1 From Theory and Practice

A decade ago, a group of people was researching and studying at the University of East London for their MSc Computing & Design that focused on the production of spatial organization and perception through the medium of computation. Computing was regarded as an instrument for perceiving, representing, and generating spatial phenomena, and thus challenged traditional production of architecture. Projects were linked to design briefs because the master’s program was coupled with a diploma design studio. Projects were to balance computing with design requirements as the Royal Institute of British Architects (RIBA) were examining students as well as a representative of the academic field of design computation (then John Frazer). A series of computational approaches were investigated for spatial properties of architectural design for the first time such as L-systems and genetic programming (Coates et al., 1999), swarms (Miranda, 2000) or neural networks (Derix et al., 2001). Theories were applied to explain the use of computation within a scientific context with logical consequence on spatial and social phenomena such as Maturana’s autopoiesis, Luhmann’s theory of social systems, or van Foerster’s second order cybernetics. The reading of spatial theories was encouraged to inform the design of algorithms like Perec, Hillier, Arnheim, or Norberg-Schulz. The course became a template for many university programs such as UCL’s MSc Virtual Environment or the AA’s DRL. The projects were largely seen as academic research with little relevance to the industry.

1.1 Tools vs Paradigms

From 2004, some members from UEL’s Computing & Design program formed the Aedas|R&D Computational Design and Research group (CDR) to apply these academic approaches to industrial briefs and open up the industry itself (‘educating the client’) to the potentials of generating spatial organizations through computing. The notion of space as a component of architecture was hardly relevant as a driver for design over the last 15 years in the global industry. Especially, computation is seen as a ‘problem-solving’ tool to support structural, geometric, climatic, or statistical aspects of traditional workstages, not questioning the stages through the new medium per se. Architects use computing as a way to erase differences with engineers, not to enhance their own knowledge of the key aspects of architecture: occupation and space.

This is not surprising as academia deals with computation in an either completely mystified or de-mystified way, reinforcing the existing roles within design disciplines. The former sells computation as artistic expressions of the new paradigm of complexity but offering only geometry via existing software (for example ‘morphogenesis’ by Generative Components or Grasshopper) and pseudo-philosophy. The latter develops serious scientific simulations for discrete aspects of design through research funding that often produce opaque mathematical models, inadequate for design practice. The former being
regarded by industry as a marketing tool, the latter as interesting research, supporting distinct design stages – none as an applicable design methodology or paradigm.

2 Losing the Jargon

First attempts to implement academic holistic models representing appropriate simulations of cybernetic systems failed as they did not synchronize with existing design workflows and shut out other designers and stakeholders. Such all-inclusive models are still currency and in fact, it seems the trend is increasing to build simulations that integrate as many scales, parameters and stakeholder participations as possible. This Feature Creep is driven by industry-philosophy such as ‘life cycle design’ or ‘building information models’ and tends to look at a project from an observer point of view rather than the designer’s (top-down rather than bottom-up).

As a reaction to this separation into simulation and (not with) design, developments became increasingly bespoke to design stages, acting as automation of design mechanism. This initially allowed for active participation and synchronization in design stages but also submitted computation to the role of tools, not methodology (Miranda and Derix, 2009). Consequently, the language of communication about computational developments changed to professional terms, losing the theoretic terminology applied in academia. This was reflected in the teaching at the master’s at UEL where theory became subordinate to crafting: learning to read and write code and techniques before learning to philosophize. The change of curriculum at UEL bore immediate fruits as students had un-mitigated access to theories of computation by understanding how to craft notions of complexity themselves from scratch before mapping them into abstract paradigms.

3 Emerging Performances

The distinction into tools and methodology appears important to convey the problem of function that is usually attributed to tools. Tools could be perceived as automation of mechanism with fixed functions. While tools could be used for other purposes, they have been designed with a specific purpose in mind. Computation on the other hand doesn’t serve a specific function but as a universal machine could simulate any mental tool if its function is abstracted adequately through a computable representation. Computation is about the design of representations and their dynamic organization. It doesn’t have properties that extend anything else other than itself (binary representation and states). As such, a designer could simulate any condition via computation if he knows how to create adequate representations and organize their states, including spatial phenomena.

Initially, this might sound plausible and many seminal works about the nature of artificial design (Simon, 1969) and the complexity of design representation exist (Rittel & Webber, 1973). But attempts of knowledge transfer from sophisticated academic research into design computation (as opposed to technology!) to the design industry have so far left few traces. Software used in the architectural and urban design industries are based on automation and technological support, not design methodology. Further, that type of software is often not developed by academics but by practitioners themselves.
Two areas traditionally identified by academia for design computation are layout automation and urban simulation, as it appears to represent the ‘right problem to solve’ by computation, i.e. a combinatorial set of statistical quantities and some abstractable geometry. But as Liggett concludes in his survey about layout automation, no development has been successfully integrated by architectural practice (Liggett, 2000). That still holds true no matter if the publication is already 10 years old. Equally, urban simulations have found their application so far only at planning levels for land-use evaluation, demographics or transport, not at urban morphology levels where designers search spatial solutions.

