On Computational Design and Critical Thinking in Architecture

Thorsten Michael Loemker¹, Katharina Richter²
¹Technical University of Dresden, Germany,
²Bauhaus-Universität Weimar, Germany
http://www.arch.tu-dresden.de/cala/
¹thorsten.loemker@tu-dresden.de, ²katharina.richter@uni-weimar.de

Abstract: The paper examines the potential of existing traditional approaches in knowledge engineering against the background of current trends in the use of information technology in architectural design. The focus of this debate will be put on the commonalities of these approaches. The key concept they share is learning from the past, be it humans or be it machines. The vision they arise from is thereby as old as mankind. It is the wish to augment human thinking by artificial devices. The aim of the paper is to remove the scales from the eyes of those who neglect the past, be it in the architectural design process or in the use of Information Technology in architecture. The paper understands itself as a position paper and was moreover inspired by the announcement of the conference theme.

Keywords: Architecture; computational design; parametric modeling; experience-based design, rule-based design.

Introduction: On architectural design problems

Not only since prime of the Design Methodology Movement back in the 1950s and 60s but way before architectural professionals as well as researchers of various disciplines tried to give answers to the eternal question on what constitutes the nature of architectural design problems. Scientists and practitioners from areas as different as architecture, philosophy, psychology, cybernetics, cognitive sciences, artificial intelligence and information technology in general involved themselves in the problem to formulate tools aiming at supporting the design process. Up to now their engagement brought us close to a fairly good understanding of this question as well as of related topics, such as the nature of the design process. They delivered findings, which are widely agreed on. What seems to be not much challenged nowadays is not so much the question on how but rather on what an architectural design problem is! This statement also becomes apparent when we spot the terms computational design or digital architecture - which seems to be talk of the town these days - used in conjunction with their ostensible capability to understand and to solve design problems.

This fact urges the question on what computational means are currently used for and what (maybe) in contrast are the real world problems of architectural design. The approach to gain answers
to this provoking question is judged essential when aiming at offering fruitful computational support for architectural design. In asking this question we reflect on what we think must be at the bottom of this consideration:

Experiential knowledge in architecture”... nobody can be a good designer without the right experience.” (Jones, 1973, p. 5) Architectural design is said to be heavily relying on experiential knowledge. Therefore architects ever since aimed at gaining experience from their studies at university, from internships in architectural offices, from travelling or from discussions with others. They had everyday good and bad occurrences in their offices and thus learned their lessons. Those who also went to academia tried to pass down their experiences to students and for quite a while the whole system worked pretty well.

At a time were computers were still used as drafting devices the idea was born to map the experience of the designer into the machine. Most of these attempts failed. This might have happened because of technological grievances but also because of methodological problems. Nowadays one might think that these problems have been overcome and that this in conjunction with the enormous technological progress proved to be a real enrichment in the design process and empower a more sound architecture. But for some reason this is not yet the case.

It has to be stated that many up-to-date projects demonstrate a lack of quality. They suffer from structural damages, cost overruns and faulty designs. It seems to be obvious that the fact that cracks have emerged, leaks have sprung, drainage is faulty, mold is growing, and that snow and ice fall dangerously from the many curved surfaces and sharp edges of a building of one of the world’s leading architects, to name but one example, is an indicator for that something is going wrong.

Indeed many other buildings that have been built by less well-established architects suffer from same problems. It happens everyday and it happens everywhere. The difference is that the high-flyers today make use of sophisticated computational processes. But it appears that their intention of technology usage is restricted to merely generating stunning form.

To make it clear, we do not argue against breathtaking form nor do we argue against technology usage. It is quite the contrary, we argue for technology usage that supports the designer to achieve architectural form whilst ensuring a functioning building. Quite a few examples exist that demonstrate that these demands are not mutually exclusive. What we think is needed is an approach to make the knowledge about the How-To commonly available. In other words, it is time to learn from the mistakes of the past!

Learning from the past

In architectural design learning from the past is essential to prevent mistakes, which have been made by others beforehand. What we think is needed is to integrate this knowledge into the systems architects have at hand, let them be named BIM, CAD or whatsoever.

In his famous manifesto “CAAD’s Seven Deadly Sins” Maver states under point 3 “Déjà vu” that research themes in computer aided architectural design come and go in cycles, recognizing the fact that old ideas get sold as new ones (1995). This is not the intention here. But our intention is to make aware of the fact that not only in architectural design but also in CAAD a tendency exists to neglect the past. Therefore we aim at interconnecting research topics seemingly out fashioned with the actual situation in the use of computer technology in architecture and to let this function as a fertilizer of the current debate (see conference theme).

