architectural design, represented by tuple values in a relational database.

45.2.3 Features and Issues in the Usage of the New Modeling Applications

Obviously, for all of the three technology types mentioned above (BIM software, parametric software, database software), there are numerous interpretations and implementations. We will not dive into the details for each of these technology types in this chapter, as the focus is here on the role and function of these new interaction media within the design process. Within the scope of this chapter, it should suffice to keep in mind that each of the outlined software environments allows to build a simple or complex semantic model as a representation to interact with.

The semantic model that can be built using the outlined modeling environment typically follows the information structure that is defined by the programming code behind the corresponding modeling application.



Fig. 45.7 Revit Architecture is one of the available BIM modeling applications that allow architectural designers to model a BIM model representation of their design

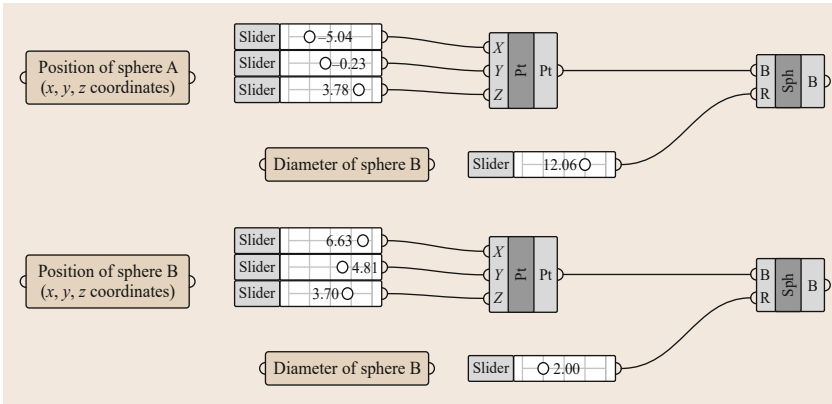


Fig. 45.8 Rhinoceros, together with the Grasshopper plugin, is an often used environment for the parametric modeling of building geometry. This environment is typically relying on nodes and sliders to represent the semantics of the designed geometry

A Revit BIM model (Fig. 45.7) is something that cannot be captured by a parametric model in Rhinoceros and Grasshopper (Fig. 45.8), because both modeling environments deploy different programming codes and corresponding information structures. Basic 2-D CAD applications enable the user to model a design in 2-D geometric object models, using lines, points, surfaces, and so forth. Basic 3-D applications allow this in 3-D, using boxes, spheres, voids, and so forth. More advanced CAD systems typically focus on information management, and thus enable the user to model a design in more informative elements, such as walls, windows, columns, beams, and so forth.

By allowing designers to model their design in a more meaningful manner (more semantic object types such as walls and doors instead of the more syntactic points and lines), designers are supposedly enabled to represent their design as a model that is much more closely related to the in-mind abstract model that they use in the design thinking process (Fig. 45.9). In concrete terms, rather than only being able to represent a design using pencil marks on a paper, semantic features in software applications allow the designers to model a semantic structure (the ontology) that reflects the in-mind structure of their designs and then use that semantic structure to represent the actual design (instantiation of the ontology). It is then easier to make decisions, as the gap between the semantic model of the design (Fig. 45.9, right) and the in-mind design model (Fig. 45.9, left) is smaller and the interpretation step that is to be made by the designer should be easier.

This functions relatively well. There are, of course, a number of issues that are commonly identified in using these modeling applications:

1. *Too much time is needed* to build the appropriate semantic structure for one's particular in-mind design model, resulting in a preference for quicker draft-

ing applications or drafting media (computer-aided drafting or sketch environments).

2. As the in-mind design model is continuously changing (cfr. co-evolution of problem and solution), one's *semantic structure is never up to date* with that in-mind design model. In other words, the semantic 3-D model (Fig. 45.9, right) is always a number of steps behind the in-mind design model (Fig. 45.9, left).
3. A need arises to share the semantic model of the design with other people, as is typically also done with sketches. However, transferring/communicating the meaning of the custom semantic structure to anyone else requires considerable effort from that other

