The Reasons for the Reality Gap in CAAD

TURK, Ziga
Univ. of Ljubljana, Faculty of Civil and Geodetic Engineering
http://itc.fgg.uni-lj.si/~zturk/ ziga.turk@itc.fgg.uni-lj.si

The term “reality gap” is used to denote the difference between the promised potential of the scientific and technical development, and the actual performance or use in the practice. In the paper, the author offers an explanation of the gap. He claims that the philosophical assumptions about the role of information technology (IT) in design have been inappropriate. There are serious limitations to the representation, exchange and communication about designed artifacts as objects with properties. A paradigm shift towards social-sciences’ research methodology is needed.

Keywords: technology transfer, philosophy of science, CAD theories, CAAD

Introduction
The term “reality gap” is used to denote the difference between the promised potential of the scientific and technical development, and the actual performance or use in the practice (Fig. 1). At a first glance it seems that the practice is sometimes slow in applying the achievements of research and development. Technological policies address this issue by promoting the “transfer to practice” and “best practice” initiatives. This traditional approach implies that the reason for the gap is in the insufficient information flow between the scientists, researchers and the users.

Another popular depiction of the way technical achievements are accepted is shown in Fig. 2 (Brandon and Betts, 1997). A graph similar to one in Fig. 2 was observed when looking at the frequency of some keywords in the CUMINCAD database (Martnes and Turk, 1999).

In the paper, the author offers another explanation of the gap. He claims that the research methodology and the philosophical assumptions about the role of information technology (IT) in design have been inappropriate. His analysis is based on the works of Thomas Kuhn (1962), Donald Schoen (1990), Martin Heidegger (1962) and Victor Frankl (1989).

Related work
The work looking at why computers are not good at assisting in conceptual design is too numerous to be covered in this short paper. Recently, for example, Tang and Gero (2001) claim that the reasons for this lies in the fact, that CAAD systems should be cognition based - they should support the four types of actions a designer takes - physical, perceptual, functional and conceptual. CAAD software is failing in areas where a designer is shifting too quickly between several types of such actions and the related intentions - faster than a software tool can follow.

The importance of freehand doodles is acknowledged in the thesis of Ellen Yi-Luen Do (1998), who also co-developed a prototype Electronic Cocktail Napkin software.

The importance of, as he calls it, unconscious design processes (the author uses the term pre-rational in this paper) is acknowledged in the thesis
of Haapasalo (2000). He distinguishes between three types of creative work - tension, incubation and heuristic working and believes that “the best solutions are achieved through intuitive tension and not stressed by conscious thinking”. In this paper it will be argued that in these cases the scientific frameworks and technical tools do not stand in the designer’s way.

**Philosophical Foundations**

To develop technologies assisting humans, it should be understood what humans, e.g. architects do. Fascinated by the 19th century’s rapid progress of the technologies, the traditional understanding of engineering has been influenced by the positivistic philosophy. The optimism is culminating in the so called techno-romanticism (Coyne, 1999). Even the intellectual work of an architect has been modeled in the same way as the Taylor’s assembly line work - input - processing - output.

Since the mid 1950s, the cognitive science has been trying to explain human, animal and machine cognition and intelligence. Cognitive science suggests that human thinking is somehow model based - that in our brains we create mental images or models of the real world around us and through the manipulation of these models we make decisions, design, achieve intelligence etc. The big promise of the artificial intelligence and the product/process modeling communities lies in the assumption that this is true and that these mechanisms could be replaced in computers.

This belief is founded on the various forms of the meaning triangle, originally conceived by Aristotle. He introduced a three-way distinction between words, “experiences in the psyche”, and things and this triangle has framed the Western thinking ever since - 2500 years later Ogden and Richards (1989) proposed a similar Ogden meaning triangle. It connects symbols, concepts and things. Each of the points on the triangle indicates a separate component that may be involved in thought or communication. The object is any entity from some real or imagined world about which an idea is held. The concept is the idea or thought of the object as held in the mind of a person. The symbol is an auditory, visual, or other form of utterance that is taken to stand for the object when communicated as part of a language. Indeed humans think in terms of concepts, but computers can do the same with words and other symbols.

