W.J. Mitchell – In Celebration of An Approach
Alexander Koutamanis
W.J. Mitchell – In Celebration of An Approach
Alexander Koutamanis

ABSTRACT

Among the founding fathers of architectural computing, the late W.J. Mitchell was arguably the one who managed to capture the imagination of the younger generations through overviews and insights that described the backgrounds and scope of the area in a way that was suitable for novices as well as people outside the area. Calling such overviews successful popularization is only part of the truth. More importantly, they formulate a coherent and comprehensive approach that allowed Mitchell to move between eras and contexts without loosing focus or sacrificing fundamental priorities. It is an approach that remains valid and usable, a source of inspiration and understanding for future generations in the area.
1. Foundation

Bill Mitchell passed away on June 11, 2010. There are many friends and colleagues eminently better suited to say something about Bill as a person and as a teacher than I. In his long and successful career he had held several positions and had worked with many other leading figures in architectural computing. My view is from farther away; it is on behalf of the wider audience that have also learned a lot from Bill.

To us, the second and third generation of CAAD, Bill was undeniably one of the founding fathers of the area, one of the handful of pioneers who had managed to establish computing in the research and teaching programs of their schools, who organized conferences and published articles and books on the subject. Some of us went to the founding fathers as graduate or PhD students, researchers or visitors but most came into contact with them at conferences or through their publications. In the then completely offline world, these few conferences and especially the scarce publications were very important for understanding what architectural computing was about. Equally important was learning what others had already done. It may sound paradoxical but back then, with fewer and less up-to-date information sources, it was more important to know what others did so as not to produce yet another theory or application on the same, tired subject.

2. Overview

At that time we were much in need of comprehensive sources that gave us overview of the area, of its principles and applications, of available methods and techniques. It is tempting to think that such overviews were easier back then. After all, there were fewer computer programs, fewer research products and a shorter history to summarize. This is certainly true but in most other respects it was tougher: architectural computing was unknown outside a few, mostly academic enclaves. It was generally viewed with scepticism and incomprehension. Even the limited number of applications and examples was a problem, as it left many lacunae in the overview, places where authors had to convince on the basis of first principles, future projections and promises. Having to explain how computers work to a largely computer-illiterate audience did not help either.

I have no doubt that many still use Mitchell’s 1977 book *Computer-aided architectural* [1] as an overview and not just for historical reasons. It was arguably the best book on the computer’s role in design in the early days of the area, a yardstick by which subsequent attempts were measured. Its emphasis on problem solving, representation, data management, industrialized building and generative systems is clearly indicative of an early stage of CAAD development but also of the intentions of its pioneers. The technology (hardware and software) it describes is to a large degree long forgotten and the domain goals have moved a lot but the book remains informative and relevant.
Its longevity lies in a couple of combinations. The first is that of theory and technology, the twin pillars of architectural computing. Especially today, with the proliferation of computers in all aspects of our professional and private lives, it is important to remember that CAAD never was a sum of computer programs. It may have been partly inspired by technological innovation but remained firmly grounded on a theoretical framework of architecture and design. Probably more than other founding fathers, Bill had the ability to present this framework in a clear, comprehensive manner and relate applications and programs to it.

The second combination concerns the audience of the overview. It is one thing to present an overview to the interested and initiated and another to make it accessible to a wider audience. Popularization (in this sense) was one of Mitchell’s strong points. He could present his own work and that of others in a coherent, relevant prose that made effortless reading, even with complex issues. This is another important lesson for today, as more and more books seem destined for a small public of researchers and practitioners already familiar with the subject matter. Even worse, they may be motivated by adherence or lip service to particular tendencies or specific architects. This is a far cry from overviews that attempt to inform and educate in scientific matters and with a scientific attitude. Architectural computing has been legitimized as an area and specialization thanks to this attitude but it is once again in danger of being relegated to technical support or dismissed as an esoteric plaything in the service of arbitrary goals.

3. Transition

The theoretical framework of architectural computing was even more prominent in Mitchell’s *The logic of architecture* [2], a book I have called “perhaps the best account of computerization in architecture to-date” [3]. In the same review I also called it a transitional book, little realizing what this transition meant. Obviously, every CAAD researcher and teacher had been working towards making computing accepted and integrated in architecture. Despite such valiant efforts, computers became ubiquitous in architectural offices and studios for different reasons: as computers became more affordable and popular in general office and entertainment use, it was a matter of time before they replaced drafting tables. It was more of a defeat than a victory for CAAD, which was no longer the exclusive custodian of an expensive, exotic commodity. Legitimization was no longer the issue; CAAD specialists now had to prove their added value, even justify their very presence in academic and professional environments.