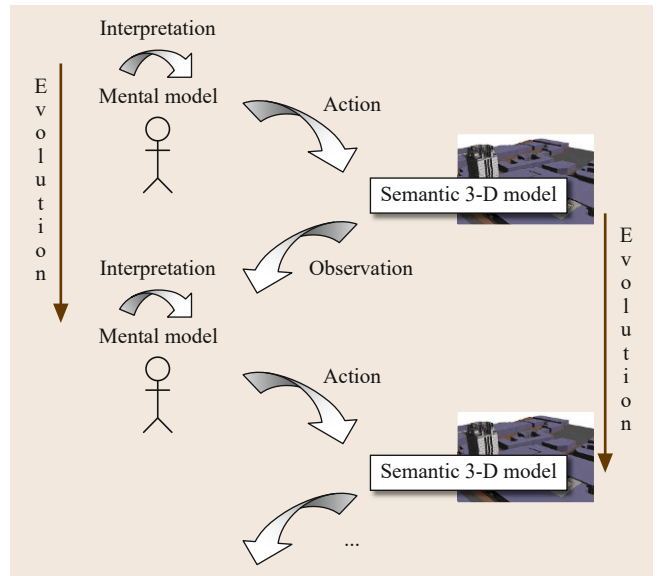


Fig. 45.9 The design process, as schematically represented earlier, indicating how semantic 3-D models are used by architectural designers

person as the presented semantic structure never matches with his own in-mind model. This is related to the well-known *interoperability* problem (see for reference [45.52–56]).

When considering the theories presented in the first section of this chapter, it is rather obvious and understandable that these three main difficulties emerge. If the abstract in-mind design model changes at every single snapshot of interaction with some kind of interaction medium (Fig. 45.9), of course, the representation on the medium with which is interacted is outdated at every single moment in time. Hence, it would also be a futile attempt to make a complete representation of an abstract in-mind design model in any of the available 3-D modeling applications. Note that it would likewise

be a futile attempt to make such a complete representation in one paper sketch.

Furthermore, in order to get information into another information structure (the interoperability challenge), no matter in what kind of information structure it was originally captured, one *always* requires interpretation if it is to be done properly. Thus, this requires human effort. The best option in this context of interoperability problems is to at least make flexible and intuitive tools available so that a designer can at least do the required interpretation effort in a relatively smooth and efficient manner and transform information from one semantic schema into another. Semantic web technologies might provide some of such flexible and intuitive tools, as is indicated in [45.56–59].

45.3 Implementing and Using ICT for Design and Construction

Considering the above writings in this chapter, modeling applications will remain to be used as alternative media for interaction by architectural designers and construction experts, no matter the amount of semantics they allow designers to represent. The semantic structures of these applications might to some extent match or resemble the in-mind model of the designer, but it will always fall short in comparison. That being considered, there is, first and foremost, a need for a flexible, intuitive, and above all, pragmatic usage of modeling applications in architectural design and construction. Many examples of such usage strategies exist in practice. Unfortunately, this pragmatic approach remains to include the effects caused by limited interoperability of information [45.53], namely an increased loss of time and an increased number of construction errors during the construction process due to the necessary remodeling of information from one environment to another.

45.3.1 Pragmatic Usage of Semantic Modeling Applications

A first example of this pragmatic approach is documented in [45.60] and elaborates on the pragmatic usage of BIM systems in the construction of the Port House in Antwerp. Initially, an integrated BIM approach [45.50] was targeted in this project, using Revit Architecture as a central BIM environment. The BIM model would then serve as a central reference model containing and providing information for all project partners. The construction company tried to be faithful to the BIM idea, but they gradually shifted to a more pragmatic software usage approach. A BIM model was

engineered and maintained as a reference model by the construction team in charge of the whole construction process. Depending on the background and software usage approach of the project partners, the information in this BIM model was interpreted and provided to the related project partners according to the semantic structures they were using and the medium in which they were working. Communication of information thus included, for instance, Excel spreadsheets, PDF documents, and partial 3-D models in various file formats. In importing and exporting these documents to and from the modeling environment, human interpretation was required in the form of manual conversion efforts. Nevertheless, this human interpretation step produced a desirable result within a foreseeable and plannable time span.

The pragmatic software usage approach outlined here and in [45.60] requires a very good balance between automatic (technological reformatting) and manual procedures (human interpretation steps). The information structures of applications can be integrated either by implementing project-specific software components or by manual modeling. The key to a pragmatic usage of (semantic) modeling applications is finding the right balance between these automatic and manual procedures (Fig. 45.10), so that it fits the current design situation.