This appears to stem from two conditions: a lack of understanding ontological representations made by designers and a technological ambition to create holistic simulations. The objects to be represented in all layout simulations described by Liggett - for example HeGeL, LOOS or Wright - are driven by computational techniques of close-packing and providing full convergence of solutions before allowing the user to inspect the results. Although HeGeL already employs heuristics as evaluation criteria, it still overlooks the nature of layout generation as a multi-stage and dynamic heuristics process, meaning that no single system, no matter how complex, can satisfy the design search space for layouts. Layout design happens through many representations for topologies, approximations of shapes for rooms, grouping of activities, organizational hierarchies, operational arrangements, design intentions and intuitive adaptation for under-constrained components like circulation through nongeneric heuristics. Even social dynamics for communication, micro-climate, and cognitive qualities are starting to inform the constraint space. Technological ambition sidelines the designer into a seed-watch-evaluate role who feels his intentions and heuristics are not participating in the search. This is particularly dangerous for urban design where current urban simulation developments bundle so many scales and performances based on the ambition to create city information models (CIM) on the template of building information models (BIM) that complete black-boxes result where the designer adapts to the software, not the other way around.

Wicked problems (Rittel & Webber, 1973) like layout or urban design require the experience of designers to negotiate the many explicit and implicit aspects that can be represented through computation. Particularly, when design aspects are not discursive and the amount of data is large, the key organizing principle of designers and design teams are their learned heuristics, not performance indicators and data sets. While computation shouldn’t imitate analogue heuristics, it can express its own search mechanism via visualization of processing steps. If a designer can interfere with computational heuristics and observe the search struggle, the opportunity for identification between designers’ analogue and computational heuristics are given. This type of empathetic coupling (or ‘structural coupling’ as Maturana calls it in his autopoietic theory) enables the validation for wicked problems when no explicit goals are set. It constructs narratives communicated between designers that eventually produce generalized abstract approaches to design tasks. For computational designers, such abstract search mechanisms are meta-heuristics. Meta-heuristics can not be applied immediately but need to be adapted through implementation into design contexts. In other words, meta-heuristic algorithms are not to be taken literally and have no inherent automatic aesthetic but produce meaning when situated. Layout or urban design is composed of many design scales and aspects that designers have developed many heuristics for. Each aspect is complex in itself and thus a multitude of computational simulations need to be negotiated by a designer and evaluation spread across a series of applications.

One of the key problems with layout design represents the non-quantitative nature of circulation. Designers don’t necessarily want close-packed topological diagrams as they know that the layout will change many times over and no specific cost-function or geometry can be given to circulation. Few explicit performance criteria exist for circulation other than the regulatory maxima for area and fire routes, while inherent performances to do with spatial, social, and cognitive properties rest within the forming intentions of the designer. Circulation emerges from a series of negotiated implicit and explicit design searches. Where lightweight computational simulations exist that can be assembled into integrated hybrid workflows, design intentions for implicit architectural properties can gradually emerge through the empathetic
coupling of heuristics. The lightweight applications must be limited in functionality (parsimonious) and visualize in real-time the ‘intentions’ of the simulation as it searches, to render its heuristics transparent. Like many analogue precedents, models need to be interactive and responsive during run-time. Frei Otto’s models as well as interactive workshops in urban design (‘charrettes’) show how to learn from autonomous processes: the models represent independent organizations and can not be linearly affected but only ‘nudged’. It’s a kind of dynamic ‘stygmeric’ (4) system where designers’ interferences are absorbed visually, allowing all participating stakeholders to understand the organizing principles of the autonomous system. Knowledge can be extracted via the interaction between designer and model, or the model and its context. Models generate diagrammatic ‘states’ as outputs, not optimized results.

As the organizing logic is of simulations distributed across the constituent parts that produce consensual states as outputs, it becomes difficult to attach the often desired explicit ‘performances criteria’. While some basic measures can be extracted, many desirable states must emerge as a validation of the assumptions that a computational representation and its processes is built around. For most architectural design aspects, many desired performances must emerge, not be hard-coded as sum measures of some parts.

A common reaction to the above arguments is that this is ‘old hat’ by now. But one can hardly see this approach practiced as it requires constant learning of dedicated teams that can construct frameworks of computational heuristics within live design contexts (hence industry has an edge over academia in this field of research). Only when these new design heuristics based on computation have been learned and generalized into frameworks through many repetitions and on complex building types (another issue with computational design education are overly simple building types) can new workflows emerge that non-computational designers trust and couple with.