Books like *The logic of architecture* represented the theoretical reply: they propagated architectural computing as an approach with a theoretical and methodical content that provided answers to fundamental and critical issues. That the computer was an integral part of these answers was a
matter of the approach, not vice versa. It was a confident approach that believed in its own potential, not one that became subservient to arbitrary morphological tendencies or sought justification in technological metaphors. Incidentally, the technical reply did not hold stand for long: the expertise CAAD specialists had in the effective and efficient use of computers diminished rapidly with every new program or version that appeared on the market and with end users’ increasing proficiency in computer use. CAAD’s technical head start was never enough to ensure a lasting influence.

Despite its focus on the troublesome notion of “architectural languages”, The logic of architecture typifies the quest for formal expression in architectural design. It presents a sequence of detailed arguments towards a firm theoretical foundation of CAAD: “a comprehensive, rigorously developed computational theory of design”. The main difference with earlier attempts lies in the purpose of the theory: it is for the benefit of architecture in general but also aims in particular at guiding the development of new, relevant software that will enhance the utility and status of CAAD. Mitchell seems to realize that the world is catching up with academia and tries not only to establish a theoretical framework but also to connect it directly to important problems. He brings together different questions, observations and solutions that derive from various sources into a computational system that manages to avoid simplistic solutions and overreliance on technology. Much more than Computer-aided architectural design, The logic of architecture is selective and at times personal - substantially different from an overview of the area.

4. Scope

The popularization of computers also brought on an increase of scope. It meant that new elements came into architectural computing, not in the least the Internet. These added to the possibilities of architectural design but also allowed CAAD to extend its application area to new ventures such as information design. I must admit to an initial reluctance to consider such extensions as part of architectural computing. If it was unrelated to architecture, there was no way I could rely on domain knowledge, I reasoned. In the end it does not matter: architecture’s loss could be someone else’s gain - usually computing’s in general. If our skills are applicable to other tasks, if our understanding of spatial matters (including metaphors) and visualization can be used outside architecture, then we have also proved a point, namely that what we have been doing has relevance to the modern digital world.

Mitchell understood these changes and the potential of existing specialist knowledge to general computer use, adapting his priorities and preoccupations accordingly but abandoning little of his general frame of reference [4]. A first indication came with The reconfigured eye, an almost encyclopaedic celebration of digital imaging [5] that explored the use of images in communication, the way they can convey meaning and guide
interpretation. It was followed by *Digital design media*, a book firmly based on the tradition of CAAD overviews [6]: a concise introduction that presents and explains fundamental issues and principles (as opposed to enumerating software and hardware implementations). It forms a practical companion to *The logic of architecture*, primarily intended for CAAD specialists and learners, yet is clearly aware of the changing context and roles of the field and less constrained by the technological state of the art than other overviews. Its analytical, open-minded character reveals possibilities beyond the original goals of CAAD – not what one must do but what one can do in architectural computing. It is an implicit acknowledgement that the landscape of the area has proved less predictable than originally envisaged by its pioneers. The last chapter, entitled “What was computer-aided design” suggests that the transition was by then over, even though the new context was far from clear.

This new context became the subject of Mitchell’s subsequent books [7–9]. In a series of essays he explored facets of the emerging combination of the digital and the physical world from the dual perspective of architecture and computing. This new combination places architecture and urbanism in the new context of widespread information networks that stimulate mobility of data rather than people. Bill’s interest in the transfer of activities that used to require purpose-built spaces to electronic environments can be seen as futuristic fascination with technology but also as a gentle yet enthusiastic introduction to architectural questions of the (near) future: information is not just a commodity but also a force shaping society and the environment. Urban life is transformed by the convergence of the digital and the physical. On the one hand, the proliferation of computing and information processing devices and services is changing places of work, entertainment and habitation; on the other, activities that used to take place in buildings designed and constructed for that purpose are increasingly migrating to digital environments – from contiguous space to information networks. As a CAAD specialist, Mitchell appears capable of explaining the background and effects of the new social and technological phenomena to the public in general from the dual viewpoint of a computing specialist and an architect.

5. Extensions

Given the focus of activities at Media Lab’s Smart Cities research group and The MIT Design Laboratory Bill had been leading, *Reinventing the automobile* [10] came as no surprise to those who follow CAAD conferences. Even though the focus is on new forms of vehicles, with Bill’s backgrounds and preoccupations, it was only natural that the context of the vehicles is not only the Internet as a connected environment but also the city as a primary beneficiary of new ideas on mobility and transportation. The main argument of the book is that the proposed vehicles can contribute to the solution of a number of problems relating to sustainability without reducing the freedom and scope of movement afforded by personal, externally-powered
means of transportation. It is a view that also applies to buildings but one could argue that the automobile industry and related consumer behaviour present more opportunities for quick, effective transformation than the building sector. Still, it remains an area with spectacular failures (including the Sinclair C5 which is not mentioned in the book).