Similar approaches appear to be followed in other large architectural firms that concentrate on geometrically or semantically complex architectural projects. In most cases, the balance shifts towards the usage of automatic methods, as larger and more complex projects often benefit from automatic procedures in terms of

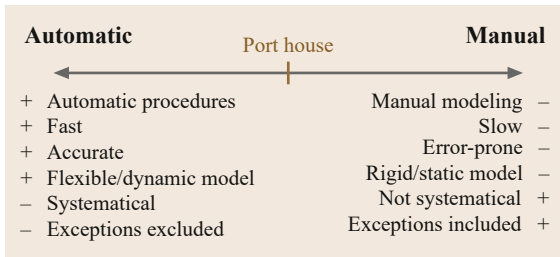


Fig. 45.10 Indication of the balance between advantages and disadvantages of using manual and automatic procedures in information-handling in the Port House architectural design project (after [45.60])

efficiency or return on investment. A good example is the Specialist Modelling Group (SMG) in Foster + Partners, which is a group that appears to concentrate on optimizing information exchange and complex modeling for specific projects [45.61]. It appears to be confirmed in [45.60] that this not necessarily requires a standard information management approach. Pragmatically constructing a common agreement between project team members and combining manual and automatic methods with an expert group of programmers, process modelers and/or communication specialists can prove to be just as effective.

45.3.2 The Usage of Design Agents or Assistants

Thus, for architectural designers and construction specialists, the better option in using information technologies is to consider these technologies as yet another set of available media with which they can interact as part of their reflective practice. As with all media, there are certain rules, advantages and disadvantages that characterize each medium. One should thus carefully consider what medium to consider for what purpose. Something that all media have in common, nonetheless, is that they can all capture but a fragment of our in-mind abstract knowledge.

One might wonder whether there really is no alternative, whether information systems will really not be capable, almost as if by definition, to capture an abstract model similar to the way in which human beings do so. As indicated in the first section of this chapter, there is only one thing that stands in the way of such a development and that is the element of *interpretation, analogical reasoning* or *abductive reasoning* (see Fig. 45.5). This seems to be one of the main capacities that distinguishes man from computer. Key design actions require interpretation, including: the mapping of incoming semantic information to its own semantic structures, or the construction of context-specific and purposeful shape

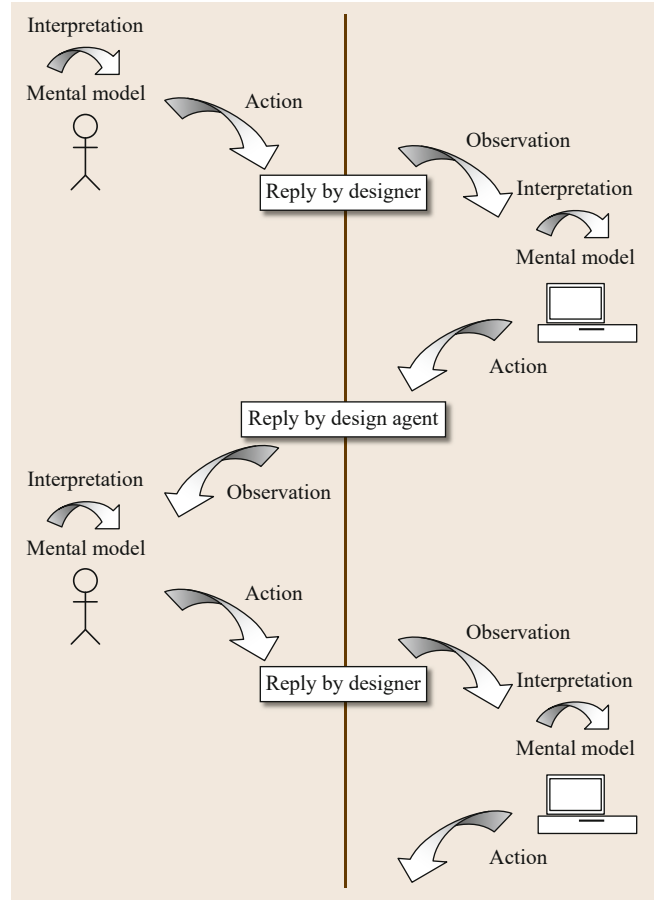


Fig. 45.11 Our original schema of the design process, adapted in order to communicate how the interaction between designer and computer-based agent could be taking place

grammars and creation of appropriate design decisions while relying on and continuously adapting this shape grammar, or the performance of multidimensional optimization or satisficing of design constraints. In order for a computer to perform these actions, it will first have to be able to interpret incoming information and understand it using a mechanism that involves a form of abductive, analogical or appositional reasoning.