Architecture demonstrates itself as science, arts and technology and in some cases even as mercantilist action, the planning process of a building is also subject to suchlike dichotomies. If architecture applies to the principles of Vitruvius then it is the felicitous aggregation of duality and the mastery of complexity that characterizes the architects’ work. But how is it possible to walk this tightrope and create
a balance between the disciplines? What computational instruments do we have at our disposal and how do we codify emerging digital technologies in this context?

**Rule-based design**

To dwell on the extraordinary writings of the ancient world: Vitruvius already described the first principles of an approach that we name rule-based or, more fashionable nowadays, generative design. In his work “The ten books on architecture” (Vitruvius, 1960) he reflected on the particular social importance of architecture and proposed rule-based proceedings, whose contemporary reinterpretation seems to be difficult for today’s architects to follow. He also complained about the emerging dilettantism in architecture and it is conjecturable that the aim of his rules was to enable less talented master builders to happily resolve their work. Potentially he already treated the use of rules as a means of quality assurance. Without doubt he made use of a system of rules and reverted to his knowledge of other disciplines in the design and construction of his buildings.

A leap into present time: In his capacity as master builder it has to be expressed that the application of a countless number of building codes, standards, ordinances or statutes can hardly be accomplished by the architect. Therefore, it is necessary to develop appropriate methodologies to repeal this controversial antagonism. The architect needs to have tools at his disposal that guarantee the successful processing of the overall planning. These have to provide problem-specific knowledge in terms of rules that leads to functional and plausible solutions in the planning process (Loemker, 2006). Generally spoken rules are implements that derive from experience. It is generalized knowledge that comes with experience that is mapped into a single rule. Even if many architects negate the existence of design rules it is pretty obvious that design is geared to the use of rules. Alongside objective rules that represent indispensable normative elements of the design process, subjective rules most notably influence the peculiarity of a design solution in terms of esthetical characteristics.

Today’s projects that were significantly developed through the use of computers demonstrate the use of rules. Kees Christiaanse uses them in urban planning (2004) and the buildings of MVRDV, NOX, Asymptote Architects or Frank O. Gehry are also clearly defined by the use of rules. How else would it be possible to recognize the work of a specific architect other then through the extraction of their intrinsic properties in form of rules? Not all architects admit themselves to the use of rules. An exception is Christopher Alexander whose “Pattern Language” was beloved and coevally berated (1977). But it appears that the ideas of those architects who employed rules not only in the form finding process but also in the detection of constructive, functional or economical solutions seem to not exist any longer or at least do not play a prominent role anymore.

However, current technology seems to provide all possibilities for architects to define rules and to benefit from their usage in the design process. A new way of thinking and designing under the umbrella of terms like digital architecture, computational design etc., has been established that is particularly driven by the software industry.

For instance the protagonists of a good idea and their products theoretically allow the architect to integrate building design, construction and management processes in the planning through the use of a single tool that has been named BIM. In conjunction with parametric modeling a powerful tool emerged that promises to leave nothing to be desired. But is that really so?

Indeed, BIM seems to satisfy the claims of those who want to work on an integrated model that maps all different aspects of the design process throughout all stages of the planning. But who integrates the various rules we were talking about beforehand? Where are the rules that map fire regulations, material properties, lighting conditions, aspects of building services engineering, planning law, buildings
standards, ordinances, guidelines and the many other items we have to deal with? Indeed all those things can be scripted and integrated into the programs. But do we want the standard architect to script local building codes into his software? And do we then have to script these codes for each and every different state within which we design a building? It is almost cynical, but industry leaves us with a declaration of intent, provides an empty framework and architects cheer.

**Experience-based design**

At the beginning of the 1980s a simple but plausible idea made waves between researchers that dealt with the development of knowledge-based computer systems. The crucial factor was the observance that experts of different domains intensely made use of their own experiences during problem-solving. Researchers came to the conclusion that artificial expertise could not be modeled solely through a system of rules (Schank, 1982; Kolodner, 1993; Schank, 1999). Thinking and problem solving processes were considered as reminding processes during which decisions were made through a comparison of a new situation with one or more concrete instances deriving from past situations. This assumption reformed the approach of knowledge-based systems with regard to the representation of knowledge. Till then researchers tried to record knowledge as abstract and general as possible in form of rules or models to be capable of applying this knowledge to a wide variety of different situations (Kolodner, 1993). With the publication of “Dynamic Memory Theory” Roger Schank raised a plea to this course of action (1982). He argued that knowledge has to be represented in form of concrete instances of specific episodes of experience. Thus, a formalism was developed through which experiences were able to be formalized and described in terms of three parts: problem description, solution of the problem and result (Kolodner, 1991; Kolodner, 1993).