4 Hybrid Paradigm: Value of Spatial Phenomena

 Clearly, technology advances play a crucial role in the realization of such computational approaches. Real-time processing, run-time interaction, and accessible APIs and SDKs, as well as publicly available GPL libraries support this computational design development. But it remains in the hands of experienced designers to responsibly use this technological power and resist the common temptation to increase the complexity of models.

After the experience with academic and bespoke parametric models, the CDR used two projects in 2007 – the Ground Zero Memorial Museum and Smart
Solutions for Spatial Planning – to start building an open framework for heuristic design simulations in Java and OpenGL. The projects aimed at the development of an assembly of applets that would investigate the design and representation of various spatial phenomena for architectural and urban design and the alignment with designers’ heuristics. It allowed for example the long overdue extension of Space Syntax’s 2-dimensional analytical methods into interactive real-time and dynamic – i.e. time-based – spatial simulations that architects and planners can choose to simply analyze designs with or create generative design procedures from. Currently, the framework consists of a long list of algorithms, meta-heuristics, and data structures that allow the group to design intuitively through computation and create new heuristics.

Having arrived at a minimal maturity and being able to demonstrate to clients this new design paradigm of artificial heuristics through computation, a careful return to the initial debate about the representation and generation of spatial phenomena might be possible. Particularly, the debate about the ‘value of design’ being led in the industry presently allows an entry for computational approximation of spatial phenomena. In a negative economic climate, companies struggle to create a competitive edge through ‘unique selling propositions’ (USP) paired with a wariness of complicated building structures as the only visible outcome from computation so far. A shift to hitherto implicit design values is demanded that are located within social, cultural, environmental, behavioral and cognitive conditions.

Dusting off the old arguments with new methodological insights appears to gain momentum. Theories and observations of spatial organization and perception from George Perec (Perec, 1974) to Rudolf Arnheim (Arnheim, 1977) can support the design of computational heuristics and their emerging evaluation criteria. It requires also the inclusion of previously ignored design partners like cost-consultants or developer agents who can provide insights into tacit evaluation mechanisms and who must adjust their models to more contextual conditions. The value of objects (solids) in a design is ever more moving towards the original notion of ecology (Maruyama, 1963) that can now be simulated in design searches via computation as the behavior – organizationally not statistically – of the system can be visualized and interacted with. Visualized relationships and organizational dynamics that objects (dynamically) belong to determine their value as much as performance values and material cost. Spatial value can be approximated in relation to non-discursive dynamics like occupational potential and cognitive performance. These relations can partially be made explicit through design intentions that emerge through the coupling of designers’ and computational heuristics. Circulation can be understood to have more value than just egress times, flow capacities and area. It emerges from the mediation of multiple simulated conditions.

Naturally, it is very difficult to avoid tautologies in the set up of the models and instill truly emergent properties. Many researchers have attempted to create ‘objective’ representations like the members of the Cambridge school · Haggert & Chorley, March & Steadman, etc. (Keller, 2006) · and some artists have shown how slippery the slope can be for assumed values of spatial properties · Bense (Bense, 1960) or Klein (Noever, 2004). Now, with recent developments, architects are finally in the position to allow for better arguments to use computation that sits as a spatial design paradigm between the scientific[5], problem-solving and the purely artistic. ‘Good design skills’ are not dependent on computing knowledge. But they facilitate deeper understanding of spatial conditions and how to form new design search strategies. The negotiating of artificial and analogue heuristics enables designers to find new expressions for sensitive spatial designs that will probably embody very simple geometries but subtle perceptive intricacies.

“Space between things turns out not to look simply empty” (Arnheim, 1977)
References


Endnotes


(2) There is a mis-perception that algorithms, especially heuristic algorithms like evolutionary algorithms, should produce complex-looking or organic outputs. After a lecture at the Hyperbody at TU Delft, a student asked the question: “Why don’t your designs look more funky when you are using so much computation?”

(3) And they feel that their experience – heuristics and intentions – is overridden if they cannot identify with the visualized process and result. A demonstration of Eleczkurtaj’s excellent layout simulation (Elezkurtaj and Frank, 2002) at Aedas architects in 2005 resulted in a categorical rejection as architects voiced the here mentioned reasons. The CDR has as a consequence built series of simulations dealing with distinct aspects of layout generation that the architects can assemble themselves according to brief and intentions.

(4) Stigmergy represents a kind of indirect communication system via an external environment (Bonabeau et al, 1999)

(5) Christopher Alexander’s skepticism is representative that informed Notes on the Synthesis of Form where he states: “Though design is by nature imaginative and intuitive, and we could easily trust it if the designers’ intuition were reliable, as it is it inspires very little confidence.” (Alexander, 1967).