If this is ever realized, it will result in the third of three realms of software usages in architectural design and construction support, as they were originally identified by Lawson [45.62]. We have seen the first two in the previous sections, namely, (1) the computer as a rigid problem-solving and all-knowing oracle, and (2) the computer as a draughtsman, which is simply used as yet another interaction medium while the designer remains the only decision-making and interpreting agent. In a third scenario (3), the computer is to be used as a true design agent or assistant (see also [45.57] for more in-

formation about how this relates to the earlier outlined pragmatic approach). In this scenario, designers interact with autonomously reasoning computer-based agents, as if they would interact with any other medium. Both the designer and the computer-based agent would then interpret incoming information and learn from experience. Using our schematic format of the design process again, this interaction between designer and computer-based agent would likely look as depicted in Fig. 45.11, with processes of inquiry happening in the human mind (Fig. 45.11, left) as well as in the computer agent’s information system (Fig. 45.11, right).

For this scenario to be realized, interpretation as it functions in the human mind needs to be implemented. In this regard, some effort has already been placed in the automation or implementation of abductive reasoning. It is useful to consider the number of research initiatives in the domain of abductive logic programming (ALP) [45.63]. Second, we have already looked into some pointers towards CBR in brief [45.21–23]. Unfortunately, most of these research initiatives appear to lack tangible research results. Not one of the resulting software systems appears capable of reliably simulating an interpretation step as it is produced by any human agent in an autonomous and natural manner, let alone within an architectural design context.

Some researchers have taken a different approach and have primarily considered the larger cycle that people appear to go through in any interaction with an outside world [45.64–66]. This larger cycle resembles

the original *process of inquiry* as it was outlined multiple times by Peirce [45.42]. It is hereby considered useful to not only consider abductive reasoning, but to also look for the occurrence of inductive and deductive reasoning as they were considered by Peirce [45.42]. This implementation approach makes sense as it seems to be the only valid way to let an agent learn in an autonomous manner, yet enabling it to store previous experiences so that they can be reused in interpreting future observations. Additionally, combining abductive reasoning with inductive and deductive reasoning in an iterative cycle is required if the agent’s functionality is to remain true to Peirce’s idea of a process of inquiry. When relying on this interpretation of Peirce’s process of inquiry [45.64–66], we might be able to indicate where such a cycle resides in our earlier diagrams of the design process (Fig. 45.12).

One research initiative that appeared to take on the realization of this agent approach can be found in the work of King [45.67, 68]. In this work, the goal was to build a machine that can autonomously “discover new scientific knowledge” by its capacity to “devise a hypothesis, carry out experiments, to test it and assess results” [45.68]. This endeavor to enable a machine to go through these diverse stages is very much like building a machine that is able to go through Peirce’s process of inquiry, in the sense that in the case of King’s research, the main question was also whether they were able to build a “robot scientist that can actually accomplish the entire process” [45.68]. Eventually, a machine was built that is capable of successfully constructing hypotheses, devise appropriate experiments and test them, in the domain of functional genomics, a domain in which the relations between genes and their functions are investigated. These actions are made using a core body of knowledge, resulting from a “formalization that involves over 10 000 different research units in a nested treelike structure, 10 levels deep, that relates 6.6 million biomass measurements to their logical description” [45.67].

In any case, even if this autonomous agent-based approach proves valid and a useful implementation of such an autonomous reasoning agent can be realized, it will take years before this agent takes on a form that can provide help to an architectural designer as it was theoretically outlined by Lawson [45.62]. Namely, it requires at least 25 years for brilliant human agents to learn this capacity and there is no reason to assume that computer-based agents might be able to outperform this human capacity. So, to conclude, we are best off, for now, with following the pragmatic software usage approach as it was outlined earlier in this chapter and keep relying on the architectural designers themselves as reflective practitioners and decision-makers.