From an architectural viewpoint it is nevertheless quite informative and instructive to follow the authors’ analysis of the mutual influences between automobiles of the past and future and the built environment towards vehicles designed for cities (as opposed to cities designed around cars), even though a large part of the book is on the infrastructure required by the proposed vehicles, including energy networks and charging facilities. Architecture is only too often seen as a static and isolated phenomenon rather than as adaptable container of dynamic activities or as part of wider networks and superstructures.

At another level, the book also makes interesting reading as a coherent collection of substantiated arguments for a specific direction of development that can undoubtedly attract attention and investment. It is a good example of how a cohesive, comprehensive research initiative can present a grand cause, a potentially major change of the world (including what has to be researched and demonstrated in order to reach an adequate level of persuasiveness). Is this different from what CAAD aspired for architectural design?

6. Commentary

Bill Mitchell seems to have spent his professional life evangelizing changes to designing and the built environment brought on by computers and information networks. Yet in World’s greatest architect [11] we are confronted with a subtle yet important change in perspective: Bill is now actually inhabiting the brave new world, not just projecting and explaining it but also commenting on it. The shift from futurologist to critical citizen appears to result into a less enthusiastic attitude towards the new situations he observes, still generally positive with respect to new technologies but not always pleased by the way they are applied. It is probably the most enjoyable of his books (possibly because his audience has turned into grumpy old men), in particular the essay that lends its title to the whole bundle: from a certain viewpoint it reads like an abstract of a new, radically revised version of the 1977 classic Computer-aided architectural design for the 21st century. It is a great pity that such a book will never come.

7. Fundamentals

A few years ago, in the framework of a wider project, a rather disparate group of researchers considered questions such as the state of the art in CAAD, possible directions for further development, new goals and ways to achieve them. We were generally worried by the dangers facing architectural
computing, from trivialization into a loose collection of mere tools toovereagerness to jump from one bandwagon to the next. On the positive side, we had no doubts about the potential of our area, the available knowledge and the solutions that have been developed over the years.

The way out of these problems seemed rather straightforward to us: return to the fundamentals of architectural computing [12]. By that we did not suggest a conservative attitude that reduces our endeavours to just a few, possibly esoteric subjects. There was no question of sacrificing the width of the area. What we wanted to propagate was the significance of a theoretical framework with underpinnings in the corpus of existing knowledge on architectural computing. The pioneers of CAAD became founding fathers exactly by stressing the importance of such a framework and demonstrating its applicability and utility in the solution of architectural problems. Among them, Bill Mitchell holds a special position by his production, his ability to reach a wide public and his insights. He defined an approach that remains usable and valid in a still changing world, something he has proved himself with a sequence of successful transitions with respect to scope and subject matter.

A return to the fundamentals does not imply unquestioning acceptance of tenets formulated in another era and context. On the contrary, we have to look back critically at the corpus of architectural computing and learn from it so that we do not unknowingly repeat the same solutions and mistakes (if it is at all conceivable that researchers of today can be unknowing of the inheritance of their area). We should consider not only achievements and fundamental issues that can inform and stimulate our further work but also limitations and blind spots we have to avoid. There is a lot that is relevant and usable but equally a lot that is in need of refinement, adaptation and extension if we are to formulate new approaches for the future.

Undoubtedly such new approaches can be as varied as the people who devise and apply them. Nevertheless, I am certain that the successful ones will share a number of common features. First of all, they must be forward-looking, eager to forge ahead, joyous explorations of both applications and fundamental ideas. While being firmly based on existing knowledge and experience, they should be open to new additions and challenges, unafraid of venturing into new territories. To do this they have to relate to theories that not only describe and replicate analogue processes but also explain and predict phenomena so as to specify and produce tools with real intelligence, tools we both need and deserve. Finally, despite the explosive growth of technological opportunities, they must manage to avoid trivialization and disintegration of CAAD into a number of loosely related technical tools by keeping a grand scheme of architectural computing in mind as well as in the picture.

These features are abundantly present in the work of Bill Mitchell: he is one of the few who managed to pursue such an approach and show the
rest of us that it can be done. I have no doubt that his books that gave us overview and insight should remain among the staple reading of especially newcomers to architectural computing, as a source of historical development, achievement and methodical structure.

References

Alexander Koutamanis
Delft University of Technology
Netherlands

a.koutamanis@tudelft.nl