Throughout the history of philosophy, there has been an opposition to such beliefs that culminated in the 20th century’s phenomenology of Heidegger and Gadamer. They studied the actual existence of things, how these things “come to be”, how their existence enters our consciousness. Heidegger, for example, claims that for most of the times, we do not mentally handle “objects”. His popular example is a person hammering. He claims that he does not experience a hammer, a nail and a wall as distinct objects until he is in some kind of trouble - when he faces a breakdown - for example a nail buckles, when he hits his finger, hammer head is loose etc. Only at this point the objects are constructed - subjective objects that may explain well the current problematic situation, and not some independent-of-the-situation, self-standing, “objective” objects (more on objectivity of objects in Turk, 2001). Rationalization follows only after that, well explained with the psychoanalysis of Frankl, who claims that the search for meaning is the main driving force of humanity.

Most of human actions, both in everyday life and in intellectual activities happens pre-rationally - before the real world situation is broken into objects with properties - as the person is immersed “in-the-world”
and “thrown” into a situation. In the centre of Heidegger thought is the idea of throwness. We solve most of the problems when thrown into a certain situation, context and not detached, when we observe the situation broken into objects with properties. The very instant we begin to think reflectively about a situation and analyse it in terms of object and properties we disconnect ourselves from the being-in-the-world, we try not to be thrown into a situation. We limit our view of the problem to the one that can be expressed by the objects and properties we have adopted, and create blindness for all other possible solutions.

**CAAD**

Computer Aided Architectural Design (CAAD) is a process in which architectural designs are created using a computer. CAAD history since the early 1960s has seen several technologies following the curves in Figure 2 (e.g. “expert systems” and “artificial reality”, see Martens and Turk, 1999). Typically designing has been modeled as a search activity (Figure 3) where the search space is a set of potential solutions and subsets of feasible, candidate, or constraint satisfying solutions (Galle, 1989). Problem solving is a search process (arrow) which can be more or less intelligent in terms of how the search arrow, from initial state to satisfying the goal, behaves. It is believed that by applying heuristic knowledge, experience, and insight, humans are able to reduce the search space and guide the search arrow efficiently. Computers, on the other hand, are expected to compensate some of that with rapid evaluation of several possibilities.

Various techniques have been invented that make this search more efficient or the search space seemingly unlimited. However, if the computer was to hold the representation of the design, the more semantically rich was the representation, the more limited was the potential solution space.

Typical examples of such problem solving in architecture is the design of floor layouts where certain rules govern the topology of the rooms, e.g. you do not enter the sleeping room through the kitchen. There are perhaps thousands of the possible combinations of the floor layout, but using rules or grammar one can extract the required solutions. Similarly, given the shape of the valley, “intelligent” design systems can generate several different concepts for the bridge, and, using rules, select the best solution.

Programs that do problem solving are created so that the model of the task environment is created, then it is formally represented as a database and populated with the data of a specific problem. A search procedure is implemented which, for example using generate and test strategy, finds plausible solutions.

The second influential model of designing is the so called “reflective” design (Schoen, 1990). He claims that designing is a reflective conversation between (1) the designer and his colleagues (e.g. architects, apprentices, engineers, clients) - therefore a dynamic social event, and (2) between the designer and the representation of his design (designer with the sketch, drawing or a clay model). See also Lawson (1994).

Phenomenological model of designing, however, would claim that the majority of design information is created in a pre-rational way. The easy one does not enter the designer’s consciousness because of their simplicity, the hard ones because of their complexity. Designers and scientists are well aware, that the “aha” moments, the innovative solutions to problems, often happen outside the process in which they were supposed to be solved - not in the office hours, not during “problem solving” but during some entirely different activity. This is explained with the fact that the problem solving mechanisms and the conceptual apparatus frame the designer into a limited solution space. Only after the person escapes this framework, the pre-rational processes can take place and provide a creative, intelligent solution. “The essence of

![Figure 3. Problem solving as a search (arrow) in the solution space.](image)
intelligence is to act appropriately when there is no simple pre-definition of the problem or the space of states in which to search for a solution" write Winograd and Flores (1986).

CAAD software is available for tasks that are not too easy (too much trouble to use a computer) and not too hard (not much help from a computer).