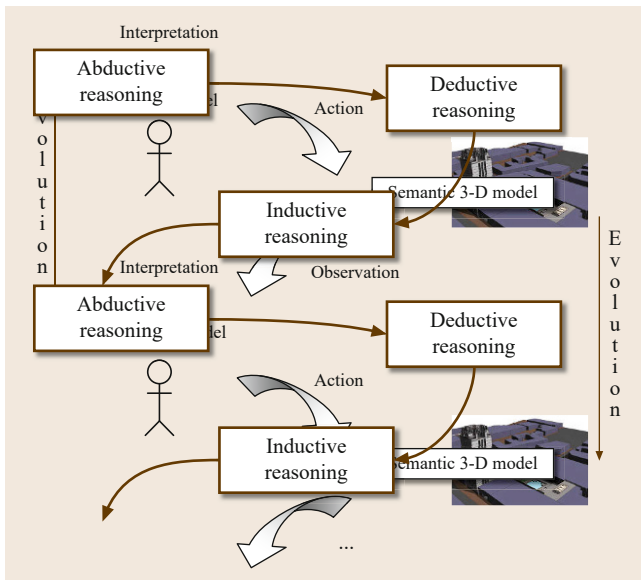


Fig. 45.12 Our original schema of the design process, with an overlay of where the abductive, deductive, and inductive reasoning might be taking place

References

- 45.1 N. Cross: Forty years of design research, *Des. Stud.* **28**, 1–4 (2007)
- 45.2 N. Bayazit: Investigating design: A review of forty years of design research, *Des. Issues* **20**(1), 16–29 (2004)
- 45.3 C.M. Eastman: New directions in design cognition: Studies of representation and recall. In: *Design Knowing and Learning: Cognition in Design Education*, ed. by C. Eastman, W. Newstetter, M. McCracken (Elsevier, Amsterdam 2001) pp. 147–198
- 45.4 H.W.J. Rittel, M.M. Webber: Dilemmas in a general theory of planning, *Policy Sci.* **4**, 155–169 (1973)
- 45.5 H.W.J. Rittel, M.M. Webber: Planning problems are wicked problems. In: *Developments in Design Methodology*, ed. by N. Cross (Wiley, Hoboken 1984) pp. 135–144
- 45.6 H.A. Simon: The structure of ill-structured problems, *Artif. Intell.* **4**, 181–201 (1973)
- 45.7 K. Dorst: Design problems and design paradoxes, *Des. Issues* **22**(3), 4–17 (2006)
- 45.8 N. Cross: *Designerly Ways of Knowing* (Springer, London 2006)
- 45.9 B. Lawson: *How Designers Think – The Design Process Demystified*, 4th edn. (Architectural Press (Elsevier), Amsterdam 2005)
- 45.10 D. Schön: *The Reflective Practitioner: How Professionals Think in Action* (Temple Smith, London 1983)
- 45.11 H.A. Simon: *The Sciences of the Artificial*, 2nd edn. (The MIT Press, Cambridge 1996)
- 45.12 N. Cross: Designerly ways of knowing, *Des. Stud.* **3**(4), 221–227 (1982)
- 45.13 M.L. Maher, J. Poon: Modelling design exploration as co-evolution, *Microcomput. Civil Eng.* **11**, 195–209 (1996)
- 45.14 J. Poon, M.L. Maher: Co-evolution in design: A case study of the Sydney Opera House, *Proc. 2nd Conf. Comput. Aided Archit. Des. Res. Asia*, Hsinchu, ed. by Y.-T. Liu, J.-Y. Tsou, J.-H. Hou (1997) pp. 439–448
- 45.15 N. Cross: Styles of learning, designing and computing, *Des. Stud.* **6**(3), 157–162 (1985)
- 45.16 L.B. Archer: Design as a discipline, *Des. Stud.* **1**(1), 17–20 (1979)
- 45.17 M. Polanyi: *The Tacit Dimension*, 1st edn. (Doubleday, New York 1967)
- 45.18 M. Polanyi: *Personal Knowledge: Towards a Post-Critical Philosophy* (Routledge and Kegan Paul, London 1958)
- 45.19 J.C. Jones: *Design Methods: Seeds of Human Futures*, 1st edn. (Wiley, Hoboken 1970)
- 45.20 T.B. Ward: Structured imagination: The role of category structure in exemplar generation, *Cogn. Psychol.* **27**, 1–40 (1994)
- 45.21 A. Aamodt, E. Plaza: Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AI Commun.* **7**, 39–59 (1994)