Rule-based systems as well as systems that work with concrete instances of experiences both rely on the usage of experiences from past problem solving situations, but they differ in the abstraction of knowledge. Whilst rules are small, independent and primarily consistent pieces of expert knowledge, concrete experiences contain large pieces of knowledge that are also often redundant with other experiences usually denoted as cases (Kolodner, 1993). Rule-based systems are applicable for disciplines that are well-known. A small amount of rules is often sufficient to successfully solve specific tasks. “Weak-Theory Domains” such as architecture are better covered through the use of systems that rely on concrete experiences (Heylighen, 2000). As stated earlier architectural design is obviously a process that is profoundly based on experiences. Thus, an amazing amount of research in the 90s tried to aim for the development of digital methods to support this course of action. Surprisingly the methods that were developed are of no relevance today.

**Reconciliation**

Computer programs that aim at supporting the architect in the planning of buildings require the integration of knowledge that can be formalized. This knowledge rests upon experiences that to some extent can be implemented in rules. However, many common problems in planning are too complex to be mapped with suchlike traditional approaches (Gritzmann and Brandenberg, 2003). This gap could be closed by means of experience based systems if their practical application would be in coincidence with the underlying theory (Richter et al, 2006). But negating the fact that the theory claims to describe experiences, whereas existing applications merely represent solutions of usually unknown problems, take the original approach ad absurdum (Richter et al, 2006; Richter and Donath, 2006). Rule-based systems as well as experience-based systems prove to be interdependent. Their conceptual methods are closely related to each other. Both methods demonstrate scope and potential for successful
implementation in the planning process. Whilst rule-based systems seem to become accepted on the basis of the fact that they can be applied to support the creative design process, experience-based systems still lead a wallflower existence. This makes it all the more surprising, as exactly these systems might be able to aid the architect in the obnoxious territories of the design process. Apparently not only technical feasibility plays an important role in the adaptability of the methods that come along with these systems, but also their general acceptance, or better: non-acceptance. Furthermore the use of suchlike systems also entails debate about a pre-defined set of criteria to measure the quality of an architectural design solution. But what we actually monitor is the exact opposite. Parametric modeling and knowledge-based systems were introduced in precisely those areas that elude a systematical assessment according to objective criteria. In this regard Rambow remarks that “the architectural society has to satisfactorily show that she performs services that are useful for the general public. … To provide proof of these services is considerably more difficult than one might think. That is to say it necessitates reaching a consensus what these services actually are and how to determine them adequately” (2001). Hence architects avoid verifiable evaluation of their buildings. “And even if technological possibilities have never been more promising, arbitrariness is the problem, or in other words: The necessity to make a choice without helpful criteria” (Fromm, 2002). Conclusion

In summary it has to be said that the possibilities that arise from the use of computational tools with regard to the solution of real world problems in architectural design are by far under-utilized. Nonetheless, manufacturers of digital tools and especially BIM-technologies ballyhoo their programs with concentrated passion as sustainable products, whereas their appliance degenerates to mere visualization and form finding tools with the sole advantage of a common building model.

But not only has the architects’ general attitude towards new technologies to be questioned. It applies to the manufacturers product lines in equal measure. The more so as the amount of tools available to support sustainable planning is manageable to put it mildly. The architect can only be indirectly accused of this development. Nonetheless, it is his liability to exert pressure on the industry to develop appropriate tools and it is likewise his responsibility to define rateable criteria to seriously and objectively evaluate the outcome of his work to prevent repeating mistakes from the past. But as one of the last bastions, architecture still hides behind fallacious assertion that no criteria or principles for architectural quality can be defined. She insists upon an antiquated claim for originality, which describes buildings as being of good quality only if they apply to the prevalent canon of architectural form or if they are judged as being significant by those who carry weight. Jäger puts it more drastic: “Success of those architecture stars does not rely on the fact that they master complexity, it relies on the fact that they negate complexity” (2002). Therewith he certainly has a point. Or would you like to work in an ecological high-tech building without functioning ventilation or mandate an architect who is time and again surprised by the cost of the buildings he designs?

References