**The reality gap of CAAD software**
The promise of CAAD software has been that software will play a vital role in the entire design process. In the last 40 years, however, we can claim that indeed software has replaced the drawing boards for the design representation, but had little influence in the design process itself or in the designed products. It seems that conceptual design and most of the hard design problems are not solved with software. CAAD software is made in such a way, that it can represent the designs - it allows the designer to “enter” the design into a computer (e.g. ArchiCAD) or to analyse the design for various features (e.g. SAP2000) or code compliance. Through the evolution of CAAD systems, the representations are increasingly complex and “powerful” - the is more and more of the different components out of which designs can be composed, carrying and increasing amount of semantic information.

From the phenomenological perspective the reasons for limited or unsuccessful use of software in some design tasks are:

**Representation.** CAAD software predetermines the objects in which the designer is supposed to think. In this way it limits the solution space and encourages fixation into designs encouraged or supported by the software. Phenomenology claims that the actual recognition of objects is a most creative and intelligent human action. Component or object based CAAD software predetermines these objects, thus shifting the creative actions from the designer to the programmer or even a standardisation body like IAI/IFC.

**Situatedness.** Pre-rational human behaviour is exhibited in reactions to “being-in-the-world”. When a designer is locked in front of a computer screen, into some computer software, the “world” is severely limited. The designer is “in-a-model”, “in-a-tool”, “in-a-media” but not thrown “in-the-world”. Cocktail napkin, on the other hand, is very portable in easy to handle.

**Communication.** The traditional goal of communication is to move information from one point to the other. Using standardised schema and formats, the information flow in the building process should have been streamlined. What has been ignored, however, is that the actors in the process (e.g. architects, engineers, manages) have different ontologies, let alone the schemas. The goal of communication is not to share some information but to engage the listener into some action or activity - as in the speech acts. The sender, by sending some information to the receiver, is trying to make him do something - what, how and when is subject to the negotiation supported by some workflow systems.

**Particularism vs. holism.** The traditional human approach to understanding big problems is by breaking them into several smaller ones and then studying the pieces. This is very popular in sciences as well - also because of very Darwinist reasons - breaking up a problem means creating topics for several scientists, own PhD students, thus growing the size of that particular topic. Scientists looking at designing and information technology in construction in general are interested in looking at the parts, the pieces, and in growing their complexity, while the practise must create a whole product, combining these parts together.

**A line is just a line?**
The common theme in the above is related to the representation of the designed artefacts. For several reasons, the progress is measured by how complex the representations and the schemata in the software are. A typical sales pitch of the community developing standardised schemas for architecture and engineering is that a line is primitive, no one knows, what it is, while a wall is something that has a distinct meaning. This can and should be turned around: A wall is just a wall, a line can be anything, from a ray of sunshine, breeze of wind and, of course a wall as
well. In a study (Turk, 2001) the author argued that the main representation problems in CAD are related to the fact that in CAD we do not model the physical but the intentional objects or the so called “second order entities” (Schutz, 1962). Natural sciences, for example, concern itself with first order constructs, which are not changed or influenced by observation or intellectual manipulation. The second order constructs are “constructs of the constructs made by actors” and are therefore influenced and changed by the observer. The author claims that in the IT in architecture, we deal with second order constructs.

Conclusions

In the past, the expectations from IT have been too optimistic, because the nature of engineering work was misinterpreted - as rational problem solving in a solution space that can be modelled. The reality gap has been particularly wide in those areas where this rationality and modeling plays a vital role - e.g. in expert systems, decision support technology, product modeling. IT has not penetrated the areas in which it gets in the way, between the designer and her “being-in-the-world”. Pencil and paper are still a smaller barrier than a computer screen and mouse - besides, they are available anytime, anyplace. On the other hand, the reality gap was smaller in areas that enhanced or simulated the “being-in-the-world” e.g. the Web broadens the designer’s horizons, informal/unformatted communication (email, phone) strengthens the social collaboration and commitment negotiation, visualization creates virtual worlds with which a designer can communicate etc.

As for the future of research, most of the current research in the use if IT in architecture and engineering is shaped on that of the hard sciences - as if the object of interest was an “objective”, “self standing” item like a planet, animal or an atom. But it is not; although the designed objects have physical presence, they are interpreted by the actors in the design process and by the clients - they are taken for “objects” in different subjective ways. The object of inquiry should also be a human - the engineer, the architect, the technician - who is supposed to be assisted by the technology and who designs, engineers, learns and interacts with other humans. Objects of study are soft, second order entities as ones handled in social sciences.

References