- 45.22 J. Kolodner: An introduction to case-based reasoning, *Artif. Intell. Rev.* **6**, 3–34 (1992)
- 45.23 J. Kolodner: *Case-Based Reasoning* (Morgan-Kaufman, Burlington 1993)
- 45.24 J. Kolko: Abductive thinking and sensemaking: The drivers of design synthesis, *Des. Issues* **26**, 15–28 (2010)
- 45.25 K. Dorst: The core of ‘design thinking’ and its application, *Des. Stud.* **32**, 521–532 (2011)
- 45.26 D. Schön: Generative metaphor: A perspective on problem-setting in social policy. In: *Metaphor and Thought*, ed. by A. Ortony (Cambridge Univ. Press, Cambridge 1979)
- 45.27 N. Cross: Descriptive models of creative design: Application to an example, *Des. Stud.* **18**(4), 427–455 (1997)
- 45.28 J.E. Bogen: The other side of the brain II: An appositional mind, *Bull. Los Angel. Neurol. Soc.* **34**(3), 135–162 (1969)
- 45.29 N. Cross: The nature and nurture of design ability, *Des. Stud.* **11**(3), 127–140 (1990)
- 45.30 A. Heylighen: Building memories, *Build. Res. Inf.* **35**, 90–100 (2007)
- 45.31 M.L. Maher, P. Pu: *Issues and Applications of Case-Based Reasoning in Design* (Lawrence Erlbaum Associates, Mahwah 1997)
- 45.32 G. Goldschmidt: The dialectics of sketching, *Des. Stud.* **4**, 123–143 (1991)
- 45.33 G. Goldschmidt: On visual design thinking: The vis kids of architecture, *Des. Stud.* **15**(2), 158–174 (1994)
- 45.34 K. Grace, R. Saunders, J.S. Gero: Interpretation-driven visual association, *Proc. 2nd Int. Conf. Comput. Creativ. Mexico City*, ed. by D. Ventura, P. Gervás, F.D. Harrell, M.L. Maher, A. Pease, G. Wiggins (Univ. Autonoma Metropolitana, Mexico City 2011) pp. 132–134
- 45.35 K.J. Holyoak, D. Gentner, B.N. Kokinov: Introduction: The place of analogy in cognition. In: *The Analogical Mind: Perspectives From Cognitive Science*, ed. by D. Gentner, K.J. Holyoak, B.N. Kokinov (The MIT Press, Cambridge 2001)
- 45.36 B.N. Kokinov: Analogy is like cognition: Dynamic, emergent and context sensitive. In: *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*, ed. by K.J. Holyoak, D. Gentner, B.N. Kokinov (NBU Press, Sofia 1998) pp. 96–105
- 45.37 T.B. Ward: Analogical distance and purpose in creative thought: Mental leaps versus mental hops. In: *Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences*, ed. by K.J. Holyoak, D. Gentner, B.N. Kokinov (NBU Press, Sofia 1998)
- 45.38 D. Gentner, B.F. Bowdle, P. Wolff, C. Boronat: Metaphor is like analogy. In: *The Analogical Mind: Perspectives from Cognitive Science*, ed. by D. Gentner, K.J. Holyoak, B.N. Kokinov (The MIT Press, Cambridge 2001)
- 45.39 D.R. Hofstadter: Analogy as the core of cognition. In: *The Analogical Mind: Perspectives from Cognitive Science*, ed. by D. Gentner, K.J. Holyoak, B.N. Kokinov (The MIT Press, Cambridge 2001)

- 45.40 G. Lakoff, M. Johnson: The metaphorical structure of the human conceptual system, *Cogn. Sci.* **4**(2), 195–208 (1980)
- 45.41 M. Douglas, B. Isherwood: *The World of Goods* (Allen Lane, London 1979)
- 45.42 C.S. Peirce: *Collected Papers of Charles Sanders Peirce*, Vols. 1–6 (Eds. Charles Hartshorne & Paul Weiss) (1931–1935), Vols. 7–8 (Ed. Arthur W. Burks) (1958) (Harvard Univ. Press, Cambridge 1958)
- 45.43 L.J. March: The logic of design and the question of value. In: *The Architecture of Form*, ed. by L.J. March (Cambridge University Press, Cambridge 1976) pp. 1–40
- 45.44 B. Lawson: Cognitive strategies in architectural design, *Ergonomics* **22**(1), 59–68 (1979)
- 45.45 M.A. Boden: *The Creative Mind: Myths and Mechanisms*, 2nd edn. (Routledge, London 2004)
- 45.46 A.T. Purcell, J.S. Gero: Drawings and the design process, *Des. Stud.* **19**, 389–429 (1998)
- 45.47 N. Cross: Natural intelligence in design, *Des. Stud.* **20**(1), 25–39 (1999)
- 45.48 J. Pallasmaa: *The Thinking Hand: Existential and Embodied Wisdom in Architecture* (Wiley, Hoboken 2009)
- 45.49 J. Conklin: *Dialogue Mapping: Building Shared Understanding of Wicked Problems* (Wiley, Hoboken 2005)
- 45.50 C.M. Eastman, P. Teicholz, R. Sacks, K.K. Liston: *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Architects, Engineers, Contractors, and Fabricators* (Wiley, Hoboken 2008)
- 45.51 J.J. Shah, M.T. Rogers: Functional requirements and conceptual design of the feature-based modelling system, *Comput.-Aided Eng. J.* **5**, 9–15 (1988)
- 45.52 K.H. Veltman: Syntactic and semantic interoperability: New approaches to knowledge and the semantic web, *The New Rev. Inf. Netw.* **7**, 159–184 (2001)
- 45.53 M.P. Gallagher, A.C. O'Connor, J.L. Dettbar, L.T. Gilday: *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry, NIST Report GCR 04–867* (National Institute of Standards and Technology, Gaithersburg 2004)
- 45.54 S. Gerbino: Tools for the interoperability among CAD systems, *Proc. 13th ADM – 15th INGEGRAF Int. Conf. Tools Methods Evol. Eng. Des. Cassino* (2003)
- 45.55 T. Pazlar, Ž. Turk: Interoperability in practice: Geometric data exchange using the IFC standard, *J. Inf. Technol. Constr.* **13**, 362–380 (2008)
- 45.56 P. Pauwels: Supporting decision-making in the building life-cycle using linked building data, *Buildings* **4**(3), 549–579 (2014)
- 45.57 P. Pauwels, R. De Meyer, J. van Campenhout: Design thinking support: Information systems vs. reasoning, *Des. Issues* **29**(2), 42–59 (2012)
- 45.58 P. Pauwels, R. De Meyer, J. van Campenhout: Interoperability for the design and construction industry through semantic web technology. In: *Semantic Multimedia*, LNCS, Vol. 6725, (Springer, Berlin, Heidelberg 2010) pp. 143–158
- 45.59 P. Pauwels, D. van Deursen, J. De Roo, T. van Ackere, R. De Meyer, R. van de Walle, J. van Campenhout: Threedimensional information exchange over the semantic web for the domain of architecture, engineering and construction, *Artif. Intell. Eng. Des. Manuf.* **25**, 317–332 (2011)
- 45.60 P. Pauwels, P. Present, T. Strobbe: A pragmatic approach towards software usage in construction projects: The Port House in Antwerp, Belgium, *Proc. 9th Eur. Conf. Prod. Process Model. eWork eBus. Archit. Eng. Constr. Reykjavik*, ed. by G. Gudnason, R. Scherer (Taylor Francis, Boca Raton 2012) pp. 509–512
- 45.61 B. Peters, X. De Kestelier: The work of Foster and Partners Specialist Modelling Group, *The Bridges Conf. Math. Connect. Art Music Sci.* (2006)
- 45.62 B. Lawson: Oracles, draughtsmen, and agents: The nature of knowledge and creativity in design and the role of IT, *Autom. Constr.* **14**, 383–391 (2005)
- 45.63 D. Poole: Probabilistic Horn abduction and Bayesian networks, *Artif. Intell.* **64**(1), 81–129 (1993)
- 45.64 P. Pauwels, R. Bod: Including the power of interpretation through a simulation of Peirce's process of inquiry, *Lit. Linguist. Comput.* **28**(3), 452–460 (2013)
- 45.65 P. Pauwels, R. Bod: Architectural design thinking as a form of model-based reasoning. In: *Model-Based Reasoning in Science and Technology*, Studies in Applied Philosophy, Epistemology and Rational Ethics, ed. by L. Magnani (Springer, Berlin, Heidelberg 2013) pp. 583–608
- 45.66 A. Aliseda: *Abductive Reasoning: Logical Investigations Into Discovery and Explanation*, Synthese Library, Vol. 330 (Springer, Dordrecht 2006)
- 45.67 R.D. King, J. Rowland, S.G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L.N. Soldatova, A. Sparkes, K.E. Whelan, A. Clare: The automation of science, *Science* **324**, 85–89 (2009)
- 45.68 R.D. King: Rise of the robo scientists, *Sci. Am.* **304**, 72–77 (2011)